**Università degli studi di Napoli “Federico II”**

**Dipartimento di Ingegneria Industriale**

**Dottorato di ricerca in “Ingegneria Aerospaziale Navale e della Qualità” - XXVIII Ciclo**

Settore scientifico disciplinare ING-IND/05:
Impianti e Sistemi Aerospaziali

# Advanced LIDAR-based techniques for autonomous navigation of spaceborne and airborne platforms

*Ing. Roberto Opromolla*

**Tutor**

*Prof. Michele Grassi*

*Prof. Giancarlo Rufino*

*Prof. Giancarmine Fasano*

**Coordinatore**

*Prof. Luigi De Luca*

*“The good thing about science*
*is that it is true whether*
*or not you believe in it”*

Neil deGrasse Tyson

# Table of contents

# Abstract

The main goal of this PhD thesis is the development and performance assessment of innovative techniques for the autonomous navigation of aerospace platforms by exploiting data acquired by electro-optical sensors. Specifically, the attention is focused on active LIDAR systems since they globally provide a higher degree of autonomy with respect to passive sensors. Two different areas of research are addressed, namely the autonomous relative navigation of multi-satellite systems and the autonomous navigation of Unmanned Aerial Vehicles. The global aim is to provide solutions able to improve estimation accuracy, computational load, and overall robustness and reliability with respect to the techniques available in the literature.

In the space field, missions like on-orbit servicing and active debris removal require a chaser satellite to perform autonomous orbital maneuvers in close-proximity of an uncooperative space target. In this context, a complete pose determination architecture is here proposed, which relies exclusively on three-dimensional measurements (point clouds) provided by a LIDAR system as well as on the knowledge of the target geometry. Customized solutions are envisaged at each step of the pose determination process (acquisition, tracking, refinement) to ensure adequate accuracy level while simultaneously limiting the computational load with respect to other approaches available in the literature. Specific strategies are also foreseen to ensure process robustness by autonomously detecting algorithms' failures. Performance analysis is realized by means of a simulation environment which is conceived to realistically reproduce LIDAR operation, target geometry, and multi-satellite relative dynamics in close-proximity. An innovative method to design trajectories for target monitoring, which are reliable for on-orbit servicing and active debris removal applications since they satisfy both safety and observation requirements, is also presented.

On the other hand, the problem of localization and mapping of Unmanned Aerial Vehicles is also tackled since it is of utmost importance to provide autonomous safe navigation capabilities in mission scenarios which foresee flights in complex environments, such as GPS denied or challenging. Specifically, original solutions are proposed for the localization and mapping steps based on the integration of LIDAR and inertial data. Also in this case, particular attention is focused on computational load and robustness issues. Algorithms' performance is evaluated through off-line simulations carried out on the basis of experimental data gathered by means of a purposely conceived setup within an indoor test scenario.

**KEYWORDS**: LIDAR; spacecraft relative navigation; uncooperative pose determination; sensor modeling; sensor simulation; spacecraft relative dynamics design; Unmanned Aerial Vehicles; localization; mapping

# Chapter 1 - Introduction

## 1.1 Autonomous relative navigation of multi-satellite systems

Advanced applications like On-Orbit Servicing (OOS) [1, 2] and Active Debris Removal (ADR) [3, 4] have recently received growing attention by researchers operating in the space domain.

Indeed, OOS missions represent the only way to have a reliable physical access to a space vehicle while it is still in orbit, giving the possibility to carry out extremely important activities such as satellite inspection or repair, satellite relocation and satellite restoration or augmentation. As a consequence, OOS may lead to a significant reduction of the risk of mission failure and of the mission cost, an increase of the mission performance, lifetime and of the mission flexibility, as well as enable new missions [1]. On the other hand, several studies [5, 6] have shown the need of ADR together with the adoption of passive mitigation measures to stabilize the population of debris in orbit, respectively by removing man-made non-functional space objects thanks to an autonomous active spacecraft and by designing satellites for future operations so that they are able to autonomously de-orbit at the end of their operative life. This is of utmost importance to prevent the triggering of the "Kessler syndrome" [7] which is worldwide seen as the major threat to the possibility to develop future space missions.

OOS and ADR are related to the concept of multi-satellite systems since they both require a servicing spacecraft, typically known as the chaser, to perform autonomous relative navigation maneuvers, such as rendezvous and docking and monitoring, in close-proximity of an assigned target which is an operative satellite and a debris, respectively. These maneuvers require development and implementation of technologies and techniques for short range pose determination. Specifically, the target-chaser relative pose, namely the set of parameters which represent the relative attitude and position between the two platforms, needs to be estimated in real time and with high accuracy in order to represent a reliable input to carry out control of the relative motion. Although some solutions have already been implemented by means of demonstration missions for OOS of cooperative spacecraft [8] and for Formation Flying (FF) [9], the state of the art still lacks of consolidated resources suitable for most of OOS applications as well as for ADR since they require additional technical challenges to be dealt with. Firstly, the target may be uncooperative, meaning that it is not equipped with a dedicated communication link and there are not easily recognizable artificial markers on its surface, e.g. Light Emission Diodes (LEDs) or Corner Cube Reflectors (CCRs), located according to a specific known pattern. Secondly, it may be freely tumbling, i.e. characterized by an uncontrolled rotational dynamics, thus being difficult to be approached by the chaser due to the necessity to compensate for the angular velocity. Finally, due to the long exposure to the space environment (e.g. space radiation, possibility of impact of small non-tractable debris fragments), it may suffer physical damages as well as optical degradation of its surfaces, thus looking different than expected. These are the reasons why, up to now, the related activities are limited to planning of demonstration mission [10, 11] and to proofs of concept [12].

In this framework, mainly due to the uncooperative nature of the targets, pose determination can be tackled by relying only on measurements provided by Electro-Optical (EO) sensors and on algorithmic solutions which exploit the knowledge of the target geometry (model-based techniques). Specifically, EO sensors can be classified into active and passive systems, e.g. respectively Light Detection and Ranging (LIDAR) technologies, and monocular or stereovision cameras operating in the visible and infrared bands. On the other hand, depending on the typology of selected EO sensor, model-based algorithms can be further ranked into monocular and three-dimensional (3D) techniques. Here, LIDAR have been preferred to passive sensors, in spite of the related higher mass, cost, power consumption, and lower frame rate, mainly because of their robustness to the variable illumination conditions typical of the space environment and their capability to easily discriminate the target from the background (segmentation) independently of the light conditions. Hence, the main goal of this research activity is the development and performance assessment of innovative model-based algorithms, which are based on the processing of 3D data (point clouds) provided by an active LIDAR, for pose determination of an uncooperative space target.

The proposed approaches are aimed at providing advancements to the state of the art in terms of

- accuracy level, by minimizing the estimation error in the pose parameters;
- computational load, by keeping the processing latencies at minimum;
- robustness and integrity of the designed architecture, by foreseeing strategies for autonomous failure detection as well as for safe transition between the different steps of the pose determination process.

In addition, given the complexity (from both the technical and the economical points of view) of performing demonstration mission in space as well as of realizing realistic experimental setup, the availability of numerical simulation environments is of crucial importance to test and validate technologies and techniques required for ADR and OOS missions. Hence, the performance of the approaches presented in this thesis is assessed within a simulation environment which is conceived to realistically reproduce the operation of a LIDAR, multi-satellite relative dynamics, target geometry and to implement pose determination algorithms. In particular, as regards the relative dynamics simulation, an original method to design relative trajectories for target monitoring, suitable for both OOS and ADR activities, is also introduced. These trajectories are conceived to satisfy not only safety requirements, but also relative navigation ones by wisely selecting the perspective at which the EO sensor on-board the chaser observes the target, thus providing meaningful and not ambiguous datasets.

## 1.2 Autonomous navigation of UAVs in complex environments

Unmanned Aerial Vehicles (UAV) have nowadays become essential tools to perform a wide range of both military and civil applications, e.g. border patrol, coastal surveillance, crime fighting, hurricane and polar ice cap monitoring, forest fire detection, natural disasters response, aerial photography, crop dusting, package delivery, and pipeline and powerline monitoring. One of the main guidelines for researchers working on UAVs is the necessity to improve their level of autonomy when flying in complex environments, i.e. unknown areas, potentially dangerous to the human life and full of static and/or mobile obstacles, which can be both indoor and outdoor. This is of crucial importance since the autonomy of an UAV, defined as the capability of accomplishing their assigned missions by carrying out, without any external control, sensor data integration, perception, situational awareness, communication, planning, decision-making, and execution, represents the best way to unleash their full potential, thus leading to further advanced applications [13].

From this perspective, one important issue to deal with is the necessity to make the UAV capable of navigating autonomously and safely in any scenario. This has been achieved, in the case of widely open outdoor areas, by integrating inertial data from an Inertial Measurement Unit (IMU) with position information provided by Global Navigation Satellite System (GNSS), e.g. GPS, by means of a sensor fusion architecture [14, 15]. This concept, known as GPS-INS, has been applied successfully to both fixed wing and rotary UAVs [16, 17], e.g. by exploiting the Extended Kalman Filter (EKF) approach. On the other hand, in several environments, respectively indicated as GPS-denied and GPS-challenging, the GPS signal may be completely absent (i.e. indoor) or unreliable due to multipath, absorption and jamming phenomena (e.g. in urban or natural canyons and under forest foliage). In these cases, autonomous navigation must be performed by relying on the integration of inertial data with measurements provided by EO sensors or by other ranging systems. This can lead the UAV to acquire advanced capabilities of state estimation (i.e. knowledge of the vehicle's state in terms of attitude, position and velocity to be sent to the flight control unit) and perception (e.g. target detection and tracking).

One possibility to autonomously navigate a Micro-UAVs (MAVs) flying in GPS-denied or GPS-challenging areas is given by simultaneous localization and mapping (SLAM), which is the process allowing a vehicle to reconstruct its own trajectory in terms of both position and attitude (localization), while simultaneously building a map of the area surrounding its travelled path (mapping) [18, 19]. The limitations in terms of payload, power, and processing resources together with the necessity to deal with a 6-Degree-of-Freedom (6-DOF) dynamics, make the implementation of SLAM on board MAVs much more challenging than for marine or ground vehicles [20]. As regards the selection of the ranging sensor to be installed on the MAV, active systems (LIDAR, RADAR and ultrasonic rangefinders) are recommended over hybrid (RGB-depth cameras) and passive (monocular and stereo cameras) ones since they are less sensitive to ambient light variation and they can work at any time (day and night), thus providing an higher degree of autonomy. Moreover, they produce directly 3D representations of the scene without requiring any computationally expensive image processing step. Among active systems, the attention is focused on LIDAR. Indeed, they can get larger distance measurements than ultrasonic rangefinders and depth camera, and they are lighter, less power consuming and provide more understandable information than RADAR.

For these reasons, this part of the thesis has the purpose of providing innovative techniques for the two main steps of SLAM (localization and mapping), which are based on the integration of laser scan data provided by a two-dimensional (2D) LIDAR, and inertial measurements given by a low cost IMU, namely the vehicle's acceleration, angular velocity and attitude. Specifically, also in this case, the issues of the algorithms' real-time implementation, and of the necessity of ensuring robustness against failures are addressed. Algorithms' performance is evaluated through off-line simulations based on real data collected by exploiting an experimental setup purposely design to operate carried by hand within an indoor test scenario.

It is worth outlining that the work relevant to this area of research has been realized during a period of six months spent at Cranfield University (UK) in the framework of the "Program STAR - Linea 2" financially supported by *UniNA* and *Compagnia di San Paolo*.

## 1.3 Thesis organization

The present thesis is organized as follows.

Chapter 2 presents the concept of pose determination and the different existing approaches, as well as an overview about the state of the art of algorithms used in space applications. This is aimed at providing an overall classification of the existing techniques and at highlighting the current limitations. Then, the proposed pose determination architecture is described entering in the detail of each step of the process (i.e. acquisition tracking, refinement, transition from acquisition to tracking).

Chapter 3 illustrates the different parts composing the simulation environment by which the performance of the proposed approach for pose determination is evaluated: the LIDAR measurement simulator which includes geometric, detection and noise models; the relative dynamics generator which exploits the target-chaser trajectory provided by the safe relative motion design method presented in chapter 4; the criteria for target selection and modeling. An overview of LIDAR systems adopted in space application is also provided to justify the modeling choices.

Chapter 4 starts with the introduction of the concept of safe trajectories during close-proximity operations. Afterwards, the innovative analytical approach, which is adopted to generate the target-chaser trajectories for the numerical simulations, is presented in detail.

Chapter 5 contains all the results of the numerical simulations realized to evaluate the performance of the proposed pose determination algorithms, together with the definition of the criteria adopted for performance analysis. Also the effectiveness of the autonomous failure detection strategy is assessed.

Chapter 6 presents an overview of the state of the art in the field of SLAM using LIDAR systems. Then, the innovative techniques for localization and mapping, based on LIDAR and inertial data integration, are illustrated. Finally, the experimental setup and the indoor scenario for gathering data is described, and the results of the off-line implementation of the proposed algorithms are analyzed to determine their capabilities.

Finally, chapter 7 contains a discussion suitable for summarizing the achieved results and providing anticipations about the further developments of the research activity.

# Chapter 2 - Spacecraft pose determination in close-proximity

## 2.1 Introduction

The capability to perform autonomous relative navigation between two spacecraft relies on a large variety of technologies, i.e. Radio Frequency (RF), GNSS-based, and EO sensors. However, each of these categories of instruments has its own field of application in terms of inter-satellite distance as well as depending on whether the two spacecraft are cooperative or not.

If the inter-satellite distance is extremely large, i.e. from hundreds of kilometers to a few hundreds of meters, like during far-range and close-range rendezvous [21], there is no sensor capable of inferring the full relative attitude state (3 DOF). However, relative navigation can still be performed by relying on range and line-of-sight (LOS) estimates, or even exclusively on LOS measurements. For instance, RF methods can provide reliable range, range-rate and LOS measurements, as shown by several applications of the Russian Kurs system for space rendezvous [21], but the target is cooperative since several antennas must be installed at specific locations on both the satellites. Range and LOS measurements can also be given by the differential GPS approach [22], whose applicability has been demonstrated by the Spaceborne Autonomous Formation Flying Experiment (SAFE) during the PRISMA mission [23]. However, this still represents a cooperative concept since a GPS antenna and a GPS receiver must be installed on board the target, together with other antennas to ensure a reliable communication link. On the other hand, autonomous LOS tracking of an uncooperative target from about 30 km up to 3 km of distance has been demonstrated during the Advanced Rendezvous demonstration using GPS and Optical Navigation (ARGON) experiment, again in the frame of the PRISMA program, by using a simple camera and specific image processing algorithm [24].

Moreover, if a laser range finder is coupled with a passive camera, both range and LOS of an uncooperative spacecraft can be estimated at far range, as demonstrated in the frame of the Orbital Express (OE) mission [25].

In spite of these achievements, FF, OOS and ADR activities involve spacecraft relative dynamics to be controlled during operations like final approach, mating, monitoring, and station keeping, which are characterized by a narrower range of distances, e.g. from 100 m to even contact (in case docking is foreseen). Hence, they are classified as close-proximity operations and they require the full 6-DOF pose state to be estimated. In this context, although RF methods represent a potential solution to obtain the full relative attitude state in addition to relative position information with respect to a cooperative target [21], they require too complex, heavy and power consuming antennas and electronic equipment on both chaser and target vehicles. At the same time, techniques based on differential GPS do not provide an adequate accuracy level at very close range since they can suffer from multi-path effects as well as from shadowing of the navigation satellites by the target structure. Hence, EO sensors represent the best choice to fly in close-proximity also because their performance tends to improve as the range reduces. Of course, the specific instrument as well as the measurement principle, e.g. active or passive sensing, must be carefully selected depending on the application.

Spacecraft pose determination is the problem of computing the rigid rotation (3 DOF) and translation (3 DOF) aligning the reference frames related to the two satellites involved. In this respect, when trying to follow the evolution of the relative pose of a moving object by exploiting measurements from a given EO sensor, the general architecture of the pose determination process, shown in Figure 2.1, is composed of two main steps, namely acquisition and tracking.

Pose acquisition is performed when the first dataset is provided by the adopted sensor and no a-priori information about the target relative position and attitude is available. Pose tracking means updating the pose parameters, as soon as new datasets are acquired, by taking the knowledge of pose estimates from one or more previous time instants into account. Pose tracking can also be augmented (in terms of accuracy level and computational efficiency) by introducing an additional step known as pose refinement. It is worth outlining that the pose refinement step can be designed to operate either by receiving in input both the sensor data and the previous pose estimates [26] (like in Figure 2.1), or by relying only on this latter information [27-31] (like in Figure 2.2).

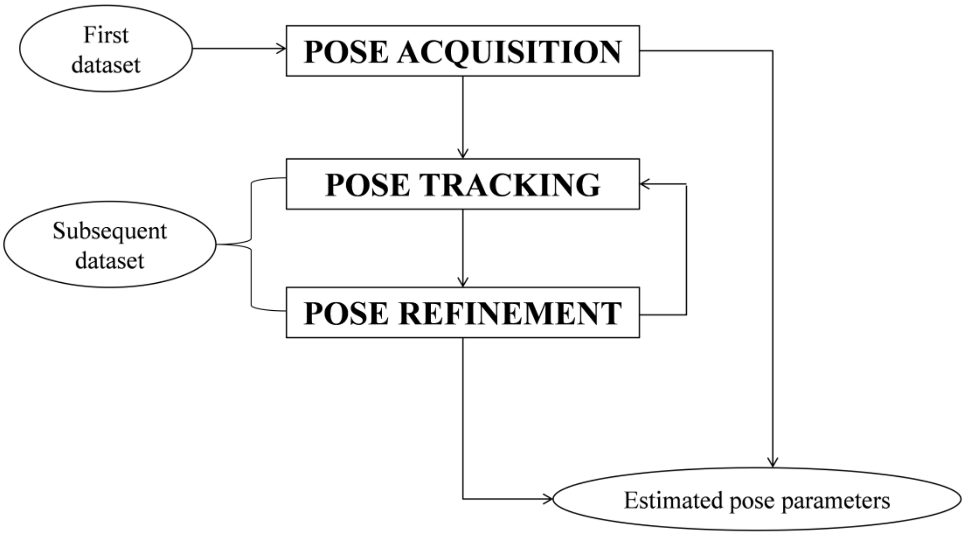


**Figure 2.1 - Logical scheme of the process for pose determination of a moving object. Pose refinement initializes the tracking step and updates the estimated pose state.**

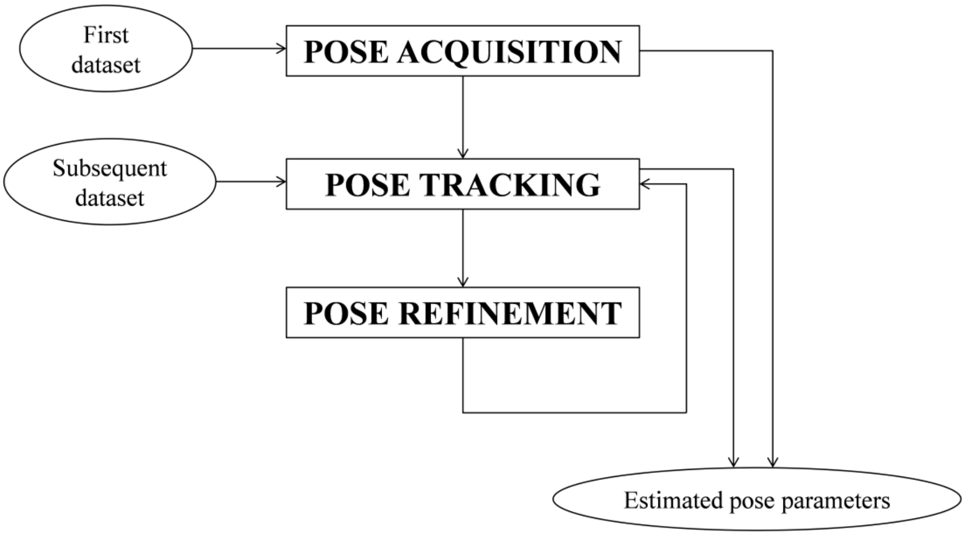


**Figure 2.2 - Logical scheme of the process for pose determination of a moving object. Pose refinement initializes the tracking step.**

In the context of spacecraft relative navigation, it can be clearly stated that the acquisition step is the most critical one of the process. Indeed, the estimated relative state can even be characterized by coarse accuracy, provided that it falls in the field of convergence of the tracking algorithm, but it must be obtained fast enough to avoid losing track of the moving target. Hence, performance analysis of pose acquisition algorithms is mainly focused on computational efficiency aspects and on reliability over extremely wide portions of the pose space. On the other hand, tracking algorithms search for convergence within a narrow area of the pose space thanks to the knowledge of previous estimates of the relative state. For this reason, they are typically quite fast and must be designed with particular attention to the accuracy level.

In the case of both cooperative and uncooperative targets, pose determination algorithms based on EO measurements can be classified in two categories.

- Monocular techniques rely on passive vision systems, like Charged Coupled Device (CCD) cameras, to estimate the target pose on the basis of angular measurements only.
- 3D techniques exploit active LIDAR or passive stereovision systems to obtain 3D information about the target, e.g. respectively point clouds and range images. Of course, this is pretty straightforward for LIDAR, while it involves complicated processing steps, e.g. image rectification and disparity map calculation, if stereocameras are used.

The former category is related to the problem of estimating the relative position and orientation of the camera with respect to the observed scene, which is known as extrinsic camera calibration [32]. Indeed, if the position and attitude of the target in the scene is a priori known, the target pose with respect to the camera can be derived. If *n* correspondences between real world and image points are found, the extrinsic camera calibration can be performed by solving the Perspective-*n*-Point (P*n*P) problem [33, 34], for which the following remarks must be made.

- If *n* is less than 3, the problem is under-constrained and it admits infinite solutions.

- If *n* is equal to 3, the problem is under-constrained, but, if the points are not collinear, the upper bound of ambiguous solutions is limited to four.
- If *n* is equal to 4 or 5 more than one real solutions exists. However, if 4 image-real world correspondences of coplanar points are available , the existence of one unique solution can be mathematically demonstrated [33].
- If *n* is larger than 5, the problem is linearly determined.

In the case of cooperative targets, this concept can be applied since they are typically equipped with active LEDs or passive CCRs mounted on their external surface (or on their docking interface) according to specific known patterns. Hence, image-real world point correspondences can be attained by exploiting the bright projections of these point targets on the focal plane of the camera. For instance, in the framework of the PRISMA space program, pose determination in close-proximity is accomplished by imaging the LEDs located on the target surface at assigned positions by using a monocular camera, i.e. the Vision Based Sensor (VBS) [35]. Specifically, an analytical solution to the P4P problem, which is based on the volume measurement of tetrahedra composed of point triplets [36], is used to determine the relative positions between two consecutive LEDs in the reference frame relative to the camera (observation unit vectors). Hence, being the same information also a-priori known in the reference frame relative to the target (reference unit vectors), the relative attitude can be derived by means of deterministic or stochastic approaches [37] and, as a consequence, the relative position is obtained by a simple vector combination.

If 3D sensors are used, the solution to pose determination of cooperative targets is even more straightforward since the relative positions between consecutive artificial markers can be directly extracted from the measurements without any complex processing, like it is done in [38] by acquiring image pairs from a stereovision system.

A significant limitation of cooperative techniques is that lack or miss detection of a marker may lead to system failure. Moreover, cooperative targets are peculiar of FF missions, while for most of OOS and ADR applications, the target satellite is uncooperative. Hence, the research activity presented in this thesis is focused on the issue of uncooperative pose determination.

## 2.2 Uncooperative pose determination: model-based algorithms

Uncooperative pose determination is generally handled by model-based algorithms, whose basic concept is to compare the data extracted from measurements of the selected EO sensor with similar information derived from a target model, typically stored on board. Of course, if the target satellite is damaged so that the available model does not correctly represent its actual geometry, this issue could be overcome by building the model directly on board [39].

Model-based algorithms can be ranked in two categories.

- Feature-based methods [40-56] are based on the extraction from the acquired dataset of geometric features, such as corners, lines, curves, and contours.
- Appearance-based methods [57-59] are based on the analysis of the shape and texture of the acquired datasets, which leads to the generation of the so-called appearance model.

In this survey, point-based techniques [53-56], which directly exploit raw data from 3D sensors, are ranked as feature-based since they rely on the same concepts with the advantage given by the absence of the feature extraction step.

For both feature-based and appearance-based approaches, different types of techniques can be used depending on their role within the pose determination process (acquisition or tracking), as well as on the typology of data available (monocular or 3D).

### 2.2.1 Feature-based methods

If feature-based methods are selected, a solution to the pose determination problem can be obtained either by means of iterative techniques (e.g. least-squares methods), which optimize a purposely defined objective function, or by exploiting the Template Matching (TM) algorithm. It is worth outlining that while the former approach is typically used for pose tracking, the latter one in most cases is suitable for both the steps of the pose determination process.

While the idea behind iterative techniques is quite straightforward, it is necessary to clarify the meaning of TM. This method derives from the concept of searching, within a 2D dataset (monocular image) or a 3D dataset (range image or point cloud), for specific features and/or specific image sections, which can be matched to an assigned template [60]. The template can have the same size as the available dataset, or it can occupy only a limited area of it, while the matching function is carried out by exploiting a correlation approach. Specifically, different kinds of correlation laws exist, among which the sum of absolute differences [61], the normalized cross correlation [62], and the distance transform [63] are most commonly used for monocular data, while mean square distance metrics [53, 54] and binary correlation [55] are suitable for 3D data. In the framework of pose determination tasks, TM requires the generation of a database of templates by sampling the 6-DOF pose space. Each template corresponds to a specific set of relative position and attitude parameters, and a correlation function is used to establish the degree of similarity between each template and the acquired dataset. Hence, the pose solution is given by the set of parameters related to the template for which the correlation function is optimized (i.e., maximized or minimized). This procedure is clarified by the flow diagram of Figure 2.3.

An example of TM algorithm applied to extract the initial target pose from monocular data can be found in [41]. Each template, indicated as prototype view, is generated by projecting the silhouette (i.e. the contour) of the target model onto the image according to an assigned set of pose parameters. Unlike other features, silhouettes can be robustly extracted independently of the characteristics of the target, which may be un-textured or slightly transparent. The peculiarity of this approach is that it aims at limiting the computationally expensive search in the pose space by building the database of images with a hierarchical structure in which similar prototype views are clustered at the lower levels of the hierarchy.

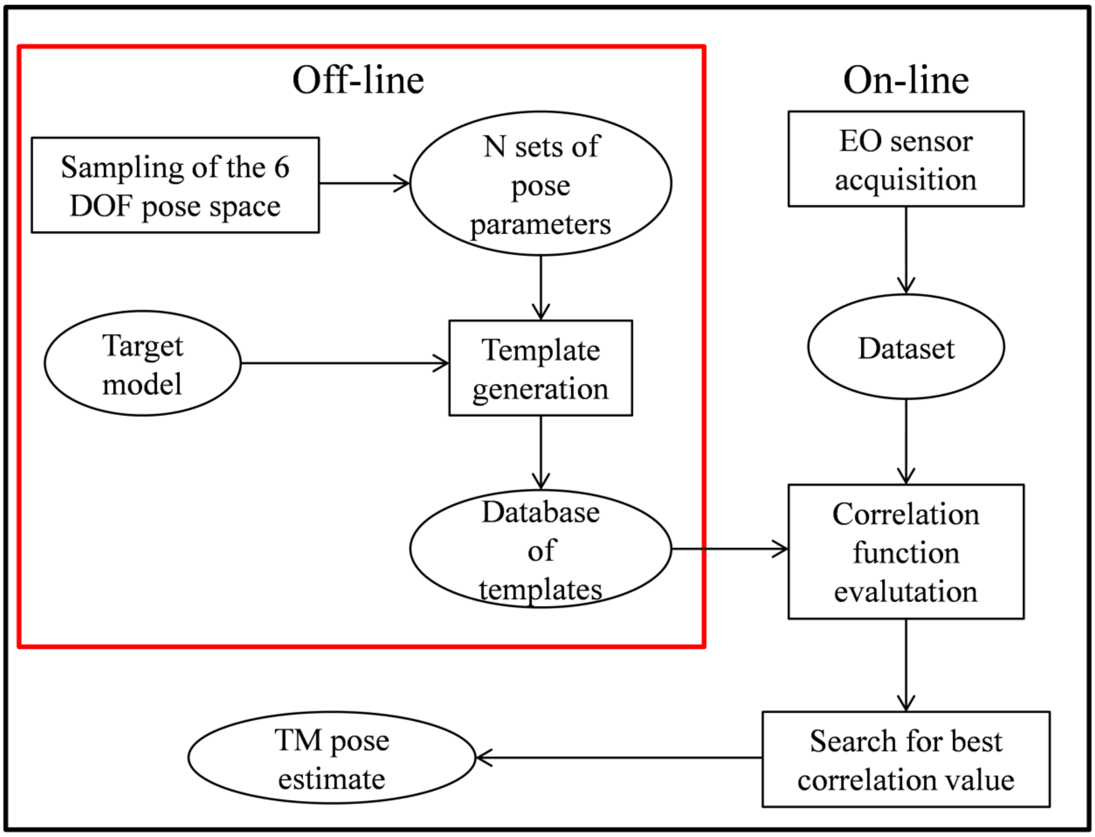

Off-line
On-line
Sampling of the 6 DOF pose space
N sets of pose parameters
Target model
Template generation
Database of templates
EO sensor acquisition
Dataset
Correlation function evalutation
Search for best correlation value
TM pose estimate

**Figure 2.3 - Logical scheme of the classical application of the TM concept to pose determination tasks.**

As regards TM approaches based on 3D data, many algorithms [53-56] exist which are specifically tailored to the pose determination of uncooperative spacecraft flying in close-proximity. These techniques do not require an initial guess and foresee specific solutions to improve computational efficiency and data storage issues connected to the basic TM concept (see sub-chapter 2.2.4 for more detail).

Another example of TM approach operating on 3D datasets but capable of performing only pose tracking, is the Bounded Hough Transform (BHT) method [42]. In this case, the model is a set of voxels, indicated as voxel occupancy. Voxels are the basic elements in 3D images. They are volumetric small regions of the space filled with binary values to indicate if these regions are empty or occupied. The templates are 3D binary arrays obtained by applying the corresponding pose transformations to the voxel occupancy. Classical pose tracking algorithms search for the pose solution in a continuous space closely to the initialization. On the other hand, the BHT method restrains the search to a discrete space obtained by sampling the continuous one around the available initial relative state. Moreover, the correlation between the acquired datasets and the template is computed by exploiting the classification concept which is inherited from the Hough Transform (HT) [64].

Moving on to the iterative (non-linear) techniques, they generally foresee the following steps if monocular sensors are used.

- The model is represented as a sparse set of features like points (e.g. raw data, or extracted image descriptors like corners), lines (e.g. edges) or curves (e.g. circles and ellipses).
- The same features are extracted in the acquired image.
- The model is projected onto the image according to the initial pose solution.
- The actual pose is obtained as the transformation that provides the best fit (alignment) between projected and image features

The latter step is typically performed by minimizing a squared metric function. A point-based approach relying on the Scale Invariant Feature Transform (SIFT) can be found in [43], edge features are used in [44, 45], while ellipses are extracted in [46].

An iterative but linear solution to the problems of searching for model to image correspondences and pose estimation is provided by the SoftPOSIT algorithm [47]. This technique is composed of two steps. Firstly the soft-assign [49, 50] algorithm determines correspondences between image and model points/lines. Afterwards, the POSIT (POS with Iterations) algorithm [51] iteratively estimates the pose.

Specifically, it uses the POS (Pose from Orthography and Scaling) algorithm to approximate the perspective projection with a scaled orthographic projection, thus finding a coarse estimate of the relative state by solving a linear system. Hence, the approximated pose is used to compute better scaled orthographic projections of the feature points, and the process is iterated until convergence. In [48], the SoftPOSIT applicability was extended to the case in which lines are detected in the image instead of points; this is important since lines are typically more stable than points, i.e. more invariant to changing in lightning conditions and are less likely to be produced by clutter and noise. Although the POSIT approach can provide an initial estimate of the pose, this is typically too coarse, thus potentially leading to failures or slow convergence. Hence, the SoftPOSIT algorithm is used exclusively for the pose tracking step of the pose determination process.

If 3D sensors are used, feature-based algorithms for pose determination are typically point-based since the raw data can be used without needing any complex and time consuming detection method. The collected dataset, expressed in the sensor reference frame, is rotated and translated on the basis of the initial pose. Hence, similarly to monocular approaches, the actual pose is obtained minimizing an error function defined by a comparison with a point-based representation of the target model. The best example of this approach is given by the Iterative Closest Point (ICP) algorithm [52]. It is an iterative technique able to find the best rigid transformation to align two datasets (registration) by minimizing a cost function which measures the similarity level between corresponding elements. Although the ICP concept is proposed to register any kind of sets of measurements, which can be composed of points, lines, or planes, it is mostly used to align a measured point cloud to a model point cloud for pose determination tasks.

### 2.2.2 Appearance-based methods

In order to avoid the necessity of complex feature extraction steps, an alternative solution for pose determination consists in applying appearance-based approaches.

They are specifically tailored for pose estimation or recognition of objects having complex articulated shapes, e.g. human hands or faces, when dealing with 2D data (no LIDAR). A distinction can be made between Active Appearance Models (AAM) methods [57] as well as solutions based on the Principle Component Analysis (PCA) [65].

The original AAM method states that the appearance of an object can be defined by its shape, i.e. set of 2D landmark points of the object image, as well as by its texture, i.e. set of intensity values of the pixels lying inside the shape. Given this definition the pose estimation problem is solved by finding the best transformation that fits the observed appearance (current image) and the expected appearance (model).

Mittrapiyanuruk et al. [58] have introduced and tested in laboratory two AAM algorithms able to track the pose of a moving target respectively from images generated by a stereovision system (3D technique) and a single camera (monocular technique). Both these techniques need an initial guess of the pose thus not being suitable for pose acquisition.

The PCA is a technique used to analyze multidimensional datasets. Specifically, it aims at deriving their principal directions, i.e. the main related information, by studying eigenvectors and eigenvalues of the associated covariance matrix. For this reason, PCA-based methods for pose determination are generally known as eigenspace approaches. Instead of considering every possible view of the target (like it is done by TM-based techniques), only a limited number of views, forming a basis in the eigenspace, are considered. The pose of the object at the time of interest is computed by representing the actual image collected by the sensor as a linear combination of the reference images forming the eigenspace. An example of monocular PCA-based approach is presented in [59]. Also PCA-based methods can be used for pose tracking if a reasonable initial guess of the pose is available.

Both AAM and PCA-based methods require a preprocessing stage performed on a series of sample images. In the former case, it is needed to build the appearance model, while in the latter one, it is used to create the eigenspace and to perform segmentation of the object from the image background. Another difference is given from the fact that PCA involves the minimization of a non-linear error function, while the algorithms based on AAM are linear, though iterative.

### 2.2.3 Comparison and synthesis

A compact survey of model-based techniques suitable for determining the pose of an uncooperative target with respect to an observer, by using EO sensors, is provided in Table 2.1.

| **Model-based technique** | **Methodology** | **Data type** | **Acquisition** | **Tracking** | **References** |
|---|---|---|---|---|---|
| **Feature-based** | TM-based | 2D | YES | YES | [41] |
| | | 3D | YES | YES | [53-56] |
| | BHT | 3D | NO | YES | [42] |
| | Iterative non-linear | 2D | NO | YES | [43-46] |
| | | 3D | NO | YES | [52] |
| | Iterative linear (softPOSIT) | 2D | YES | YES | [47-51] |
| **Appearance-based** | AAM | 2D | NO | YES | [57, 58] |
| | | 3D | NO | YES | [57, 58] |
| | PCA-based | 2D | NO | YES | [59] |

**Table 2.1 - Survey of model-based techniques for uncooperative pose determination.**

According to this survey, only TM-based methods and the softPOSIT algorithm are suitable for both acquisition and tracking. The former ones are typically not fast enough for tracking due to the necessity to search in the entire 6-DOF pose space. An exception to this statement is given by the BHT, being a TM approach specifically tailored for tracking applications. On the other hand, the initial solution provided by softPOSIT can be too coarse to ensure safe transition from acquisition to tracking.

Feature-based methods represent a more convenient solution to uncooperative pose determination with respect to appearance-based ones. Indeed, they are more reliable to large variations of the pose which can occur in space and can cause lost of validity of the appearance model. Also, they can provide solutions to both the acquisition and tracking issues, and they are applicable independently of the EO technology adopted (while appearance-based techniques are intrinsically related to the processing of single camera images also in the case of the stereo approach in [58]).

Appearance-based methods have the advantage of not requiring feature detection steps, thus being particularly convenient in cluttered background. However, this is not an issue in space applications where in most cases, segmentation of the target with respect to the background is extremely easy.

This explains why feature-based algorithms are selected for spacecraft pose determination in close proximity, as it is shown in the next sub-chapter.

### 2.2.4 Space applications

This sub-chapter contains a survey of model-based techniques conceived for pose determination of non-cooperative spacecraft by processing measurements of EO sensors. Their performance has been assessed by means of numerical simulations, off-line runs on data gathered during previous space missions, off-line or real-time runs on data gathered by means of experimental test-beds.

Firstly, monocular approaches, both for acquisition and tracking, are addressed.

Full monocular-based pose determination architecture is proposed by Astrium Satellites in the framework of a program for optimization and implementation of sensors and navigation solutions onboard a debris-removal vehicle named "The Debritor" with the main objective to ensure high safety proximity maneuvers [66]. Specifically, pose acquisition is performed by applying a customized version of the silhouette TM algorithm [41]. Indeed, modifications are necessary to be compliant to the specific features of TM approaches (valid for both monocular and 3D algorithms), when they are adopted for spacecraft pose determination. Specifically, since the database is generated during a preprocessing learning stage carried out off-line, it must be stored on board the chaser. Hence, two issues arise, which are indicated hereunder.

- Necessity to limit the computational cost related to the search in the 6-DOF database to avoid losing track of the target.
- Necessity to restrain the amount of on-board data storage.

In [66], these issues are dealt with by building a 2-DOF database (the view points are sampled on a sphere as it can be seen in Figure 2.4) with a hierarchical structure. Once an edge-based matching stage is completed, the remaining unknowns are obtained through the segmentation of the silhouette of the object. This strategy accelerates the algorithm's convergence but it is spread over multiple frames, thus increasing the risk of losing track of the target. Pose tracking is performed by applying the edge-tracking method in [45] modified to improve robustness with respect to outliers. Algorithms' performance is evaluated on synthetic images of debris, i.e. an Ariane 4 upper stage and a Spot family satellite, and real images of the Soyuz TMA-12 spacecraft taken during its rendezvous phase with the International Space Station (ISS). The former ones have been simulated using an Astrium rendering tool called "Surrender" (based on classical rendering functions, adapted to peculiarity of space environment and debris properties), coupled with a simplified space dynamics simulator to reproduce trajectories and satellite dynamics. Range and tumbling rate of the simulated targets have been set respectively variable from 10 m to 100 m and from 0°/s to 2°/s, in order to be consistent with the typical requirements of the final phase of an ADR mission. The algorithm provided good results averaged over several tests. Specifically, the attitude error was around 1°, while the position error varied approximately from 50 cm to 5 cm as the range decreased from 50 m to 5 m. Additional results can be found in [67, 68]. As regards the computational load, the authors claim the algorithms to be still at a prototype level since they are able to run in real time but only on ground.

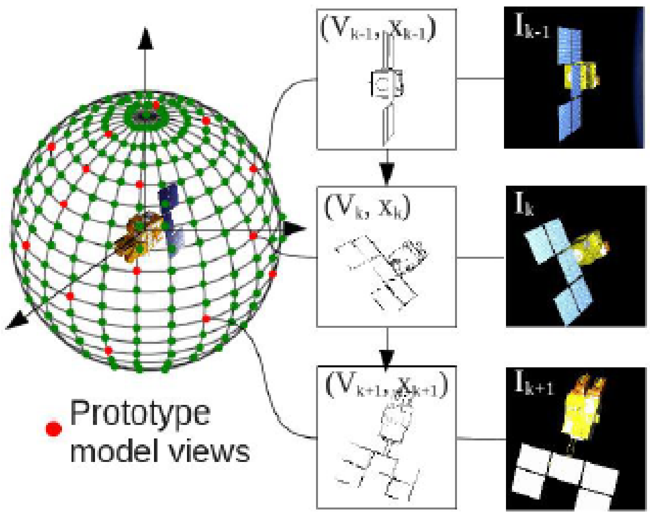


**Figure 2.4 - Template (prototype model view) generation by sampling a 2-DOF relative attitude space (spherical coordinates) for silhouette TM [66].**

Another approach for spaceborne pose determination of non-cooperative targets based on monocular images is proposed in the framework of a program from the German Space Agency (DLR) [69]. Acquisition is accomplished by exploiting a TM method based on the concept of perceptual grouping [70]. Perceptual groups are combinations of lines and points to be extracted from both the model and the acquired images and then to be matched. These entities are more robust descriptors than edges or corners, thus ensuring a more reliable matching process. In this case, the database is given by all the possible perceptual groups extracted from the model, and the initial pose solution is given by the viewpoint which gives the best alignment between model and image perceptual groups. On the other hand, pose refinement and tracking are performed by applying, respectively, a multi-dimensional Newton-

Raphson method and a weighted iterative batch least-squares estimator with a-priori information. Both these techniques try to find the best alignment between corresponding perceptual groups, which are re-matched after the initialization. This pose determination architecture is summarized by the flow diagram in Figure 2.5.

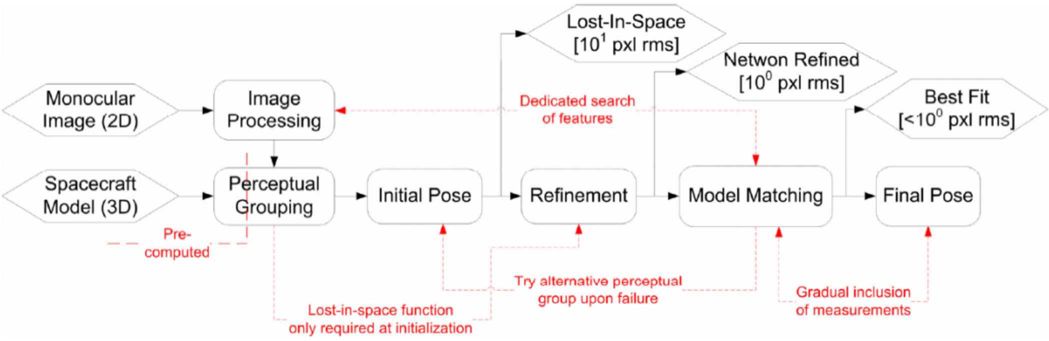

Monocular Image (2D)
Spacecraft Model (3D)
Image Processing
Perceptual Grouping
Pre-computed
Initial Pose
Lost-in-space function only required at initialization
Lost-In-Space [$10^1$ pxl rms]
Dedicated search of features
Refinement
Netwon Refined [$10^0$ pxl rms]
Try alternative perceptual group upon failure
Model Matching
Best Fit [<$10^0$ pxl rms]
Gradual inclusion of measurements
Final Pose

**Figure 2.5 - Spaceborne monocular-based pose determination architecture [69].**

The performance of this approach is evaluated by runs over real images collected during the ARGON experiment of the PRISMA mission [24]. The authors claim the tracking algorithm to be reliable enough (accuracy of 10 cm and 10° in position and attitude, respectively) provided that the initial angular error is kept below 40°. Moreover, they state the major limiting factor for accuracy and reliability to be in the image processing module, which is now based on the HT.

Liu and Hu addressed the problem of finding the pose of non-cooperative cylinder-shaped spacecraft by processing images from a single camera [46]. A coarse estimate of the relative orientation of the symmetry axis and of the relative translation vector (4 DOF) is obtained by exploiting projective geometry, after having matched ellipses extracted from the acquired images to the ones in the model [71]. Hence, the solution is refined by optimizing a non-linear objective function. The full pose (6 DOF) is then computed by exploiting information provided by the non-symmetric components of the spacecraft, e.g. antennas and solar arrays. This method does not need any initial guess. The accuracy of this approach is assessed by exploiting real space images of the Soyuz as well as synthetic images of the same target generated by the Satellite Tool-Kit (STK). A significant issue is given by the worsening of the relative position accuracy as the rotation of the cylinder with respect to its transversal axes increases. Moreover, the method would probably suffer from partial surface occlusion which may deny ellipses to be detected.

Additional spaceborne feature-based techniques are specifically tailored to pose tracking [26, 72, 73], thus requiring an initial pose estimate.

Kelsey et al., from the Scientific Systems Company Inc. (SSCI), have designed both the HW and SW components of a Vision System for Autonomous Rendezvous and Docking (VISARD) for OOS applications [26]. Given a coarse pose initialization and images from a single camera, the system is able to perform pose refinement by exploiting the edge-tracking approach from [44], and pose tracking, by using an EKF. Performance analysis of VISARD algorithms is provided by processing images obtained from an experimental setup which allows reproducing various scenarios, e.g. space-based rendezvous and proximity operations, aircraft mid-air refueling, and ground-based visual servoing. Scaled replicas of existing satellites are used as target objects, namely a 1/20 scale Delta II second-stage rocket body model, a 1/72 scale Soyuz satellite model, and a Magellan 1/24 scale model. Example of VISARD images are shown in Figure 2.6.

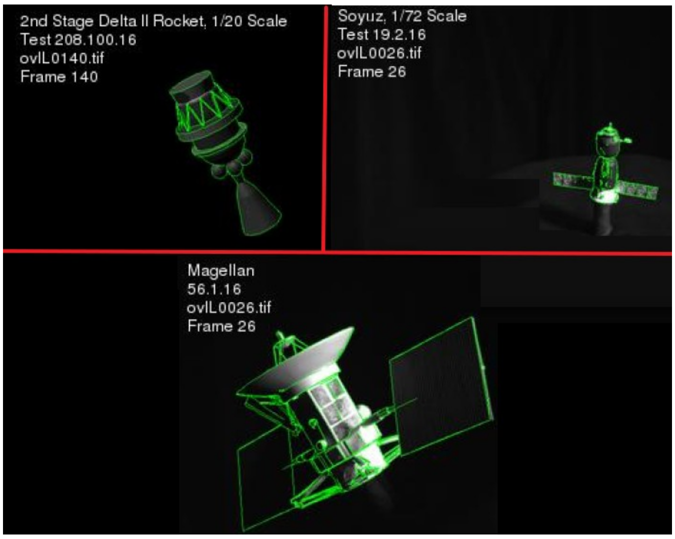


**Figure 2.6 - Scaled satellite models used for VISARD experiments.**

The pose refinement algorithm is stated to be able to handle errors in the initial pose parameters up to 30° in attitude and 10% of the range in position. However, algorithm performance significantly reduces as the initialization error increases, i.e. the standard deviation of the angular error is 1.33° for initial errors below 2° and it becomes 10.22° if the initial error is below 30°.

Cropp et al. estimate the pose of an uncooperative target by matching a pre-generated 3D line model with lines detected in the image [72]. At first, image lines are detected with sub-pixel accuracy thanks to the HT. Then, heuristics are used to generate a list of correspondences between image and model lines. This is aimed at reducing the computational cost since otherwise all the possible matches would have to be processed. However this represents also a significant limitation of the algorithm, as these heuristics are strictly related to the geometry of the considered target. Once the correspondences have been found, two least-squares error functions, which satisfy specific geometric constraint [74], are minimized to get the relative attitude and position. A method based on Random Sample Consensus (RANSAC) [33] is used to improve algorithm robustness against incorrect matches. The relative pose is computed several times, considering different subsets of line correspondences, within the initial global set. The quality of each estimated pose is computed by projecting the model on the image plane and evaluating the differences in position and orientation between the corresponding lines. Hence the best-fit pose parameters are determined and a Gauss-Newton minimization is performed to improve the solution. A series of numerical simulations are realized, considering UoSAT (a microsatellite from Surrey Space Center) as target. They show the capability of the algorithm to attain an average rotation error less than 5° and an average translation error of about 2% of the target range (maximum 10 m). However, the authors state that the accuracy could be improved by applying filtering techniques exploiting information from multiple successive frames.

Another pose tracking algorithm [73] is proposed by a Spanish company (GMV Aerospace and Defense) in the framework of the COBRA IRIDES experiment which aims at modifying the attitude motion of a non-cooperative satellite by means of the interaction between the thruster exhaust gases and the target itself [75]. Specifically, a modified version of the edge tracking approach provided in [44] is adopted. It envisages an original solution to extract from the model only salient features, i.e. silhouette, creases or boundaries which are candidates to match image edges, as well as to remove hidden lines. In order to test algorithm performance, real images from the PRISMA and Picard missions are taken, and the pose is initialized by manually matching model and image corners as shown in Figure 2.7. The analysis of the results shows that the algorithm provides a tracking error of a few degrees and tens of cm in the relative attitude and position, respectively.

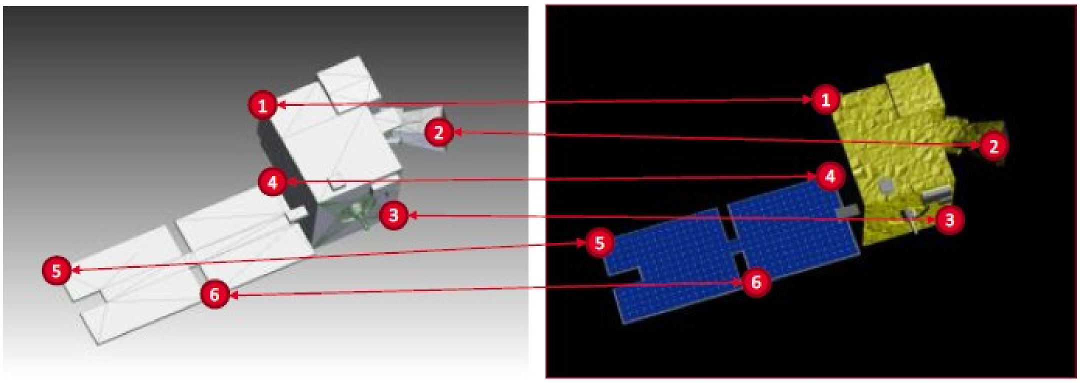


**Figure 2.7 - Manual pose acquisition based on model-image point correspondences, in the case of the Picard satellite [73].**

Moving on to 3D techniques, several pose determination architectures [56, 76, 77] have been proposed, capable of relying on point clouds measured by active LIDAR and/or passive stereo vision system.

Neptec has developed a vision system for autonomous on-orbit rendezvous and docking that does not require the use of cooperative markers on the target spacecraft. The system uses efficient model-based algorithms [56], developed in collaboration with the Canadian Space Agency (CSA), to provide 6-DOF relative pose information in real time by processing point clouds from the active TriDAR 3D sensor [78]. Pose acquisition is performed by an innovative object localization algorithm, namely the polygonal aspect hashing (PAH). It is a point-based technique which allows accelerating the search for the pose solution within a 6-DOF database built off-line, by restraining it to those sets of parameters for which the target model surfaces are aligned to one or more polygons (composed of about 4/6 points) extracted from the input data, as shown in Figure 2.8. Given the point-correspondences for each polygon, a pose solution is computed. However, the algorithm outputs the one which minimizes an error function. The process can be repeated with more polygons from the same input point cloud or from newly acquired point clouds over time in order to improve accuracy. Pose tracking, instead, is performed by means of a customized version of the ICP algorithm.

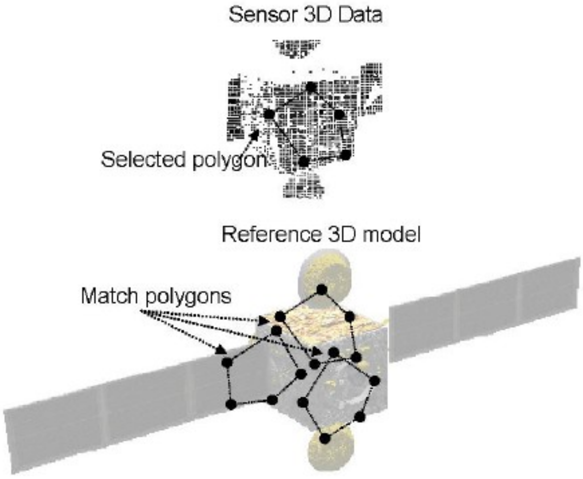


**Figure 2.8 - Pose acquisition by polygon matching [56].**

Algorithms' performance is evaluated by means of off-line runs over simulated and real sensor data of a 1/2 scale model of a Pressurized Mating Adapter (PMA) which is connected to a lateral port of a Node of the ISS. Specifically, the robustness of the PAH technique against reduction in the size of the point cloud, occlusion and spatial resolution is tested. The algorithm is considered successful if the errors remain 20 cm and 10° in position and attitude respectively. High values of the success rate are obtained if the number of matched polygons is large enough (more than 4) despite the reduction in point cloud size and sensor resolution (due to the increase in the relative distance), while the effect of occlusions starts being relevant causing a drop in performance when the target visibility goes below 50%. As regards the ICP algorithm, it provides accuracies below 1 cm and 1° in relative position and attitude, respectively [79].

Jasiobedki et al. designed a vision system [76] that was proven capable of determining the pose of a known satellite for operations at medium and short range by using model-based 3D algorithms able to process data from a terrestrial scanning LIDAR, i.e. the Optech ILRIS-3D [80], or two stereo cameras [81]. As regards pose acquisition, a TM approach is presented which addresses the issue of computational efficiency by exploiting the idea of splitting the computation of the pose parameters in two phases, like it is done in [66]. Specifically, the PCA is applied in order to find the orientation of the target main axis. Hence, a 3D binary TM algorithm [55] is used to look in a 4-DOF database for the best estimate of the remaining rotation and the relative position parameters. It is clear that a major limitation of this approach is that it is specifically tailored for objects having an elongated shape. Also in this case, pose tracking is performed by means of a customized version of the ICP algorithm. Algorithms' performance is evaluated within a testbed developed at MacDonald, Dettwiler and Associates Ltd. (MDA). The experimental setup, depicted in Figure 2.9, includes two industrial robots, one holding the instrument, the other holding an exact 1/5 scaled replica of Radarsat-2.

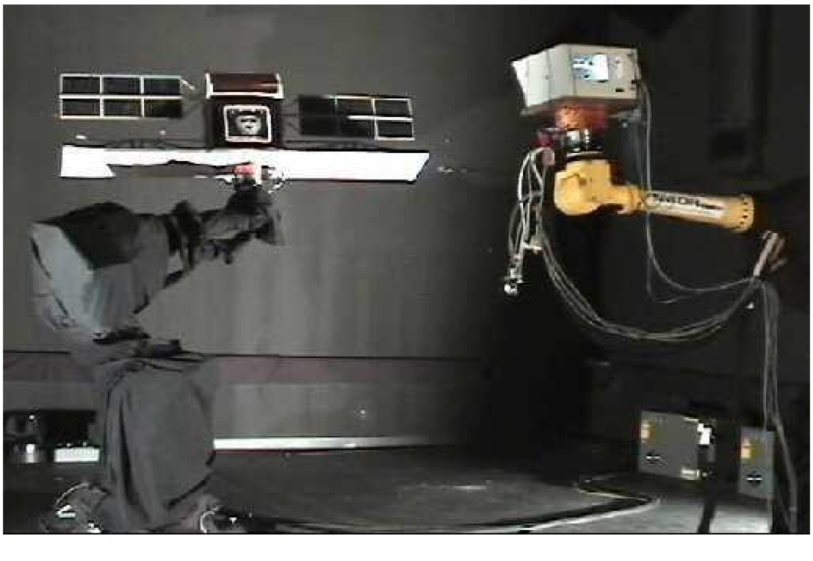


**Figure 2.9 - MDA experimental setup [76] to test uncooperative pose determination techniques.**

The two robots can follow predefined trajectories so that not only the acquisition but also the tracking operational mode can be tested. The illumination system can simulate direct sun light and Earth albedo, and actual space surface materials are used to create realistic effects. Algorithms' performance depends on the sparseness of the point cloud. However, an asymptotic behavior is reached when the number of points goes above 2000. Specifically, pose acquisition shows sub-degree accuracy in relative attitude and an accuracy between 3 cm to 5 cm in relative position. On the other hand, tracking accuracy is below 1 cm and 0.4° in position and attitude, respectively, independently of data sparseness.

In [42], performance of several variant of the BHT methods is compared to the ICP algorithm in terms of computational load and accuracy level for pose tracking. Also a hybrid techniques which applies the BHT and then refines its solution by means of the ICP algorithm is tested. This is done by using both simulated LIDAR data, obtained thanks to the Virtual Reality Modeling Language (VRML), as well as real data gathered by the same experimental setup as in [76]. Results, obtained using the Radarsat satellite as well as freeform objects (e.g. a duck, a dinosaur, a molecule model) as targets, show that the hybrid method provides the best performance. Similar values of the accuracy level and computational load are obtained independently of the target shape

Sommer and Ahrns [77] developed a conceptual Guidance Navigation and Control (GNC) system layout for rendezvous operations toward a non-cooperative but known space vehicle. An active LIDAR system is considered as the main relative navigation sensor, whose measurements are processed by 3D model-based techniques for pose determination. Pose acquisition is performed by applying a TM approach which exploits small range images (30x30 points) to improve the computational efficiency, while pose tracking applies the ICP algorithm. Performance is tested within a numerical simulation environment which reproduces LIDAR operation and target-chaser relative dynamics for a linearly approaching trajectory. The selected target is ENVISAT, being a perfect example of large debris in low Earth orbit. The TM approach provides an attitude error of about 5°. The tracking algorithm ensures an accuracy of few centimeters for the position and less than 1° for the attitude while the relative range varies from 50 m to 10 m. Below 10 m, the performance significantly worsens due to the reduction of the target visibility.

In addition to purely monocular and 3D techniques, an hybrid approach to pose tracking of uncooperative spacecraft, i.e. characterized by the simultaneous use of both passive and active technologies, has been conceived and tested [82]. Two different sensors are used. A Photonic Mixer Device (PMD), i.e. a 3D Time-of-Flight (TOF) camera, is used to collect range images which are processed to measure the target distance in the sensor boresight direction and the relative rotation with respect to this direction. The remaining 3 DOF, i.e. the cross-boresight component of the relative position vector and the relative rotation around the boresight axis, are obtained by processing monocular images from an high-resolution grayscale camera. The proposed algorithm compares the acquired ranges and the monocular images to a target model represented as a set of plane and outer lines. Hence, the camera and the PMD respectively perform edge-tracking and plane-tracking. This technique has been extensively tested, within the DLR facility called EPOS (European Proximity Operations Simulator) shown in Figure 2.10, using a scaled mock-up of the rear part of a Geostationary Earth Orbit (GEO) satellite as target.

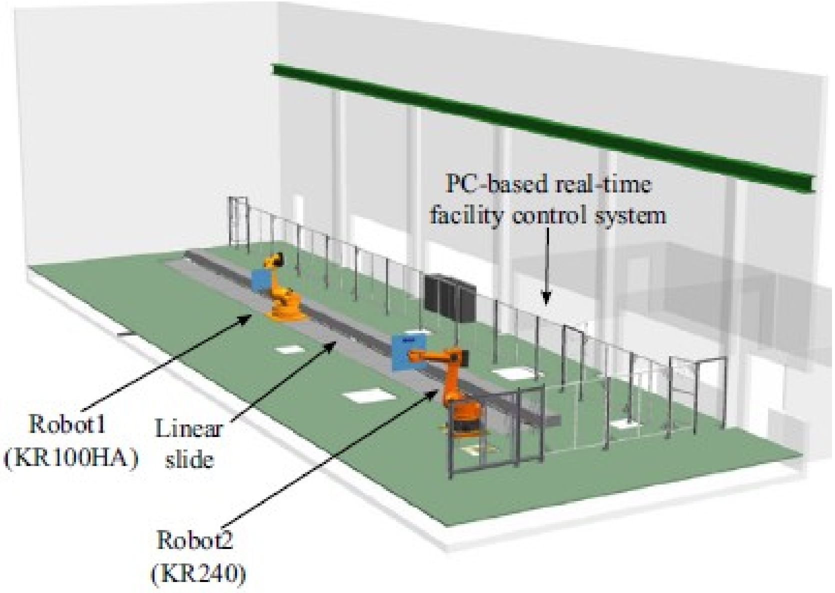


**Figure 2.10 - EPOS facility at DLR [82].**

An approaching maneuver is simulated with relative distance varying from 22 m to 4 m. If no artificial disturbance is applied to the relative trajectory, the algorithm provides sub-cm and sub-degree accuracy in relative position and attitude. If sine disturbances are introduced, the accuracy level reduces for any estimated DOF. However the performance worsening is more relevant to the DOF computed by means of the PMD, potentially due to its limited resolution.

This survey is summarized in Table 2.2. Globally, both active and passive technologies give promising and comparable results in terms of pose estimation accuracy. Passive sensors are lighter, less expensive and power consuming, and provide larger frame rates than active ones. However, LIDAR have fundamental advantages in terms of

- capability to discriminate target from background (segmentation);
- level of autonomy.

Indeed, segmentation can be negatively affected by the presence of the Earth in their Field of View (FOV) of passive sensors, which are also more sensitive to the variability of the illumination conditions. This latter aspect can prevent the extraction of robust features. During space operations without ground-control, the necessity to increase the autonomy level as much as possible is certainly a driving factor. Moreover, recent developments in innovative detectors (e.g. the Avalanche Photo Diode, APD), compact scanner systems (e.g. the Micro-Opto-Electro- Mechanical Systems, MOEMs), and high-power and short-pulse laser sources have pushed LIDAR applications in space. Hence, this thesis deals with the development and the performance assessment of 3D model-based algorithms by focusing on LIDAR, although the proposed concepts are easily extendable to the case of passive stereo-vision systems. It is also important to state that active systems typically have low resolution thus providing sparse dataset. On one side, this allows significantly reducing the computational load, which is further limited by the fact that most 3D techniques are point-based thus relying on direct processing of raw data. On the other side, this could limit the algorithms' accuracy level. Hence, it is essential to demonstrate capability of the proposed techniques to provide performance comparable to state-of-the-art approaches while dealing with highly sparse point clouds, occluded target images, and highly variable target shape.

| Experiment owner | EO sensor | Pose determination | Performance analysis criterion | Target |
|---|---|---|---|---|
| Astrium France | CCD camera | Acquisition and tracking<br>Silhouette TM [41]<br>Edge-tracking [45] | Off-line runs on synthetic and real images from past missions | Ariane 5 upper stage<br>Spot family satellite<br>Soyuz TMA-12 |
| DLR | CCD camera | Acquisition and tracking<br>Perceptual group matching [70]<br>Edge-tracking [69] | Off-line runs on real images from past missions | TANGO (target of PRISMA mission) |
| Beihang University | CCD camera | Acquisition and tracking<br>Ellipse matching and iterative refinement [46] | Off-line runs on real images from past missions | Soyuz |
| SSCI | CCD camera | Tracking<br>Edge-tracking [44] | Real-time runs on real images from experimental setup | Delta II second-stage<br>Soyuz<br>Magellan |
| University of Surrey | CCD camera | Tracking<br>Line-to-line matching [72] | Off-line runs on simulated images | Uosat microsatellite |
| GMV | CCD camera | Tracking<br>Edge-tracking [44] | Off-line runs on real images from past missions | TANGO<br>Picard |
| Neptec | LIDAR | Acquisition and tracking<br>PAH [56]<br>ICP [52] | Off-line runs on real point clouds from past missions | PMA |
| MDA Robotics | LIDAR<br>Stereo-camera | Acquisition and tracking<br>3D binary TM [55]<br>ICP [52] and BHT [42] | Real-time runs on real images from experimental setup | Radarsat 2<br>Freeform objects |
| Astrium Germany | LIDAR | Acquisition and tracking<br>TM [77]<br>ICP [52] | Off-line runs on simulated images | ENVISAT |
| DLR | PMD<br>CCD camera | Tracking<br>Hybrid camera-LIDAR approach [82] | Real-time runs on real images from experimental setup | GEO satellite mock-up |

**Table 2.2 - Survey of model-based techniques for pose determination of uncooperative space targets.**

## 2.3 Pose determination architecture

Hereinafter, the model-based algorithms developed to estimate the relative attitude and position of a servicing spacecraft with respect to a non-cooperative target satellite during close-proximity maneuvers are presented. The algorithms are 3D, meaning that they are designed to operate on 3D point clouds which can be provided by either an active LIDAR (directly) or a passive stereovision system (through stereo processing).

To this aim, the following rules are adopted concerning the mathematical notation: italic type is used for scalar quantities and quaternions, italic type with a single underline is used for other vectors, and italic type with double underline is used for matrixes.

For the sake of mathematical simplicity but without losing generality of the exposition, the proposed algorithms estimate the relative position and attitude of a target reference frame (TRF) with respect to a sensor reference frame (SRF) which is relative to the EO system installed on board the chaser. Indeed, the rigid rotation and translation between the SRF and a chaser reference frame is a-priori known by construction. The relative pose vector ($\underline{p}$) is composed of 6 parameters: $\underline{T}$ is the 3D relative position vector of the chaser with respect to the target and expressed in SRF, while the rotation matrix from TRF to SRF ($\underline{\underline{R}}$) is derived by a 321 sequence of Euler angles (i.e., yaw, $\gamma$, pitch, $\beta$, and roll, $\alpha$) or equivalently by the unit quaternion, $q$.

The logical scheme of the adopted pose determination architecture is represented in Figure 2.11. The pose acquisition block receives the first acquired dataset in input and estimates the initial pose vector ($\underline{p}_0$) without requiring any a-priori solution. Different techniques are proposed, which are described in detail in sub-chapter 2.4. Then, $\underline{p}_0$ is adopted to initialize the pose tracking block which estimates the time evolution of the pose parameters by processing new measurements. For the tracking step, different versions of the same ICP-based algorithm are described in sub-chapter 2.5. This architecture follows the concept expressed in Figure 2.2, meaning that pose tracking is augmented by a pose refinement step aimed at improving the accuracy of the tracking initialization only on the basis of previous estimates. However, this architecture is improved with respect to the solutions which can be found in the literature, as it includes strategies to enhance algorithms' robustness toward possible failures. Indeed, on one side, an additional block is introduced to manage the safe transition from acquisition to tracking by solving any possible ambiguity in the pose estimation process (see sub-chapter 2.6). On the other side, the pose solutions provided by both acquisition (after the transition step) and tracking algorithms are subject to a supplementary step foreseen for autonomous failure detection.

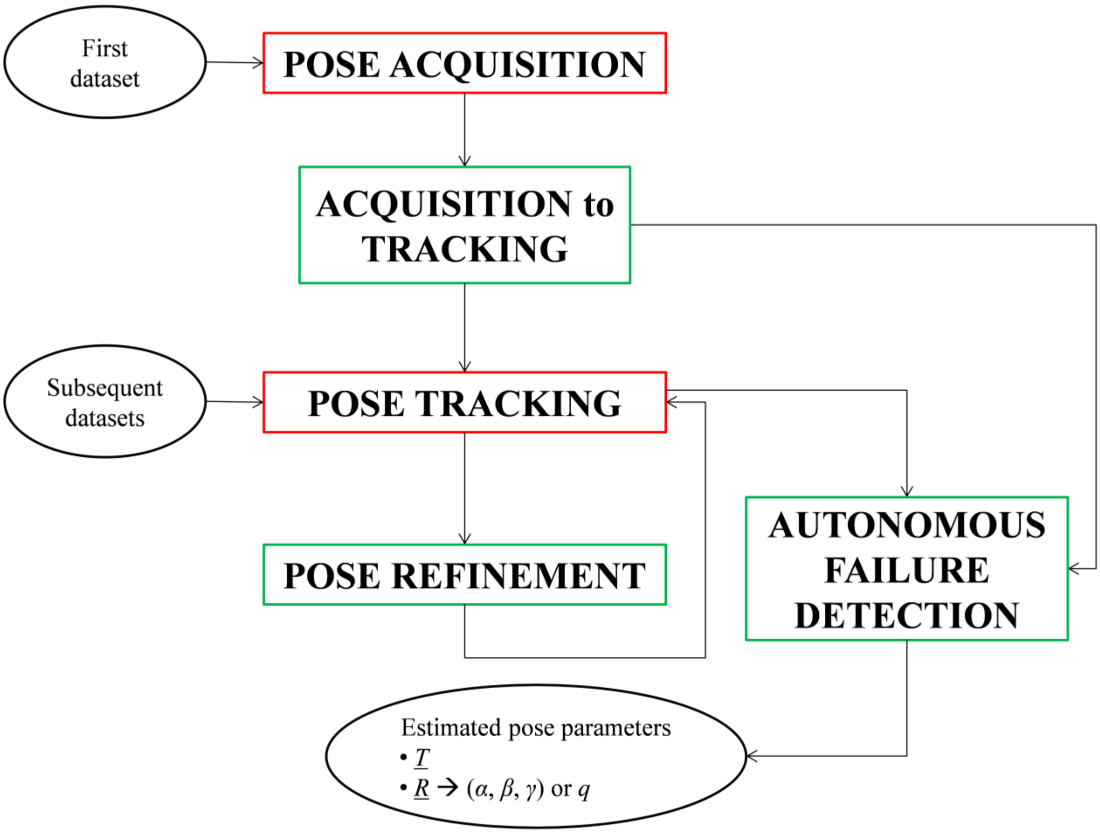


**Figure 2.11 - Logical scheme of the architecture proposed for uncooperative pose determination. Blocks containing model-based algorithms are highlighted in red. Blocks introduced to enhance robustness and efficiency of the proposed architecture are highlighted in green.**

## 2.4 Pose acquisition

### 2.4.1 3D on-line Template Matching

The 3D on-line Template Matching algorithm, shortly indicated as on-line TM, is a model-based technique able to estimate $\underline{p}_0$. All the previously mentioned TM approaches operating on 3D data [55, 56, 77] exploit off-line processing to build and organize the on-board database, and each of them foresees a different strategy to reduce the computational cost of the on-board processing. On the other hand, the on-line TM aims at improving the computational efficiency of the pose acquisition step with respect to the state-of-the-art approaches, while simultaneously cutting down the amount of data storage. Indeed, these aspects are both of utmost importance for close-proximity operations in space.

The on-line TM uses the idea of splitting the search for the pose solution within the relative state space in two phases [55, 66], but in an innovative way. As soon as the sensor point cloud ($\underline{\underline{P}}$) is acquired by the available LIDAR/stereovision system, the target initial relative position ($\underline{T}_0$) is estimated by exploiting a centroiding approach as shown by eq. (2.1),

$$\underline{T}_0 = -\underline{P}_C = -\frac{1}{N_P}\left(\sum_{i=1}^{N_P} x_i \quad \sum_{i=1}^{N_P} y_i \quad \sum_{i=1}^{N_P} z_i\right) \tag{2.1}$$

where $\underline{P}_C$ = ($P_{Cx}$, $P_{Cy}$, $P_{Cz}$) and $N_p$ are respectively the centroid and the size of $\underline{\underline{P}}$, while $x_i$, $y_i$ and $z_i$ are the SRF coordinates of the i$^{\text{th}}$ measured point ($\underline{P}^{\text{i}}$).

Hence, the search for the remaining unknowns, i.e. the initial Euler angles ($\gamma_0$, $\beta_0$, $\alpha_0$), can be restrained to a 3-DOF database. This procedure provides a significant reduction of the number of templates to be generated and compared to the sensor data thus improving the computational efficiency. Moreover, the database can be created dynamically since each template is generated on-line, i.e. directly on board, just before being compared to $\underline{P}$. In this way, the only data to be stored on board are the geometrical information about the target needed to generate the templates.

A flow diagram which describes in detail the main steps of the on-line TM is presented in Figure 2.12.

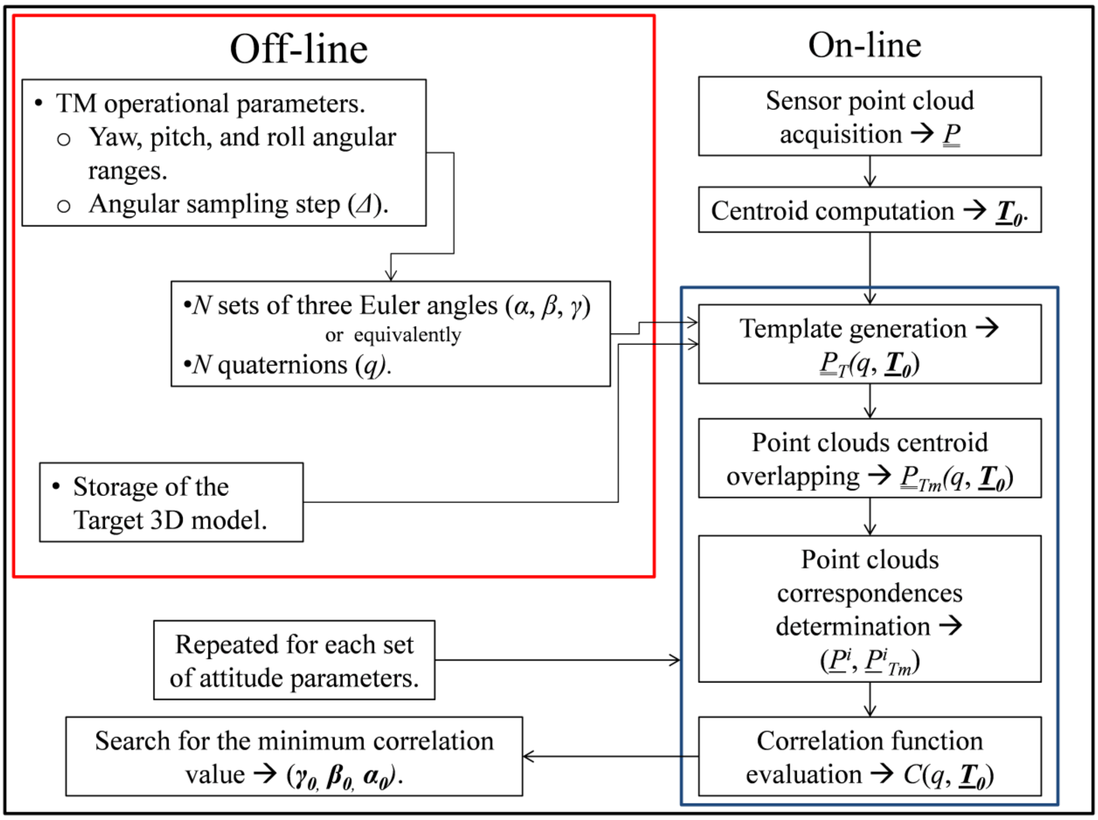

Off-line
• TM operational parameters.
○ Yaw, pitch, and roll angular ranges.
○ Angular sampling step ($\Delta$).
•$N$ sets of three Euler angles ($\alpha$, $\beta$, $\gamma$)
or equivalently
•$N$ quaternions ($q$).
• Storage of the Target 3D model.
On-line
Sensor point cloud acquisition → $\underline{P}$
Centroid computation → $\boldsymbol{T_0}$.
Template generation → $\underline{P}_T(q, \boldsymbol{T_0})$
Point clouds centroid overlapping → $\underline{P}_{Tm}(q, \boldsymbol{T_0})$
Point clouds correspondences determination → ($\underline{P}^i$, $\underline{P}^i_{Tm}$)
Correlation function evaluation → $C(q, \boldsymbol{T_0})$
Repeated for each set of attitude parameters.
Search for the minimum correlation value → ($\boldsymbol{\gamma_0}$, $\boldsymbol{\beta_0}$, $\boldsymbol{\alpha_0}$).

**Figure 2.12 - Flow diagram of the 3D on-line TM architecture. The estimated initial pose parameters are highlighted by bold type.**

It is clear that, unlike traditional TM approaches, the off-line actions (enclosed in the red box) are limited to the definition of the algorithm operational parameters, and the storage of the target geometric model. As regards the algorithm operation, it is necessary to assign

- the range of variation of the Euler angles, i.e. (−179°, 179°) for $\alpha$ and $\gamma$, and (−89°, 89°) for $\beta$, where the boundaries are left out to avoid considering ambiguous triplets;
- the angular sampling step ($\Delta$) with which the above defined intervals are sampled.

Of course, the lower the value of $\Delta$ is, the larger the number of templates ($N$) becomes.

Moving to the on-line stage, once $\underline{T}_0$ is estimated by applying eq. (2.1), four steps, enclosed in the blue box in Figure 2.12, are iterated for each sampled set of Euler angles (or, equivalently, for each corresponding $q$).

Firstly, a template, i.e. a point cloud ($\underline{\underline{P}}_T$), is built by the 3D sensor measurement simulator (see sub-chapter 3.3.1 for details about the template generation process).

Secondly, $\underline{\underline{P}}_T$ is translated so that its centroid ($\underline{P}_{CT}$) is aligned to $\underline{P}_C$, thus maximizing the point-cloud overlapping. This step is necessary in order to cope with the misalignment due to the fact that the estimation error in $\underline{T}_0$, which is used to generate the templates, can even be of the order of a few meters (depending on the size and shape of the point cloud). If not eliminated, this misalignment could easily make the algorithm produce wrong relative attitude solutions. The output of this overlapping step is a modified template ($\underline{\underline{P}}_{Tm}$) computed by applying eq. (2.2) to any element in $\underline{\underline{P}}_T$.

$$\underline{P}^j_{Tm}(q,T_0) = \underline{P}^j_T(q,T_0) - \underline{P}_{CT}(q,T_0) + \underline{P}_C \tag{2.2}$$

$\underline{P}^j_{Tm}$ is the j^th^ element of $\underline{\underline{P}}_{Tm}$, and $j$ varies from 1 to the size of $\underline{\underline{P}}_T$ ($N_t$).

Thirdly, template-sensor correspondences are determined by means of the Nearest Neighbor (NN) approach, i.e. each point in $\underline{\underline{P}}$ is associated to the closest one in $\underline{\underline{P}}_{Tm}$ according the Euclidean metric. At least in theory, the NN approach could be applied by inverting the roles of the two point clouds (i.e. associating each point in $\underline{\underline{P}}_{Tm}$ to the closest one in $\underline{\underline{P}}$). However, the first solution is adopted since the second one involves the possibility of excluding some measurements from the pose estimation process, thus potentially leading to performance worsening.

Fourthly, the level of similarity between the two point clouds is established by computing a correlation function (*C*), which is defined in eq. (2.3) as the mean square distance of corresponding template-sensor points.

$$C(q,T_0) = \frac{1}{N_P}\sum_{i=1}^{N_P}\left|(\underline{P}^i - \underline{P}^i_{Tm}(q,T_0))\right|^2 \tag{2.3}$$

$\underline{P}^i_{Tm}$ is the element of the modified template corresponding to $\underline{P}^{\mathrm{i}}$.

Finally, once this iterative on-line procedure is repeated for each given set of Euler angles, the relative attitude solution is the triplet associated to the template which minimizes *C*.

### 2.4.2 3D on-line Fast Template Matching

Although the on-line TM technique provides a significant reduction of the number of templates to analyze, it is still highly time consuming. Hence, a variant of this algorithm is introduced, namely the 3D On-line Fast Template Matching, which is shortly indicated as on-line fast-TM.

The main idea of the on-line fast-TM is to exclude, from the evaluation of the correlation function, those templates which are potential candidates to produce large value of *C*, meaning that they are badly correlated with $\underline{\underline{P}}$. Indeed, they can be a-priori recognized by analyzing a parameter which measures the point cloud distribution with respect to the boresight axis of the adopted sensor. Given a generic point cloud ($\underline{\underline{PC}}$), this parameter, $D_{BOR}$, can be defined as the mean distance from the sensor boresight axis, as shown in eq. (2.4),

$$D_{BOR}(\underline{\underline{PC}}) = \frac{1}{N_{PC}} \sum_{i=1}^{N_{PC}} \sqrt{(x_{PC}{}^{i})^2 + (y_{PC}{}^{i})^2} \tag{2.4}$$

where $N_{PC}$, $x_{PC}{}^{i}$ and $y_{PC}{}^{i}$ are the size and the SRF cross-boresight coordinates of its i$^{th}$ point, respectively. So, referring again to the flow diagram in Figure 2.12, the on-line fast-TM requires executing the two final steps in the blue box only if the condition on $D_{BOR}$, defined by eq. (2.5), is satisfied.

$$\frac{\left|D_{BOR}(\underline{\underline{P}}) - D_{BOR}(\underline{\underline{P}}_{Tm})\right|}{D_{BOR}(\underline{\underline{P}})} \leq \tau \tag{2.5}$$

Equation (2.5) states that all the templates characterized by a distribution with respect to the sensor boresight direction which differs from the one of $\underline{\underline{P}}$ by more than a given threshold ($\tau$), must be neglected. Of course the lower the selected $\tau$ is, the more the number of non-correlated templates increases, and, consequently, the larger the amount of saved computational time becomes. However, as it will be shown in detail by the results in chapter 5, if $\tau$ is too low the algorithm may exclude also potential good candidates to minimize the correlation function. Hence, the selected threshold must ensure the better trade-off between the computational time saving and the necessity to attain a negligible loss of performance.

Also, the computational time saving provided by this technique is affected by the fact that each template must be built and overlapped to sensor point cloud before being able to decide whether its distribution is compatible to the measured one. This means that the percentage of time saved thanks to this approach is always lower than the percentage of non-correlated templates.

### 2.4.3 3D PCA-based on-line Template Matching

Due to the limitations of the on-line fast-TM, it is interesting to consider the possibility of conceiving different techniques which exploit information obtained from the distribution of the measured point cloud to further improve the computational efficiency while simultaneously keeping at the same level (or even improving) algorithm performance.

To this aim the 3D PCA-based on-line Template Matching, shortly indicated as PCA-TM, is introduced. This technique exploits the centroiding approach, as well as the PCA and TM concepts to obtain an extremely large reduction of the computational time with respect to the previously defined methods. The search for the initial pose solution is subdivided in three stages.

Firstly, $\underline{T_0}$ is estimated by exploiting eq. (2.1), as done for both the on-line TM and on-line fast-TM.

Secondly, if the target has an elongated shape (which is typical of most active spacecraft and debris) its main axis ($\underline{e_M}$) can be estimated by exploiting the PCA. Indeed, it states that the principal directions of an assigned dataset are given by the eigenvectors of the associated covariance matrix. Thus, $\underline{e_M}$ is identified by the eigenvector corresponding to the maximum eigenvalue of the covariance matrix ($\underline{\underline{Q}}$) associated to the measured point cloud. This latter quantity is computed by eq. (2.6).

$$\underline{\underline{Q}} = \frac{1}{N_P}\begin{bmatrix} \sum_{i=1}^{N_P}(x_i - P_{Cx})^2 & \sum_{i=1}^{N_P}(x_i - P_{Cx})(y_i - P_{Cy}) & \sum_{i=1}^{N_P}(x_i - P_{Cx})(z_i - P_{Cz}) \\ \sum_{i=1}^{N_P}(y_i - P_{Cy})(x_i - P_{Cx}) & \sum_{i=1}^{N_P}(y_i - P_{Cy})^2 & \sum_{i=1}^{N_P}(y_i - P_{Cy})(z_i - P_{Cz}) \\ \sum_{i=1}^{N_P}(z_i - P_{Cz})(x_i - P_{Cx}) & \sum_{i=1}^{N_P}(z_i - P_{Cz})(y_i - P_{Cy}) & \sum_{i=1}^{N_P}(z_i - P_{Cz})^2 \end{bmatrix} \tag{2.6}$$

If the estimated main target axis direction, expressed in SRF, is assumed to correspond to the positive z-axis ($\underline{z}$) of the TRF, the relation between $\underline{e_M}$ = ($e_{Mx}$, $e_{My}$, $e_{Mz}$), and $\underline{z}$ is given by eq. (2.7), where the expression for the relative rotation matrix is obtained by setting to zero the yaw initial rotation.

$$\begin{bmatrix} e_{Mx} \\ e_{My} \\ e_{Mz} \end{bmatrix} = \begin{bmatrix} \cos(\beta_0) & 0 & -\sin(\beta_0) \\ \sin(\alpha_0)\sin(\beta_0) & \cos(\alpha_0) & \sin(\alpha_0)\cos(\beta_0) \\ \cos(\alpha_0)\sin(\beta_0) & -\sin(\alpha_0) & \cos(\alpha_0)\cos(\beta_0) \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.7}$$

So, the roll and pitch angles can be estimated thanks to eq. (2.8).

$$\begin{cases} \alpha_0 = \tan^{-1}(\dfrac{e_{My}}{e_{Mz}}) \\ \beta_0 = \sin^{-1}(-e_{Mx}) \end{cases} \tag{2.8}$$

Finally, it is necessary to compute $\gamma_0$, which represents the unresolved rotation around the target main axis.

This task is accomplished by means of an on-line TM approach in which a 1-DOF database of templates is built directly on-board by sampling the range of variation of $\gamma$, i.e. (-179°, 179°), with a fixed angular sampling step (still indicated by $\Delta$), which is the unique tunable parameter within the PCA-TM. Also in this case the number of templates is inversely related to $\Delta$.

For each of the $N$ values of $\gamma$, the four steps of the on-line TM, enclosed in the blue box in Figure 2.12, are repeated to evaluate the degree of similarity between the corresponding template and $\underline{\underline{P}}$. In this case the correlation function ($C_{PCA\text{-}TM}$) is given by eq. (2.9).

$$C_{PCA-TM}(\gamma,\alpha_0,\beta_0,T_0) = \frac{1}{N_P}\sum_{i=1}^{N_P}\left|(\underline{P}^i - \underline{P}^i_{Tm}(\gamma,\alpha_0,\beta_0,T_0))\right|^2 \tag{2.9}$$

Hence, $\gamma_0$ is set as the value of the yaw angle that minimizes $C_{PCA\text{-}TM}$. Although this procedure is very straightforward, a further problem arises due to the uncooperative nature of the target. Indeed, the PCA allows estimating, without ambiguity, the target main axis, but its direction remains undetermined since there is no way to establish whether $\underline{e}_M$ corresponds to the positive or negative z-axis in TRF. Thus, two different solutions can be obtained for roll and pitch angles, indicated as ($\alpha_{01}$, $\beta_{01}$) and ($\alpha_{02}$, $\beta_{02}$), by applying twice eq. (2.7) and eq. (2.8) considering both the positive and negative z-axis direction. As a consequence of this ambiguity, the TM part of the algorithm has to be run twice thus leading to a double solution also in the yaw angle (i.e. $\gamma_{02}$ and $\gamma_{02}$). At the end of this process, two values of $\underline{p}_0$ are available, which are listed in eq. (2.10),

$$\begin{cases} \underline{p}_{01} = (\alpha_{01}, \beta_{01}, \gamma_{01}, \underline{T}_0) \\ \underline{p}_{02} = (\alpha_{02}, \beta_{02}, \gamma_{02}, \underline{T}_0) \end{cases} \tag{2.10}$$

where $\beta_{01}$ and $\beta_{02}$ are opposite while $\alpha_{01}$ and $-\alpha_{02}$ are supplementary angles. The PCA-TM overall architecture is summarized by the flow diagram in Figure 2.13.

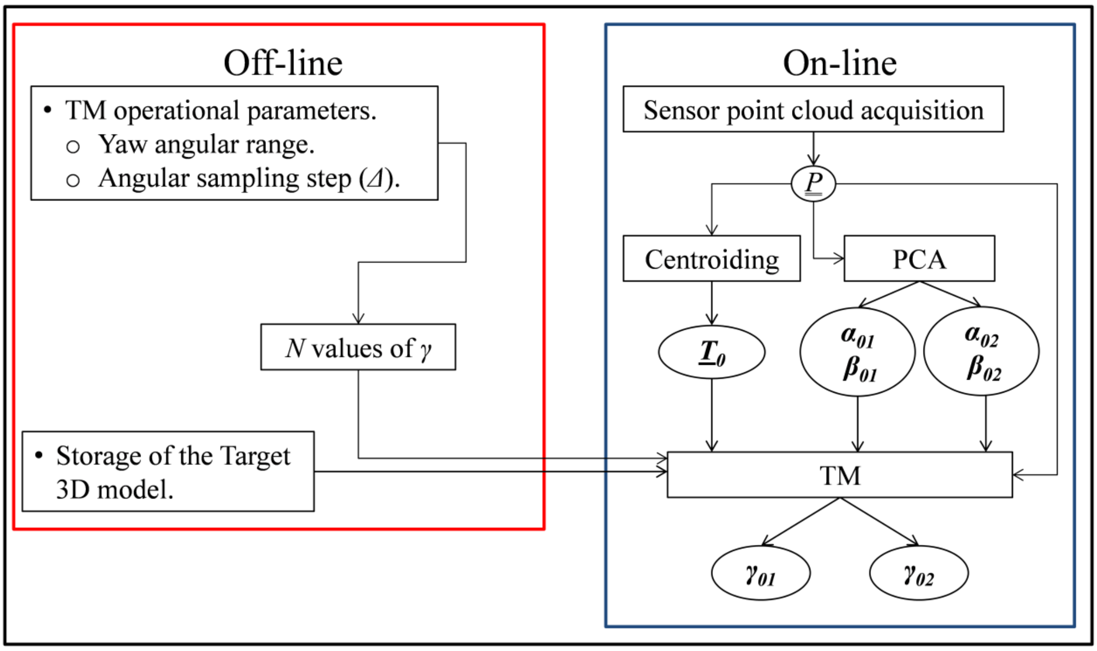

Off-line
• TM operational parameters.
o Yaw angular range.
o Angular sampling step (Δ).
N values of γ
• Storage of the Target 3D model.
On-line
Sensor point cloud acquisition
P
Centroiding
PCA
T0
α01 β01
α02 β02
TM
γ01
γ02

**Figure 2.13 - Flow diagram of the 3D PCA-based on-line TM architecture. The estimated initial pose parameters are highlighted by bold type.**

In spite of the necessity to solve the ambiguity between the two solutions in eq. (2.10), which is carried out by the acquisition-to-tracking transition step (see sub-chapter 2.6), the PCA-TM is an extremely promising approach. Indeed, the on-board data storage is restricted just like for the on-line TM and on-line fast-TM. Moreover, the PCA-TM provides a significant advantage in terms of computational efficiency with respect to both the previously proposed approaches. This depends on the fact that the TM-based search is limited to the 1-DOF database thus cutting down the number of analyzed templates. Indeed, by varying *Δ* from 10° to 60°, *N* ranges from 26011 to 196 for the on-line TM, while it goes from 37 to 7 for the PCA-TM.

Finally, it is worth outlining that the PCA-TM robustness can be enhanced by introducing an additional step which verifies the consistency of the direction estimated by the PCA. Although most potential targets for OOS and ADR applications have an elongated shape, there could be some pose conditions at which the target principal direction is occluded so that the measured point cloud is not able to image it correctly. However, these conditions can be identified, before applying the TM step, by analyzing the shape of $\underline{\underline{P}}$ thanks to the PCA. The basic idea is that the ratio ($r$) between the maximum and minimum eigenvalues of $\underline{\underline{Q}}$ is a direct measure of the elongation of the object observed from a specific point of view. Hence, the higher the value of $r$ is, the more reliable the estimated $\underline{e}_M$ is. The validity of this procedure will be demonstrated by exploiting results from numerical simulations in sub-chapter 5.2.3. A modified architecture for the PCA-TM, including this consistency check, is shown in Figure 2.14. According to this scheme, if $r$ is lower than a safety threshold ($\tau_\lambda$), selected by analyzing off-line the target geometry, it is necessary to wait for a certain amount of time, during which the true pose may evolve toward most favorable conditions.

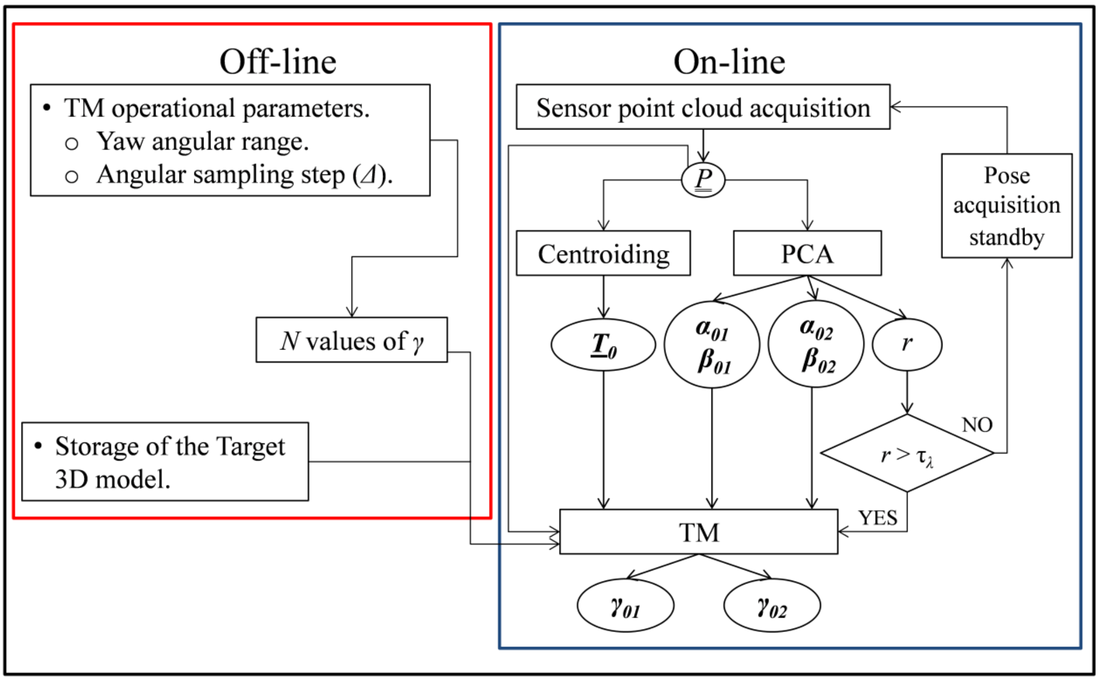


**Figure 2.14 - Flow diagram of the 3D PCA-based on-line TM architecture modified to include PCA consistency check. The estimated initial pose parameters are highlighted by bold type.**

## 2.5 Pose tracking

### 2.5.1 Customized Iterative Closest Point algorithm

The ICP algorithm is largely applied to deal with the problem of pose determination since it is a simple and fast solution to register two datasets. Specifically, it allows finding, through an iterative process, the best estimate of the rigid rotation and translation necessary to align a point-based representation of the target to $\underline{\underline{P}}$. The ICP procedure comprises several different phases for each of which many variants have been derived [83]. However, three mandatory steps can be identified.

- Initialization. An initial guess of the pose parameter is used to express both $\underline{\underline{P}}$ and $\underline{\underline{P}}_M$ in the same reference.
- Matching. Correspondences between model and sensor points are determined.
- Selection and minimization of error metric function. It monitors the convergence of the algorithm.

The proposed customized ICP algorithm [31] is obtained by selecting specific solutions for the above-mentioned stages. However, it also foresees the possibility to activate an additional weighting step after matching, and it includes a strategy for autonomous failure detection.

With the exception of the tracking phase starting in which the initial guess is the one provided by pose acquisition, each time a new dataset is available, the first ICP iteration is initialized by a prediction algorithm [31], which is in charge of the pose refinement step within the pose determination architecture depicted in Figure 2.11. It is a linear kinematic-only filter which updates the values of the previous estimates of the Euler angle on the basis of their time derivative, and it is aimed at both accelerating algorithm's convergence and improving its accuracy. The subscripts

"INIT" and "EST" in eq. (2.11) indicate respectively the initial guess and the ICP estimate for each Euler angle.

$$\begin{cases} \alpha_{INIT}^{t=t_i} = \alpha_{EST}^{t=t_{i-1}} + \left.\dfrac{d\alpha_{EST}}{dt}\right|_{t=t_{i-1}} (t_i - t_{i-1}) \\ \beta_{INIT}^{t=t_i} = \beta_{EST}^{t=t_{i-1}} + \left.\dfrac{d\beta_{EST}}{dt}\right|_{t=t_{i-1}} (t_i - t_{i-1}) \\ \gamma_{INIT}^{t=t_i} = \gamma_{EST}^{t=t_{i-1}} + \left.\dfrac{d\gamma_{EST}}{dt}\right|_{t=t_{i-1}} (t_i - t_{i-1}) \end{cases} \tag{2.11}$$

As regards the matching step, two different methods, namely the NN and the Normal Shooting (NS), are compared in terms of performance accuracy and computational efficiency. While the former method associates each sensor point to the closest one in a point-cloud generated off-line from the target model, i.e. the model point cloud ($\underline{P}_M$), the latter one generates $\underline{P}_M$ dynamically, i.e. on-line at each algorithm iteration, by projecting the sensor points on the planes corresponding to the closest target model surface according to the local normal. The difference between these two methods to determine sensor-model point correspondences is highlighted in Figure 2.15, considering a 1D problem so that, in the NS case, the target model is a single line (instead of a plane) whose normal is *n*.

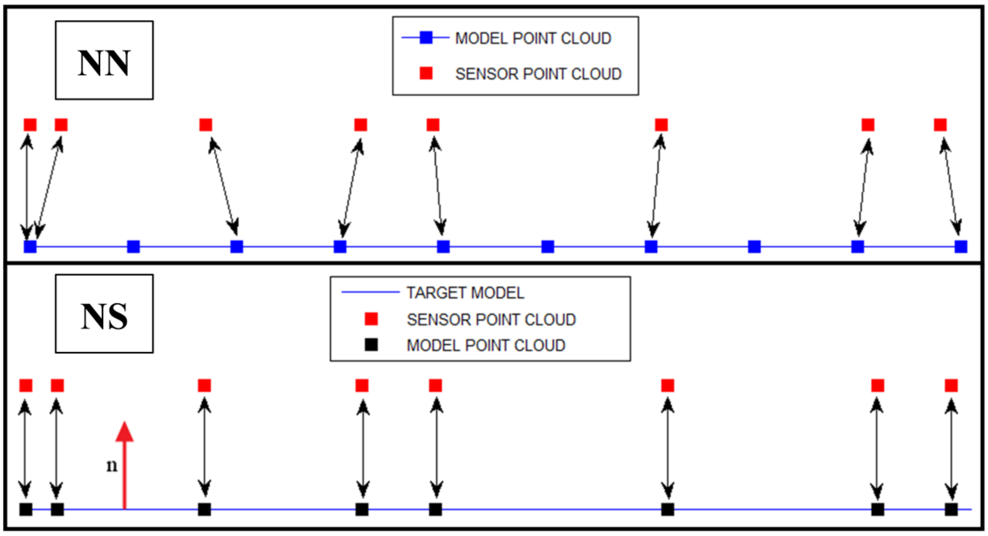
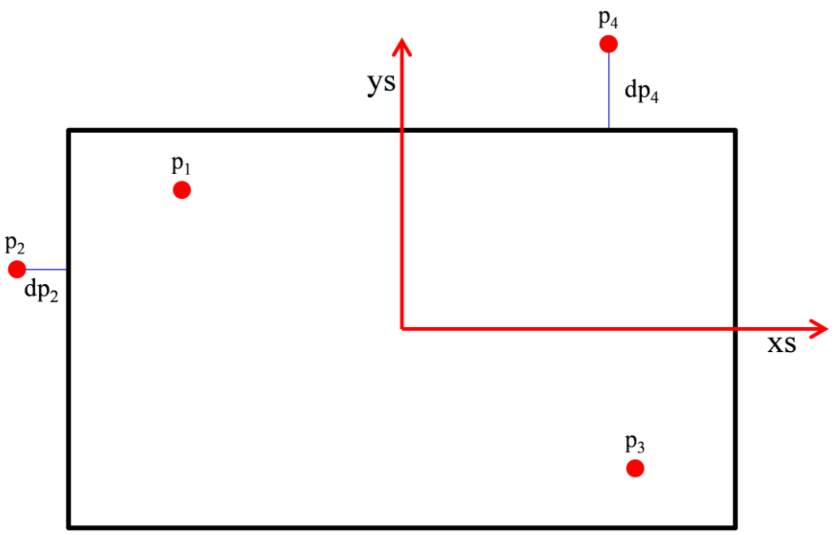


**Figure 2.15 - NN and NS solutions to the ICP matching step.**

If the NN approach is adopted $\underline{\underline{P}}_M$ is pre-processed to build a K-D tree [84]. This is done to accelerate the NN search which otherwise could be very time consuming since the size of $\underline{\underline{P}}_M$ is typically one, two or even three order of magnitude larger than $\underline{\underline{P}}$ (depending on how densely the surfaces of target 3D model are sampled). The correct choice for the off-line generation of $\underline{\underline{P}}_M$, in terms of discretization, is typically driven by a trade-off between accuracy and computational efficiency.

It is now necessary to clarify a problematic which arises from the implementation of the NS concept. Indeed, due to potential errors in the initial guess used to rotate and translate $\underline{\underline{P}}$ from the SRF to the TRF, some elements of $\underline{\underline{P}}_M$ could fall outside of the physical boundaries of the target surfaces (see Figure 2.16). The simplest solution could be to eliminate all the sensor/model point pairs for which this phenomenon occurs. However, this could lead to lose too many information thus compromising the effectiveness of the algorithm. Hence, a distance threshold ($\tau_{NS}$) is defined to verify the consistency of projected model points with respect to the target 3D model, as it can be seen in Figure 2.16.



$p_1$ → inside → **OK**

$p_2$ → outside→ $dp_2 < \tau_{NS}$ → **OK**

$p_3$ → inside → **OK**

$p_4$ → outside→ $dp_4 > \tau_{NS}$ → **NOT OK**

**Figure 2.16 - Check of model point consistency for the NS.**

Numerical simulations are performed to evaluate the ICP sensibility to the discretization level of the target model, for the NN, and to the value of $\tau_{NS}$, for the NS, whose results are presented in sub-chapter 5.3.3.

After matching, a weighting step can potentially be introduced. It consists in assigning different weights ($w$) to the matched sensor-model point pairs. Specifically, the weighting law shown in eq. (2.12) is selected, which assigns lower weights to pairs characterized by greater distance ($d_{ICP}$). In this equation, the index $i$ identifies each point pair and max$\{d_{ICP}\}$ is the maximum pair distance.

$$w^i = 1 - \frac{d_{ICP}{}^{i}}{\max\{d_{ICP}\}} \tag{2.12}$$

As regards the selection of the error metric function, also indicated as cost function ($f$), it is defined in eq. (2.13) as the mean squared distance of corresponding model-sensor points.

$$f(q) = \frac{1}{N_P}\sum_{i=1}^{N_P}\left|\underline{P}^i_M - \underline{\underline{R}}(q)^T(\underline{P}^i + \underline{T})\right|^2 \tag{2.13}$$

In eq. (2.13), $\underline{P}^i{}_M$ is the model point corresponding to $\underline{P}^i$. This cost function is minimized through a closed-form solution based on the unit quaternion, as proposed by Horn [85]. Firstly, the covariance matrix of the sensor-model point pairs ($\underline{\underline{Q}}_{SM}$) is derived by eq. (2.14) where $\underline{\underline{P}}_{CM}$ is the centroid of the model point cloud, and it is used to build a symmetric 4x4 matrix ($\underline{\underline{Q}}_{\Sigma}$) according to eq. (2.15).

$$\underline{\underline{Q}}_{SM} = \frac{1}{N_P}\sum_{i=1}^{N_P}\left[(\underline{P}^i - \underline{P}_C)\cdot(\underline{P}_M{}^i - \underline{P}_{CM})^T\right] \tag{2.14}$$

$$\underline{\underline{Q}}_{\Sigma} = \begin{bmatrix} tr\left(\underline{\underline{Q}}_{SM}\right) & \underline{\Delta}_Q{}^T \\ \underline{\Delta}_Q & \underline{\underline{Q}}_{SM} + \underline{\underline{Q}}_{SM}{}^T - tr\left(\underline{\underline{Q}}_{SM}\right)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix} \tag{2.15}$$

In eq. (2.15), $tr(\underline{\underline{Q}}_{SM})$ is the trace of $\underline{\underline{Q}}_{SM}$, while $\underline{\Delta}_{Q}$ is the column vector composed of the out-of-diagonal elements of $\underline{\underline{Q}}_{SM}$ - $\underline{\underline{Q}}_{SM}^{T}$. Once the optimal quaternion ($q_{ICP}$) is estimated as the eigenvector corresponding to the maximum eigenvalue of $\underline{\underline{Q}}_{\Sigma}$, $\underline{T}$ is updated by means of eq. (2.16).

$$\underline{T} = \underline{\underline{R}}(q_{ICP})\underline{P}_{CM} - \underline{P}_{C} \tag{2.16}$$

All the above-described stages of this implementation of the ICP algorithm must be iterated since a convergence criterion is met. Specifically, the stop condition is reached when the time derivative of the cost function between two subsequent iterations goes below a threshold of $10^{-6}$ m$^{2}$. Moreover, in order to limit the achievable computational time, a maximum number of 30 iterations is considered for each run of the technique. The value of the cost function at convergence is indicated as $f_{CONV}$.

Once the convergence criterion is met, the pose solution must be subject to the autonomous failure detection block shown in Figure 2.11, in order to move on to subsequent time step. The autonomous failure detection strategy is ICP-based, as it relies on the fact that the value of $f_{CONV}$ is a measure of the algorithm's accuracy level, i.e. the higher $f_{CONV}$ is, the coarser the estimated pose gets [31]. This means that when $f_{CONV}$ is larger than a specific threshold ($f_{LIM}$), the ICP pose solution is not reliable and the algorithm must be applied to the next set of measurements using the same initial guess. The logical scheme describing this failure detection strategy is shown in Figure 2.17. If the failure test is not satisfied for a fixed number of subsequent applications of the ICP algorithm, the tracking is considered lost and the pose must be re-initialized again by means of the acquisition step.

Finally, it is important to notice that this autonomous failure detection strategy is applicable also after the pose acquisition step. For these cases, the "Customized ICP algorithm" block in Figure 2.17 is not representative of the tracking phase but of the transition step from acquisition to tracking as it is clarified in the next sub-chapter.

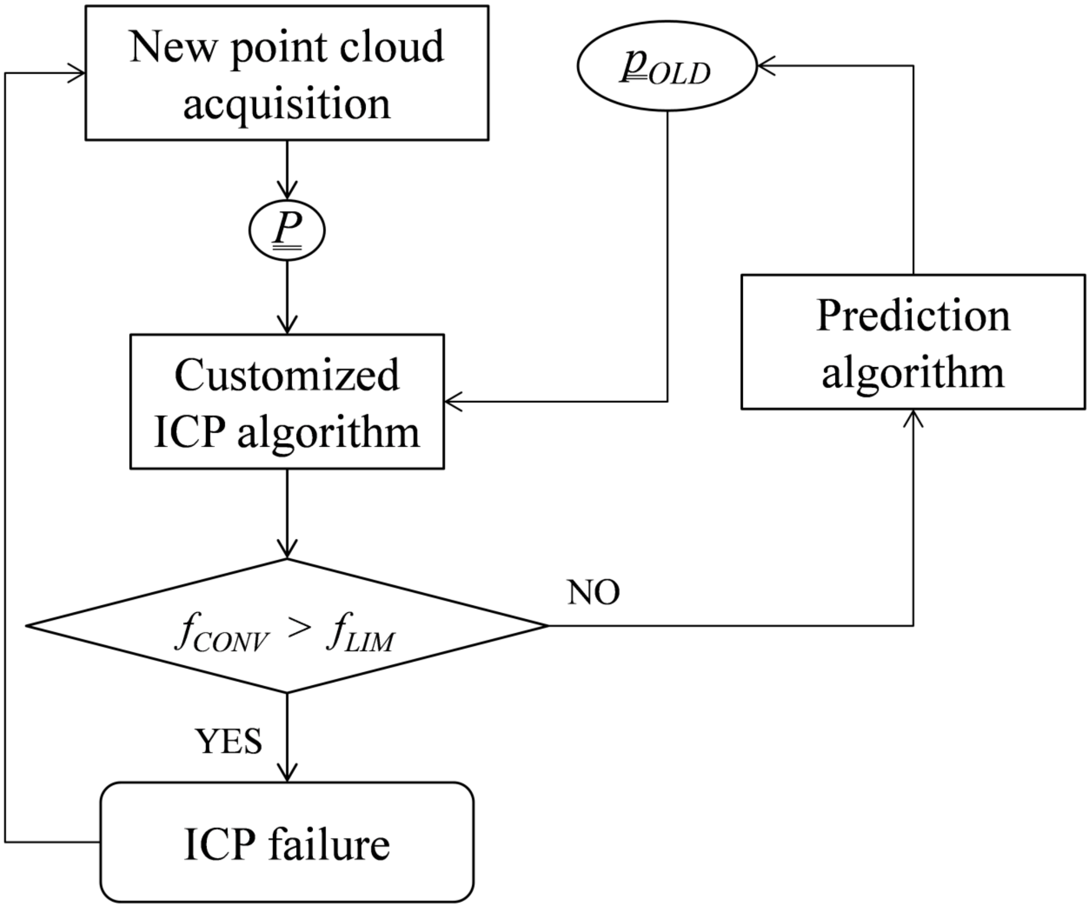


**Figure 2.17 - Autonomous failure detection strategy of the proposed pose determination architecture.**

## 2.6 Transition from acquisition to tracking

In order to ensure a safe transition from acquisition to tracking, an ICP-based strategy is adopted. Again, the basic idea is to exploit the relation between the accuracy level attained by the ICP algorithm and the value of $f_{CONV}$.

If the on-line TM or the on-line fast-TM are in charge of pose acquisition, once the initialization is available, the transition simply consists in applying the ICP algorithm and then verifying whether the pose solution is reliable to move on to the tracking step by means of the failure detection strategy. This substantially means that if the condition $f_{CONV} < f_{LIM}$ is not satisfied the transition is considered not successful and the pose acquisition must be applied again on a subsequent dataset provided by the available sensor. It is worth outlining that the maximum iteration number is set to 100 during this transition step, as a further safety criterion. This is necessary since the pose parameter gap that the ICP algorithm has to fill is larger than when the tracking phase has already started, due to the coarse accuracy of the pose estimate provided by the acquisition step.

If the PCA-TM is adopted, the acquisition-to-tracking transition requires a strategy to identify which is the correct pose solution between the two possible ones listed in eq. (2.10). This strategy is summarized by the flow diagram in Figure 2.18.

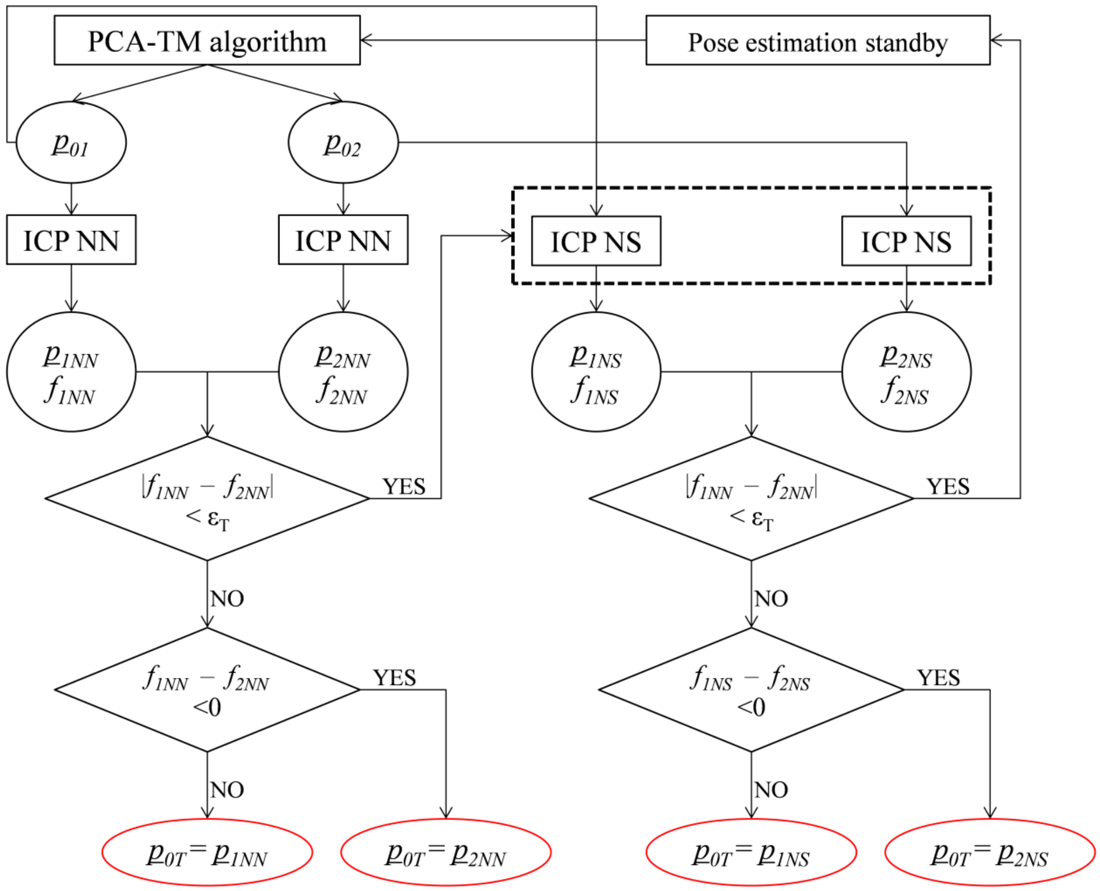

PCA-TM algorithm
Pose estimation standby
$\underline{p}_{01}$
$\underline{p}_{02}$
ICP NN
ICP NN
ICP NS
ICP NS
$\underline{p}_{1NN}$ $f_{1NN}$
$\underline{p}_{2NN}$ $f_{2NN}$
$\underline{p}_{1NS}$ $f_{1NS}$
$\underline{p}_{2NS}$ $f_{2NS}$
$|f_{1NN} - f_{2NN}| < \varepsilon_T$
YES
$|f_{1NN} - f_{2NN}| < \varepsilon_T$
YES
NO
NO
$f_{1NN} - f_{2NN} < 0$
YES
$f_{1NS} - f_{2NS} < 0$
YES
NO
NO
$\underline{p}_{0T} = \underline{p}_{1NN}$
$\underline{p}_{0T} = \underline{p}_{2NN}$
$\underline{p}_{0T} = \underline{p}_{1NS}$
$\underline{p}_{0T} = \underline{p}_{2NS}$

**Figure 2.18 - Flow diagram describing the strategy adopted to safely manage the transition from acquisition to tracking in the case of the PCA-TM. Possible final outputs are highlighted in red.**

The first step of the process consists in applying twice the ICP algorithm by exploiting the NN approach, taking as initial guess both $p_{01}$ and $p_{02}$. This provides two possible solutions ($p_{1NN}$ and $p_{2NN}$) as well as the corresponding values of the error metric function at convergence ($f_{1NN}$ and $f_{2NN}$). If the latter values differ of more than a threshold ($\varepsilon_T$) of $10^{-6}$ m$^2$, the ambiguity between $p_{1NN}$ and $p_{2NN}$ can be solved. Thus, the pose solution corresponding to the lowest between $f_{1NN}$ and $f_{2NN}$ is taken as input of the pose tracking stage ($p_{0T}$).

If the NN approach provides ambiguous pose estimates, i.e. $|f_{1NN} - f_{2NN}| < \varepsilon_T$, the procedure is repeated, starting from $p_{01}$ and $p_{02}$, but applying the NS approach. This leads to a couple of solutions ($p_{1NS}$ and $p_{2NS}$), so that $p_{0T}$ can be determined by comparing the corresponding values of the error metric function at convergence ($f_{1NS}$ and $f_{2NS}$). Also in this case, if the solution is still ambiguous, i.e. $|f_{1NS} - f_{2NS}| < \varepsilon_T$, it is necessary to wait for a certain amount of time, during which the true pose may evolve toward most favorable conditions, and to execute the PCA-TM algorithm on an updated set of 3D measurements.

In conclusion, if the ambiguity is solved, also for the PCA-TM the final check of the transition process is to apply the ICP and wait for the outcome of the failure detection condition as shown in Figure 2.17.

# Chapter 3 - Models and simulation environment

## 3.1 Introduction

One of the main issues related to research activities on the subject of relative navigation between spacecraft flying in close-proximity is the lack of reliable and effective means by which innovative technological and algorithmic solutions can be proved. Indeed, the amount of open-access data about uncooperative OOS or ADR space activities is extremely limited, and the capability to realize meaningful tests on ground is restrained by the difficulty to reproduce, within an experimental testbed, realistic conditions, especially in terms of illumination and relative dynamics. Hence, in order to fully assess the performance of technologies and algorithms for uncooperative pose determination, it is necessary to integrate experimental tests with numerical simulations. Indeed, realistic software environments ensure the possibility to reproduce a much wider range of operational conditions with respect to an experimental setup.

For these reasons, the purpose of this chapter is to describe the numerical simulation environment, developed in MATLAB, designed to evaluate the performance of the pose determination algorithms presented in sub-chapter 2.4 and 2.5. This simulation environment, whose architecture is depicted in Figure 3.1, is able to realistically reproduce

- highly variable target-chaser relative dynamics during relative navigation operations like rendezvous, station keeping and monitoring,
- and the operation of a LIDAR capable of acquiring 3D point clouds.

Also, it includes the implementation of the above-mentioned model-based algorithms.

Firstly, the input data about target and chaser absolute dynamics are processed within a relative dynamics simulator to obtain the true relative position and attitude parameters.

Secondly, these latter data, together with the LIDAR operational and noise parameters, and the target 3D model, are exploited by a complex LIDAR simulator which includes modeling of all the related geometric, detection and noise aspects, thus generating realistic point clouds.

Finally, the pose determination algorithms are applied to get estimates of the pose parameters, so that their performance in terms of estimation accuracy and computational efficiency can be evaluated.

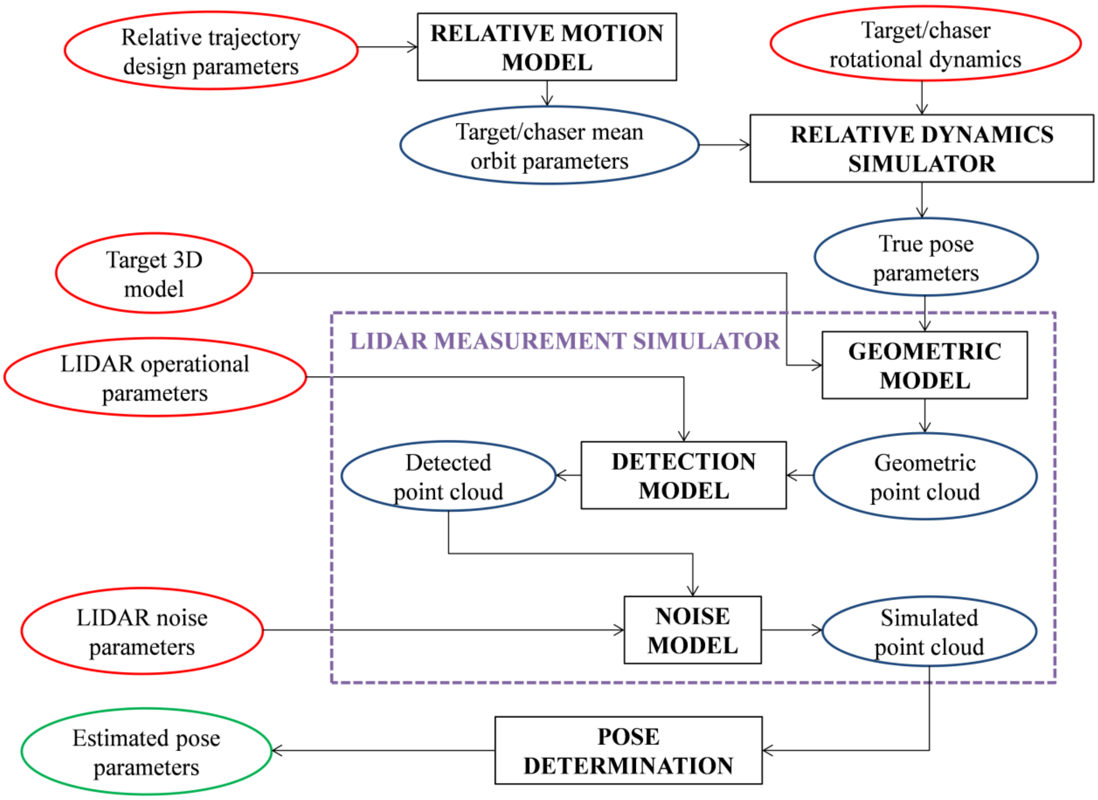

Relative trajectory design parameters
RELATIVE MOTION MODEL
Target/chaser rotational dynamics
Target/chaser mean orbit parameters
RELATIVE DYNAMICS SIMULATOR
Target 3D model
True pose parameters
LIDAR MEASUREMENT SIMULATOR
GEOMETRIC MODEL
LIDAR operational parameters
Detected point cloud
DETECTION MODEL
Geometric point cloud
LIDAR noise parameters
NOISE MODEL
Simulated point cloud
Estimated pose parameters
POSE DETERMINATION

**Figure 3.1 - Architecture of the simulation environment for performance evaluation of spacecraft pose determination algorithms. The inputs, the intermediate data, and the final output are contained in red, blue and green circles respectively. The violet dashed rectangle identifies all the blocks composing the LIDAR measurement simulator.**

All the aspects relative to the above-described simulation environment are organized in this chapter as follows.

Sub-chapter 3.2 describes the measurement principle of LIDAR systems and contains a review about spaceborne LIDAR adopted during missions involving relative navigation maneuvers.

Sub-chapter 3.3 describes the LIDAR measurement simulator including details about how all the related geometric, detection and noise aspects are modeled.

Sub-chapter 3.4 is focused on the characterization of the simulated LIDAR system motivating the choices of its operational parameters and noise data.

Finally, sub-chapter 3.5 presents the selection of the targets adopted within the numerical simulations and the method adopted to create their 3D models, which depends on the information required by the pose determination algorithms.

Sub-chapter 3.6 describes the relative dynamics simulator which generates the true pose from the absolute dynamics information of the chaser and the target. As highlighted in Figure 3.1, the simulator includes a relative motion model which is used to determine the target and the chaser mean orbit parameters, given the relative trajectory design data in input. For the sake of clarity of the presentation, this model is presented in chapter 4.

## 3.2 LIDAR system overview

The term LIDAR is used to indicate a large variety of sensors that are basically able to measure distances by illuminating a target with a laser and analyzing the backscattered radiation. The main components of a LIDAR system are the laser source (operating in the ultraviolet, visible and IR regions of the electromagnetic spectrum), the optics, the detector and the control electronics.

The basic typology of LIDAR is given by laser range finders which use a single detector to measure the distance travelled by a non-steerable laser beam. While these systems are able to compute exclusively range and LOS of the target, 3D LIDAR can provide 6-DOF relative information by acquiring data, e.g. point clouds, within an assigned FOV. As regards the technological solutions 3D LIDAR can be ranked into three main groups: scanning devices (e.g. single-point, slit or pattern-projection scanners), sensors based on detector arrays, and spatial light modulators [86-88].

Scanning LIDAR have typically one detector and they are able to change the direction of a single narrow laser beam by means of lenses, mirrors or other devices moved by high-speed and high-precision galvanometers, thus providing high-resolution point clouds. As regards the scanning pattern, several solutions exist ranging from the classical raster scan (depicted in Figure 3.2) to more complicated patterns, e.g. Lissajous, rosette and spirals, which are able to cover the FOV faster since they do not require stopping the scanning mirrors at the end of a line [89].

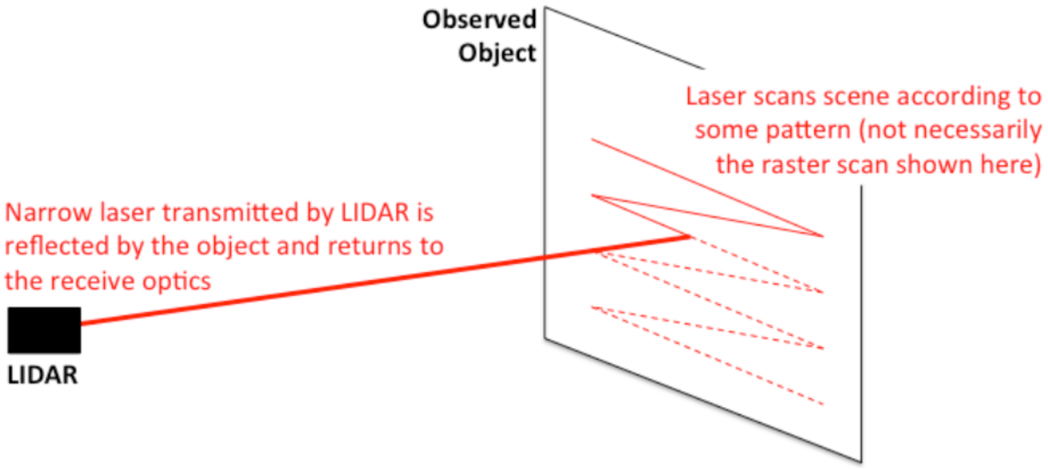

Observed Object
Laser scans scene according to some pattern (not necessarily the raster scan shown here)
Narrow laser transmitted by LIDAR is reflected by the object and returns to the receive optics
LIDAR

**Figure 3.2 - Conceptual representation of scanning LIDAR measurement principle [88].**

These sensors are relatively easy to calibrate since the user only needs to be concerned about the light sensitivity and timing for one detector. However, they contain moving parts which can be potential source of hardware failure, they ensure poor frame rate due to the time needed to scan the FOV and, most of all, the acquired point cloud can be affected by undesirable blur effects due to the target motion while the sensor scans its FOV.

The second major class of LIDAR contains those instruments which illuminate the entire scene with a single broad laser beam and use a detector array to individually measure the TOF of the light backscattered from each pixel direction (see Figure 3.3). The receiver optics and detector array operate in much the same way as a conventional camera. So, they do not have moving parts and their images are not affected by noticeable motion blur. However, scannerless LIDAR have more detectors, thus being more challenging to calibrate, and the fabrication limit on the size of the detector arrays restrains the size of the 3D point cloud, thus getting poor and fixed spatial resolution.

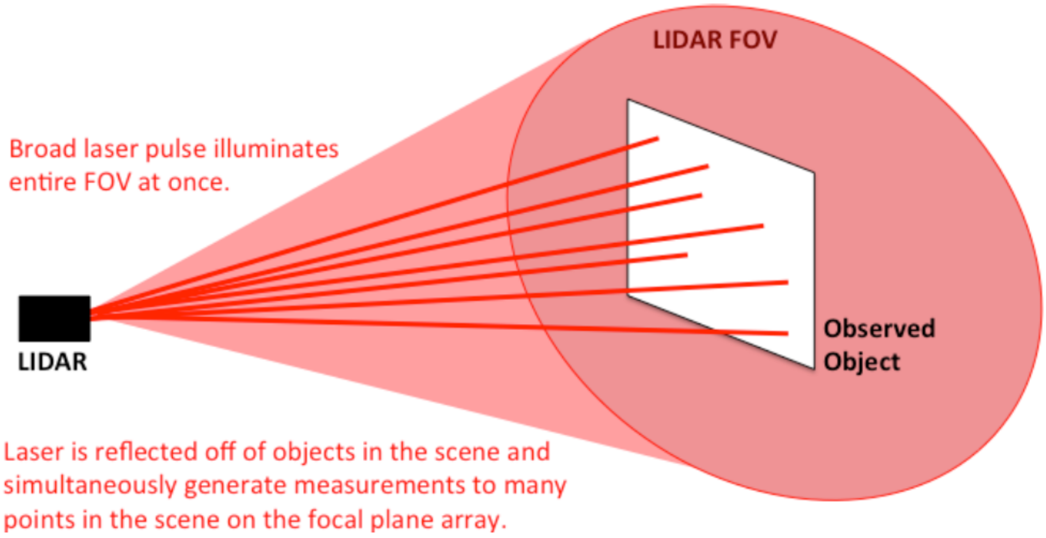

LIDAR FOV
Broad laser pulse illuminates entire FOV at once.
LIDAR
Observed Object
Laser is reflected off of objects in the scene and simultaneously generate measurements to many points in the scene on the focal plane array.

**Figure 3.3 - Conceptual representation of the measurement principle for scannerless LIDAR [88].**

Spatial light modulators basically operate by illuminating portions of the scene according to a pre-defined pattern and then sensing the return with a single detector. The backscattered radiation is processed to obtain meaningful information by using compressed sensing algorithms [90]. While the former two categories have been already used for relative navigation on board previous spacecraft and for flight experiments, spatial light modulators are innovative instruments, still under development, and far from practical space application. For these reason, the attention is focused on the first two categories.

Regarding the typology of laser source, LIDAR can be divided in pulsed or continuous-wave (CW) systems. This distinction is valid to laser range finders as well as to scanning and scannerless instruments. In particular, sensors based on a detector array which rely on a pulsed laser source are called flash LIDAR.

CW LIDAR compute distances by measuring the phase difference between an amplitude-modulated (AM), more frequently used, or a frequency modulated (FM) emitted signal (the reference signal) and the reflected echo, thus exploiting the heterodyne principle. The major related issue is that their applicability is limited to close-range applications by the phase integer ambiguity [88]. CW LIDAR can also exploit a different measuring principle, namely the triangulation method. As shown in Figure 3.4, triangulation consists in emitting a laser beam which is reflected by the target and focused by a lens on a CCD detector. The location where the reflected beam is backscattered on is a function of the target range. This procedure ensures to attain millimeter or even sub-millimeter accuracy, which unfortunately decreases as the inverse of square root of the distance [91]. For this reason, CW LIDAR based on triangulation are typically used at close range, e.g. below 15 m.

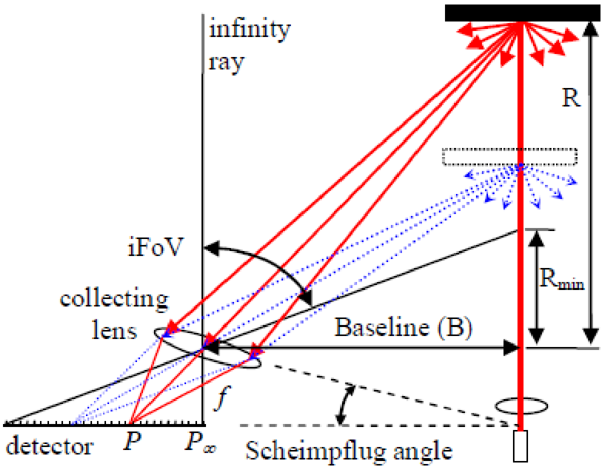


**Figure 3.4 - Principle of triangulation for distance measurement [92].**

On the other hand, pulsed LIDAR are direct-detection system which compute distances by measuring the time delay between transmitted and received laser pulses typically exploiting Gallium Arsenide TOF counters. These systems, also known as TOF LIDAR, are intrinsically simpler since the laser source is incoherent. TOF LIDAR represent the most convenient choice when the interval of operating distances is very large, i.e. from a few meter to a few kilometer, since they can provide constant accuracy level as a function of the range. However, the accuracy provided by the TOF principle is limited since the range resolution depends on the achievable time resolution. For instance, in order to get an accuracy of centimeter order, a time resolution of tens of picoseconds is needed thus requiring complicated and expensive electronics.

For the sake of completeness, it is worth outlining that an additional method, based on pseudo-random number (PRN), can be used for TOF computation. It consists in encoding a PRN sequence onto the laser and then performing an autocorrelation with the sensed return. However this method is typically not used for spaceborne LIDAR.

### 3.2.1 Spaceborne LIDAR for relative navigation

An extended (but not necessarily exhaustive) review of active laser-based sensors used or in development to perform relative navigation maneuvers between a chaser satellite and a cooperative/uncooperative target is presented hereunder.

Laser Range Scanner

The Laser Range Scanner (LARS) [91, 93, 94], developed by the CSA, is a 3D sensor for space applications capable of doing surface imaging, target ranging and tracking. It was realized to cooperate with the Canadian Space Vision System (CSVS, designed to help astronauts to assemble the ISS) on the STS-52 shuttle flight in 1992 to test the capability of 3D space inspection and surface reflectance monitoring of the ISS [78]. LARS is a single point laser scanner able to operate at short range, i.e. from 0.5 m to 10 m, by using the triangulation method, as well as at long range, i.e. 10 m to 10 km, by using the TOF method, thus exploiting the complementary accuracy performances of the two techniques at different ranges. The triangulation and TOF modes respectively use CW and pulsed laser sources. The resolution at short range is sub-millimeter but it reduces gradually with distance, i.e. it becomes 1 cm at 10 m of range. At long range the TOF method provides a constant accuracy of about 3 cm. The capability of LARS to scan the entire FOV is guaranteed by two high speed galvanometers. LARS is particularly suitable for operating in space, where surfaces have high contrasts and can generate specular reflections, since it guarantees an extremely wide dynamic range of intensities. Moreover, it has clear advantages over microwave systems in terms of size, mass, power and precision. In the case of a single target, the algorithm which directly process LARS data can reach a refresh rate of 137 Hz in tracking mode.

NEAR Laser Rangefinder

The NEAR Laser Rangefinder (NLR) [95, 96] is a TOF-based laser altimeter developed by the Applied Physics Laboratory (APL) at John Hopkins University and installed on board the Near Earth Asteroid Rendezvous (NEAR) spacecraft that was launched in space in 1996. The main mission objective was to provide information about composition, mineralogy, morphology, internal mass distribution and magnetic field of the asteroid Eros 433. The NLR is a bi-static system, i.e. composed of separate transmitter and receiver, able to send infrared laser pulses to the surface of the asteroid, receive the backscattered ones (detected by an APD) and record the TOF (measured by a Gallium Arsenide chip) in order to compute the distance. The system architecture is accurately described in Figure 3.5.

With regards to its main specifications, the pulse duration is 15 ns and its frequency varies between 0.125 Hz and 8 Hz. During the one-year observation of Eros, approximately 11 million measurements were obtained from NLR. It provided range measurements with 31.2 cm resolution and less than 6 m accuracy. Altimeter data combined with orbital tracking allowed the volume and the mass of Eros to be estimated to a precision of 0.01 % and 0.0001 %, respectively.

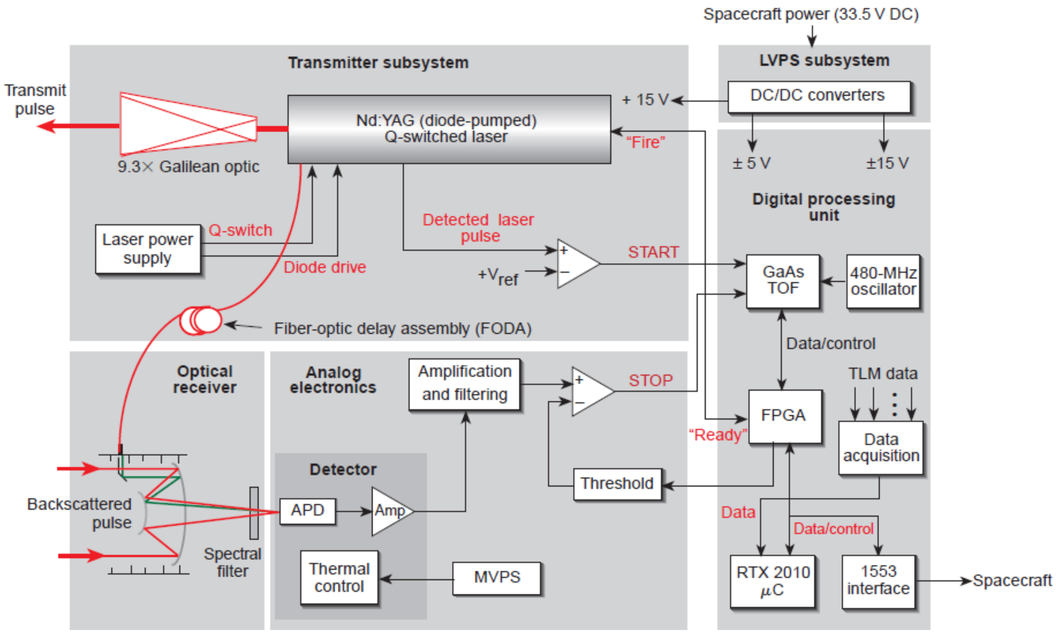


**Figure 3.5 - System architecture of the NLR [95].**

Rendezvous Laser RADAR

The Rendezvous laser Radar (RVR) [97, 98] was designed by the National Space Development Agency of Japan (NASDA) in the framework of the ETS-VII mission started in 1997. During this mission the RVR was the main sensor used for relative navigation between the chaser and the target satellites at intermediate and close range, i.e. between 0.3 m and 660 m. The RVR is a CW FM system. It emits a laser beam in a specified angle without any scanning system, by using a near-infrared laser diode which radiates pulsed light (810 nm wavelength) in an 8.5° cone. The laser light is reflected by CCRs on the target and detected on the chaser using a CCD camera and an APD. The RVR estimates the LOS angle by processing the CCD image and computes the relative range by comparing the phase difference between transmitted and received beam. One great property of this system is that it can function under the optical interference of the Sun and other active optical sensors.

Laser Dynamic Range Imager

The Laser Dynamic Range Imager (LDRI) [99], developed by Sandia National Laboratories (SNL), is a scannerless range 3D imager used to remotely measure vibration of the ISS structure and to determine the structural modal frequencies and amplitudes. It is characterized by six modes of operation, a 40° FOV, range resolution of 2.5 mm at approximately 45 m and a 7.5 Hz update rate. The LDRI operation consists in illuminating the scene with an amplitude modulated CW diffuse source, and imaging the average reflected intensity on a CCD detector, composed of 640 x 480 pixels. The sensor flew on the space shuttle flight STS-96 and provided range measurements and video of the newly installed P6 solar array panel radiators. Additional measurements of curvature in the solar array panels demonstrated the potential for on-orbit characterization or inspection of structures. Space Shuttle flights results are reported in [99].

Laser Camera System

The Laser Camera System (LCS) [79, 100] was the critical system, developed by Neptec under contract by the National Aeronautics and Space Administration (NASA), for repair decisions during two Space Shuttle missions for ISS assembly, namely STS-118 and STS-122. Although a prototype had already been tested on several previous missions starting from the STS 105, the development of this sensor was a crucial point for NASA, especially after the Space Shuttle Columbia tragedy in 2004. The LCS, depicted in Figure 3.6, is a high precision triangulation 3D laser scanner designed to carry out on-orbit inspection of the spacecraft.

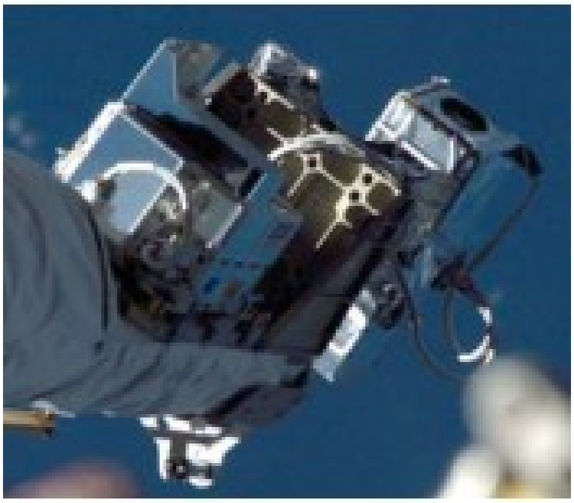


**Figure 3.6 - LCS mounted on the ISS robot arm for inspection purposes [78].**

Specifically, it was used to scan the shuttle's thermal protection system (TPS) at stand-off ranges in the interval (1, 3) m to detect cracks less than 1 mm thick and facilitate clearance of the shuttle for landing. Two moving mirrors, driven by high-precision galvanometers, steer a continuous laser beam on a target in space, creating a three-dimensional representation of the scanned area. The diffusely reflected light is captured by a collecting lens which focuses the energy on a linear detector array (LDA). The LCS has two operational modes.

- Imaging mode is mainly adopted for inspection tasks. The system scans an object and registers a voxel for each illuminated point. Each one of these voxels is identified by four parameters, namely the two galvanometer angles, the detected peak and the intensity of the reflected signal. From these data it is possible to compute the spatial coordinates of the considered point by triangulating the intersection between the projected and reflected ray paths.
- Centroid acquisition mode is used to determine the position of discrete target points on an object by projecting Lissajous patterns on circular targets with strong black and white contrasts.

The LCS also demonstrated capability to track retro-reflectors on the target surface.

Laser Mapper

The Laser Mapper (LAMP) [101, 102] was developed by the NASA's Jet Propulsion Laboratory (JPL) for five guidance and navigation applications.

- Capture of a Mars sample in Mars orbit.
- Hazard avoidance during smart landing on Mars.
- Traverse planning for Mars rovers.
- Rendezvous or docking with another spacecraft in earth orbit.
- Small body landing/exploration/mapping.

The system was used as the primary relative navigation sensor during the demonstration mission XSS-11 in order to perform the Autonomous Rendezvous Experiment (ARX). The LAMP is a TOF laser which operates by emitting short and high-power pulses of infrared light which are pointed toward the target by a two-axis gimbaled mirror that allows the system to cover an area of 10° x 10° in 1 s. The laser pulses hit the target and a small amount of the light is reflected back to the instrument, collected by a telescope and sent to the APD in order to stop the count of the TOF. A rendezvous software is combined with the LAMP in order to acquire and track the target providing range and LOS measurement. The tracking operation consists in continually scanning the FOV in a small window around the target. The size of the sub-window is determined by the angular aspect of the object being tracked (the larger the angular extent of the target is, the bigger the scan window gets). The centroid of each scan is computed to estimate LOS and range. During the mission XSS-11, for the ARX experiment, a set of retro-reflectors was mounted on the surface of the orbital sample that could be identified by the instrument up to a distance of 5 km. However, the LAMP is able to identify also a generic Lambertian surface up to a distance of 2.5 km, although the photon budget is significantly reduced.

### Rendezvous and Docking Sensor

The Rendezvous and Docking Sensor (RVS) [103] was developed by Jena-Optronik for the European Space Agency (ESA) and the Japanese Space Agency (JAXA), as well as for the American Cygnus program, in order to support and control the automated docking of unmanned transfer vehicles with the ISS. It was first used in space on board the first Automatic Transfer Vehicle (ATV-001), although the RVS prototype had already been qualified in orbit with two Space Shuttle Missions (STS-84 and STS-86) docking to the MIR space station in 1997. The RVS, depicted in Figure 3.7, is a TOF laser range finder combined with a galvanometric scanning system able to measure range and LOS of a target from a distance of about 1500 m. It uses a mirror system to send short laser pulses toward the target, e.g. the Russian module of the ISS (where the ATV docks), that are reflected toward the RVS by special retro-reflectors installed on its surface.

Jena-Optronik has recently developed the new generation version of sensors for rendezvous and docking, namely the RVS-3000 and RVS-3000 3D, which are both lighter in weight and less power consuming than the basic RVS. The former system is able to operate in cooperative mode up to a further range (3000 m), but it also measures target range and LOS in an uncooperative mode up to 100 m of range. The RVS-3000 flew as an experiment on the last European transport vehicle ATV-005. The RVS 3000 3D represents a further advancement since it is able of providing automatic target identification and tracking as well as 3D imaging (point cloud representation).

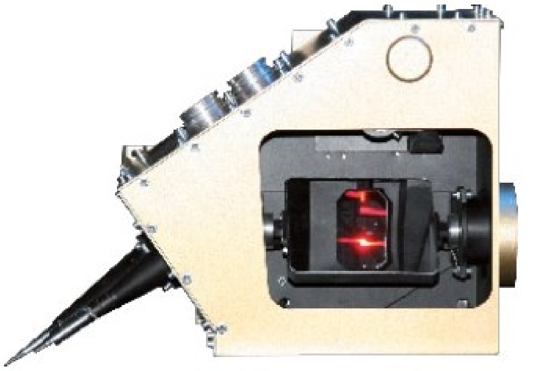


**Figure 3.7 - Lateral view of the RVS [103].**

TRIDAR

The Triangulation LIDAR sensor (TRIDAR) [92], depicted in Figure 3.8 and developed by Neptec, was used during the Space Shuttle missions STS-128 (2009), STS-131 (2010) and STS-135 (2011). During STS-128 TRIDAR provided astronauts with real-time guidance information during rendezvous and docking with the ISS. During STS-131 TRIDAR tracked the ISS for rendezvous and docking as well as for undock and fly-around operations. During STS-135 TRIDAR repeated its tracking demonstration with improved performance from hardware and software upgrades. It represents the first vision system used in space able to automatically acquire and track a space target in an uncooperative way that means using only knowledge about its shape. TRIDAR operates at distances between 0.5 m and 2000 m. It combines a short-range, high precision and auto-synchronous triangulation sensor, i.e. a modified version of the previously described LCS, with a mid-to-long-range TOF LIDAR sensor in the same unit (in a way similar to LARS). The two sensors share the same optical path and control electronics resulting in a compact package with multi-range and multi-role capabilities. The data collected by the system can be used to perform 6-DOF real-time tracking of the target pose. A further version of the TRIDAR has been selected to support the operations of autonomous rendezvous and docking for the Hubble Robotic Vehicle (HRV). This system will operate at short range, i.e. below 10 m, and mid-range, i.e. between 10 m and 150 m.

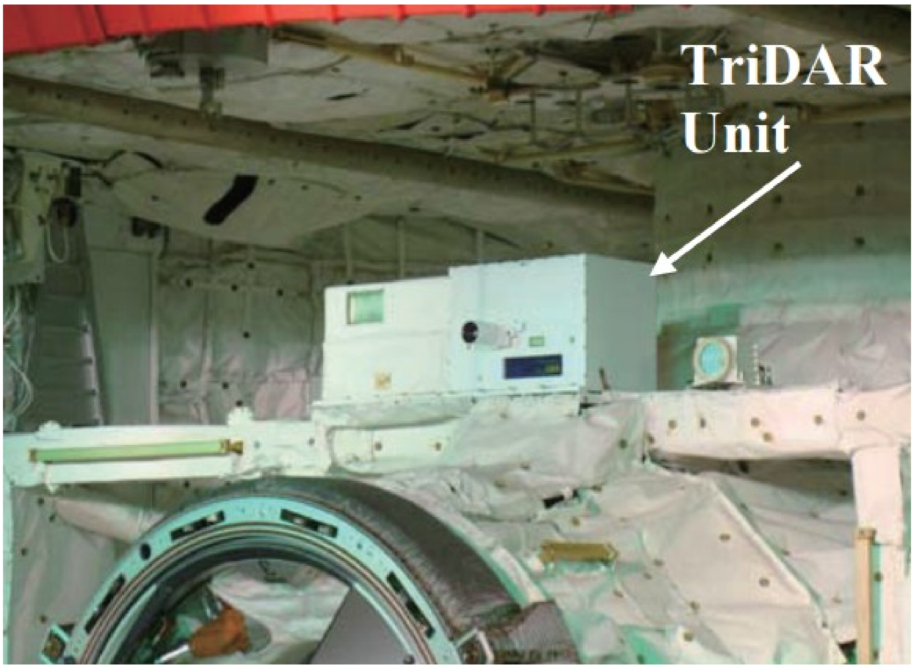


**Figure 3.8 -TRIDAR mounted on the shuttle's cargo bay [104].**

Hawkeye

Hawkeye [78] is the laser range finder recently developed by Neptec, qualified for use in Geosynchronous Orbit and specifically designed to provide ranging as well as satellite situational awareness for FF applications. Hawkeye, depicted in Figure 3.9, is able to measure the target range up to a nominal distance of 30 km, although its operational range could exceed this limitation depending on the target size and surface characteristics (large size and highly reflecting materials). Since it weighs less than 3 kg and consumes low power (2.2 W), it is particularly suitable for application on small satellites.

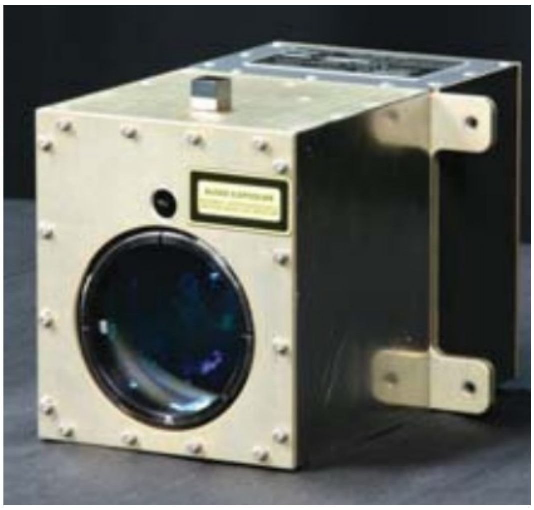


**Figure 3.9 - Hawkeye laser range finder [78].**

DragonEye 3D Flash LIDAR Space Camera

The DragonEye Space Camera [105] is a lightweight flash LIDAR, developed by Advanced Scientific Concepts Inc. (ASC Inc.), characterized by a full array of 128 x 128. Each pixel is able to measure TOF at a frequency up to 30 Hz, allowing 16300 3D range data and intensity points to be generated as 3D point-cloud images or video streams in real time. This system, designed under contract by NASA, is conceived for autonomous rendezvous and docking applications, and a prototype successfully flew during STS-127 and STS-133 missions for performance assessment. This flash LIDAR is composed of three main items contained within a compact small case depicted in Figure 3.10.

- One 3D sensor engine.
- One laser illuminator with lens/diffuser.
- One cooling surfaces/vacuum enclosure.

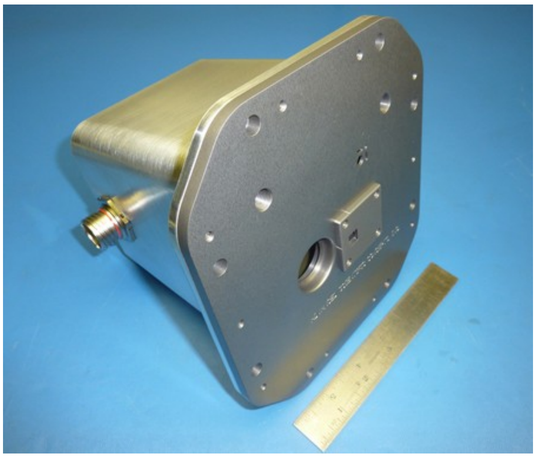


**Figure 3.10 - DragonEye Flash LIDAR Space Camera [104].**

GoldenEye 3D Flash LIDAR Space Camera

The GoldenEye Space Camera [105] is another flash LIDAR developed by ASC Inc. It is designed for deep space, geosynchronous or landing operations, during which it admits two configurations, namely the Space Operations LIDAR (SOLID) and the Geosynchronous 3D (GEO-3D). The choice between SOLID and GEO-3D depends on the application requirements.

An example of deep space operation using SOLID GoldenEye is given by the OSIRIS-REX Asteroid Sample Return mission. The SOLID GoldenEye Space Camera has similar specification and performance if compared to the DragonEye one. However, it is heavier and more power consuming, but it is also able to provide a wider operational range, i.e. up to 3000 m instead of 1500 m.

A summary of the documented information about the sensors listed above is given in Table 3.1(a) and Table 3.1(b), where the symbol *N/F* (not found) is used to indicate those specifications which are not open-access. Moreover, the symbol *N/A* (not applicable) is used for the "FOV" slot of all the laser range finders, i.e. NLR, RVR and Hawkeye, since, by definition, they take measurements from a single direction.

| **System (developer)** | **Operational mode** | **Technology & measurement principle** | **Operational range (m)** | **Documented accuracy** |
|---|---|---|---|---|
| **LARS**<br>(CSA) | cooperative | Scanning<br>- CW Triangulation<br>- pulsed TOF | - 0.5 ÷ 10<br>- 10 ÷ 10000 | - sub-mm<br>- 3 cm |
| **NLR**<br>(APL) | non-cooperative | Laser range finder<br>Pulsed TOF | < 160000 | < 6 m |
| **RVR**<br>(NASDA) | cooperative | Laser range finder<br>CW FM | 0.3 ÷ 660 | 23 cm (3$\sigma$)<br>0.02° (3$\sigma$) |
| **LDRI**<br>(SNL) | non-cooperative | Scannerless<br>CW AM | < 45 | 0.25 cm |
| **LCS**<br>(Neptec) | - cooperative<br>- non-cooperative | Scanning<br>CW Triangulation | 1 ÷ 10 | 0.1 mm ÷ 5 mm (1$\sigma$) |
| **LAMP**<br>(JPL) | - cooperative<br>- non-cooperative | Scanning<br>Pulsed TOF | - < 5000<br>- < 2500 | - 10 cm (bias)<br>2.6 cm (3$\sigma$)<br>0.034° (3$\sigma$)<br>- *N/F* |
| **RVS**<br>(Jena-Optronik) | cooperative | Scanning<br>Pulsed TOF | 1 ÷ 1500 | 0.01 m ÷ 0.5 m (bias)<br>0.01 m ÷ 0.1 m (3$\sigma$)<br>0.1° (bias)<br>0.1° (3$\sigma$) |
| **RVS-3000**<br>(Jena-Optronik) | - cooperative<br>- non-cooperative | Scanning<br>Pulsed TOF | - 1 ÷ 1500<br>- 1 ÷ 100 | *N/F* |
| **TRIDAR**<br>(Neptec) | non-cooperative | Scanning<br>- CW Triangulation<br>- Pulsed TOF | 0.5 ÷ 2000 | *N/F* |
| **Hawkeye**<br>(Neptec) | non-cooperative | Laser range finder<br>Pulsed TOF | 50 ÷ 30000 | 2 m (bias)<br>45 mm (3$\sigma$) |
| **DragonEye**<br>(ASC Inc.) | non-cooperative | Scannerless<br>Pulsed TOF | < 1500 | 10 cm (bias)<br>15 cm (3$\sigma$) |
| **GoldenEye**<br>(ASC Inc.) | non-cooperative | Scannerless<br>Pulsed TOF | < 3000 | 10 cm (bias)<br>15 cm (3$\sigma$) |

**Table 3.1(a) - Survey of currently existing and under development spaceborne LIDAR systems (part a).**

| **System (developer)** | **Mass (kg)** | **Power Consumption (W)** | **Operating wavelength (nm)** | **FOV (°)** | **Data rate (Hz)** |
|---|---|---|---|---|---|
| **LARS**<br>(CSA) | 6 | 6 | 1540 | 30x30 | *N/F* |
| **NLR**<br>(APL) | 5 | 15 | 1064 | *N/A* | 0.125 ÷ 8 |
| **RVR**<br>(NASDA) | *N/F* | *N/F* | 810 | *N/A* | 2 |
| **LDRI**<br>(SNL) | 2.3 | 37 | 807 | 40x40 | 7.5 |
| **LCS**<br>(Neptec) | 12.1 | 75<br>(max) | 1500 | 30x30 | 5 |
| **LAMP**<br>(JPL) | 5.9 | 35 | 1064 | 10x10 | 2 |
| **RVS**<br>(Jena-Optronik) | 14.5 | 40 ÷ 70 | 910 | 40x40 | *N/F* |
| **RVS-3000**<br>(Jena-Optronik) | 8 | 30 ÷ 50 | 1500 | 40x40 | 1 ÷ 4 |
| **TRIDAR**<br>(Neptec) | 15.9 | 70 | 1540<br>(TOF)<br>1400<br>(triangulation) | 30x30 | *N/F* |
| **Hawkeye**<br>(Neptec) | 3 | 2.2 ÷ 20 | 1064 | *N/A* | *N/F* |
| **DragonEye**<br>(ASC Inc.) | 3 | 35 | 1570 | 45x45 | 5 ÷ 30 |
| **GoldenEye**<br>(ASC Inc.) | 6.5 | 50 | 1570 | 45x45 | 5 ÷ 10 |

**Table 3.1(b) - Survey of currently existing and under development spaceborne LIDAR systems (part b).**

By looking at these data it is possible to make the following statements.

Those systems which are designed to deal with cooperative targets are characterized by extremely large intervals of operating distances and precise and accurate estimates of relative position and angular parameters. Indeed, this depends on the much larger intensity of the backscattered radiation which can be collected at the detectors' surface if the target is covered by retro-reflectors with respect to the one obtained by imaging Lambertian (uncooperative) surfaces. However, also the sensors designed for non-cooperative applications are able to provide satisfying accuracy levels but within a narrower range of distances. Among them, scanning systems like LCS and LAMP are better performing than flash LIDAR. The real limitation of LIDAR is given by the related specifications in terms of weight and demanded power. From this point of view, scannerless systems have some advantages being lighter and slightly less power consuming than scanning sensors. Due to the fact that the FOV is imaged instantaneously by all the pixels composing the detector, scannerless systems can also provide more frequent measurements but they are characterized by poorer resolution. All the listed LIDAR have comparable characteristics in terms of the size of the FOV. Specifically, it is limited around 30° and 40° both in azimuth and elevation. On one side, this choice certainly allows reducing the amount of data to be processed and, consequently, the computational load. However, it also determines the necessity to improve pose determination algorithms performance in terms of computational efficiency in order to avoid losing track of the target. Indeed, this phenomenon can occur when dealing with uncooperative non-controllable targets, i.e. debris, which are typically characterized by a fast tumbling motion.

As a result of this review, the operation of a pulsed TOF LIDAR is selected to be reproduced by the simulator presented in the next sub-chapter. This choice is consistent with the willingness of assessing the performance of the developed pose determination algorithms against highly variable conditions in terms of relative dynamics. Indeed, the TOF measurement principle provides reasonable and constant accuracy within an extremely wide interval of relative distances, unlike triangulation, AM and FM methods which are relative to CW LIDAR.

## 3.3 LIDAR measurement simulator

The purpose of this sub-chapter is to describe the simulator designed to realistically reproduce the operation of a pulsed TOF LIDAR. The simulator is composed of three blocks, i.e. the geometric, the detection and the noise models, whose details will be presented in the following. Two distinct operative modules are foreseen, as it can be seen in Figure 3.11, each of which having a specific goal.

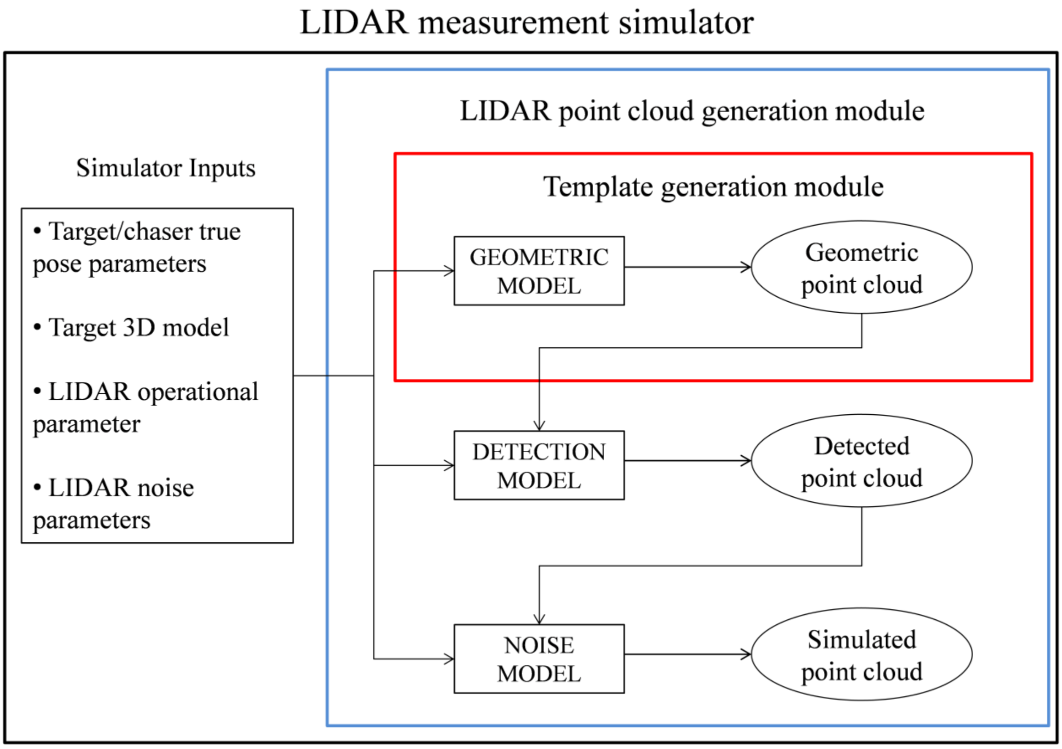


**Figure 3.11 - LIDAR measurement simulator: LIDAR point cloud and template generation modules.**

The template generation module (enclosed in the red box in Figure 3.11) relies exclusively on the geometric model and it is part of the on-board software implementing the proposed acquisition algorithms, as previously shown in Figure 2.12.

The LIDAR point cloud generation module (enclosed in the blue box in Figure 3.11) exploits all the models composing the simulator in order to generate the synthetic point clouds which are then processed by the pose determination algorithms presented in sub-chapters 2.4 and 2.5.

### 3.3.1 Geometric model

Ray tracing is a technique, commonly adopted in computer graphics, able to generate synthetic 2D images by determining the interceptions (tracing) between the path travelled by the light along the directions identified by all the pixels composing the camera focal plane and virtual objects located within a pre-defined 3D environment [106]. The geometric model of the LIDAR measurement simulator is based on an algorithm which aims at extending the principle of ray tracing to generate 3D synthetic images of an assigned target. The input required by this algorithm is limited to the true relative position and attitude parameters between the TRF and SRF as well as to a model representation of the target (whose details are discussed in sub-chapter 3.5).

Firstly, the algorithm determines whether the planar surfaces composing the target model are in sight with respect to the LIDAR or occluded. This is done by computing, for each surface, a visibility parameter ($I_S$), defined as the dot product between the vector identifying the position of the geometric center of the considered surface with respect to the sensor ($\underline{P}_{CS}$) and the corresponding normal unit vector ($\underline{n}_S$), as shown by eq. (3.1) where all the quantities are expressed in SRF.

$$I_S = \underline{P}_{CS} \cdot \underline{n}_S \tag{3.1}$$

All the surfaces for which $I_S$ is positive or zero shall be considered occluded, thus being able to identify a non-zero number ($N_S$) of visible planar surfaces.

The following step consists in determining the interceptions between all the laser beam directions (which sample the observed FOV) and the planes corresponding to each of the $N_S$ visible surfaces. Specifically, the range of the interception ($R$) is obtained by solving the system of equations composed of the parametric representation of a straight line and the Cartesian representation of a plane, as shown in eq. (3.2),

$$\begin{cases} x = DC_x R \\ y = DC_y R \\ z = DC_z R \\ n_{Sx}x + n_{Sy}y + n_{Sz}z + d_S = 0 \end{cases} \Rightarrow R = \frac{-d_S}{n_{Sx}DC_x + n_{Sy}DC_y + n_{Sz}DC_z} \qquad (3.2)$$

where $d_S$ is the known term in the Cartesian representation of a flat surface; $x$, $y$, and $z$ are the Cartesian coordinates of the target point hit by the laser beam; $n_{Sx}$, $n_{Sy}$ and $n_{Sz}$ are the coordinates of $\underline{n}_S$; $DC_x$, $DC_y$ and $DC_z$ are the direction cosines of the LOS identified by the laser beam.

At this point, each laser beam will have $N_S$ possible interceptions but most of them are not consistent with the actual size of the target, meaning that all the interceptions falling outside of the geometrical boundaries of the target surfaces must be discarded. This is done by verifying that a condition of consistency is met. Specifically, the distance of the computed interception to the center of the surface must be upperly limited by a scalar parameter, depending on the shape (e.g. rectangular, circular, elliptical) of the surface. Hence, if more than one point still represent plausible interceptions, the ambiguity is solved by considering the one characterized by the minimum value of $R$.

At the end of this procedure, an ideal, i.e. purely geometric, point-cloud representation of the target can be obtained, although the real output is given by the ideal values of $R$.

The method described above is valid if the target model is exclusively composed of planar surfaces. However, its applicability can be easily extended to more complicated models including curve (3D) surfaces (e.g. paraboloids, hyperboloids, cylinders). Specifically, this is done by substituting the Cartesian representation of a flat surface in eq. (3.2) with the Cartesian representation of the geometric curve of interest. Moreover, the 3D surfaces are always considered in sight. If the target model includes both curve and planar surfaces, the interceptions of any laser beam direction with the two typologies of surface must be computed (and verified for consistency) separately. Hence, potential ambiguities are again solved by considering the interception with the minimum value of *R*.

### 3.3.2 Detection model

Once the geometric point cloud is computed, the LIDAR detection process is simulated in detail taking all the radiometric aspects into account. This establishes whether the backscattered laser beams are detected or not, thus practically extracting only the detected values of *R*. To this aim the probability of detection ($P_D$) of each received echo is evaluated by using eq. (3.3), as a function of the probability of false alarm ($P_{FA}$) and the Signal to Noise Ratio (*SNR*) [107].

$$P_D = 0.5\{1 + erf[(0.5 + SNR)^{0.5} - 0.5\ln\frac{1}{P_{FA}}]\} \tag{3.3}$$

The $P_{FA}$ is expected to be very low for spaceborne LIDAR, so it is set equal to $10^{-4}$. On the other hand, the derivation of the *SNR* can be carried out by first recalling the concept of the LIDAR equation.

LIDAR equation

The LIDAR equation governs the process of propagation of a laser signal and it basically provides the mathematical relation between the amount of detected power ($Pw_{DET}$) as a function of the transmitted one, in a way analogous to the original

RADAR equation [108]. The LIDAR equation is here derived by taking advantage of the following assumptions.

- The target is extended (well-resolved), meaning that the surface area which contributes to the target reflectivity is limited by the size of the illuminating beam rather than by the target dimension.
- The retro-reflection angle ($\theta_{REF}$), identified by the power backscattered toward the detector, is equal to the angle of incidence of the transmitted laser beam ($\theta_{INC}$).

The former assumption is valid for spaceborne applications in the range of distances of interest (tens of meters) since the area covered by the laser spot is much lower than the size of the target due to the extremely limited value of the beam divergence ($\theta_B$) typical for LIDAR. The latter assumption is justified by the fact that LIDAR have typically a monostatic configuration, i.e. the displacement between the transmitting and receiving components is neglible.

Firstly, the instantaneous power transmitted by the laser source ($Pw_{TR}$) is computed in eq. (3.4), in the case of a pulsed LIDAR, as a function of the average laser power ($Pw_{AVG}$), the pulsed repetition frequency ($PRF$) and the pulse width ($t_W$).

$$Pw_{TR} = \frac{Pw_{AVG}}{PRFt_W} \tag{3.4}$$

As regards the incident power ($Pw_{INC}$), if the incident laser beam is perpendicular to the target surface ($\theta_{INC}$ = 0°), it can be obtained by simply multiplying $Pw_{TR}$ with $\tau_A$, i.e. the atmospheric transmission loss factor. However, this latter parameter, which models the loss of power due to the absorption and scattering by atmospheric molecules along the propagation direction of the transmitted signal, can be set equal to 1 for spaceborne applications. Hence, in the most general case ($\theta_{INC} \neq 0°$), the loss of incident power is ruled by the cosine of $\theta_{INC}$ as it is shown in Figure 3.12.

At this point, it is necessary to compute the amount of power reflected by the target surface ($Pw_{REF}$). In the most general case, this can be done by introducing the bidirectional reflectance distribution function ($BRDF_\lambda$), which relates the energy incident on a point within an illuminated area to the reflected one as a function of the directions of incidence and reflection, the material and roughness of the local surface, and the wavelength of the incident radiation ($\lambda$) [109]. In the context of the analyzed application, the statements listed below can be made.

- Since the target is well-resolved, the surface material can be considered homogeneous over the illuminated area, and the points of incidence and reflection are coincident.
- The incident radiation transmitted by the LIDAR is uniform and isotropic, e.g. in terms of $\lambda$, within $\theta_B$.

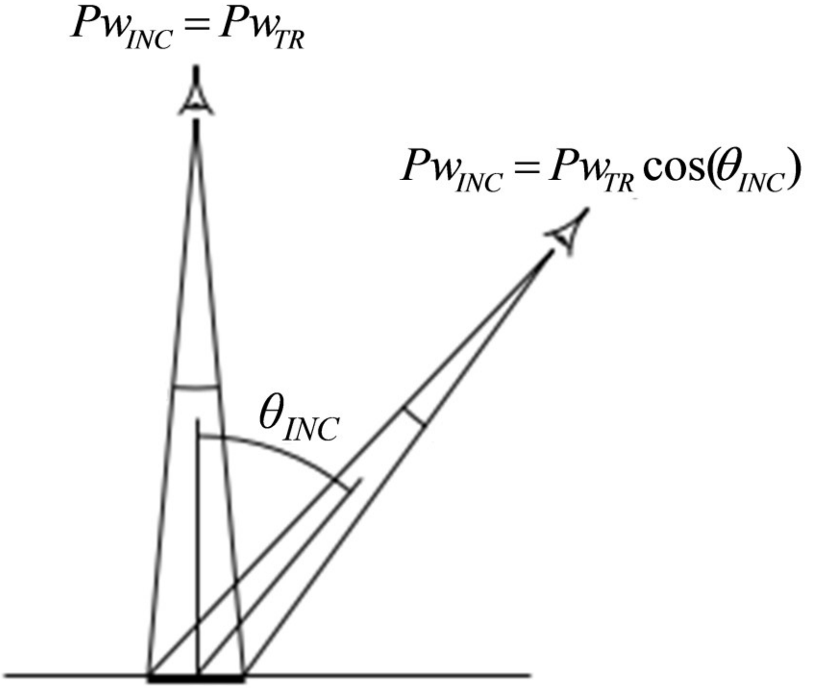


**Figure 3.12 - Lambert's cosine law. The loss of power due to absorption and scattering by atmospheric molecules is negligible in space.**

Under these considerations, the $BRDF_\lambda$ and, consequently, the reflection coefficient ($\rho_{T\lambda}$), i.e. the ratio between the reflected and incident power, are constant within the illuminated area and depend on $\lambda$ and on the surface material. Moreover, the relation between $BRDF_\lambda$ and $\rho_{T\lambda}$ is given by eq. (3.5),

$$\rho_{T\lambda} = BRDF_{\lambda}\theta_R \tag{3.5}$$

where $\theta_R$ is the solid angle in which the backscattered radiation is dispersed. Hence, $Pw_{REF}$ can be computed by exploiting eq. (3.6)

$$Pw_{REF} = \rho_T Pw_{INC} = \rho_T Pw_{TR}\cos(\theta_{INC}) \tag{3.6}$$

Once $Pw_{REF}$ is given, the power arriving at the receiver per unit of area ($I_R$) can be computed thanks to eq. (3.7),

$$I_R = \frac{Pw_{REF}}{\theta_R R^2}\tau_A = \rho_{T\lambda}\frac{Pw_{TR}\cos(\theta_{INC})}{\pi R^2} \tag{3.7}$$

where the effect of $\tau_A$ during the propagation of the returning signal is still negligible and $\theta_R$ is set equal to π since the surfaces of space targets can be considered Lambertian thus ensuring purely diffusive reflection.

At this point, if the receiver area ($A_R$) is identified by the aperture diameter ($D_{\mathrm{A}}$), the $Pw_{DET}$ can be determined by using eq. (3.8) which takes the power loss due to the transmittance of the optics ($\tau_O$) into account. Specifically, the optics of a LIDAR system is typically composed of an optical band-pass filter, whose bandwidth ($\Delta\lambda$) is centered at the wavelength of the laser source.

$$Pw_{DET} = I_R A_R \tau_O = I_R \frac{\pi D_A^{\ 2}}{4}\tau_O \tag{3.8}$$

Hence, a final expression for $Pw_{DET}$ is given by eq. (3.9) which can be obtained by substituting eq. (3.7) into eq. (3.8).

$$Pw_{DET} = \rho_{T\lambda}\frac{Pw_{TR}\cos(\theta_{INC})D_A^{\ 2}}{4R^2}\tau_O \tag{3.9}$$

*SNR* derivation for LIDAR

The *SNR* is typically defined as the power ratio between the signal (meaningful information) and the background noise. However, an alternative definition is here considered which allows computing the *SNR* as the ratio between the signal expected value and the noise standard deviation [108].

The signal expected value is the average number of photoelectrons ($\mu_{SIG}$) produced by the detector when a laser beam (backscattered from the target) falls on its surface. This quantity is strictly related to the number of photons ($\Gamma$) hitting the detector during the integration time of its circuit ($\Delta t$), which can be put equal to $t_W$ for this application [108]. However, the photons arrive at random times thus introducing an uncertainty in the value of $\Gamma$. This effect is known as photon counting noise and it can be evaluated by modeling $\Gamma$ as a Poisson random variable whose mean ($\mu_\Gamma$) is equal to the detected energy, given by eq. (3.9), divided by the energy per photon, given by the Planck's law. Hence, $\mu_{SIG}$ can be computed as in eq. (3.10) by including the effect of the quantum efficiency of the detector ($\eta$) which is a measurement of a device's electrical sensitivity to light, i.e. the ratio between electron generation rate and photon incident rate.

$$\mu_{SIG} = \eta\mu_\Gamma = \eta \frac{Pw_{DET}t_W}{h\nu} \tag{3.10}$$

In the equation above, $h$ is the Plank constant and $v$ is the frequency of the electromagnetic radiation.

As regards the noise standard deviation, the main phenomena affecting the performance of LIDAR are the laser speckle, the thermal noise and the background noise.

The laser speckle noise effect is a statistical fluctuation of the light arriving at the detector caused by interference occurring there from a large collection of independent coherent radiators. As a consequence of this phenomenon, if the object imaged by a laser system is composed of surfaces that are rough on the scale of optical wavelengths (as most objects are), the image is found to have a granular appearance, with a multitude of bright and dark tiny spots that bear no apparent relationship to the macroscopic scattering properties of the object [110]. However, since interference is a phenomenon which is related to the degree of coherence of the electromagnetic waves and TOF LIDAR are direct-detection systems which do not need coherent laser sources, the laser speckle noise contribution can be neglected for the analyzed case.

The thermal noise is a problem which arises at the detector level since any object at a temperature different than 0 K radiates photons. The variance of the number of thermal noise electrons ($\sigma_{TH}^{2}$) depends on the detector temperature (*Temp*) and on the capacitance of the detector circuit (*Ca*) [108], as shown in eq. (3.11),

$$\sigma_{TH}^{2} = \frac{k_b TempCa}{q_e^{2}} \tag{3.11}$$

where $k_b$ is the Boltzmann constant, while $q_e$ is the electron elementary charge. In some cases, *Ca* is not available within the detector's specification. Hence, the thermal noise can be derived as a function of the average number photoelectrons due to the dark current ($\mu_{DARK}$), i.e. the limited amount of electric current which flows on the detector surface even if no photons are entering the device [108].

The background noise is a measure of the total number of unwanted photons collected by the detector but not originated by the laser transmitter. These background photons do not carry any information concerning the range to the target thus contributing to the system measurement noise. The total number of unwanted photoelectrons can be modeled as a Poisson random variable whose variance ($\sigma_{BACK}^{2}$) can be obtained by adding the dark current contribution to the background one as done in eq. (3.12),

$$\sigma_{BACK}^{2} = \eta \frac{Pw_{BACK} t_W}{h\nu} + \mu_{DARK} \tag{3.12}$$

where $\mu_{DARK}$ is given by eq. (3.13) if $i_D$ is defined as the mean intensity of the dark current.

$$\mu_{DARK} = \frac{i_D t_W}{q_e^2} \tag{3.13}$$

Equation (3.12) relies on the fact that a Poisson distribution has equal mean and variance, and it is obtained by substituting $Pw_{DET}$ in eq. (3.10) with the power detected from background ($Pw_{BACK}$). This latter quantity mainly depends on the photons coming from the sun and reflected on the target surface toward the LIDAR detector according to eq. (3.14),

$$Pw_{BACK} = \rho_{T\lambda} \frac{S_{IB} \Delta\lambda dA D_A^2}{4R^2} \tau_O \tag{3.14}$$

where $S_{IB}$ is the solar irradiance level at mean Earth-Sun distance in units of W/(m$^2$µm) of electromagnetic bandwidth, and $dA$ is the surface area of the target illuminated by the laser computed by means of eq. (3.15).

$$dA = \frac{\pi (R\theta_B)^2}{4} \tag{3.15}$$

Once all the main sources of noise have been statistically modeled, the *SNR* can be computed by means of eq. (3.16), where the overall noise standard deviation is approximated as the square root of the sum of the variances of each individual noise contributions

$$SNR = \frac{\mu_{SIG}}{\sqrt{\sigma_{TH}^2 + \sigma_{BACK}^2}} \tag{3.16}$$

This approximation is valid since the individual terms can be considered statistically uncorrelated [108]. However, the final expression for the *SNR*, adopted in the simulator, derives from the use of an APD as detector. Indeed, the APD is able to produce a surplus of photoelectrons, i.e. the avalanche, from a single incoming photon . Hence, the exact amount of photoelectrons induced by each photon is given by the APD Gain ($G_{APD}$), thus obtaining the following expression for the $SNR_{APD}$.

$$SNR_{APD} = \frac{G_{APD}\mu_{SIG}}{\sqrt{\sigma_{TH}^2 + G_{APD}\sigma_{BACK}^2}} \tag{3.17}$$

It is worth outlining that, if APDs are used, $\eta$ measures the probability with which the avalanche occurs.

Detection process simulation

Once the *SNR* is evaluated, the detection process can be simulated by adopting a statistical approach similar to the one used in [111]. For each point of the geometric point cloud, the $P_D$ is computed using eq. (3.3) and a random number is extracted from a uniform distribution in the interval (0, 1). All the geometric points and consequently the ideal values of $R$ for which the random extraction produces a number larger or equal to the corresponding value of $P_D$ is considered to be part of the detected point cloud, thus extracting only the detected values of $R$.

### 3.3.3 Noise model

The reproduction of the operation of a pulsed TOF LIDAR can be considered complete if the 3D position of the points composing the detected point cloud is modified by taking all the main sources of noise into account. Specifically, it is necessary to consider the range uncertainty, the pointing uncertainty and the effect caused by the presence of outliers within the acquired data. The solutions adopted to model these noise contributions are described hereunder.

The range uncertainty is directly dependent on the error in measured TOF and it is modeled as a Gaussian white noise (0, $\sigma_{RANGE}$) to be added to the detected values of $R$ corresponding to each backscattered laser beam.

The pointing uncertainty is the angular error between the ideal laser beam direction ($\underline{L}_{IDEAL}$) and the real one ($\underline{L}_{NOISE}$) and it is reproduced by adopting a method similar to the approach proposed in [112]. Specifically, the angular separation between $\underline{L}_{NOISE}$ and $\underline{L}_{IDEAL}$ ($\nu_{LOS}$) is extracted from a normal distribution (0, $\sigma_{LOS}$). Then, $\underline{L}_{NOISE}$ is rotated around $\underline{L}_{IDEAL}$ of an angle ($\varepsilon_{LOS}$) extracted from a uniform distribution in the interval (0, 2π). This procedure is summarized in Figure 3.13.

It is now necessary to clarify the meaning of $\underline{L}_{IDEAL}$ and $\underline{L}_{NOISE}$. The former parameter identifies the directions along which the reflected light is ideally collected. These directions are used to determine the true ranges of interception and consequently the geometric point cloud. On the other hand, $\underline{L}_{NOISE}$ represents the noised directions provided in output by the sensor due to its pointing uncertainty. These directions are combined with the noised values of $R$ to obtain the simulated point cloud. Hence, this uncertainty causes a deformation of the simulated point cloud with respect to the geometric and detection ones.

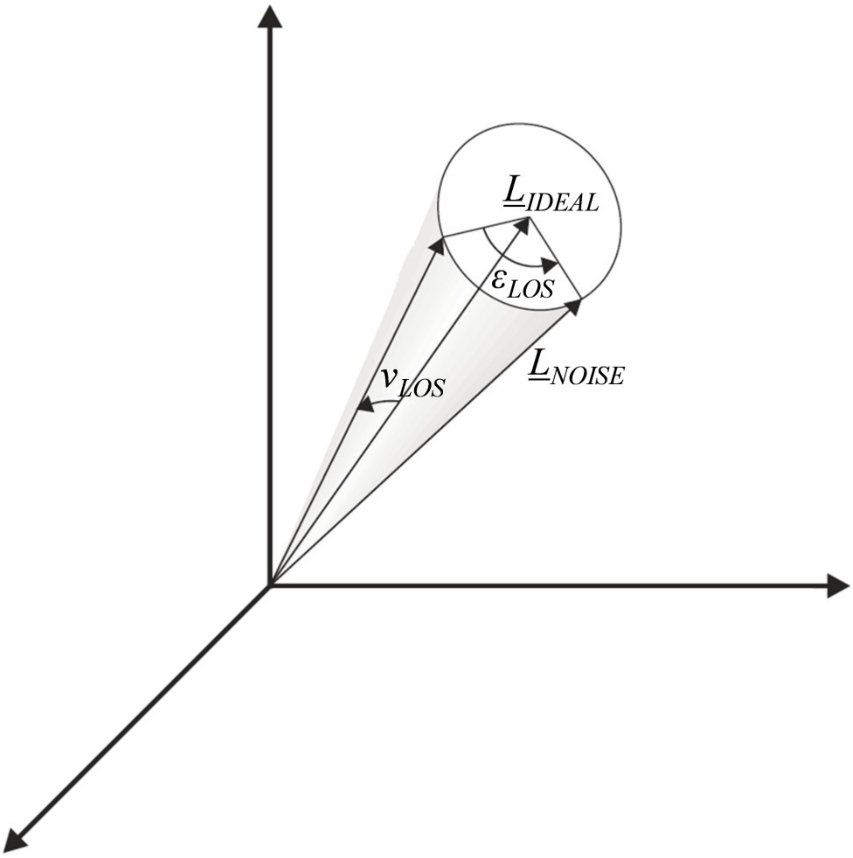


**Figure 3.13 - Modeling adopted for the pointing uncertainty of the simulated LIDAR.**

It is worth outlining that the effect of pointing and range uncertainties could be simulated in a different way. Firstly, the true ranges of interception are computed relying on the noised directions. Secondly, the noised ranges are obtained by superimposing the effect of $\sigma_{RANGE}$ on the true ones (after having discarded the misdetections). Finally, the simulated point cloud is generated by combining the noised ranges with the ideal directions. However, the pointing uncertainty is typically so low for spaceborne LIDAR, that the two procedures do not provide significant variations in the result.

In conclusion, the presence of outliers within the measured point cloud is also reproduced. Specifically, it is modeled as an assigned percentage of points whose range uncertainty is four times larger than $\sigma_{RANGE}$ [56]. This allows considering the effects on the LIDAR measurements of possible multipath phenomena which can occur on the surfaces of space targets, typically because they might be covered by several external devices which are not represented within the target 3D model exploited by the simulator. Of course, different percentages can be assigned to different kinds of surface.

## 3.4 Characterization of the simulated LIDAR system

The purpose of this sub-chapter is to present (and motivate if necessary) the selection of all the operational and noise parameters adopted within the previously described models to realistically generate 3D point clouds provided by a pulsed TOF LIDAR.

As regards the technological solution, the simulator is designed to be flexible, meaning that it can be used to reproduce the operation of both scanning and flash LIDAR, since the only difference is in the values of $\underline{L}_{IDEAL}$. Indeed, for scanning sensors, the ideal laser beam directions are identified by the scan pattern. Specifically a raster scan pattern is implemented in the numerical simulations realized for performance assessment of the developed pose determination algorithms since it allows to uniformly sample the FOV. However more complicated patterns, e.g. Lissajous, rosette and spirals, able to ensure faster scan rate, can also be implemented. On the other hand, the ideal directions of flash LIDAR are identified by the centers of each pixel.

A squared FOV of 40°x40° is selected to be imaged by the LIDAR at 1 Hz measurement rate. These values are consistent with the specifications of spaceborne systems, as previously shown in Table 3.1(b). Also the angular resolution ($\delta_{LOS}$) is selected to be the same in azimuth and elevation, and it is set to 1° during the numerical simulations performed. These values correspond to an extremely poor resolution if compared to typical performance of both flash LIDAR (e.g. $\delta_{LOS}$ is around 0.35° for both the DragonEye and the GoldenEye cameras) and scanning LIDAR (for which $\delta_{LOS}$ can even be lower than 0.01°). This choice is justified by the willingness of evaluating the performance of pose estimation algorithms against extremely challenging conditions in terms of sparseness of the point clouds. Indeed, if the representation of the target is so sparse, the size of the point clouds is significantly reduced with respect to the one exploited by the pose determination algorithms available in the literature. Hence this choice, on one side, allows improving the computational efficiency (less data to process), but on the other side it makes the pose determination problem more prone to ambiguity issues. In conclusion, it is worth outlining that both FOV and $\delta_{LOS}$ can be freely tuned by the user for additional performance analyses.

The detection model is characterized by assigning all the operational parameters of the LIDAR transmitter and receiver considering as reference typical data of systems adopted in space applications.

As regards the output signal wavelength, two alternative choices are available since LIDAR are active EO sensors operating in the near infrared (NIR) or in the short-wavelength infrared (SWIR) bands of the electromagnetic spectrum. A 1064 nm wavelength is generally used for remote sensing applications, like for the NLR. However, by looking at Table 3.1(a), it is possible to see that the same value of $\lambda$ is used also by the LAMP and the Hawkeye. Also other systems, i.e. the LDRI and the RVS, operate in the SWIR band, but adopting lower wavelengths, respectively 807 nm and 910 nm. This choice could represent a significant issue regarding the possibility to use them during manned space activities since the corresponding wavelengths are not eye-safe. This problem can be solved by selecting greater wavelengths, i.e. in the NIR band. Indeed, this region is not only eye-safe (i.e. the strong absorption by water present in the eye reduces the optical power reaching the retina, thus increasing the maximum permissible exposure level for the transmitted power) but it also allows improving the *SNR* since it is characterized by a relevant reduction of $S_{IB}$. For instance, $S_{IB}$ is 300.8 W/(m$^2$μm) if λ is 1500 nm, while it becomes more than 50 % larger, i.e. 613.9 W/(m$^2$μm) if λ is 1064 nm [113]. This motivates the choice of a 1540 nm wavelength for this SNR computation like in the case of the LARS and the TRIDAR in their TOF operational mode. The corresponding value of $S_{IB}$ is 268.6 W/(m$^2$μm).

As shown by eq. (3.4), the LIDAR transmitted power depends on three parameters. The $Pw_{AVG}$ is set to 1 mW which represents a conservative choice considering that $Pw_{AVG}$ varies in the interval (1, 100) mW for the LARS while it is constantly kept equal to 150 mW for the LCS. On the other hand, the selection of the *PRF* must be done avoiding ambiguities between return signals. To this aim, the Pulse Repetition Interval (*PRI*) must be longer than the round-trip time of the pulse to the maximum detectable distance ($R_{MAX}$), thus obtaining a maximum allowable value for the *PRF* ($PRF_{MAX}$), as stated by eq. (3.18).

$$PRF = \frac{1}{PRI} < PRF_{MAX} = \frac{c}{2R_{MAX}} \tag{3.18}$$

In spite of this limiting condition, the $PRF_{MAX}$ is very high at the relative distances typical of close-proximity operations, i.e. some tens of meters. For instance, it is 1MHz if $R_{MAX}$ is 150 m. Hence, the PRF is set equal to 10 kHz, which is in line with typical performance of spaceborne laser scanners, e.g. 18 kHz for the LARS, 10 kHz for the LAMP, and varying in the interval (8, 12) kHz for the TRIDAR. Finally, the $t_W$ is set equal to 10 ns like for the LARS. It is worth nothing that this value is consistent with the choice adopted for the *PRF* since the $t_W$ is a few orders of magnitude smaller than the *PRI*, thus avoiding ambiguities. An additional parameter which is relative to the transmitting LIDAR components is the laser beam divergence which is set equal to 0.02° like for the LAMP.

As regards the LIDAR receiver, it is necessary to select the aperture diameter, the detector, and the optical band-pass filter. Firstly, the aperture size is set equal to 4 cm which is in line with the 2.5 cm of the LARS and the 5 cm of the LAMP [5]. Secondly, the selected detector is the G8931-20 InGaAs APD produced by Hamamatsu [114] since it works in the interval (900, 1700) nm thus being well suited to be used within spaceborne LIDAR. Finally, an optical band-pass filter, produced by ThorLabs Inc., is selected [115]. Specifically, it has a center wavelength of 1540 nm and $\Delta\lambda$ equal to 24 nm. In correspondence of this wavelength, the optical filter provides a maximum value of $\tau_O$ equal to 0.3898. In conclusion, all the data selected to simulate the detection process are summarized in Table 3.2.

| **LIDAR transmitter** | |
|---|---|
| $\lambda$ (nm) | 1540 |
| $Pw_{AVG}$ (mW) | 1 |
| $PRF$ (kHz) | 10 |
| $t_W$ (ns) | 10 |
| $\theta_B$ (°) | 0.02 |
| **LIDAR receiver - Aperture** | |
| $D_A$ (m) | 0.04 |
| **LIDAR receiver - G8931-20 InGaAs APD** | |
| $\eta$ | 0.7247 |
| $G_{APD}$ | 10 |
| $Ca$ (pF) | 1.5 |
| $Temp$ (K) | 273.15 |
| $i_D$ (nA) | 150 |
| **LIDAR receiver - Optical band-pass filter** | |
| $\Delta\lambda$ (nm) | 24 |
| $\tau_O$ ($\lambda$) | 0.3898 |

**Table 3.2 - Specifications of the simulated pulsed TOF LIDAR.**

The performance of the LIDAR detection process as a consequence of the previously described characterization is shown in Figure 3.14. Specifically, the $P_D$ is evaluated as a function of $\theta_{INC}$ at different ranges. In order to perform this analysis, a very low value, i.e. 0.2, is selected for $\rho_{T\lambda}$.

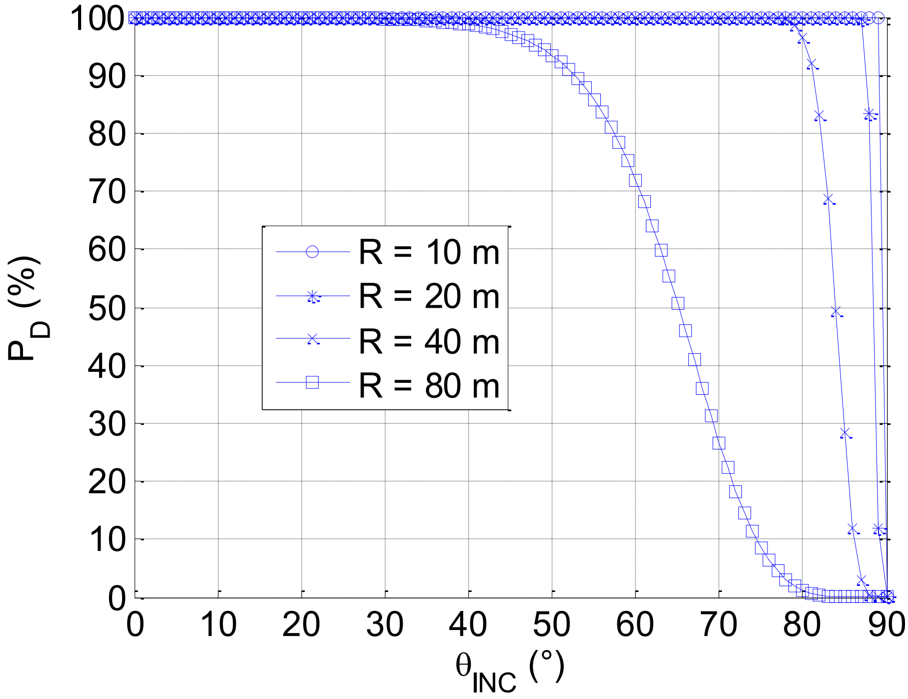


**Figure 3.14 - Probability of detection of the simulated LIDAR as a function of $\theta_{INC}$ at different ranges.**

By looking at Figure 3.14, it is possible to state that as the inclination of the incident beam increases, if the range is below 40 m, the probability of misdetection is practically zero for any value of $\theta_{INC}$ lower than 80°. Hence, for each value of $R$ it is possible to identify a threshold of $\theta_{INC}$ below which the probability of misdetection becomes different from zero. This threshold decreases rapidly as a function of range. For instance, it goes from 80° to 40° if the range is increased from 40 m to 80 m.

However, the effect of range can be better seen by representing the $P_D$ as a function of $R$ at different values of $\theta_{INC}$, like it is done in Figure 3.15.

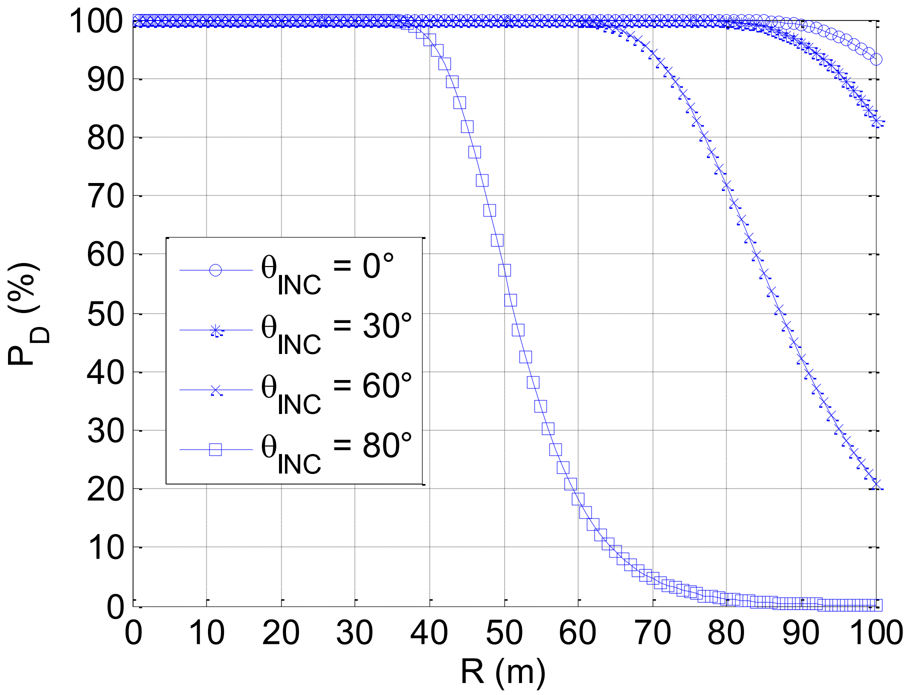


**Figure 3.15 - Probability of detection of the simulated LIDAR as a function of *R* at different inclinations of the laser beam.**

If the laser beam is perpendicular to the target surface ($\theta_{INC}$ = 0°), the configuration selected for the LIDAR (which is quite conservative with respect to typical performance of spaceborne systems especially in terms of emitted power) ensures the probability of misdetection to be equal to zero in the interval of distances (0, 90) m. Of course, in a way complementary to the data contained in Figure 3.14, the extension of this interval reduces as $\theta_{INC}$ increases. An additional statement obtained from this analysis is that, having assigned to the $P_{FA}$ the value of $10^{-4}$, the *SNR* must be at least 70 dB in order to obtain a $P_D$ that is reasonably close to 1.

Finally, as regards the characterization of the noise model, the values of $\sigma_{RANGE}$ (25 mm), $\sigma_{LOS}$ (0.007), and the percentage of outliers (5% to 7%) are chosen to be representative of a typical performance of 3D active sensor [56]. As it will be described in detail in chapter 5, these values are set for most of the performed simulations since the basic aim is to assess algorithm's performance against pose variability in the 6 DOF relative state space. However, the effect of variation of these data is also analyzed by means of ad-hoc simulations.

## 3.5 Target modeling and selection

As discussed in chapter 2, the problem of pose determination of an uncooperative target can be carried out by relying on measurements provided by active or passive EO sensors which must be processed by means of purposely developed model-based algorithm. It is clear that the strategy adopted for target modeling is strictly related to the specific algorithm exploited. Hence, this sub-chapter firstly describes in detail the target modeling approaches required by the algorithms developed for pose acquisition and tracking. Secondly, the targets selected for the numerical simulations, as well as their corresponding models, are presented.

### 3.5.1 Target modeling

As regards the algorithms proposed for pose acquisition, i.e. the on-line TM, the on-line fast-TM, and the PCA-TM, the target model is needed exclusively for template generation. This function is carried out by the corresponding module of the LIDAR measurement simulator which exploits only the geometric model (see sub-chapter 3.3.1). Specifically, it requires the target to be represented as the combination of 2D (planar) and 3D surfaces. Planar surfaces are basic 2D geometric entities, i.e. rectangles, squares, circles and ellipses, since they allow describing with enough level of detail the shapes of most of the existing satellites and debris. They are identified by the set of information listed below (vector quantities are expressed in the TRF).

- The normal unit vector of the corresponding plane.
- The position vector of the geometric center of the surface with respect to the TRF origin.

- One or more scalar parameters describing the size of the surface, i.e. the length of two sides (one side) for a rectangle (square), the length of the radius (semi-axes) for a circle (ellipse).

These latter data represent the scalar parameters mentioned in sub-chapter 3.3.1 necessary to check the consistency of the computed interceptions.

As regards the 3D surfaces, paraboloids, hyperboloids and cylinders can be used. For instance, in the latter case, the required data are limited to the radius and the height of the cylinder.

As regards the different versions of the ICP algorithm adopted for pose tracking, the same target modeling strategy is needed if the NS approach is used for the matching step, since it dynamically builds the model point cloud by projecting the measured points, expressed in TRF, on the closest surface of the target. On the other hand, the NN approach requires each surface to be discretized as a set of points, thus building the model point cloud off-line.

In order to finalize the generation of the simulated LIDAR point clouds used to test the entire pose determination process, additional data (whose selection is part of the target modeling strategy) are needed to carry out the detection process, namely the values of $\rho_{T\lambda}$ for any surface included in the model. In the context of space applications, the materials are generally opaque, so $\rho_{T\lambda}$ can be obtained by using eq. (3.19),

$$\rho_{T\lambda} = 1 - \alpha_{T\lambda} \tag{3.19}$$

where $\alpha_{T\lambda}$ is the absorption coefficient which is also is equal to the emissivity ($\varepsilon_{T\lambda}$) for a surface in thermodynamic equilibrium at constant temperature (Kirchhoff law).

In conclusion, it is extremely important to underline that the idea behind the generation of the satellite models used in the simulator is to capture only the macroscopic features of the real objects. This choice not only limits model complexity but also allows performing a conservative analysis of proposed approach performance. Indeed, preserving in the model only the macroscopic features of the target does not help solving ambiguous poses, thus offering the possibility of demonstrating approach effectiveness and robustness. However, the possibility that not modeled target surface details affect measurement accuracy (e.g. by multipath) is properly considered by means of the percentage of outliers randomly introduced within the measured point clouds.

### 3.5.2 Target selection

Performance of the pose determination algorithms developed in this thesis is evaluated considering targets of different typology and shape, which may be representative of space debris or still active serviceable satellites. Specifically, the following targets have been selected.

- ENVISAT, which is a huge low Earth orbit spacecraft recently (2012) declared as a debris.
- One satellite of the COSMO-SkyMed constellation (shortly indicated as CSM), which is a medium-size active satellite.
- One abandoned Kosmos 3M 2nd stage debris, i.e. a rocket body (RB).

As regards ENVISAT and CSM, the simplified models are given by an assembly of cuboid-shaped elements which represent the main body, the solar arrays, the synthetic aperture radar (SAR) antenna, and the related appendixes. Hence, these models are exclusively composed of planar rectangular surfaces. On the other hand, RB is modeled as a simple cylinder, thus being composed of two planar surfaces (i.e. the top and bottom circles) linked by a cylindrical lateral surface. For all these targets, the dimensions are consistent with information provided in the literature and they are collected in Table 3.3 [116-120]. For each of these targets, the TRF is defined as a body-centered reference frame with the origin in the geometric center of their main body, and axes indicated as $x_{TRF}$, $y_{TRF}$ and $z_{TRF}$. Examples of model point clouds generated for ENVISAT, CSM and RB is shown in Figure 3.16.

| **ENVISAT** | | | |
|---|---|---|---|
| Surface | Length along $x_{TRF}$ (m) | Length along $y_{TRF}$ (m) | Length along $z_{TRF}$ (m) |
| Main body | 4 | 4 | 10 |
| SAR antenna | 0.1 | 10 | 1.3 |
| Solar array | 0.1 | 6 | 15 |
| Appendix<br>SAR antenna - Main body | 0.5 | 0.1 | 0.1 |
| Appendix<br>Solar array - Main body | 6 | 0.1 | 0.1 |
| **CSM** | | | |
| Surface | Length along $x_{TRF}$ (m) | Length along $y_{TRF}$ (m) | Length along $z_{TRF}$ (m) |
| Main body | 1.5 | 3 | 1.5 |
| SAR antenna | 1.5 | 0.1 | 5.5 |
| Solar array 1 | 0.1 | 1.5 | 6 |
| Solar array 2 | 0.1 | 1.5 | 6 |
| Appendix<br>Solar array 1 - Main body | 0.1 | 0.1 | 1.5 |
| Appendix<br>Solar array 2 - Main body | 0.1 | 0.1 | 1.5 |
| Communication antenna 1 | 0.7 | 0.1 | 0.1 |
| Communication antenna 2 | 0.7 | 0.1 | 0.1 |
| **RB** | | | |
| Top and bottom circles | Radius (m) | | 1.2 |
| Cylindrical lateral surface | Height along $z_{TRF}$ (m) | | 6.5 |

**Table 3.3 - Geometrical dimension of the components of the simplified models generated for the selected targets.**

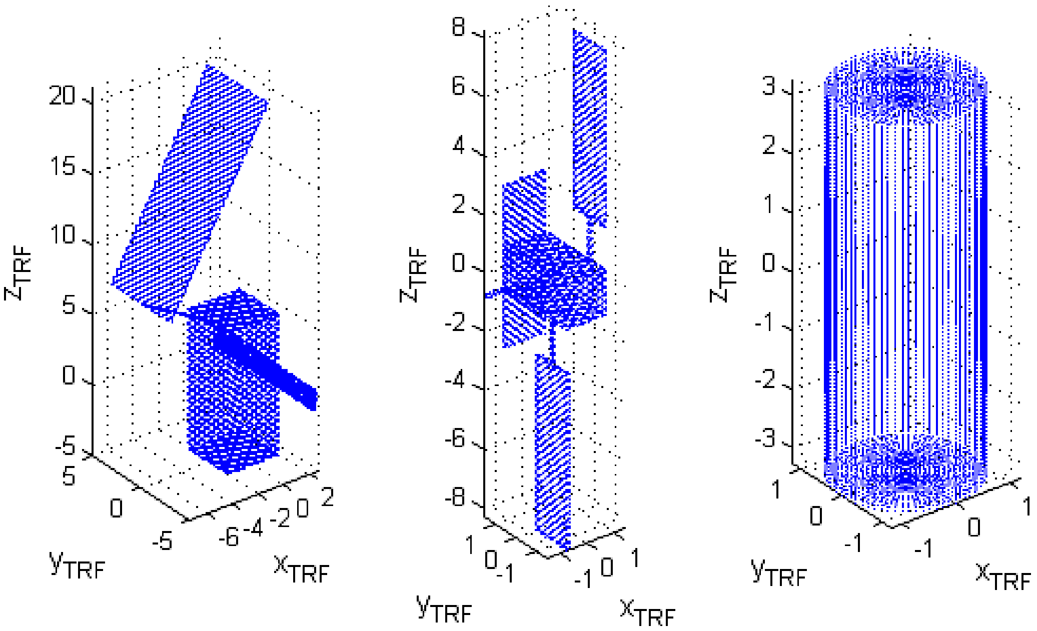


**Figure 3.16 - Model point clouds for ENVISAT (left), CSM (center), and RB (right), expressed in TRF.**

Additional information that must be included in the target modeling is the choice of the surface materials since it determines the values of the emissivity and, consequently, of the corresponding reflection coefficients.

Both for ENVISAT and CSM, a low value of the reflection coefficient (0.15) is conservatively selected for the surfaces of the main body and appendixes. Indeed, this can partially reproduce the reduced amount of power backscattered toward the detector due to the various devices typically installed on the external surface of the real target and not considered in the simplified model, which can cause multipath phenomena and can affect the uniformity of the radiation backscattered by a given satellite surface. Instead, the SAR antennas are made of reflective metallic materials, e.g. aluminum alloys, thus being characterized by high values of the reflection coefficient (0.97). Finally, as regards the solar arrays, fused silica cover (0.17) is adopted for the solar cell surface, while the other surfaces are assumed to be covered by Silverized Teflon (0.34). This produces an average value of the reflection coefficient (computed assigning weights proportional to the surface areas of each element) of 0.26 for ENVISAT and 0.38 for CSM. Instead, for the RB the reflection coefficient has been considered constant and equal to 0.4. These choices can be considered conservative since the satellite surface optical properties degradation due to exposition to the space environment typically leads to an average value of the reflection coefficient of about 0.6 [121].

## 3.6 Relative dynamics simulator

The relative dynamics simulator is used to generate the true target-chaser pose parameters. These data are not only inputs for the LIDAR simulator but they also represent the truth to which the outputs of the acquisition and tracking algorithms are compared. As it is shown in Figure 3.1, this module of the simulation environment requires the following inputs.

- The mean orbit parameters of the target and the chaser.
- The absolute rotational dynamics of the target and the chaser.

The first input is provided by a relative motion model, discussed in chapter 4, as a function of some geometrical design parameters which are used to assign the shape and size of the desired relative trajectory. Specifically, the simulated scenarios are inspired by close-proximity operations in space, e.g., rendezvous and docking, station keeping and monitoring. However, target monitoring scenarios are addressed with particular attention since they allow observing the target from multiple perspectives thus being able to generate much more variable sets of relative position and attitude parameters over which to test acquisition and tracking capabilities of the proposed approaches. These mean orbit parameters are used to numerically propagate the relative trajectory taking the main gravitational and perturbing forces into account.

With regards to the second input, the chaser is assumed to be three-axis stabilized with its boresight axis pointed toward the target geometric center. On the other hand, the target attitude is propagated by integrating the Euler's equations of the rotational dynamics by considering only the perturbation determined by the gravity gradient torque. This calculation requires the inertia tensor ($\underline{\underline{I}}$) of the target to be determined. While the inertia tensor of ENVISAT is taken from [122], where the assigned values are chosen to be on the upper end of the expected ones, the inertia tensors of CSM and RB are computed on the basis of the geometrical and mass characteristics of the simplified models defined in sub-chapter 3.5.2. The results are collected in Table 3.4.

| | | | |
|---|---|---|---|
| **ENVISAT** | | | |
| $\underline{\underline{I}}$ (kg m$^2$) | 129000 | 4500 | 1500 |
| | 4500 | 125000 | 1800 |
| | 1500 | 1800 | 17000 |
| **CSM** | | | |
| $\underline{\underline{I}}$ (kg m$^2$) | 5375 | 0 | 265 |
| | 0 | 3780 | 0 |
| | 265 | 0 | 2277 |
| **RB** | | | |
| $\underline{\underline{I}}$ (kg m$^2$) | 5705 | 0 | 0 |
| | 0 | 5705 | 0 |
| | 0 | 0 | 1058 |

**Table 3.4 - Inertia tensor for the selected targets.**

As regards the propagation of the absolute rotational dynamics of the considered targets, ENVISAT and RB are assumed to be gravity gradient stabilized thus having their minimum inertia axis (i.e. $z_{TRF}$) aligned with the radial direction of their Target Orbital Reference Frame (TORF). Specifically, these targets will have their TRF perfectly aligned to the TORF, i.e. $y_{TRF}$ is anti-parallel to the orbital angular momentum vector and $x_{TRF}$ is consequently pointed to get a right-handed reference frame.

On the other hand, CSM is assumed to be three-axis stabilized and performing yaw steering (being an active satellite for Earth observation). Hence, a non-zero relative attitude exists between the TRF and the TORF, which is represented by the 321 sequence of Euler angles defined hereunder.

- The rotation around the 3rd axis is the yaw steering ($\gamma_{STEERING}$) which can be computed by applying eq. (3.20),

$$\gamma_{STEERING} = 90 - arctg\left(\frac{\sin(i_{CSM})\cos(n_{CSM}t)}{-\cos(i_{CSM}) + (n_{CSM}/\Omega_E)}\right) \qquad (3.20)$$

  where $i_{CSM}$ and $n_{CSM}$ are the mean orbit inclination and the orbital angular velocity of CSM, $t$ is the time and $\Omega_E$ is the Earth angular velocity.
- The rotation around the 2nd axis is 0°.
- The rotation around the 1st axis is 90°.

# Chapter 4 - Relative trajectory design

A fundamental requirement to be satisfied by autonomous orbital maneuvers, which involve two space objects flying in close-proximity, is that the related relative trajectories must be designed to minimize the risk of collisions. In particular, the required level of safety cannot be considered successfully attained if it is exclusively assured by actively controlling the absolute dynamics of the servicing spacecraft with respect to an uncooperative target (OOS or ADR applications), or the relative dynamics of two-body formations (FF applications).

This concept was first emphasized by the partial failure of the Demonstration of Autonomous Rendezvous Technology (DART) project, which was carried out by NASA in 2005 [123]. The aim of this mission was to demonstrate the capability of a controllable platform (the DART spacecraft), which was launched from ground, to perform rendezvous as well as other maneuvers like station keeping, circumnavigation and collision avoidance around a non-maneuverable and uncooperative target (the MUBLCOM satellite), which was already on orbit. Unfortunately, the DART spacecraft collided with its target during the proximity-operation phase, due to a malfunctioning of its relative navigation system. Specifically, the investigations performed by NASA allowed discovering that the transition point from the GPS-based to the vision-based relative navigation was missed due to errors in the estimation of the relative state. Hence, these errors did not allow the DART autonomous computer to control the undesired drift of the relative trajectory which occurred in the along-track direction. However, the necessity to avoid introducing relative motion that makes the servicing platform pass through the target vehicle's along-track (also known as V-bar) at any point in subsequent orbits was fully recognized only thanks to the lessons learned during the OE mission. Indeed, a collision was accidentally avoided during an abort maneuver and loss of navigation information thanks to the unintentional presence of out-of-plane motion [8].

These missions clearly demonstrated the necessity to design relative trajectories which never intersect the along-track direction of the space object around which it is desired to maneuver, thus making them passively safe, meaning that absence of collision is ensured even without accurately controlling the relative dynamics. The first example of trajectory design which satisfies this constraint is given by the concept of safety ellipses (SE) which can be obtained by setting the phase difference between the in-plane and the out-of-plane relative motion to 90° [21]. An example of SE with the target located at its center is shown in Figure 4.1. Specifically, the trajectory is represented in the Target Hill Reference Frame (THRF), which is defined as follows: the x-axis is in orbit radial direction, the z-axis is parallel to the orbital angular momentum vector (cross-track) and the y-axis is perpendicular to the plane composed of the previous two axes and directed to obtain a right-handed coordinate system (along-track).

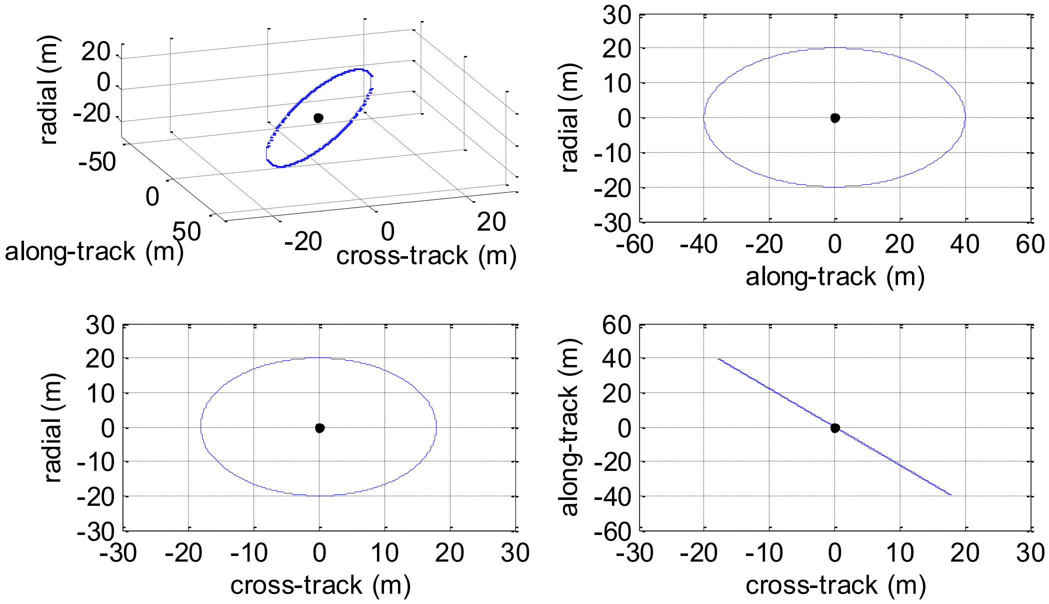


**Figure 4.1 -Example of SE. The blue lines represent the relative trajectory and the corresponding projections on the planes of the THRF. The black dots identify the target.**

The SE concept has been exploited by many researcher over the last few years. Specifically, Naasz [124] proposes a mathematical description of the SE in Hill's coordinate, which is based on phase-space considerations. This model is used in the framework of the Hubble Space Telescope Robotic Servicing and De-orbit Mission, in order to obtain a trajectory which optimizes the solar power collection since the primary design requirement is to keep the boresight axis of the relative navigation sensor, installed on board the de-orbiting module, always pointed toward the Hubble Space Telescope. On the other hand, Gaylor and Barbee [125] derive the equations of motion for the SE by exploiting geometric and kinematics considerations and they use this concept to design trajectories for safe target separation and circumnavigation. One successful implementation of this concept can be found in the PRISMA mission. Specifically, the safe orbit control module foresees a sequence of maneuvers necessary to put the main spacecraft on a SE designed so that it is always kept outside of an ellipsoidal avoidance region centered at the target [126].

A fundamental limitation of the SE concept as it has been introduced in the literature, is that the servicing spacecraft is allowed passing above/below the target during the relative orbit, as it can be clearly seen by looking at Figure 4.1. This could be the cause of collisions especially in the case of close motion with respect to a target which is an uncooperative freely tumbling object. Moreover, the capability to adequately control the relative motion during close-proximity maneuvers is strictly related to the accuracy with which the relative position and attitude are estimated. Hence, the capability to design relative trajectories which not only satisfy safety constraints but also optimize the target visibility (inside the FOV of the sensor in charge of relative navigation) for pose determination purposes, is crucial.

In this context, this chapter presents an analytical approach conceived to design different typologies of trajectories for safe target monitoring. These trajectories are then used within the simulation environment which assesses the performance of the developed pose determination algorithms. Numerical simulations, whose results are shown in chapter 5 (see sub-chapter 5.2.3 and sub-chapter 5.3.2) are specifically addressed at showing how this analytical approach can also be used to optimize the trajectory in terms of target observation conditions. Indeed, this can be done by carefully selecting the orbit design parameters so that the overall target shape is imaged for most of the time without occlusions (optimal point of view) and without losing too much detail (optimal interval of relative distances).

## 4.1 Analytical approach

The presented method has the crucial advantage that the design process is carried out by assigning high-level constraints regarding the stability of the relative motion as well as the size and the shape of the trajectory in order to directly derive the differences between the target-chaser mean orbit parameters. The main idea is to compare the classical form of the Hill's equation [127], which is expressed in rectangular relative motion coordinates, with the relative motion model in [128] which instead is developed in mean orbit parameters and it is based on a time explicit formulation. This model is particularly indicated to the case of spacecraft flying in close-proximity in Low Earth Orbit (LEO), as it is valid for targets moving along low-eccentricity orbits and it includes the perturbation effect due to secular Earth oblateness (J2).

This approach is exploited to generate three different typologies of relative trajectory. Firstly, a stable circular formation around any kind of target (it can move either on a circular or a low eccentricity orbit) is considered. Although this kind of relative motion potentially ensures safe target monitoring as it allows keeping constant the relative distance, it does not exclude potential collisions due to unintentional along-track drifts of the target when the chaser passes ahead or behind it. For this reason, two additional trajectories are proposed, namely the minimum-range-variation SE ($SE_{\Delta Rmin}$) and the multiple-requirement SE ($SE_{MR}$). These trajectories represent improved versions of the classical concept of SE since they involve the chaser to pass neither above/below nor ahead/behind the target, thus intrinsically preventing collision in the case of unintentional drifts in along-track direction. Of course, while for the circular formation the only geometrical design parameter is its radius, multiple geometrical constraints are foreseen by the modified

SE strategies. Before entering the detail of the determination of these trajectories, the relative motion model expressed in terms of the differences in the mean orbit parameters as well as the classical formulation of the relative in Hill's variable (rectangular coordinates), are briefly recalled.

### 4.1.1 Relative motion models

The nomenclature here adopted is typical of FF applications and is relevant to two-body formations. Specifically, the target and the chaser are indicated as chief and deputy, respectively, and the classical symbolism is adopted for the mean orbit parameters: *a* is the semi-major axis, *e* is the eccentricity, *M* is the mean anomaly, *Ω* is the right ascension of the ascending node, *ω* is the argument of perigee, *i* is the orbit inclination, and $\dot{u}$ is defined as $\dot{M}+\dot{\omega}$. With regards to these parameters, no subscripts are used for the chief, while the subscripts "D" and "0"are used to refer to the deputy and the initialization, respectively. Moreover, the symbol *δ* indicates the difference between deputy and chief parameters. Finally, it is fundamental to clarify that the symbols *x*, *y* and *z* are used to indicate the Hill's coordinates in THRF exclusively in the content of this chapter.

The relative motion model in terms of differences in the mean orbit parameters is shown in eq. (4.1), where *t* is the time variable.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \cong a \begin{bmatrix} \frac{\delta a}{a} - \delta e \cos(M_0 + \delta M_0 + \dot{M}_D t) + 2e \sin\left(\frac{\delta M_0}{2}\right) \sin\left(M_0 + \dot{M}_D t + \frac{\delta M_0}{2}\right) \\ 2\delta e \sin(M_0 + \delta M_0 + \dot{M}_D t) + 4e \sin\frac{\delta M_0}{2} \cos\left(M_0 + \frac{\delta M_0}{2} + \dot{M}_D t\right) \\ + \delta(\omega_0 + M_0) + \delta\Omega_0 \cos i + t\left(\delta\dot{u} + \delta\dot{\Omega}\cos i\right) \\ -(\delta\Omega_0 + \delta\dot{\Omega} t)\sin i \cos(\omega_{D0} + M_{D0} + \dot{u}_D t) + \delta i \sin(\omega_{D0} + M_{D0} + \dot{u}_D t) \end{bmatrix} \tag{4.1}$$

For the sake of clarity of the exposition, the model's equation can be re-arranged as shown in eq.(4.2),

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \cong \begin{bmatrix} x_{off} + A_x \cos\left(\dot{M}_D t + \varphi_x\right) \\ y_{off} + 2A_x \sin\left(\dot{M}_D t + \varphi_x\right) + t y_{dr} \\ A_z(t)\cos\left[\dot{u}_D t + \varphi_z(t)\right] \end{bmatrix} \tag{4.2}$$

where $y_{dr}$ is the drifting term, $x_{off}$ and $y_{off}$ are the radial and along-track offset, $A_x$ and $A_z$ are the amplitude of the radial and cross-track oscillations, $\varphi_x$ and $\varphi_z$ are the phases of the radial and the out-of-plane oscillations. These quantities can be computed as shown in eq. (4.3).

$$\begin{aligned}
x_{off} &= \delta a \\
A_x &= a\sqrt{\left[\delta e + 2e\sin^2\left(\frac{\delta M_0}{2}\right)\right]^2 + \left[e\sin\left(\delta M_0\right)\right]^2} \\
\varphi_x &= M_{D0} + \tan^{-1}\left(\frac{e\sin\left(\delta M_0\right)}{\delta e + 2e\sin^2\left(\frac{\delta M_0}{2}\right)}\right) \\
y_{off} &= a\left[\delta(\omega_0 + M_0) + \delta\Omega_0 \cos i\right] \\
y_{dr} &= a\left(\delta\dot{u} + \delta\dot{\Omega}\cos i\right) \\
A_z(t) &= a\sqrt{\delta i^2 + \left[\left(\delta\Omega_0 + \delta\dot{\Omega}t\right)\sin i\right]^2} \\
\varphi_z(t) &= \omega_{D0} + M_{D0} + \tan^{-1}\left\{\frac{\delta i}{\left[\delta\Omega_0 + \delta\dot{\Omega}t\right]\sin i}\right\}
\end{aligned} \tag{4.3}$$

Since the following sub-chapters require the duality between this representation and the one in rectangular coordinate to be exploited, the Hill's equations are recalled in eq. (4.4),

$$\begin{aligned}
&\ddot{x} - 2n\dot{y} - 3n^2 x = 0 \\
&\ddot{y} + 2n\dot{x} = 0 \\
&\ddot{z} + n^2 z = 0
\end{aligned} \tag{4.4}$$

where $n$ is the orbital angular velocity of the chief. The general solution is given by eq. (4.5).

$$
\begin{aligned}
&x = (\dot{x}_0 / n)\sin(nt) - (3x_0 + 2\,\dot{y}_0/ n)\cos(nt) + 4x_0 + 2\dot{y}_0 / n \\
&y = (2\dot{x}_0 / n)\cos(nt) + (6x_0 + 4\dot{y}_0 / n)\sin(nt) - (6nx_0 + 3\dot{y}_0)t - 2\dot{x}_0 / n + y_0 \\
&z = (\dot{z}_0 / n)\sin(nt) + z_0\cos(nt)
\end{aligned}
\quad (4.5)
$$

The equations above can also be written as shown in eq. (4.6), if the secular term is nullified, thus getting stable relative trajectories.

$$
\begin{aligned}
x &= \sqrt{x_0^{\,2} + \frac{\dot{x}_0^{\,2}}{n^2}}\cos\left(nt + arctg\left(-\left(\frac{\dot{x}_0}{n}\right)/\,x_0\right)\right) \\
y &= 2\sqrt{x_0^{\,2} + \frac{\dot{x}_0^{\,2}}{n^2}}\sin\left(nt + arctg\left(-\left(\frac{\dot{x}_0}{n}\right)/\,x_0\right)\right) \\
z &= \sqrt{z_0^{\,2} + \frac{\dot{z}_0^{\,2}}{n^2}}\cos\left(nt + arctg\left(-\left(\frac{\dot{z}_0}{n}\right)/\,z_0\right)\right)
\end{aligned}
\quad (4.6)
$$

From the analysis of the Hill's equations, it is known that the elimination of the secular term is equivalent to set *δa* (which is the radial offset) equal to zero [128]. The representation in mean orbit parameters, instead, shows that if the chief is located on a low-eccentricity orbit and if the effect of J2 is considered, the relative motion stability is ensured by nullifying $\delta\dot{u}$ and $\delta\dot{\Omega}$, considering that these latter parameters depend not only on *δa*, but also on *δe* and *δi*. In this context, it can be demonstrated that the effect of *δe* is negligible [127]. Hence, the drift of the relative trajectory can be prevented by setting to zero both *δa* and *δi*.

### 4.1.2 Design of the circular formation

If the chief is on a circular orbit, circular formations can easily be designed in Hill's variables as shown in [128]. The procedure is briefly summarized hereunder. Four conditions are given by imposing zero along- track drift and offset, see eq. (4.7) and eq. (4.8) respectively, and that the cross-track oscillation has the same phase and $\sqrt{3}$ times the amplitude if compared to the radial oscillation, as in eq. (4.9) and eq. (4.10).

$$\dot{y}_0 = -2x_0 n \qquad \text{(no along-track drift)} \qquad (4.7)$$

$$y_0 = 2\dot{x}_0 / n \qquad \text{(no along-track offset)} \qquad (4.8)$$

$$z_0 = \pm\sqrt{3}x_0 \qquad (4.9)$$

$$\dot{z}_0 = \pm\sqrt{3}\,\dot{x}_0 \qquad (4.10)$$

Equation (4.7) is a stability constraint, while the remaining equations determine the shape of the trajectory. Equation (4.8) could also be re-written to consider a freely assigned along track offset. Although, the mathematical procedure would be exactly the same, this case in not considered in the following as it is of lower importance in the context of monitoring scenarios. Indeed, if the chief is at the center of the relative trajectory, it is easier for the deputy to observe it from every point of view. The signs of eq. (4.9) and eq. (4.10) have to be the same to ensure the same phasing between radial and cross-track oscillations. The choice of the sign determines the inclination (±30°) of the plane of relative orbit, which contains the along-track axis, with respect to the along-track/cross-track plane. For instance, eq. (4.11) is obtained by selecting the "plus" sign.

$$\begin{aligned} x &= (\dot{x}_0 / n)\sin nt + x_0 \cos nt \\ y &= (2\dot{x}_0 / n)\cos nt - 2x_0 \sin nt \\ z &= \sqrt{3}(\dot{x}_0 / n)\sin nt + \sqrt{3}x_0 \cos nt \end{aligned} \qquad (4.11)$$

Choosing the values of the initial radial position and velocity of the deputy with respect to the chief, is equivalent to assigning the radius ($R_C$) and the phasing ($\theta_C$) of the relative orbit, as shown in eq. (4.12).

$$\begin{aligned} x_0 &= (R_C / 2)\cos\theta_C \\ \dot{x}_0 &= -(R_C\, n / 2)\sin\theta_C \end{aligned} \qquad (4.12)$$

By combining the conditions composing eq. (11), or equivalently eq. (12), it is possible to directly relate $R_C$ to the initial radial position and velocity, and consequently to $A_x$.

$$R_C = 2\sqrt{{x_0}^2 + \frac{{\dot{x}_0}^2}{n^2}} = 2A_x \tag{4.13}$$

Although this procedure is very straightforward, it is extremely limiting in OOS and ADR missions which do not allow freely selecting the chief (target) orbit, which, for instance, can be non-circular. Moreover, if orbital perturbations are not included the condition on the secular term set in Hill's variable do not necessarily prevent the relative motion from drifting. Hence, the following analytical approach can be used if the chief is on a small eccentricity non-keplerian (J2 secular effects included) orbit. Before entering the detail, it is worth outlining that due to the previous assumptions on $\delta a$ and $\delta i$, which are set equal to zero to ensure stability of the relative trajectory, the time drift (caused by non-keplerian effects) of the differences in the mean orbit parameters, which represent the unknowns of this problem, can be considered negligible. This is a crucial statement to make it possible the parallelism between the Hill's and mean orbit parameters formulations of the relative motion. The remaining conditions are required to determine the circular shape. Specifically, the along-track offset is set to zero by looking at eq. (4.3), thus obtaining the dual of eq. (4.7).

$$\delta(\omega_0 + M_0) + \delta\Omega_0 \cos i = 0 \tag{4.14}$$

Again. if an along-track offset was desired, eq. (4.15) should be considered, instead.

$$\delta(\omega_0 + M_0) + \delta\Omega_0 \cos i = \frac{y_{off}}{a} \tag{4.15}$$

At this point, $R_C$ is linked to $\delta e$ and $\delta M_0$ by substituting the expression of the amplitude of the radial oscillation contained in eq.(4.3) within eq. (4.13).

$$R_C = 2A_x = 2a\sqrt{\delta e^2 + 4e^2 \sin^2\left(\frac{\delta M_0}{2}\right)\cos^2\left(\frac{\delta M_0}{2}\right) + 4e^2 \sin^4\left(\frac{\delta M_0}{2}\right) + 4e\delta e \sin^2\left(\frac{\delta M_0}{2}\right)} \tag{4.16}$$

By combining the relations in eq. (4.3) with eq. (4.9) and eq. (4.10), it is possible to express the conditions on the amplitude and phase of the radial and cross-track oscillation in terms of differences between mean orbit parameters. The result is shown by eq. (4.17) and eq. (4.18), respectively.

$$\delta\Omega_0 \sin i = \pm\sqrt{3}\sqrt{\delta e^2 + 4e^2 \sin^2\left(\frac{\delta M_0}{2}\right)\cos^2\left(\frac{\delta M_0}{2}\right) + 4e^2 \sin^4\left(\frac{\delta M_0}{2}\right) + 4e\delta e \sin^2\left(\frac{\delta M_0}{2}\right)} = \pm\sqrt{3}\,\frac{R_C}{2a} \quad (4.17)$$

$$\omega_{D0} = \omega_0 + \delta\omega_0 = \tan^{-1}\left(\frac{e\sin(\delta M_0)}{\delta e + 2e\sin^2\left(\frac{\delta M_0}{2}\right)}\right) \quad (4.18)$$

This procedure has led to six constraints just as shown in terms of rectangular coordinates. In that case, it was possible to select two degrees of freedom, i.e. $R_C$ and $\theta_C$. On the other hand, in this latter case, the condition on $\delta i$ involves a constraint on the value of the $\theta_C$ (which results to be dependent on the initial position of the chief). This is the reason why $R_C$ becomes the unique tunable design parameter. Once $R_C$ is selected, the remaining differences in mean orbit parameters can be determined. Specifically, $\delta\Omega_0$ can be extracted from eq. (4.17) where the choice of the sign still determines the inclination (±30°) of the relative orbital plane.

$$\delta\Omega_0 = \pm\sqrt{3}\,\frac{R_C}{2a\sin i} \quad (4.19)$$

An expression for $\delta\omega_0$ is obtained from eq. (4.14).

$$\delta\omega_0 = -\delta\Omega_0 \cos i - \delta M_0 \quad (4.20)$$

However, this relation is not directly applicable since $\delta M_0$ is still unknown. Hence, it is necessary to combine eq. (4.18) with eq. (4.20) and to express the resulting relation in terms of $\delta e$.

$$\delta e = \frac{e\sin(\delta M_0)}{\tan(\omega_0 - \delta\Omega_0 \cos i - \delta M_0)} - 2e\sin^2\left(\frac{\delta M_0}{2}\right) \quad (4.21)$$

Equation (4.21) is then substituted into eq. (4.16), resulting in a non-linear equation in the only unknown $\delta M_0$, which can be solved with a numerical approach (e.g. Newton-Raphson).

### 4.1.3 Design of the modified safety ellipses

The SE$_{\Delta\text{Rmin}}$ is conceived to combine the classical SE concept with an additional constraint regarding the variation of the distance along the relative trajectory. The fundamental idea is that the design of one stable and safe trajectory, which also keeps limited the variation of the target-chaser relative distance, can be extremely useful. Indeed, if a particular value of the range is demonstrated to be advantageous in terms of target observability, e.g. resolution and coverage in the observed field-of-view, this modified SE allows the chaser to move in a restricted interval of distances from the target around this specific value. This can be accomplished by carefully tuning the two design parameters, which univocally define the SE$_{\Delta\text{Rmin}}$, namely $A_x$ and the minimum distance in the radial/cross-track plane ($dxz_{MIN}$). The analytical procedure necessary to design the SE$_{\Delta\text{Rmin}}$, is based on the assumption that the difference in frequency between in plane and out-of-plane oscillations is negligible, which means neglecting the perigee precession rate. Although this latter statement is valid only over a limited time interval, it is typically long enough to cover the applications addressed by this work. Hence, the following formulation for the relative motion model of eq. (4.1) and eq. (4.2) can be used,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \cong \begin{bmatrix} A_x \cos(\Gamma_x(t)) \\ 2A_x \cos\left(\Gamma_x(t) + \dfrac{\pi}{2}\right) \\ A_z \cos[\Gamma_x(t) + \Delta\varphi_z] \end{bmatrix} \tag{4.22}$$

where $\Delta\varphi_z$ is the phase difference between the radial and cross-track oscillations. This quantity, together with the argument $\Gamma_x$, can be computed as shown in eq. (4.23).

$$\begin{aligned} \Gamma_x &= \dot{M}_D t + \varphi_x \\ \Delta\varphi_z &= \omega_{D0} - \tan^{-1}\left( \frac{e\sin(\delta M_0)}{\delta e + 2e\sin^2\left(\dfrac{\delta M_0}{2}\right)} \right) \end{aligned} \tag{4.23}$$

The SE$_{\Delta\mathrm{Rmin}}$, determined by the proposed approach, is the trajectory ensuring the minimum variation of range among all the ones identified by the assigned values of $A_x$ and $dxz_{MIN}$, which simultaneously has the in-plane motion of the circular formation defined by $A_x$. Firstly, it is necessary to define the difference in the cross-track oscillation between the desired trajectory and the corresponding circular ($\Delta z_C$), as shown in eq. (4.24).

$$\Delta z_C(t) = A_z \cos(\Gamma_x(t) + \Delta\varphi_z) - \sqrt{3} A_x \cos(\Gamma_x(t)) \tag{4.24}$$

This relation can be manipulated by applying basic trigonometric relations, thus obtaining eq. (4.25).

$$\Delta z_C(t) = \sqrt{A_z^2 + 3A_x^2 - 2\sqrt{3} A_z A_x \cos(\Delta\varphi_z)} \cos\left(\Gamma_x(t) + \tan^{-1}\left(\frac{A_z \sin(\Delta\varphi_z)}{A_z \cos(\Delta\varphi_z) - \sqrt{3} A_x}\right)\right) \tag{4.25}$$

Hence, the minimum-range condition is equivalent to minimize the amplitude of $\Delta z_C$ ($A_{\Delta z_C}$) with respect to $A_z$. For the sake of mathematical simplicity, but without losing generality of the exposition, the square value of $A_{\Delta z_C}$ is minimized, as shown in eq. (4.26).

$$\frac{\delta A_{\Delta z_C}^2}{\delta A_z} = 0 \Leftrightarrow 2A_z - 2\sqrt{3} A_x \cos(\Delta\varphi_z) = 0 \tag{4.26}$$

The condition above allows expressing $\Delta\varphi_z$ as a function of $A_x$ and $A_z$, as shown in eq. (4.27).

$$\cos(2\Delta\varphi_z) = \frac{2}{3}\frac{A_z^2}{A_x^2} - 1 \tag{4.27}$$

It is now necessary to derive $A_z$ as a function of the design parameters. To this aim, the definition of distance in the radial/cross-track plane ($dxz$), given by eq. (4.28), is minimized thus writing eq. (4.29).

$$dxz^2 = x^2 + z^2 = \frac{A_x^{\ 2}}{2}\cos^2\Gamma_x + \frac{A_z^{\ 2}}{2}\cos^2(\Gamma_x + \Delta\varphi_z) \tag{4.28}$$

$$dxz_{MIN}^2 = \frac{A_x^{\;2}}{2} + \frac{A_z^{\;2}}{2} - \sqrt{\frac{A_x^{\;4}}{4} + \frac{A_z^{\;4}}{4} + 2\frac{A_x^{\;2}A_z^{\;2}}{4}\cos 2\Delta\varphi_z} \tag{4.29}$$

Finally, by substituting eq. (4.27) within eq. (4.29), it is possible to obtain a bi-quadratic equation in which the only unknown is $A_z$.

$$\frac{A_z^{\;4}}{3} + A_z^{\;2}(dxz_{MIN}^2 - A_x^{\;2}) + dxz_{MIN}^2(A_x^{\;2} - dxz_{MIN}^2) = 0 \tag{4.30}$$

It is clear that this method is applicable only if the tuning parameters are assigned so that the discriminant of eq. (4.30) is positive, which means that $A_x$ and $dxz_{MIN}$ must satisfy the constraint given by eq. (4.31).

$$A_x > \sqrt{\frac{7}{3}} dxz_{MIN} \tag{4.31}$$

If eq. (4.31) is verified, $A_z$ and $\Delta\varphi_z$ can be computed by applying eq. (4.32) and eq. (4.27) respectively.

$$A_z = \pm\sqrt{\frac{(A_x^{\;2} - dxz_{MIN}^2) \pm \sqrt{(dxz_{MIN}^2 - A_x^{\;2})^2 - \frac{4}{3}dxz_{MIN}^2(A_x^{\;2} - dxz_{MIN}^2)}}{2/3}} \tag{4.32}$$

The ambiguity about the four possible solutions for $A_z$ must be solved. Firstly, the maximum between the two values of $A_z^2$ is selected. . Secondly, the sign plus is selected for $A_z$. If this latter choice leads to a negative value of $e$ for the chaser (at the end of the analytical procedure), the negative sign must be assigned to $A_z$, instead.

At this point, the parallelisms between the formulations of the relative motion model in eq. (4.1) and eq. (4.5), can be used to compute the differences between the target-chaser mean orbit parameters identifying SE$_{\Delta \text{Rmin}}$, considering that $\delta a$ and $\delta i$ are still set equal to zero to ensure stability of the relative trajectory. Firstly, $\delta\Omega_0$ is computed by using the definition of $A_z$ given by eq. (4.3) (recalling again the assumption about the perigee precession rate).

$$\delta\Omega_0 = -\frac{A_z}{a\sin i} \tag{4.33}$$

At this point, $\delta\omega_0$ can be determined by applying eq. (4.20), which still corresponds to zero the along-track offset. However, also in this case $\delta M_0$ and $\delta e$ are still unknown. Firstly, eq. (4.34) is obtained by combining the definitions of $\varphi_x$ and $\varphi_z$ in eq. (4.3).

$$\delta\varphi_z = \omega_0 + \delta\omega_0 - \tan^{-1}\left(\frac{e\sin(\delta M_0)}{\delta e + 2e\sin^2\left(\frac{\delta M_0}{2}\right)}\right) \tag{4.34}$$

Hence, $\delta e$ can be expressed as a function of $\delta M_0$, by combining eq. (4.34) with eq. (4.20), as shown in eq. (4.35).

$$\delta e = \frac{e\sin(\delta M_0)}{\tan(\omega_0 - \delta\Omega_0 \cos i - \delta M_0 - \delta\varphi_z)} - 2e\sin^2\left(\frac{\delta M_0}{2}\right) \tag{4.35}$$

Finally eq.(4.35) is substituted into the definition of $A_x$ given within eq. (4.3), thus obtaining a non-linear equation which is again solved numerically thanks, for instance, to the Newton-Raphson method.

Moving on to the SE$_{\text{MR}}$, it is designed, just like the SE$_{\Delta\text{Rmin}}$, so that the relative trajectory does not intercept either the along-track or the radial axes of the THRF. However, the fundamental difference is that the idea of minimizing the variation of the target-chaser relative distance is substituted by the possibility of freely selecting the cross-track separation at zero along-track ($\tilde{z}$), which results to be an extremely important degree of freedom if optimal observation conditions for pose determination are requested, as it will be demonstrated in chapter 5. Hence, the SE$_{\text{MR}}$ is univocally defined by the same design parameters of SE$_{\Delta\text{Rmin}}$, i.e. $A_x$ and $dxz_{MIN}$, with the addition of $\tilde{z}$.

This problem relies on the same assumptions used to write eq. (4.22) and eq. (4.23). The first step consists in fully characterizing the trajectory from the geometrical point of view. i.e. by computing $A_z$, and $\Delta\varphi_z$. By definition, $\tilde{z}$ can be derived by using eq. (4.36), which can then be used to find an expression for $\Delta\varphi_z$.

$$\tilde{z} = A_z \cos\Delta\varphi_z \tag{4.36}$$

$$\Delta\varphi_z = \frac{1}{2}\cos^{-1}\left(\frac{2\tilde{z}^2}{A_z^2} - 1\right) \tag{4.37}$$

By substituting eq. (4.37) in eq. (4.29), it is possible to get a second-order equation in $A_z$ whose solution is given by eq. (4.38).

$$A_z = \pm\sqrt{\frac{dxz_{MIN}^4 - A_x^{\ 2} dxz_{MIN}^2 - A_x^{\ 2}\tilde{z}^2}{dxz_{MIN}^2 - A_x^{\ 2}}} \tag{4.38}$$

Once $A_z$ is derived (by considering the positive root of the second order equation), $\Delta\varphi_z$ can be computed from eq. (4.37). Since all the geometric parameters describing the $SE_{MR}$ are known, the corresponding differences in the mean orbit parameters can be determined by exploiting exactly the same approach relative to the $SE_{\Delta Rmin}$.

In order to conclude the discussion of this sub-chapter, it is necessary to summarize the differences between the two presented trajectories. Once $A_x$ and $dxz_{MIN}$ are selected, it is clear that the design of the $SE_{\Delta Rmin}$ allows minimizing the ratio between the maximum and minimum target-chaser relative distance without a-priori knowing the corresponding value of $\tilde{z}$. However, since in the space of the geometrical parameters which identify the modified SE ($A_x$, $dxz_{MIN}$, and $\tilde{z}$) the applicability of this strategy is limited by eq. (4.31), the $SE_{MR}$ appears to be a more powerful tool to design relative trajectories for close-proximity operations which satisfy multiple requirements (hence the name) as it will further shown in the following sub-chapter.

## 4.2 Examples

Practical implementations of the previously defined analytical approach to design safe trajectories for target monitoring are here shown considering as targets the space objects selected in sub-chapter 3.5.2.

As regards ENVISAT, CSM and one of the Kosmos 3M rocket bodies, the Two Line Elements (TLE) relevant to a recent measurement (November 9th 2015) are adopted as starting point for the analysis. Firstly, three different relative trajectories are designed around ENVISAT by exploiting each one of the strategies discussed in the previous sub-chapters to compute the chaser mean orbit parameters. In order to compare these different solutions, the corresponding design parameters are assigned so that the minimum target-chaser relative distance is of (about) 25 m. While in the case of the circular trajectory, this requirement is satisfied by simply setting $R_C$ to 25 m, the $\mathrm{SE}_{\Delta\mathrm{Rmin}}$ and the $\mathrm{SE}_{\mathrm{MR}}$ require a tuning procedure of their corresponding design parameters. Specifically, as regards the $\mathrm{SE}_{\Delta\mathrm{Rmin}}$, a minimum relative distance of 25.2 m is ensured by setting $A_x$ to 21 m and $dxz_{MIN}$ to 13 m. On the other hand, a $\mathrm{SE}_{\mathrm{MR}}$ characterized by a minimum range of 25.01 m is obtained by setting $A_x$ to 16 m, $dxz_{MIN}$ to 8 m, and $\tilde{z}$ to 25 m. As a result of this design process, the mean orbit parameters of both the chaser and ENVISAT are collected in Table 4.1.

| **Mean orbit parameters** | **Chaser - circular trajectory** | **Chaser - $\mathrm{SE}_{\Delta\mathrm{Rmin}}$** | **Chaser - $\mathrm{SE}_{\mathrm{MR}}$** | **ENVISAT (9th November 2015)** |
|---|---|---|---|---|
| $a$ (km) | 7143.78192394 | 7143.78192394 | 7143.78192394 | 7143.78192394 |
| $i$ (°) | 98.31650000 | 98.31650000 | 98.31650000 | 98.31650000 |
| $\Omega$ (°) | 9.13622451 | 9.13659470 | 9.13664281 | 9.13640000 |
| $e$ | 0.00009663 | 0.00009451 | 0.00009545 | 0.00009630 |
| $\omega$ (°) | 78.47285310 | 78.09171042 | 78.25466816 | 79.49300000 |
| True Anomaly (°) | 1.02031869 | 1.40158263 | 1.23860338 | 0 |

**Table 4.1 - Initial mean orbit parameters to obtain relative trajectories around ENVISAT characterized by a minimum range of 25 m using the three proposed approaches.**

By looking at the relative trajectories which derive from the mean orbit parameters listed above, the main features of the proposed approaches can be put into evidence. The fact that the circular trajectory is only passively safe since it intersects the along-track axis of the THRF is highlighted in Figure 4.2 where the trajectory projection on the radial/cross-track plane is a straight line.

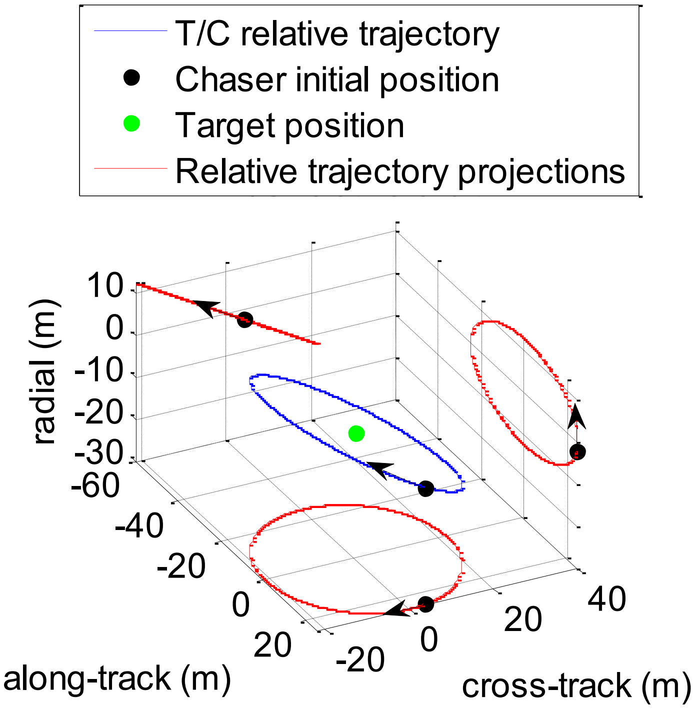


**Figure 4.2 - Circular trajectory (25 m radius) around ENVISAT, together with its projections on the reference planes of THRF.**

On the other hand, the remaining design strategies both provide relative motions of the chaser around the target which satisfy the additional safety constraint of intersecting neither the along-track nor the radial axis of the THRF, as it is shown in Figure 4.3 and Figure 4.4, where their projections on the along-track/cross-track plane are ellipses, instead of being straight lines as foreseen by the classical SE concept (see Figure 4.1).

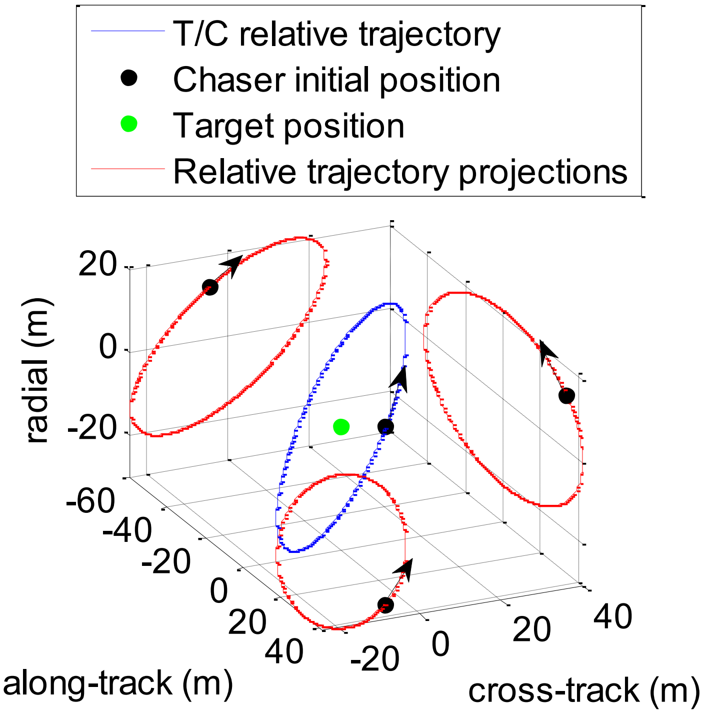


**Figure 4.3 - SE$_{\Delta R\min}$ ($A_x$ of 21 m, $dxz_{MIN}$ of 13 m) around ENVISAT, together with its projections on the reference planes of THRF.**

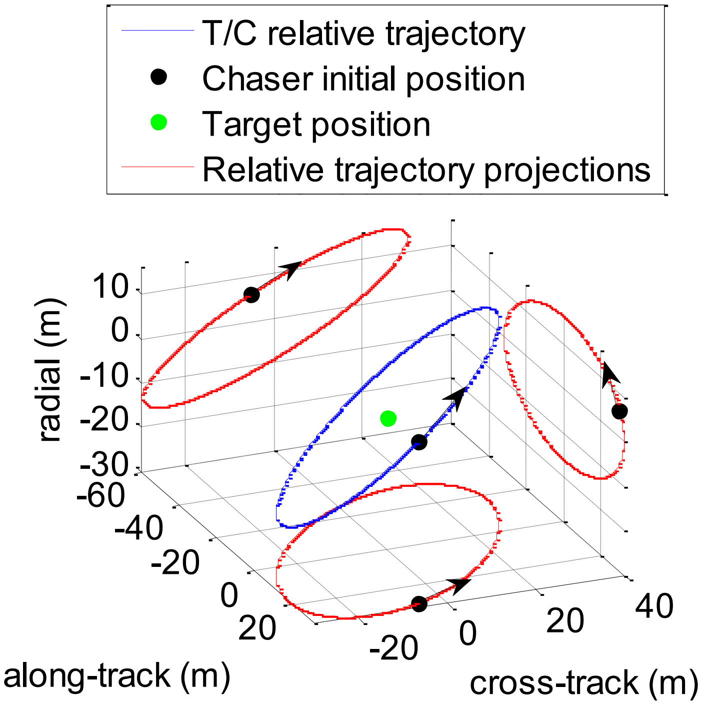


**Figure 4.4 - SE$_{MR}$ ($A_x$ of 16 m, $dxz_{MIN}$ of 8 m, $\tilde{z}$ of 25 m) around ENVISAT, together with its projections on the reference planes of THRF.**

By favoring z̃ over $A_x$, it is possible to generate trajectories characterized by very limited range variation and low values of $\varphi_{OUT}$, i. e. the out-of plane observation angle, which can be computed by means of eq. (4.39).

$$\varphi_{OUT} = \tan^{-1}\left(\frac{A_x}{\tilde{z}}\right) \tag{4.39}$$

Conversely, if $A_x$ is larger than $\tilde{z}$, both the range variation and $\varphi_{OUT}$ increase. Hence, a great advantage of this design strategy is that it gives the possibility to select the most convenient value of $\varphi_{OUT}$ to optimize the target observation conditions for relative navigation purposes, which depend on both the shape of the target and its absolute rotational dynamics. Examples of how the proper selection of $\varphi_{OUT}$ improves pose estimation performance are shown in sub-chapter 5.2.3.

Due to its higher flexibility, the $SE_{MR}$ model is used to provide examples of trajectory design for both CSM and RB for which the design parameters are tuned in order to obtain a minimum relative range to the target of about 11 m and 7 m respectively. Specifically, in the case of CSM a $SE_{MR}$ characterized by a minimum range of 11.2 m is obtained by setting $A_x$ to 8 m, $dxz_{MIN}$ to 6 m and $\tilde{z}$ to 14 m thus leading to a 29.7° value for $\varphi_{OUT}$. On the other hand, a $SE_{MR}$ around RB ensuring a minimum range of 7 m is obtained by setting $A_x$ to 5 m, $dxz_{MIN}$ to 4 m and z̃ to 14 m thus leading to a 16.6° value for $\varphi_{OUT}$. The target-chaser initial mean orbit parameters corresponding to the above defined trajectories (depicted in Figure 4.5 and Figure 4.6) are collected in Table 4.2.

| Mean orbit parameters | Chaser - $SE_{MR}$ around CSM | CSM (9th November 2015) | Chaser - $SE_{MR}$ around RB | RB |
|---|---|---|---|---|
| *a* (km) | 6997.94011260 | 6997.94011260 | 7145.76400000 | 7145.76400000 |
| *i* (°) | 97.86000000 | 97.86000000 | 74.03200000 | 74.03200000 |
| *Ω* (°) | 0.00018183 | 0 | 294.88419744 | 294.88400000 |
| *e* | 0.00118088 | 0.00118000 | 0.00432250 | 0.00432200 |
| *ω* (°) | 90.03532213 | 90 | 81.40148794 | 81.39500000 |
| True Anomaly (°) | 359.96461925 | 0 | 359.99340088 | 0 |

**Table 4.2 - Initial mean orbit parameters to obtain $SE_{MR}$ around CSM and RB characterized by a minimum range of respectively 11.2 m and 7 m..**

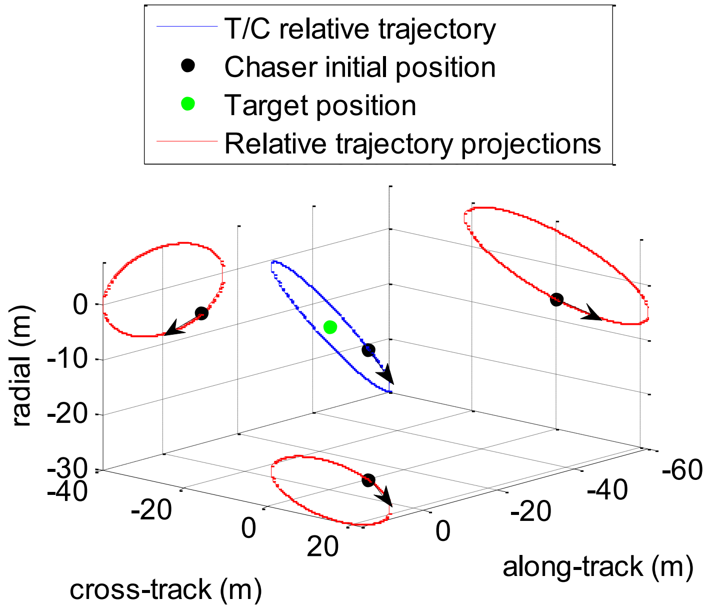


**Figure 4.5 - $SE_{MR}$ ($A_x$ of 8 m, $dxz_{MIN}$ of 6 m, $\tilde{z}$ of 14 m) around CSM, together with its projections on the reference planes of THRF.**

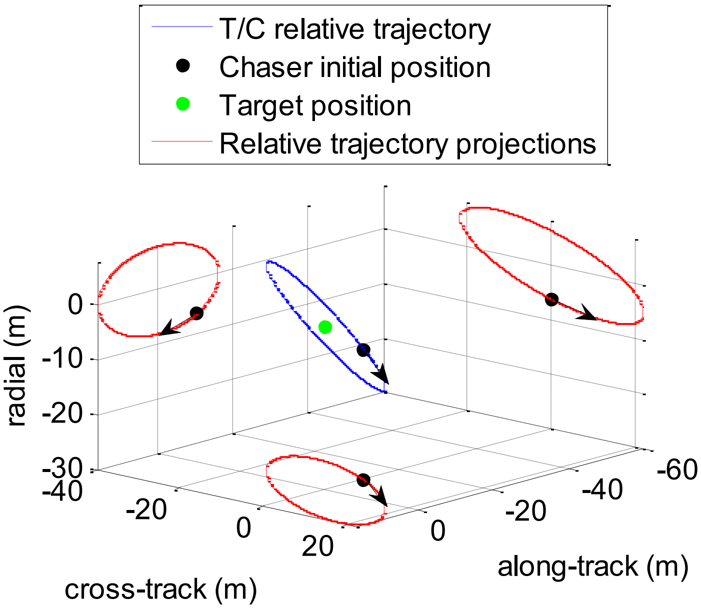


**Figure 4.6 - SE$_{\mathrm{MR}}$ ($A_x$ of 5 m, $dxz_{MIN}$ of 4 m, $\tilde{z}$ of 14 m) around RB, together with its projections on the reference planes of THRF.**

# Chapter 5 - Pose determination algorithms: performance assessment

This chapter presents and discusses the results of all the numerical simulations carried out to assess the performance of the pose determination architecture described in sub-chapter 2.3. Specifically, this performance evaluation is done considering both the pose estimation accuracy level and the computational efficiency issues. The effectiveness of the autonomous failure detection strategy and of the check for consistency of the PCA solution (which is relevant only to the PCA-TM algorithm), as well as the applicability of the analytical approach presented in chapter 4 to smartly design trajectories which optimize target observation conditions are assessed. The simulations are performed in MATLAB™ environment and run on a commercial desktop equipped with an Intel™ i7 CPU at 3.4 GHz. Of course, embedded processors used for space applications are not as fast as ground ones, due to limitations in size and weight for payload, but at the same time, the proposed algorithms have not been implemented to run in real time. Hence, although the run-time represents a good indicator of absolute computational efficiency, the most valuable results are obtained by comparing the performance of different solutions to identify the faster ones and evaluate the effect of setting parameters on the computational load. Finally, it is worth recalling that the selection of all the LIDAR specifications and the noise parameters, adopted for the numerical simulations, has been presented and motivated in sub-chapter 3.4.

## 5.1 Performance analysis criteria

Before entering the details of the numerical results, it is necessary to clarify the criteria adopted to determine algorithms' accuracy.

With regards to the pose acquisition step, effectiveness of 3D on-line Template Matching (on-line TM), the 3D on-line Fast Template Matching (on-line fast-TM) and the 3D PCA-based on-line Template Matching (PCA-TM) is verified in terms of both computational cost and accuracy level. However, the pose accuracy attained by these techniques is not a meaningful metrics since the attitude parameter space is sampled with large angular steps (tens of degrees). Moreover, the centroiding approach, exploited to determine the relative position vector, produce errors of the order of meters, depending on the fact that the centroid is estimated by considering only the visible portions of the target. This poor accuracy level is the reason why the pose solution provided by the acquisition step cannot be directly sent into a navigation filter. However, this is not a real issue since the only aim of the pose acquisition step is to obtain a pose solution which falls in the field of convergence of the tracking algorithm, meaning that pose acquisition is deemed successful if a subsequent application of the ICP algorithm is able to estimate the pose with a prefixed accuracy level. In this thesis, a threshold of 3° is adopted, as it represents approximately ten times the value of the ICP attitude accuracy level at regime [31]. The success rate of the proposed pose acquisition algorithms (respectively indicated as $SR_{TM}$, $SR_{fast\text{-}TM}$, and $SR_{PCA\text{-}TM}$) is evaluated at fixed values of the target-chaser distance (still indicated as $R$ in the following) over large numbers of sets of relative attitude parameters (the Euler angles) selected by means of random extractions from uniform distributions defined on their specific intervals of variation, i.e. (-90°, 90°)for $\beta$, (-180°, 180°) for $\alpha$ and $\gamma$. The determination of the success rate at different values of $R$ allows densely covering an extremely wide portion of the 6-DOF pose space. The effect on the success rate of varying $\Delta$, which determines the number of templates and consequently the computational load, is also analyzed.

Moving on to the pose tracking step, performance of different versions of the ICP algorithm discussed in sub-chapter 2.5 are evaluated, over specifically designed relative trajectories, and compared. Although the errors in the estimation of the relative translation vector and of the relative Euler angles with respect to the real values (deriving from the assigned relative dynamics) directly represent meaningful metrics, equivalent error parameters are also introduced to better summarize the results. Specifically, the relative attitude estimation error is represented by $\phi_{ERR}$, which is the equivalent Euler angle corresponding to the quaternion error ($q_{ERR}$) between the true and estimated ones (respectively $q_{TRUE}$ and $q_{EST}$) [129]. Once $q_{ERR}$ is obtained using eq. (5.1), in which the quaternion division is exploited, $\varphi_{ERR}$ can be computed using eq. (5.2),

$$q_{ERR} = \frac{q_{TRUE}}{q_{EST}} \tag{5.1}$$

$$\varphi_{ERR} = 2\cos^{-1}\left(q_{ERR\_S}\right) \tag{5.2}$$

where $q_{ERR\_S}$ is the scalar component of $q_{ERR}$. An additional measure of the error in the relative attitude is $\delta\varphi_{EA}$, which represents the angle between the directions of the Euler axes corresponding to the real and the estimated orientation of chaser with respect to the target. It is computed using eq. (5.3),

$$\cos\left(\delta\varphi_{EA}\right) = \sin\left(\frac{2\cos^{-1}\left(q_{TRUE\_S}\right)}{2}\right) q_{TRUE\_V} \cdot \sin\left(\frac{2\cos^{-1}\left(q_{EST\_S}\right)}{2}\right) q_{EST\_V} \tag{5.3}$$

where $q_{TRUE\_S}$, $q_{TRUE\_V}$, $q_{EST\_S}$, $q_{EST\_V}$ are the scalar and vector components of $q_{TRUE}$ and $q_{EST}$. On the other hand, the error in the relative position is measured by $|\underline{T}|_{ERR}$, which is the difference between the Euclidean norms of the true and the estimated values of $\underline{T}$, and $\vartheta_{ERR}$, which is the angle between the directions of the true and the estimated values of $\underline{T}$. Finally, as regards the computational time analysis, it is based on the evaluation of the ICP convergence time ($t_{CONV}$) and of the corresponding number of iterations ($N_{IT}$).

A statistical analysis is performed in order to compute, for each parameter of interest, the time statistics (i.e. mean, std and rms) over the assigned relative trajectory. Specifically, the incremental mean and standard deviation [130] are dynamically computed over $N_{SIM}$ simulations (each one differing from the others due to the random extractions for the LIDAR range and pointing uncertainties), at each position along the trajectory. If $X$ is the parameter of interest, the incremental mean value after $N_{SIM}$ simulations ($\mu_{XN}$) is given by eq. (5.4).

$$\mu_{X_N} = \frac{1}{N_{SIM}} \cdot [X_N + \mu_{X_{N-1}} \cdot (N_{SIM} - 1) \tag{5.4}$$

As regards the incremental standard deviation after $N_{SIM}$ simulations ($\sigma_{XN}$), it is computed using eq. (5.5),

$$\sigma_{X_N} = \sqrt{\frac{S_N}{N_{SIM} - 1}} \tag{5.5}$$

where $S_N$ is given by eq. (5.6)

$$\mu_{X_N} = \frac{1}{N_{SIM}} \cdot [X_N + \mu_{X_{N-1}} \cdot (N_{SIM} - 1) \tag{5.6}$$

## 5.2 Pose acquisition results

### 5.2.1 On-line TM and on-line fast-TM

Before starting the evaluation of the performance of the on-line TM and of the on-line fast-TM over an extremely large portion of the 6-DOF relative state space, it is necessary to determine which approach is preferable, between the NN and NS ones, to be selected for the matching step when the ICP is adopted to establish success or failure of pose acquisition.

To this aim, the performance of the on-line TM is analyzed over a set of relative poses selected along a relative trajectory, considering ENVISAT as test case. Specifically, one SE$_{MR}$ is designed around ENVISAT by setting $A_x$ to 22 m, $dxz_{MIN}$ to 20 m and $\tilde{z}$ to 10 and considering the TLE relevant to a recent measurement (January 7th 2014) as starting point for the analysis. As a result of this procedure, the initial mean orbit parameters for the chaser and the target are listed in Table 5.1.

| **Mean orbit parameters** | **Chaser - SE$_{MR}$** | **ENVISAT (7th January 2014)** |
|---|---|---|
| $a$ (km) | 7144.370 | 7144.370 |
| $i$ (°) | 98.4045 | 98.4045 |
| $\Omega$ (°) | 76.2196 | 76.2199 |
| $e$ | $1.277 \cdot 10^{-4}$ | $1.247 \cdot 10^{-4}$ |
| $\omega$ (°) | 86.6180 | 86.9910 |
| True Anomaly (°) | 0.3731 | 0 |

**Table 5.1 - Initial mean orbit parameters to obtain the SE$_{MR}$ around ENVISAT characterized by $A_x$ of 22 m, $dxz_{MIN}$ of 20 m, and $\tilde{z}$ of 10 m.**

This modified safety ellipse has $R$ ranging from about 25 m to 53 m. The SR$_{TM}$ is computed as the percentage of successful pose solutions over 241 positions equally separated in time (50 s) along two relative orbits (see the black dots in Figure 5.1). The time step is selected so that two consecutive poses are significantly different in terms of both relative attitude and position.

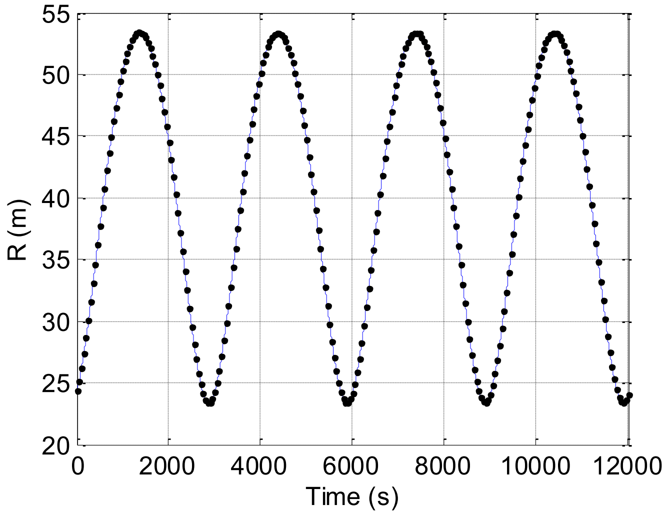


**Figure 5.1 - Time variation of the $R$ along two consecutive orbits (in blue) for the designed SE$_{MR}$. The black dots indicate the positions at which the SR$_{TM}$ is computed.**

The effect of adopting five different values of $\Delta$ (i.e. 60°, 40°, 30°,20°, and 10°) is estimated on the $SR_{TM}$, on the number of templates to be generated and consequently on the computational cost, as shown in Figure 5.2. Of course, the information about the computational cost does not include the time needed to declare success/failure by comparing the accuracy level attained by a subsequent application of the ICP with the 3° error threshold.

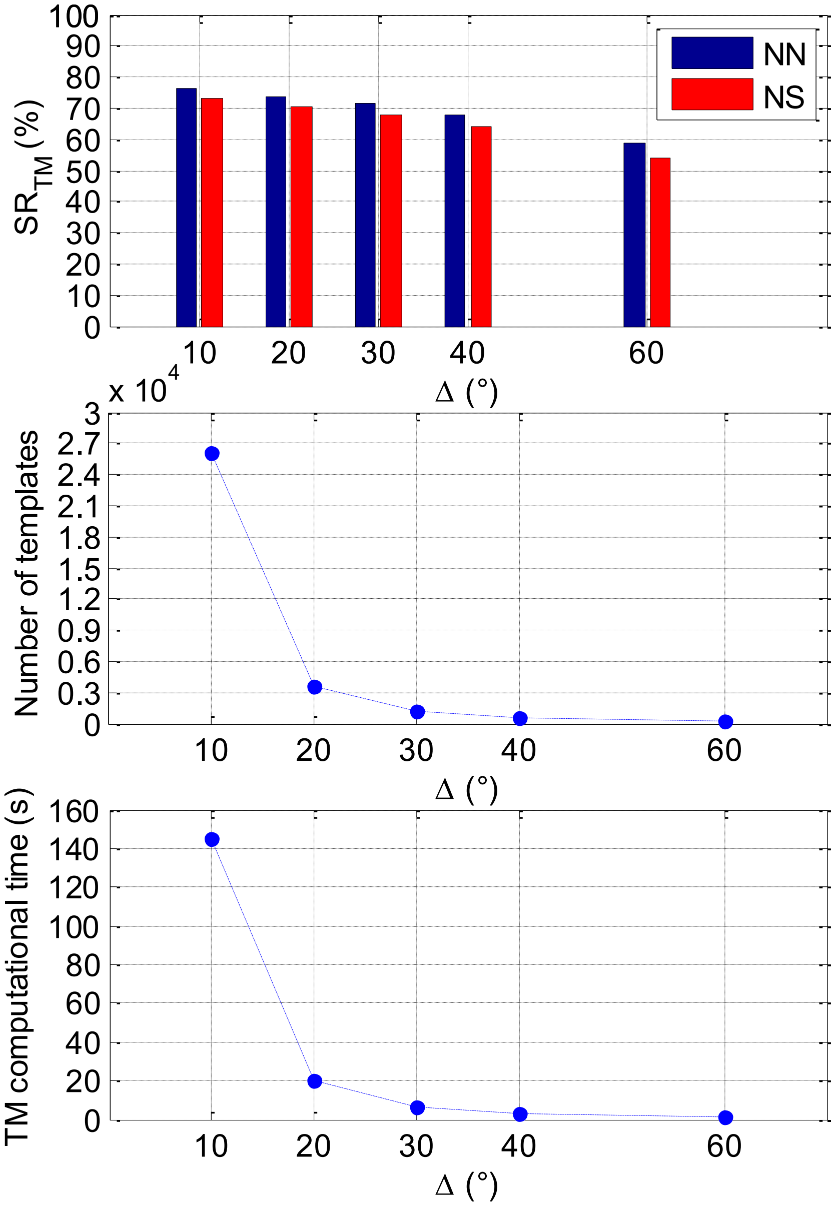


**Figure 5.2 - (Top) Effect of $\Delta$ on $SR_{TM}$ comparing the NN and NS approaches. (Center) Effect of $\Delta$ on the number of templates. (Bottom) Effect of $\Delta$ on the average computational cost.**

This analysis clearly shows that the NN approach is more effective than the NS one, as the associated value of $SR_{TM}$ is larger for any value of $\Delta$. This can be explained by remarking that highly rough initializations of the pose parameters, as the ones provided by the adopted pose acquisition strategy, can cause sensor-model point associations characterized by larger distances when projecting a sensor point on the closest model surface as requested by the NS logic, compared to the ones determined by the NN approach. This is the reason why, in the following, only the NN variant will be used to declare success/failure of the pose acquisition algorithms for performance evaluation. This result also motivates the choice of always adopting the NN method for the first application of the ICP algorithm, which represents the transition from acquisition to tracking to both the on-line TM and the on-line fast-TM.

The analyzed sets of pose parameters cover a very limited portion of the 6-DOF relative state space, being restricted by the assigned relative trajectory. Hence, although the results cannot be considered fully representative of algorithm's performance in terms of the achievable success rate, further interesting comments can be made.

As expected, a reduction of $\Delta$ produces an increase in the $SR_{TM}$ since it allows restraining the angular gap that the tracking algorithm has to compensate. Specifically, if $\Delta$ is 10°, the $SR_{TM}$ reaches its maximum value of 76.3 %, but the number of templates is so large (26000) that the computational time also becomes unacceptably high (145 s) for close-proximity flight. On the other hand, if $\Delta$ is 60°, the number of templates drops down to 196 and so does the computational time (1 s), but the $SR_{TM}$ reduces to 58.9 %. However, it is interesting to notice that the selection of intermediate values of $\Delta$ (20° and 30°) keeps the algorithm's computational time low enough (20 s and 7 s) to enable real-time operations, while simultaneously providing values of the $SR_{TM}$ very close to the maximum (73.4 % and 71.8 %). This can be explained observing that the number of templates reduces of one order of magnitude if $\Delta$ is increased from 10° to 20°, while the average estimation error in the Euler angles (evaluated considering only the successful pose estimates) remains approximately the same, as it is shown in Table 5.2.

| **Average attitude estimation error** | | | | | | |
|---|---|---|---|---|---|---|
| ***Δ* (°)** | | **10** | **20** | **30** | **40** | **60** |
| **Successful pose estimates** | | **184** | **177** | **173** | **163** | **142** |
| **Euler angles** | ***α* (°)** | 9.73 | 14.04 | 13.83 | 39.83 | 56.88 |
| | ***β* (°)** | 6.17 | 6.88 | 8.15 | 14.50 | 18.97 |
| | ***γ* (°)** | 21.40 | 22.70 | 37.45 | 47.62 | 78.45 |

**Table 5.2 - On-line TM mean estimation error in the relative Euler angles averaged over the successful pose estimates within sequence of 241 poses sampled along the designed SE$_{MR}$.**

The error in the yaw angle is larger than the one in pitch and roll for any value of *Δ*. This is motivated by the fact that ENVISAT has one principal direction and, consequently, the relative rotation angle around the corresponding axis of the TRF is more prone to produce ambiguous estimations. Of course, this is also related to the fact that the target simplified model does not include many details present on the satellite external surface. Another important property of the on-line TM can be noticed by comparing the values of the estimation error in the relative position vector when the algorithm is successful to the same ones corresponding to algorithm failures. These results are collected in Table 5.3 where $T_X$, $T_Y$ and $T_Z$ are the components of $\underline{T}$ in the SRF.

| **Relative position vector components** | | **Average position estimation error** | | | | |
|---|---|---|---|---|---|---|
| | | ***Δ* = 10°** | ***Δ* = 20°** | ***Δ* = 30°** | ***Δ* = 40°** | ***Δ* = 60°** |
| **On-line TM success** | **$T_X$ (m)** | 2.809 | 2.697 | 2.689 | 2.772 | 2.853 |
| | **$T_Y$ (m)** | 1.324 | 1.268 | 1.251 | 1.343 | 1.369 |
| | **$T_Z$ (m)** | 1.924 | 1.936 | 1.868 | 1.808 | 1.868 |
| **On-line TM failure** | **$T_X$ (m)** | 4.021 | 4.198 | 4.130 | 3.773 | 3.443 |
| | **$T_Y$ (m)** | 1.588 | 1.714 | 1.732 | 1.478 | 1.412 |
| | **$T_Z$ (m)** | 4.698 | 4.360 | 4.390 | 4.192 | 3.601 |

**Table 5.3 - On-line TM mean estimation error in the relative position vector components averaged over the successful pose estimates within sequence of 241 poses sampled along the designed SE$_{MR}$.**

It is clearly shown that error induced by the centroiding approach in the estimation of the relative position vector, which depends on the specific conditions in terms of target relative attitude, is the one that compromises the algorithm's capability to find the set of sampled Euler angles adequately close to the real triplet. For instance, the on-line TM algorithm is more likely to provide success if the error in the estimation of the along boresight component of $\underline{T}$ is kept below 2 m.

It is now necessary to determine all the potential advantages/drawbacks which can result if the on-line TM is substituted for its fast variant. Specifically, the on-line fast-TM is applied selecting a value of $\tau$ equal to 0.1 in eq. (2.5), meaning that the only templates to be considered for potential matching must have a point distribution with respect to the sensor boresight axis which is more than 10 % different from the one associated to the measured dataset. The choice of this value is driven by the necessity of not to risk losing too many templates which instead could represent potential candidates to minimize the correlation function, $C$, of eq. (2.3). Also for this analysis, ENVISAT is selected as test case. Firstly, the two methods are compared over the 241 sets of pose parameters identified along the previously define $SE_{MR}$.

Results collected in Figure 5.3, immediately show that the on-line fast-TM excludes from the evaluation of $C$ about 66 % of the generated templates for any value of $\Delta$. Although one could expect the computational efficiency to improve in a proportional way, the actual acceleration is limited to 15 % of the computational time associated to the on-line TM. This is due to the fact that the on-line fast-TM has no impact on the time required for templates generation, which represents the main contribution to the overall computational burden. In terms of success rate (SR), the results highlight that the fast-TM strategy causes a loss of performance as compared to the basic approach since it can lead to the exclusion of potentially good candidate templates. This loss of performance is affected by the value of $\Delta$. Specifically, if the angular sampling step is low enough (10° or 20°), the loss of success rate ($SR_{TM}$ - $SR_{fast\text{-}TM}$) is extremely limited (about 1 %), while it increases up to 22 % when $\Delta$ grows to 60°. However, this effect is still not clear since these results are relevant to an analysis carried out over a restricted portion of the 6-DOF relative state space.

This is the reason why the loss of success rate occurring when *Δ* is 40° (10 %) is lower than if *Δ* is 30° (14 %).

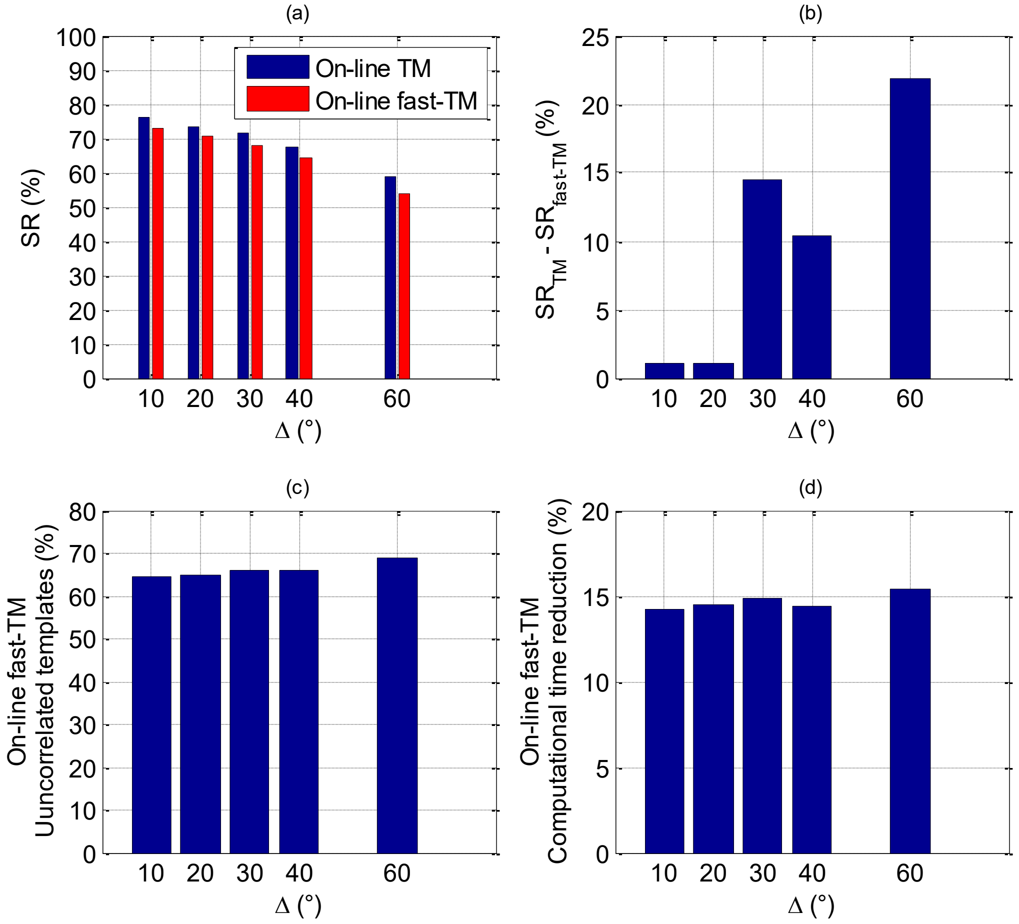


**Figure 5.3 - On-line TM vs. on-line fast-TM. (a) Success rate. (b) Loss of success rate of the on-line fast-TM. (c) Percentage of templates excluded from the correlation function evaluation. (d) Computational time reduction.**

Hence, this study must be extended over a much wider portion of the 6-DOF pose space thus obtaining absolute performance indicators. This is done by evaluating the $\mathrm{SR_{TM}}$ and the $\mathrm{SR_{fast\text{-}TM}}$ over 500 sets of randomly selected Euler angles and considering four different values of *R* (20 m. 30 m, 40 m, and 50 m), as shown in Figure 5.4.

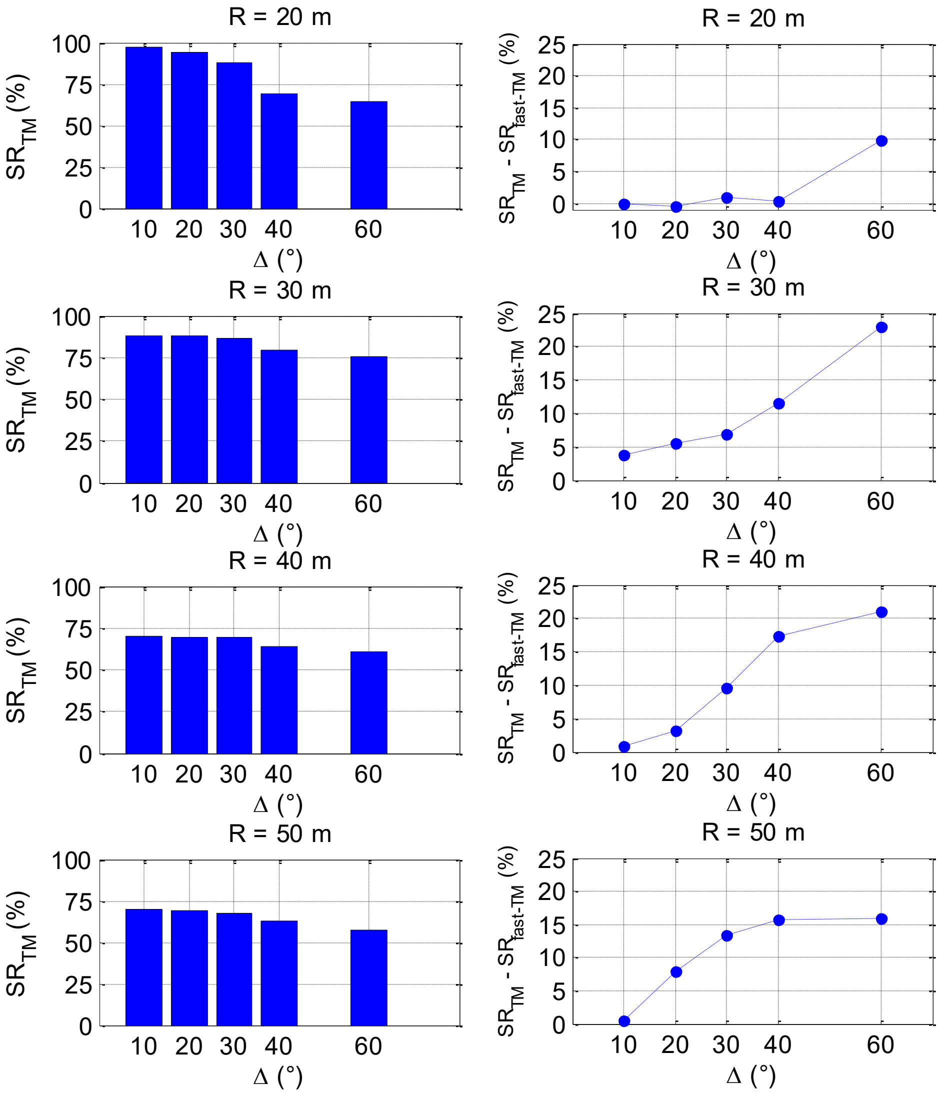


**Figure 5.4 - (Left side) SR$_{TM}$ as a function of $\Delta$ at different ranges. (Right side) Variation of SRfast-TM with respect to SR$_{TM}$ as a function of $\Delta$ at different ranges**

Firstly, it is confirmed that the SR reduces as $\Delta$ increases, as this happens independently of the value of $R$. However, it can be noticed that this effect tends to weaken at farther ranges. Specifically, if $R$ is 20 m, the SR$_{TM}$ rises from 64.8 % up to 97.2 % by changing $\Delta$ from 60° to 10°. On the other hand, if $R$ is 40 m, the SR$_{TM}$ increases from 61 % to only 69.8 % by considering the same variation of $\Delta$.

Secondly, the effectiveness of the proposed approach for pose acquisition is significantly worsened if $R$ increases at a fixed value of $\Delta$. For instance, if $\Delta$ is equal to 30°, the value of $SR_{TM}$ goes from 87.8 % at 20 m down to 67.6 % at 50 m. This phenomenon has two main causes.

- The fixed angular resolution adopted to simulate LIDAR measurements ($\delta_{LOS}$ is 1° in azimuth and elevation), causes a reduction in the size of the measured point clouds (it varies, on average, from about 490 points at 20 m to about 120 points at 50 m). Sparse point clouds can more easily give rise to ambiguous matches (they can produce similar values of *C*)
- The *SNR* of the backscattered laser beams goes down, thus increasing the probability of point misdetection.

Of course, this performance worsening at increasing *R* is independent of how well the attitude parameters space is sampled.

By focusing the attention on the right side of Figure 5.4, it is possible to notice that the increase of *R* has a negative effect also on the SR performance of the on-line fast-TM. For instance, if *R* is 20 m, the loss of success rate remains below 1 % for any *Δ* up to 40°. In particular, by setting *Δ* to 20°, the $SR_{TM}$ - $SR_{fast\text{-}TM}$ becomes -0.4 % thus meaning that the adoption of the fast variant of the proposed TM algorithm is able not only to reduce the computational load but also to slightly improve the performance. On the other hand, if *R* is 30 m, the $SR_{TM}$ - $SR_{fast\text{-}TM}$ is always positive (no SR improvement introduced by the fast variant) and gets worse for increasing *Δ* since it varies from 3.9 % ($\Delta = 10°$) to 23 % ($\Delta = 60°$).

As regards the computational load, also the time saving provided by the on-line fast-TM with respect to the basic TM algorithm is influenced by the variation of the target range. Specifically, as *R* enlarges, the size of the LIDAR point cloud reduces and so the contribution of the correlation determination task to the overall computational burden becomes less important, thus limiting the time acceleration provided by its fast variant. This is shown in Table 5.4, where the analyses performed setting *R* to 20 m and 50 m are considered. Indeed, in spite of a similar percentage of uncorrelated templates, the time acceleration ensured by the on-line fast-TM algorithm with respect to the basic approach varies on average from 24 % at closer range (20 m) to 9 % at farther range (50 m).

| *Δ* (°) | ***R* = 20 m** | |
|---|---|---|
| | **Uncorrelated templates (%)** | **Computational time reduction (%)** |
| **10** | 68.9 | 31.3 |
| **20** | 68.6 | 19.3 |
| **30** | 67.9 | 23.6 |
| **40** | 67.5 | 23.6 |
| **60** | 67.3 | 21.8 |
| *Δ* (°) | ***R* = 50 m** | |
| | **Uncorrelated templates (%)** | **Computational time reduction (%)** |
| **10** | 65.6 | 10.5 |
| **20** | 66.0 | 12.9 |
| **30** | 67.0 | 8.9 |
| **40** | 66.5 | 5.4 |
| **60** | 69.3 | 9.1 |

**Table 5.4 - Effect of the target-chaser relative range on the percentage of uncorrelated templates and on the overall time acceleration characterizing the on-line fast-TM.**

In conclusion, the possibility to substitute the on-line TM with the on-line fast-TM can be justified by the advantage attained in terms of computational efficiency only if the associated loss of success rate is kept to a minimum. Indeed, this phenomenon is clearly caused by the fact that the eq. (2.5) is a reliable measure of the similarity between the templates and the LIDAR point cloud only if enough information to perform the discrimination process are available. This condition is satisfied only if the number of templates is large (*Δ* is low), and if the sensor point cloud is not too sparse thus being able to recognize more geometric details (the sensor is sufficiently close to the target).

Finally, regarding the global performance of the proposed techniques it is possible to state that the selection of intermediate values of *Δ* (20° and 30°) provides the best compromise between the necessities to simultaneously ensure high success rates and computational time which is sustainable for real-time applications.

### 5.2.2 PCA-TM: success rate analysis

Results from the previous sub-chapter motivate why the performance of the PCA-TM is derived by using the NN matching method within the ICP algorithm used to declare success/failure. On the other hand, differently from the cases of the previously analyzed techniques, the transition step from acquisition to tracking requires the ICP algorithm to be applied with both the NN and NS approaches in order to solve any possible ambiguity in the estimated pose.

Numerical simulations are realized covering an extremely wide portion of the 6-DOF relative state space. Specifically, the $SR_{PCA\text{-}TM}$ is evaluated as the number of successful estimates over 1000 sets of randomly generated Euler angles, at different values of *R*. Firstly, it is interesting to focus on the effect of *Δ* on the PCA-TM performance in terms of success rate and computational efficiency, as done for the on-line TM and the on-line fast-TM. Specifically, ENVISAT is considered as test case, *Δ* is varied from 5° to 60°, and the target-chaser relative distance is set to 20 m and 30 m. Results of this simulations are collected in Table 5.5.

| *Δ* (°) | Number of templates | $SR_{PCA\text{-}TM}$ (%), *R* = 20 m | Average computational time (s) | $SR_{PCA\text{-}TM}$ (%), *R* = 30 m | Average computational time (s) |
|---|---|---|---|---|---|
| **5** | 73 | 96.5 | 0.889 | 89.8 | 0.913 |
| **10** | 37 | 96.2 | 0.451 | 89.6 | 0.484 |
| **20** | 19 | 96.2 | 0.238 | 89.5 | 0.240 |
| **30** | 13 | 95.7 | 0.161 | 89.5 | 0.167 |
| **40** | 10 | 91.1 | 0.125 | 89.2 | 0.129 |
| **50** | 8 | 85.3 | 0.102 | 88.4 | 0.104 |
| **60** | 7 | 78.0 | 0.089 | 87.5 | 0.089 |

**Table 5.5 - Effect of the angular sampling step on the success rate and computational time of the PCA-TM, considering ENVISAT as test case at two different values of the relative range.**

One first interesting result of this analysis is that the effect of *Δ* on the $\mathrm{SR}_{\mathrm{PCA\text{-}TM}}$ is weak, compared to the behavior observed for the $\mathrm{SR}_{\mathrm{TM}}$ and the $\mathrm{SR}_{\mathrm{fast\text{-}TM}}$. Indeed, the $\mathrm{SR}_{\mathrm{PCA\text{-}TM}}$ estimated considering a relative range of 20 m is larger than 90 % even if *Δ* is set to 40°, which means restraining the evaluation of eq. (2.3) to only 10 templates. The result obtained setting *R* to 30 m is even more impressive since the $\mathrm{SR}_{\mathrm{PCA\text{-}TM}}$ stays above 87 % independently of *Δ*. Of course, the reduction of the angular sampling step still affects the average computational time, which tends growing as a nearly linear function of the number of templates but is always kept below 1 s. By considering also the step for transition from acquisition to tracking the overall computational time is always kept 2 s.

It is now extremely important to evaluate the effect on the performance of the PCA-TM of the size and shape of the target, as a crucial part of the algorithm is the adoption of the PCA to identify the direction of the target main axis by analyzing the shape of the measured point cloud . Specifically, the analysis will be focused on the success rate since high computational efficiency is always ensured by the extremely limited number of templates. Hence, the $\mathrm{SR}_{\mathrm{PCA\text{-}TM}}$ is estimated considering the three targets presented in sub-chapter 3.5 (ENVISAT, CSM, and RB) and setting *Δ* to 30° (which provides the best combination of high success rate and low computational load). The intervals of distance is (20 m, 80 m) for ENVISAT, and (10 m, 40 m) for CMS and RB. This choices are done taking the size of each target into account, so that the number of points in the cloud goes from a few hundreds to a few tens, while the target goes from partial to full view conditions. Results of these analyses are shown in Figure 5.5, where $PC_{Coverage}$ is the ratio between the mean cross boresight dimensions of the measured point cloud and the target, while $TRGT_{Coverage}$ is the ratio between the maximum cross boresight distance covered by the target and the LIDAR swath width ($S_W$). Specifically, $PC_{Coverage}$ indicates how much the point cloud covers the target surface (averaged over the variable attitude conditions), thus being less than 1 by definition. On the other hand, $TRGT_{Coverage}$ measures how large the maximum cross boresight distance covered by the target is compared to $S_W$. Thus, a value of $TRGT_{Coverage} \leq 1$ implies that the target is completely contained in the LIDAR FOV independently of its relative orientation.

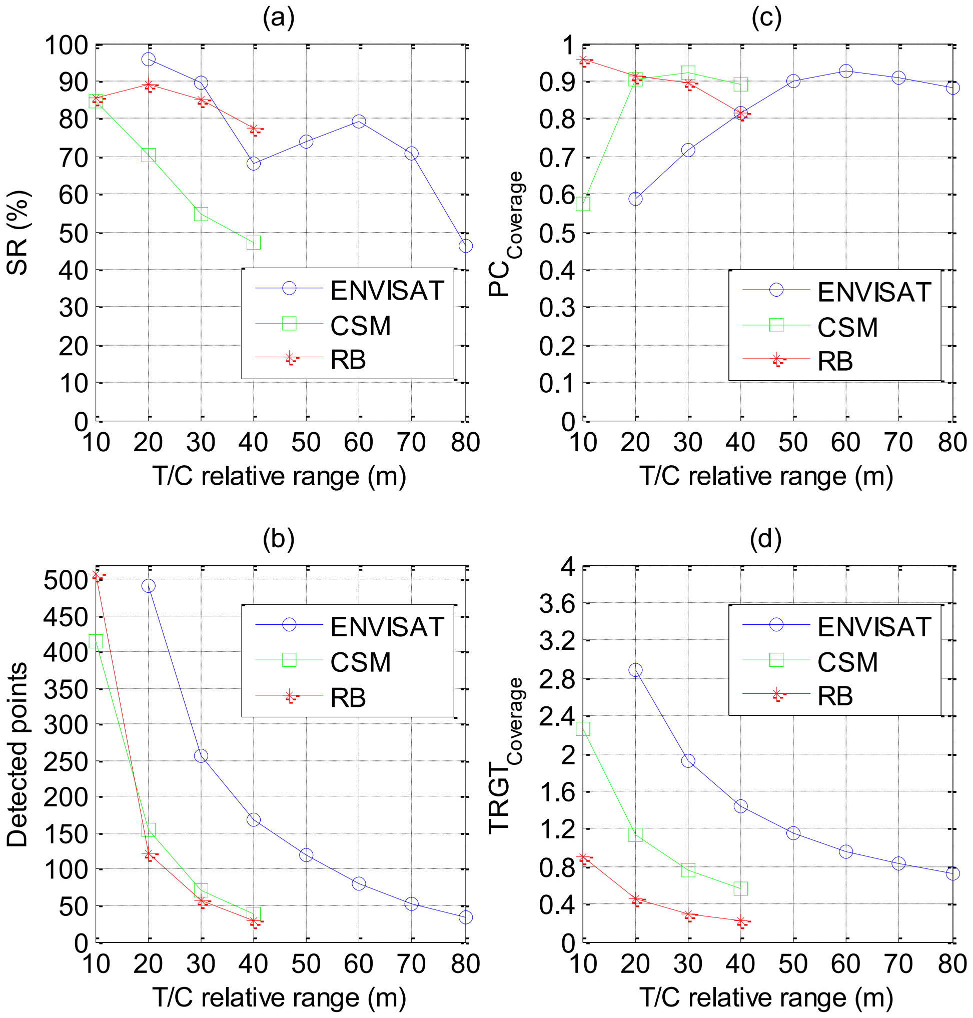


**Figure 5.5 - Performance analysis of the PCA-TM considering different targets and variable range. Effect of the target-chaser (T/C) relative range on the $SR_{PCA-TM}$ (a), the number of detected points (b), the $PC_{Coverage}$ (c), and $TRGT_{Coverage}$ (d).**

Figure 5.5-b shows that, as expected, the number of detected points drops down as the *R* grows. Although this causes an overall reduction of the success rate for any considered target geometry (see Figure 5.5-a), the performance worsening has different aspects to point out depending on the analyzed case.

As regards ENVISAT, the $SR_{PCA-TM}$ is above 68 % for any value of the *R*, excluding 80 m (46.2 %). Indeed, this latter condition is particularly unfavorable due to the limited size of the point clouds (about 30 points on average) covering an extremely large spacecraft (see dimensions of ENVISAT in Table 3.3), and to the high percentage of misdetections (around 30 % on average). Moving from 20 m to 70 m, the variation of the success rate has a peculiar behavior. Specifically, it is characterized by a local minimum (68.2 %) and maximum (79.4 %), respectively found at 40 m and 60 m. This can be explained by observing that the PCA-TM capability of successfully computing the target pose depends not only on the point cloud density, but also on the target observation geometry. Specifically, denser point clouds help the algorithm to distinguish similar (ambiguous) poses, but this task is more complex if the target, and in particular its main axis, is only partially included in the FOV. Hence, from 20 m to 40 m the reduced point cloud size dominates and is responsible of the performance worsening. On the other hand, from 40 m to 60 m, both $PC_{Coverage}$ and $TRGT_{Coverage}$ tend to 1 (meaning that the point cloud covers, on average, the entire target surface and the target is fully contained in the LIDAR $S_W$). Considering the particular shape of ENVISAT, this condition helps in solving for ambiguous poses, so that the success rate exhibits a local maximum. Of course, above 60 m, the further decrease of the $\mathrm{SR}_{\mathrm{PCA\text{-}TM}}$ is produced by the increased level of point cloud sparseness, i.e. the number of points becomes too low.

In spite of comparable values of the number of LIDAR measurements, CSM is characterized by worse performance in terms of success rate. Specifically, the $\mathrm{SR}_{\mathrm{PCA\text{-}TM}}$ goes from 84.4 %, if $R$ is 10 m, down to 46.9 %, if $R$ is 40 m. Indeed, the y-z plane of its TRF is a symmetry one (while ENVISAT is fully not symmetrical), and this makes the pose determination issue much easily subject to ambiguous assignments at relative distances farther than 20 m where CSM starts being fully contained in the sensor FOV. On the opposite, a relative distance of 10 m produces partial views of CSM ($PC_{Coverage}$ of about 60 %) and this helps solving pose ambiguities. In order to fully demonstrate that performance degradation of the PCA-TM in terms of success rate when the target is fully contained in the sensor FOV is exclusively due to the symmetry of the simplified CSM model, ad-hoc simulations are performed. Specifically, the geometry of the target is modified with respect to actual CSM shape by attaching an additional cuboid-shaped element (3 m x 0.1 m x 1.5 m) to its main body, as it is shown in Figure 5.6. Simulation results are compared to the ones obtained for CSM in the interval of $R$ (20, 40) m in Table 5.6. The significant positive effect of introducing an asymmetry in the target geometric model clearly demonstrate the before-mentioned statement. It is particularly interesting to notice that the attained values of $SR_{PCA-TM}$ are better than the ones obtained for ENVISAT in the same interval of ranges. This is mostly related to the fact that the modified CSM is totally in sight by the LIDAR while ENVISAT is only partially contained in the FOV.

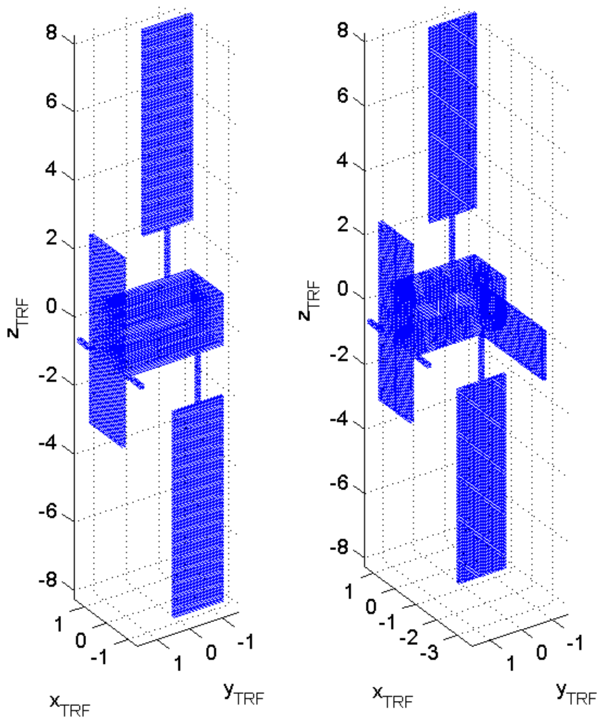


**Figure 5.6 - Model point clouds for CSM (left), and its modified version (right).**

| *R* (m) | $SR_{PCA-TM}$ (%) | |
|---|---|---|
| | CSM | Modified CSM |
| 20 | 70.4 | 94.7 |
| 30 | 54.7 | 93.1 |
| 40 | 46.9 | 89.0 |

**Table 5.6 - Success rate comparison between CSM and its modified version.**

As regards the RB, it is firstly necessary to remark that the pose acquisition algorithm based on the PCA does not require the TM procedure since the z-axis of its TRF is a symmetry one (i.e. the pose has only 5-DOF). In the considered interval of *R*, the RB is always fully contained in the $S_W$ (see Figure 5.5-d), thus the size of the point cloud compared to the target dimension (see Figure 5.5-c) reduces almost linearly for increasing range. Although this determines a global reduction of the success rate as the relative distance goes from 10 m to 40 m, the $SR_{PCA-TM}$ is slightly larger at 20 m (89.2 %) than at 10 m (85.6 %). This derives by the fact that the difference in detected points between the lateral surface and the base of the cylinder is positive and large enough, so that the shape of the point cloud allows identifying its main axis, in a wider range of Euler angles. The fact that the TM step is not required by symmetric cylinder-shaped objects makes the achievable success rate strictly related to their elongation, as it mainly determines the shape of the point cloud. Ad-hoc simulations are performed to show this effect, considering two modified versions of RB. The first one ($RB_{M1}$) is obtained by doubling the radius with respect to RB, thus halving the ratio between the inertia in the symmetry axis and the inertia in the perpendicular direction ($I_{ratio}$). Conversely, the second modified rocket body ($RB_{M2}$) is obtained by doubling the height with respect to RB, thus doubling also $I_{ratio}$. Simulation results are compared to the ones obtained for RB in the interval of *R* (20, 30) m in Table 5.7. As predictable, the higher $I_{ratio}$ is, the larger the $SR_{PCA-TM}$ becomes. Moreover, $RB_{M1}$ is so flattened that the region in the relative attitude space where the PCA is able to identify its main axis is extremely limited, thus compromising the achievable success rate.

| Target | Radius (m) | Height (m) | $I_{ratio}$ | $SR_{PCA-TM}$ (%) | |
|---|---|---|---|---|---|
| | | | | $R$ = 20 m | $R$ = 30 m |
| $RB_{M1}$ | 2.4 | 6.5 | 2.7 | 35.3 | 27.4 |
| RB | 1.2 | 6.5 | 5.4 | 89.2 | 85.1 |
| $RB_{M2}$ | 1.2 | 13 | 10.8 | 97.6 | 97.3 |

**Table 5.7 - Success rate comparison between cylinder-shaped targets with different $I_{ratio}$.**

### 5.2.3 PCA-TM: robustness improvement

Two point clouds for each the three test cases adopted as target (ENVISAT, CSM and RB), obtained considering a relative range of 30 m, are shown in Figure 5.7. Specifically, the point clouds on the left and on the right correspond to attitude conditions for which the PCA-TM provides, respectively, failure and success.

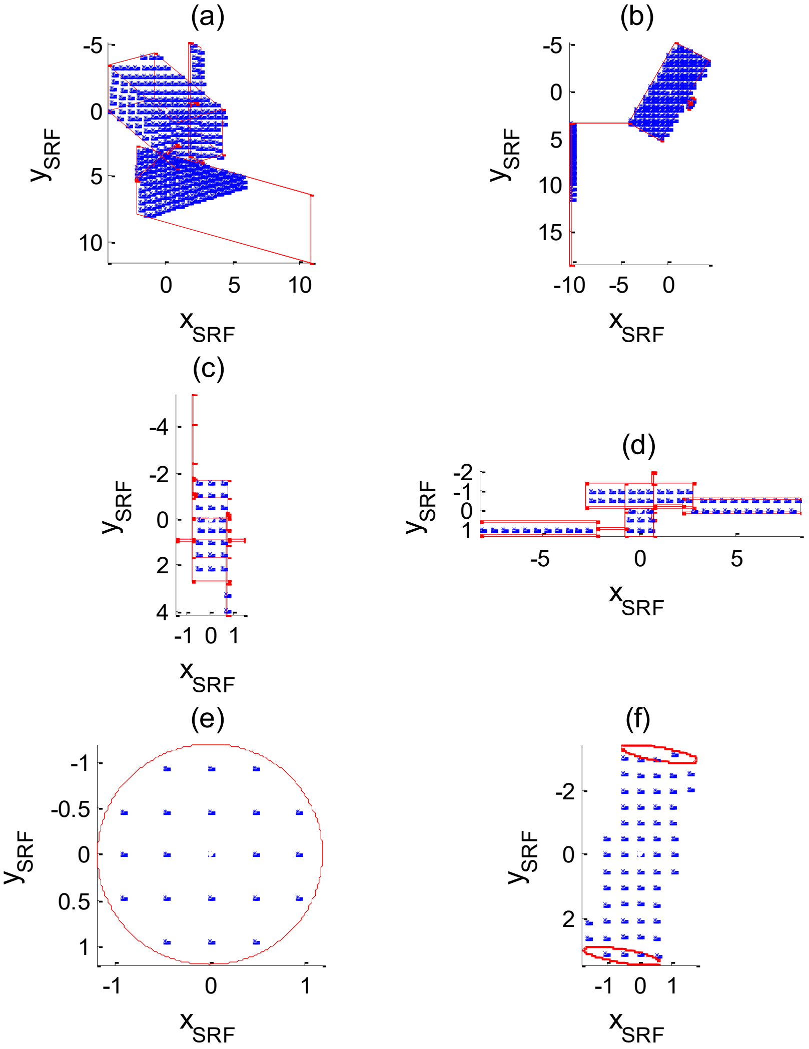


**Figure 5.7 - Examples of target point clouds at 30 m of relative range: (a) ENVISAT, $\alpha$=150°, $\beta$=-30° and $\gamma$=0°; (b) ENVISAT, $\alpha$=90°, $\beta$=30° and $\gamma$=0°; (c) CSM, $\alpha$=30°, $\beta$=0° and $\gamma$=0°; (d) CMS, $\alpha$=120°, $\beta$=90° and $\gamma$=0°; (e) RB, $\alpha$=0°, $\beta$=0° and $\gamma$=0°; (f) RB, $\alpha$=80°, $\beta$=10° and $\gamma$=0°.**

Figure 5.7 clearly shows how the success/failure of the PCA-TM strongly depends on the shape of the acquired point cloud, meaning that the measured points must be spread enough in the direction of the main axis, $\underline{e}_M$. Since this condition is certainly not satisfied if the target is viewed by the LIDAR with a very shallow angle, the algorithm's robustness against such situations must be improved. To this aim, two solutions are envisaged, which are presented in the current sub-chapter.

Firstly, it is necessary to apply the consistency check defined in sub-chapter 2.4.3 which leads to the modified architecture for the PCA-TM shown in Figure 2.14. The criterion introduced to a-priori establish whether the target main axis direction can be easily identified or not by analyzing the shape of the measured point cloud is here recalled and its applicability is assessed thanks to ad-hoc simulations. The basic idea is that the ratio ($r$) between the maximum and minimum eigenvalues of the covariance matrix, $\underline{\underline{Q}}$ defined in eq. (2.6), is a direct measure of the elongation of the object observed from a specific point of view. Hence, the higher the value of $r$ is, the more reliable the estimated $\underline{e}_M$ direction is. Specifically, the PCA is applied to 10000 point clouds obtained by assigning a fixed value to $R$ (30 m) and $\gamma$ (0°), while the couples of $\alpha$ and $\beta$ are extracted from two uniform distributions, respectively in the intervals (-180°, 180°) and (-90°, 90°). This analysis is performed for each of the assigned targets to identify the regions in the $\alpha$-$\beta$ plane where $\underline{e}_M$ is a good representation of the z-axis ($\underline{z}$) of the TRF (red regions in Figure 5.8). This condition, which is verified by applying the same ICP-based method defined in sub-chapter 5.1, is not satisfied (blue regions in Figure 5.8) for ENVISAT and CSM when their views are occluded respectively by the solar array and the SAR antenna, and for RB if the lateral surfaces are poorly in sight of the LIDAR. The behavior of $r$, depicted in Figure 5.9, clearly shows the correlation between large (low) values value of $r$ and the algorithm's success (failure). In order to visualize the effect of the relative range, the PCA-TM success/failure and of the value of $r$ in the $\alpha$-$\beta$ plane are depicted considering a larger value of the target-chaser distance (40 m), respectively in Figure 5.10 and Figure 5.11. At this farther range the correlation is still evident, as the number of failures and the portion of the $\alpha$-$\beta$ plane where $r$ becomes too low tend to enlarge. This is particularly evident for RB, as it can be seen by looking at Figure 5.9-c and Figure 5.11-c.

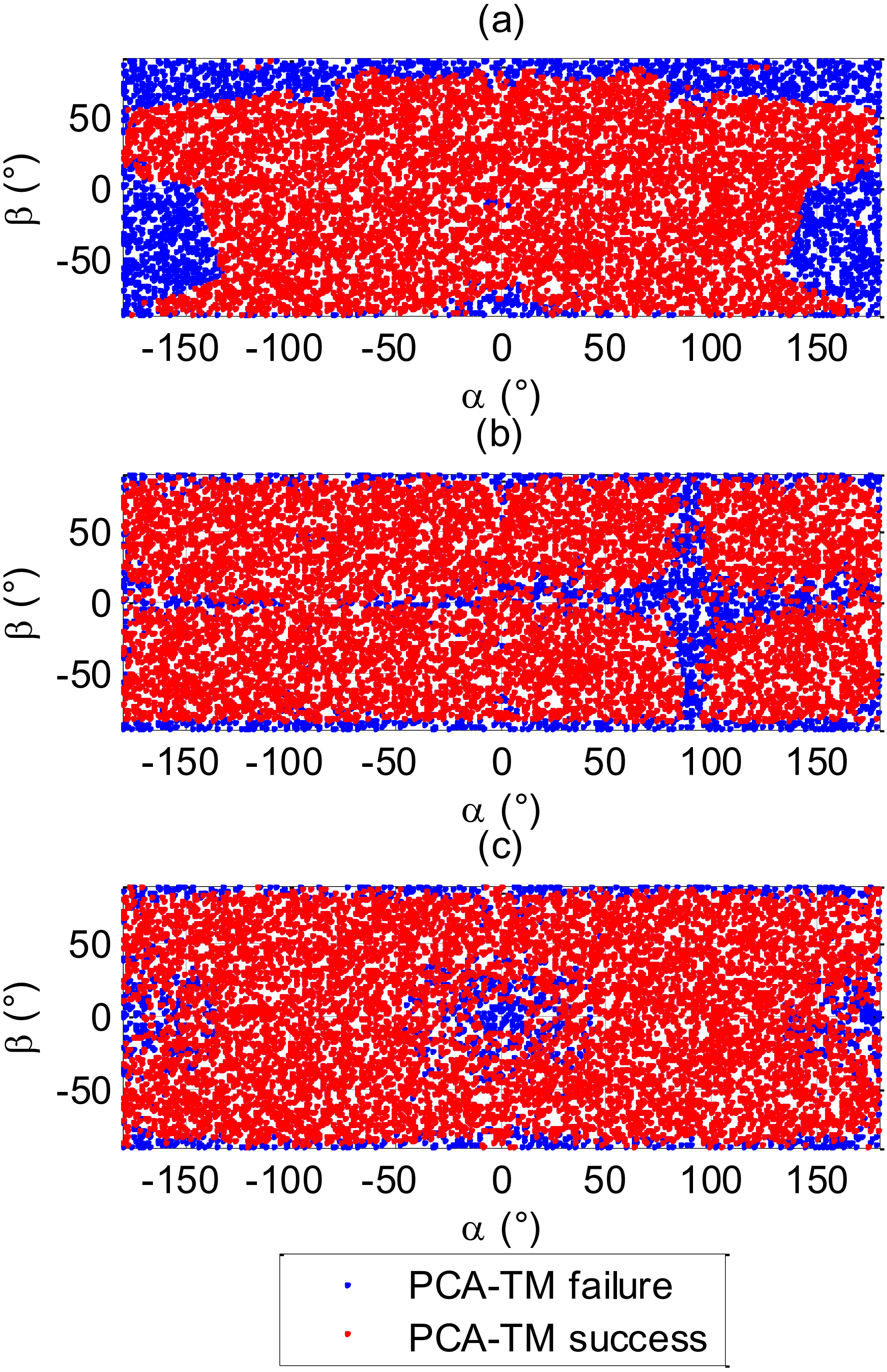


**Figure 5.8 - PCA estimation of the target main axis in the $\alpha$-$\beta$ plane, considering a relative range of 30 m. (a) ENVISAT; (b) CSM; (c) RB.**

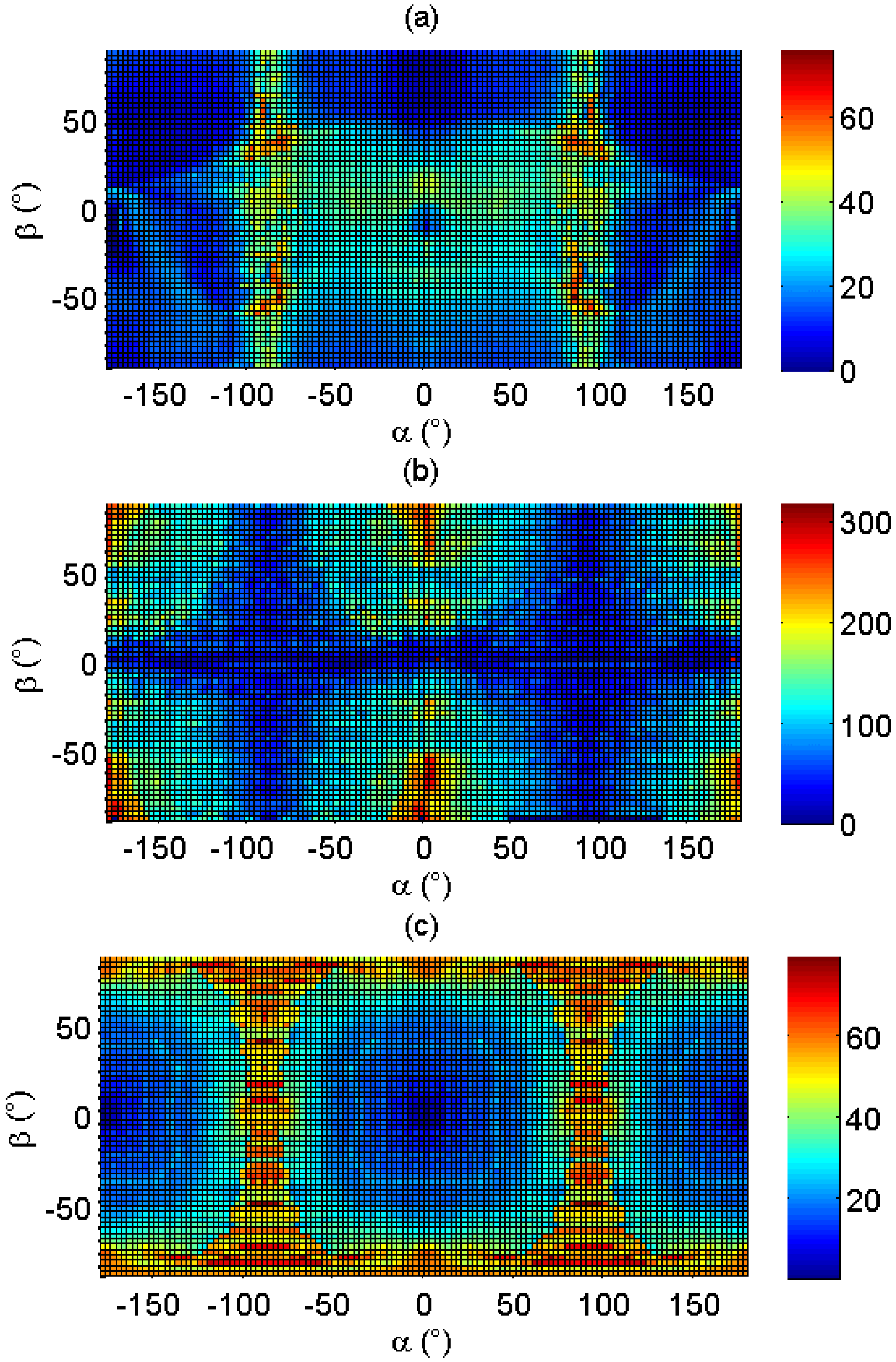


**Figure 5.9 - Behavior of *r* in the *α*-*β* plane, considering a relative range of 30 m. (a) ENVISAT; (b) CSM; (c) RB.**

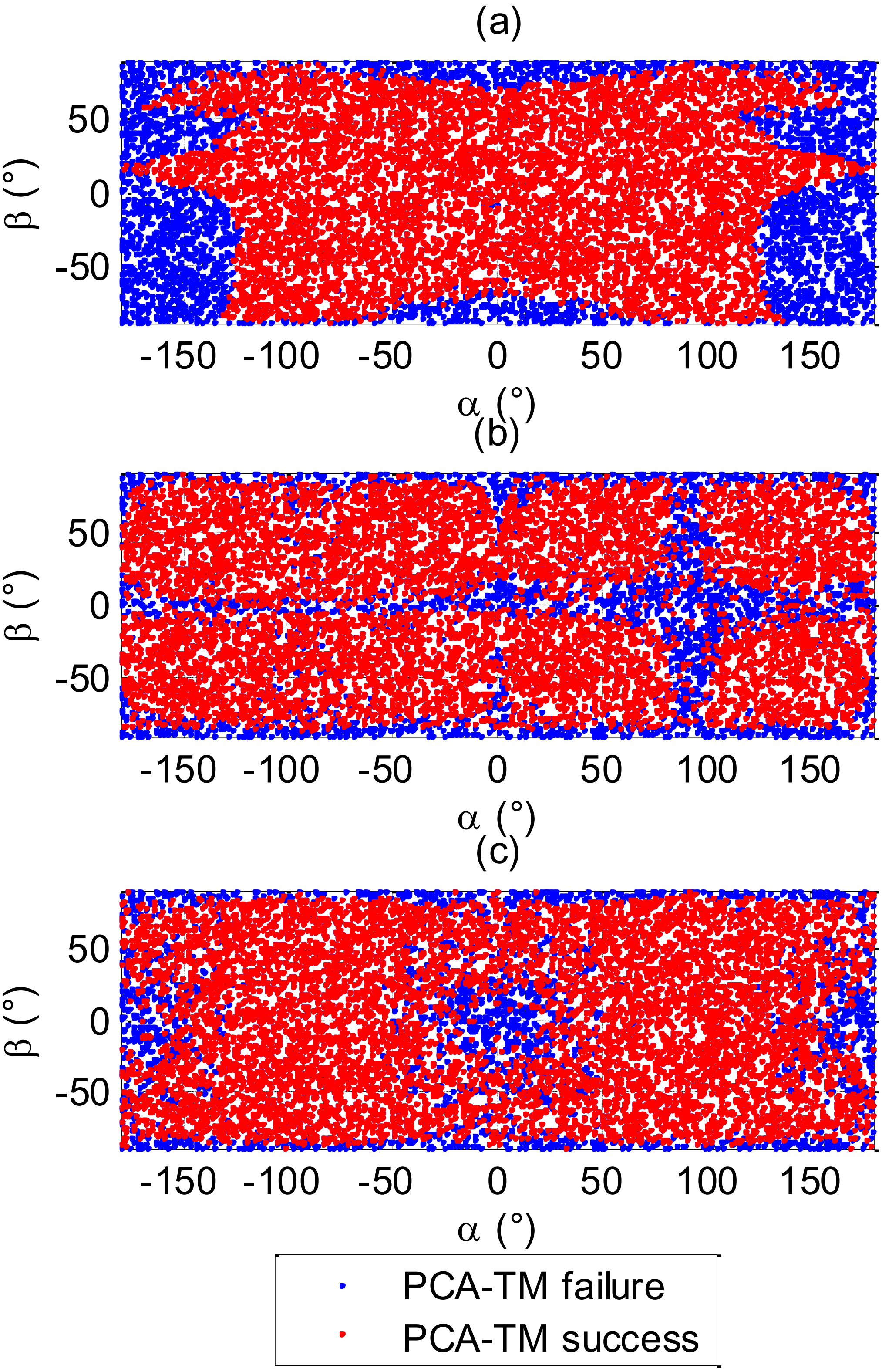


**Figure 5.10 - PCA estimation of the target main axis in the *α*-*β* plane, considering a relative range of 40 m. (a) ENVISAT; (b) CSM; (c) RB.**

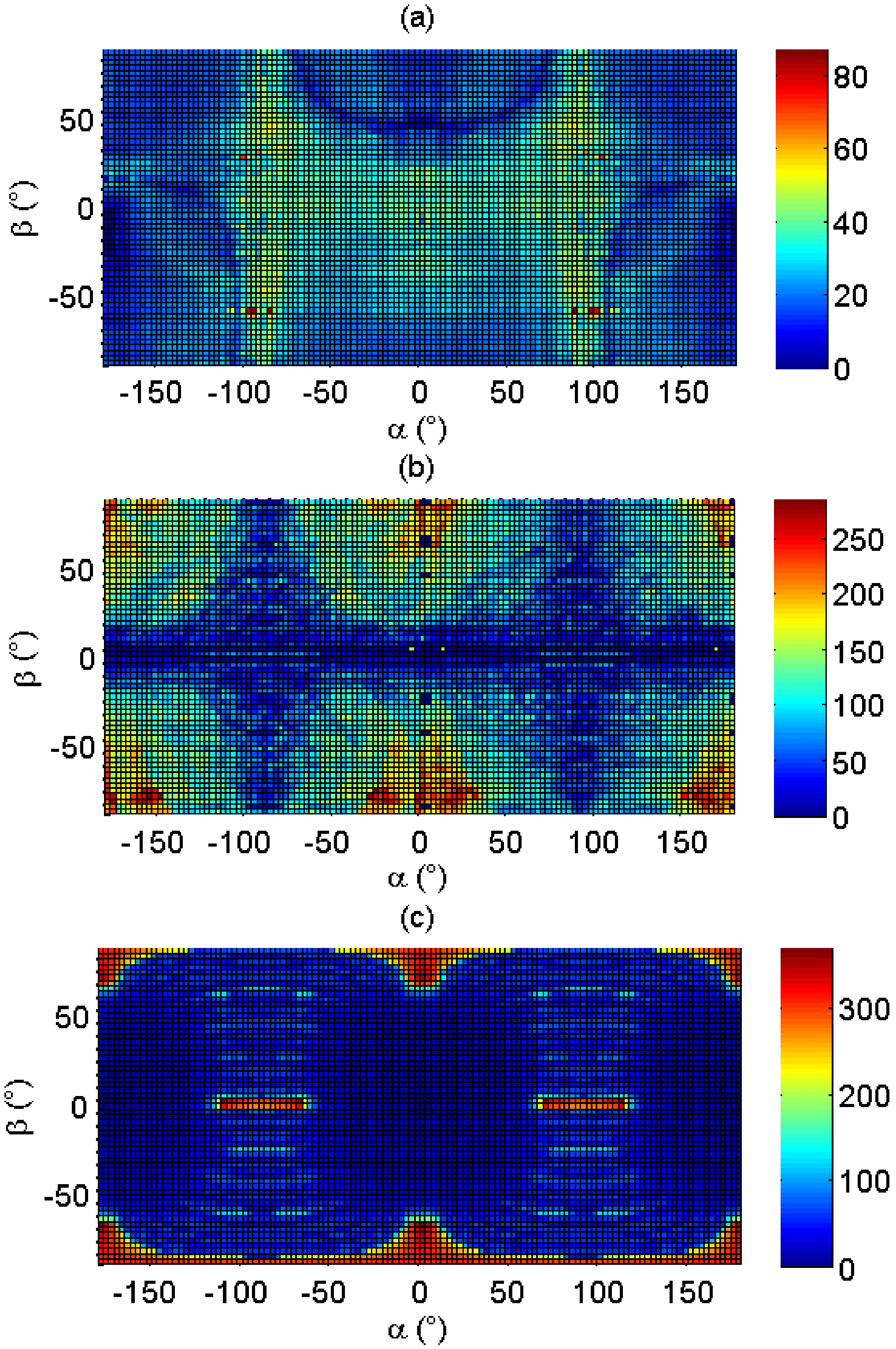


**Figure 5.11 - Behavior of *r* in the *α*-*β* plane, considering a relative range of 40 m. (a) ENVISAT; (b) CSM; (c) RB.**

Hence, the level of robustness of the pose acquisition step can be improved by running the PCA-TM algorithm only if the estimated $r$ is higher than the safety threshold, $\tau_{\lambda}$, as shown in Figure 2.14. Practically, if the LIDAR views the target at a very small angle with respect to its main axis, the PCA will probably give in output a direction different from the actual one. This produces a value of $r$ much lower than the one corresponding to a situation in which the main axis is perfectly in sight. Hence, it possible to discard this PCA solution exploiting $\tau_{\lambda}$. In that case, it is necessary to wait for a certain amount of time, during which the true pose may evolve toward most favorable viewing conditions.

The second solution to improve the robustness of the PCA-TM is to exploit the analytical approach for relative trajectory design presented in chapter 4. The main concept of this method is to derive the differences between the target-chaser mean orbit parameters by assigning high level constraints regarding the stability and the shape of the relative trajectory. Hence, the crucial point is that the target observation condition can be optimized (ensuring that its main axis is globally visible during most of the relative trajectory) by properly selecting one single geometrical parameter, i.e. the out-of plane observation angle, $\varphi_{OUT}$, defined in eq. (4.39). In order to demonstrate this point, the locus of the roll and pitch values corresponding to four relative trajectories around ENVISAT is superimposed to the Figure 5.8-a, as it is shown in Figure 5.12. The first three trajectories are defined by the initial mean orbit parameters collected in Table 4.1. In spite of having the same minimum range constraint (25 m), the values of $\varphi_{OUT}$ are different. Specifically, the 25-m-radius circular trajectory has $\varphi_{OUT}$ equal to 30°, the $\mathrm{SE}_{\Delta\mathrm{Rmin}}$ ($A_x$ is 21 m and $dxz_{MIN}$ is 13 m) has $\varphi_{OUT}$ equal to 53°, and the $\mathrm{SE}_{\mathrm{MR}}$ ($A_x$ is 16 m, $dxz_{MIN}$ is 8 m, and $\tilde{z}$ is 25 m) has $\varphi_{OUT}$ equal to 32.6°. The fourth relative trajectory is another $\mathrm{SE}_{\mathrm{MR}}$ ($A_x$ is 24 m, $dxz_{MIN}$ is 22 m, and $\tilde{z}$ is 9 m) which also has a minimum range of 25 m but $\varphi_{OUT}$ equal to 69.4°. By looking at Figure 5.12, it is possible to state that low values of $\varphi_{OUT}$ represent the most convenient design choice since they ensure the principal direction of ENVISAT to be constantly in sight of the LIDAR during their relative motion. This comes from the fact that ENVISAT is gravity gradient stabilized, thus the larger $\varphi_{OUT}$ is, the more the chaser will look at it from above/below and this observation geometry clearly prevents the LIDAR from providing enough elongated datasets during part of the relative trajectory. Hence, it is clear that the design strategy related to the $SE_{MR}$ represents the best trade-off between safety and observation geometry requirements since it also allows to directly selecting the desired value of $\varphi_{OUT}$ by properly assigning its design parameters according to eq. (4.39). On the other hand, the $SE_{\Delta Rmin}$ model satisfies the safety conditions but it does not allow to freely selecting $\varphi_{OUT}$, while the circular trajectory strategy, in spite of having the advantage of ensuring a fixed value of $\varphi_{OUT}$ (30°), has to be discarded since it does not avoid the chaser to pass ahead/behind the target.

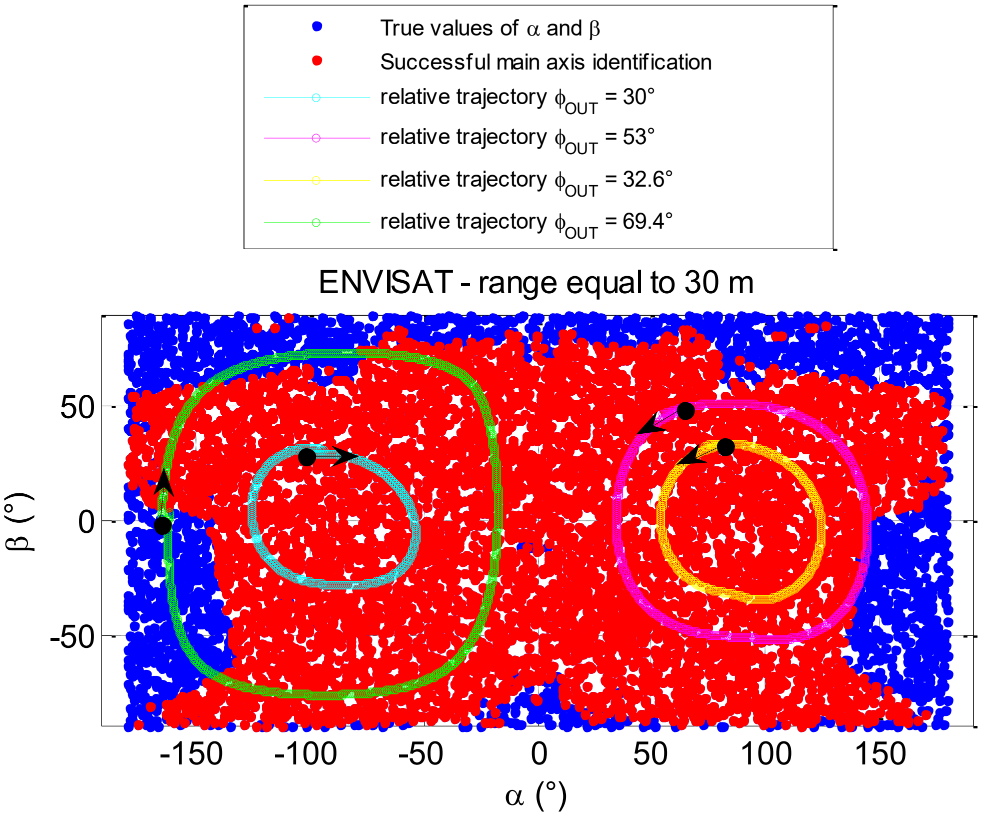


**Figure 5.12 - Locus of the roll and pitch angles of four relative trajectories around ENVISAT in the α-β plane. Red (blue) dots correspond to successes (failures) of the PCA in identifying the target main axis at a relative range of 30 m.**

In order to confirm this result, two $SE_{MR}$ which satisfy the same minimum range criterion and are characterized by different values of $\varphi_{OUT}$ are designed around CSM and RB. Specifically, the two SE around CSM ensure a minimum range of 11 m and have $\varphi_{OUT}$ equal to 51.34° and 29.74°, while the two safety ellipse around RB ensure a minimum range of 7 m and have $\varphi_{OUT}$ equal to 81.87° and 19.65°. These trajectories have been obtained selecting the design parameters collected in Table 5.8..

| **Design parameters** | **$SE_{MR}$ around CSM** | | **$SE_{MR}$ around RB** | |
|---|---|---|---|---|
| **$A_x$ (m)** | 8 | 10 | 5 | 7 |
| **$dxz_{MIN}$ (m)** | 6 | 8 | 4 | 5 |
| **$\tilde{z}$ (m)** | 14 | 8 | 14 | 1 |
| **$\varphi_{OUT}$ (°)** | 29.74 | 51.34 | 19.65 | 81.87 |

**Table 5.8 - Geometrical design parameter for two $SE_{MR}$ around CSM and RB which ensure the same minimum range constraint (11 m and 7 m respectively) but different values of $\varphi_{OUT}$.**

Hence, the analysis which determines successes and failures of the PCA in the $\alpha$-$\beta$ plane is realized for both CSM and RB considering a relative range to 20 m. This allows superimposing the above-defined trajectories to the success/failure representation as done in Figure 5.12. This is done in Figure 5.13 for CSM and in Figure 5.14 for RB. It is interesting to notice that previous results about ENVISAT are confirmed by the analysis regarding the rocket body. Indeed it is gravity gradient stabilized, so that it is possible to ensure its lateral surface to be always in sight of the LIDAR by designing a relative trajectory characterized by a low value of $\varphi_{OUT}$. On the other hand, CSM is three-axis stabilized with its main direction toward the along-track axis and the SAR antenna pointed toward the Earth. Hence, the LIDAR is able to generate elongated point clouds during most part of the relative motion almost independently of the out-of-plane observation angle. In addition, it is worth outlining that although the most critical observation condition for CSM takes place when the chaser looks at it from directly below the SAR antenna, the designed SEs are both able to avoid the corresponding region in the $\alpha$-$\beta$ plane.

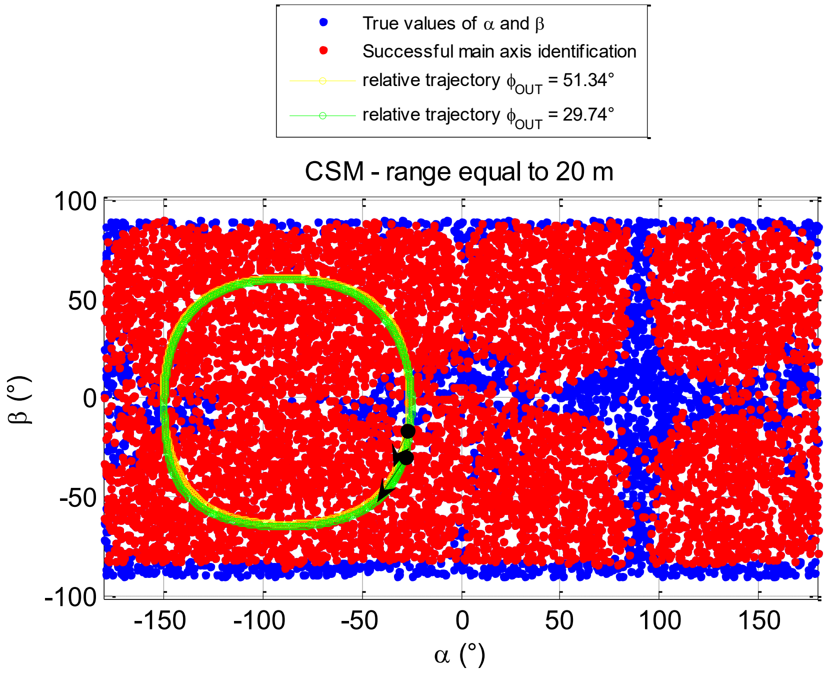


**Figure 5.13 - Locus of the roll and pitch angles of two relative trajectories around CSM in the α-β plane. Red (blue) dots correspond to successes (failures) of the PCA in identifying the target main axis at a relative range of 20 m.**

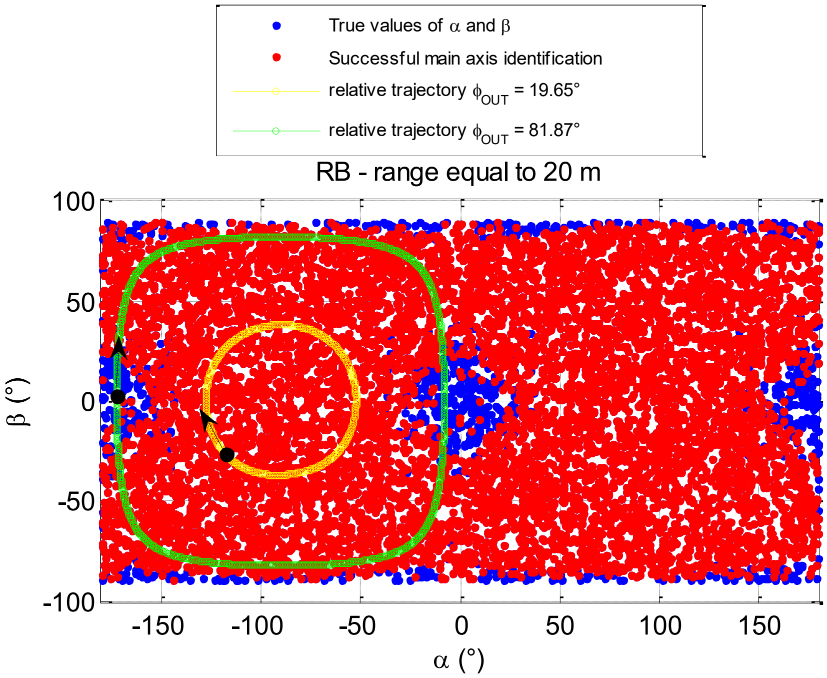


**Figure 5.14 - Locus of the roll and pitch angles of two relative trajectories around RB in the α-β plane. Red (blue) dots correspond to successes (failures) of the PCA in identifying the target main axis at a relative range of 20 m.**

### 5.2.4 Global performance summary

In order to conclude the performance assessment of the techniques proposed for pose acquisition, it is convenient to summarize and compare the achieved results in terms of success rate (computed considering highly variable relative position and attitude conditions) and computational efficiency. Specifically, Table 5.9 collects the values of $\mathrm{SR}_{\mathrm{TM}}$, $\mathrm{SR}_{\text{fast-TM}}$, and $\mathrm{SR}_{\text{PCA-TM}}$, as a function of $\Delta$, considering ENVISAT as test case and setting $R$ to 20 m and 50 m.

| ***R* = 20 m** | **Δ (°)** | **$\mathrm{SR}_{\mathrm{TM}}$ (%)** | **$\mathrm{SR}_{\text{fast-TM}}$ (%)** | **$\mathrm{SR}_{\text{PCA-TM}}$ (%)** |
|---|---|---|---|---|
| | **10** | 97.2 | 97.2 | 96.2 |
| | **20** | 94.0 | 94.4 | 96.2 |
| | **30** | 87.6 | 86.8 | 95.7 |
| | **40** | 69.2 | 69.0 | 91.1 |
| | **60** | 64.8 | 58.4 | 78.0 |
| ***R* = 50 m** | **Δ (°)** | **$\mathrm{SR}_{\mathrm{TM}}$ (%)** | **$\mathrm{SR}_{\text{fast-TM}}$ (%)** | **$\mathrm{SR}_{\text{PCA-TM}}$ (%)** |
| | **10** | 70.2 | 69.8 | 74.9 |
| | **20** | 68.8 | 63.4 | 74.7 |
| | **30** | 67.6 | 58.6 | 73.9 |
| | **40** | 62.7 | 52.8 | 74.6 |
| | **60** | 57.6 | 48.4 | 74.4 |

**Table 5.9 - Success rate comparison between the proposed techniques for pose acquisition.**

This comparison clearly shows that the PCA-TM provides best performance both at close range (20 m) and middle range (50 m). This means that this statement is valid in the case of both partial and global target visibility. Indeed, if $R$ is 20 m, the percentage of coverage of the surfaces of ENVISAT ensured by the point cloud is on average 60 %, while almost total coverage is obtained when $R$ is 50 m($PC_{Coverage}$ is on average 90 %). The success rate provided by the on-line TM and the on-line fast-TM algorithm can be kept just slightly below the ones attained by the PCA-TM only if the angular sampling step is low enough (adequate sampling of the 3-DOF relative attitude space). In particular, the value of $\Delta$, below which the reduction of $\mathrm{SR}_{\mathrm{TM}}$ and $\mathrm{SR}_{\text{fast-TM}}$ becomes too relevant, gets lower as the range increases. For instance, at close range, the $\mathrm{SR}_{\mathrm{TM}}$ and the $\mathrm{SR}_{\text{fast-TM}}$ are even larger than $\mathrm{SR}_{\text{PCA-TM}}$ if $\Delta$ is 10° and they are still comparable to this latter quantity even if $\Delta$ is 30°. On the other hand, at farther range comparable performance is attained only if $\Delta$ is 10°. This phenomenon is related to the fact the $\mathrm{SR}_{\text{PCA-TM}}$ is almost independent on the value of $\Delta$.

Table 5.10 contains the corresponding computational time analysis, where in the case of the PCA-TM the time required by the step for transition from acquisition to tracking is included.

| | | **Number of templates** | | | **Computational time (s)** | | |
|---|---|---|---|---|---|---|---|
| ***R* = 20 m** | ***$\Delta$* (°)** | **On-line TM** | **On-line fast-TM (mean)** | **PCA-TM** | **On-line TM** | **On-line fast-TM (mean)** | **PCA-TM** |
| | **10** | 26011 | 12158 | 37 | 137.99 | 123.46 | 1.53 |
| | **20** | 3610 | 1706 | 19 | 19.40 | 16.89 | 1.27 |
| | **30** | 1183 | 562 | 13 | 6.35 | 5.79 | 1.18 |
| | **40** | 500 | 238 | 10 | 2.60 | 2.46 | 1.14 |
| | **60** | 196 | 102 | 7 | 1.02 | 0.92 | 1.11 |
| ***R* = 50 m** | | **Number of templates** | | | **Computational time (s)** | | |
| | ***$\Delta$* (°)** | **On-line TM** | **On-line fast-TM (mean)** | **PCA-TM** | **On-line TM** | **On-line fast-TM (mean)** | **PCA-TM** |
| | **10** | 26011 | 8936 | 37 | 193.24 | 132.79 | 1.53 |
| | **20** | 3610 | 1229 | 19 | 23.96 | 19.34 | 1.34 |
| | **30** | 1183 | 391 | 13 | 8.11 | 6.19 | 1.28 |
| | **40** | 500 | 167 | 10 | 3.41 | 2.61 | 1.27 |
| | **60** | 196 | 60 | 7 | 1.36 | 1.06 | 1.29 |

**Table 5.10 - Computational time comparison between the proposed techniques for pose acquisition.**

The PCA-TM is characterized by a very low computational load, independently of $\Delta$, due to the extremely restrained number of templates (the TM is applied to a database obtained sampling a 1-DOF relative state space). On the other hand, the computational cost relative to the on-line TM and the on-line fast-TM is low enough (a few seconds) to make them applicable for pose acquisition during close-proximity flights, while simultaneously ensuring acceptable success rate only if *Δ* is set to 30°.

In conclusion, the PCA-TM outperforms the other techniques conceived in this thesis for pose acquisition, both in terms of success rate and computational efficiency. Indeed, this performance is comparable with the results obtained by the most advanced 3D techniques available in the literature [55, 56], but it is achieved considering higher variability of the pose conditions and of the size and shape of the target, and more relevant sparseness of the measured point clouds.

Of course, it is worth clarifying that although a very large portion of both operating spacecraft and large debris in orbit do have a principal direction, this condition may not be satisfied in a few cases. If this latter case occurs, the on-line TM and the on-line fast-TM can still be exploited for the acquisition step within the architecture proposed for pose determination of an uncooperative target since they have demonstrated good performance over an extremely wide interval of relative attitude and relative position states.

## 5.3 Pose tracking results

### 5.3.1 Performance comparison of ICP algorithm variants

The first step of the performance assessment of the ICP algorithm, described in sub-chapter 2.5, is to compare the results obtained implementing different potential strategies. Specifically, four variants are selected by coupling, or not, the NN and NS methods, used in the matching step, with the pose prediction algorithm introduced to accelerate the ICP convergence. The analysis is realized selecting ENVISAT as test case. Moreover, it is necessary to assign the relative trajectory (and the corresponding number of orbits) during which the pose must be tracked. Specifically, the relative trajectory around ENVISAT is the $SE_{MR}$ defined in sub-chapter 5.2.1 ($A_x$ is 22 m, $dxz_{MIN}$ is 20 m and $\tilde{z}$ is 10). As soon as the acquisition step is accomplished, the tracking algorithm is applied for the time interval during which the chaser describes two complete relative orbits around the target, i.e. about 12000 s. This allows testing the proposed technique in a wide interval of relative Euler angles and range, whose time variation along the relative trajectories is shown in Figure 5.15. Indeed, the time variation of $\gamma$ (see Figure 5.15-a) is slowed down due to the combination of its absolute yaw motion and the relative orbit angular velocity.

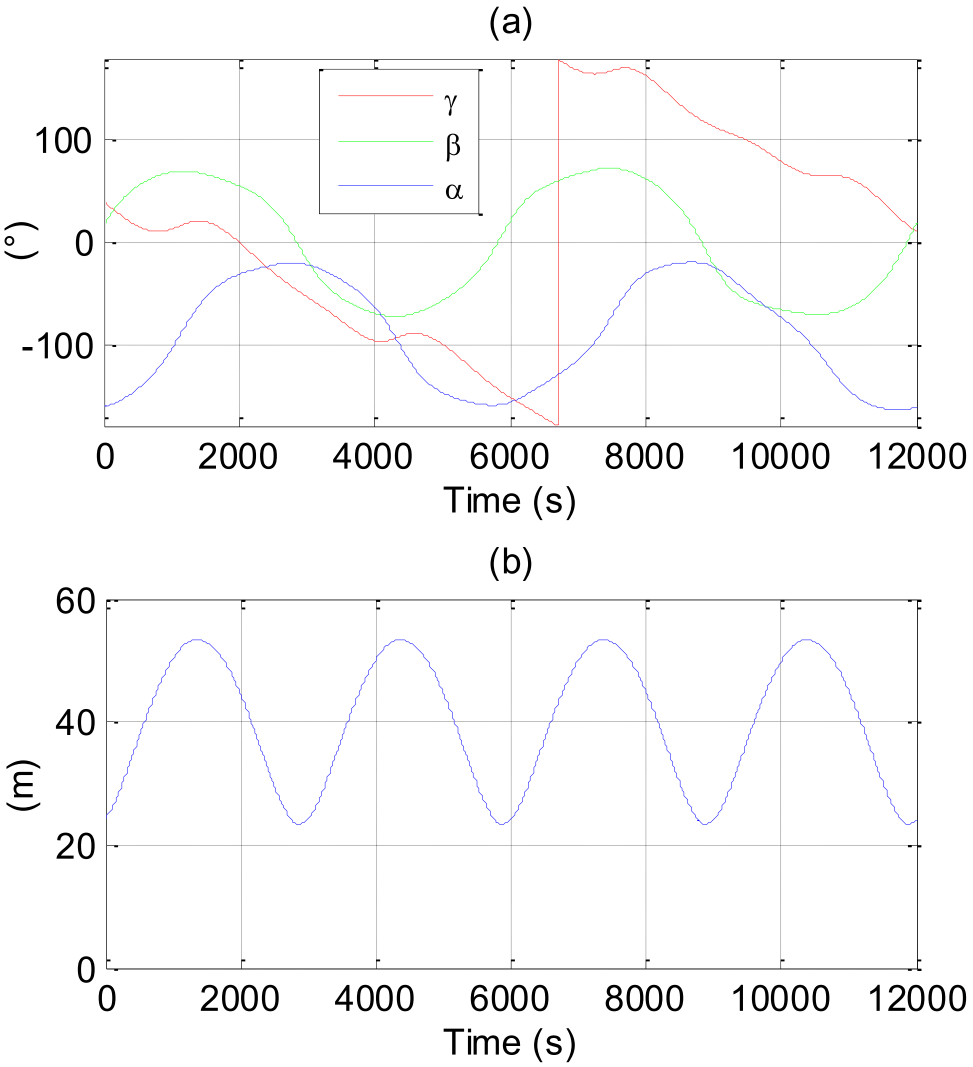


**Figure 5.15 - Time variation of the true relative pose parameters during two consecutive orbits around ENVISAT. (a) Euler angles. (b) Range.**

It is also worth noticing that while the relative range varies from 23 m to 53 m, the number of detected points varies in the interval (45, 887) while the average value is about 200. It is clear that the choice of considering a very poor resolution for the LIDAR leads to extremely challenging conditions for pose determination toward which the implemented ICP variants are tested.

The performance of the four variants of the ICP algorithm is evaluated both in terms of pose estimation accuracy and computational time. Specifically, the statistical analysis presented in sub-chapter 5.1 is implemented by setting the number of simulations to 100. Simulation results are collected in Table 5.11, where the accuracy level is measured by the rms of $\varphi_{ERR}$ and $|\underline{T}|_{ERR}$, respectively for the relative attitude and position.

| **ICP algorithm variant** | $\varphi_{ERR}$ **(°) rms** | $\lvert\underline{T}\rvert_{ERR}$ **(cm) rms** | $t_{CONV}$ **(ms) mean** | $N_{IT}$ **mean** |
|---|---|---|---|---|
| **NN** | 0.28 | 1.3 | 39 | 8.3 |
| **NS** | 0.24 | 1.3 | 19 | 9.8 |
| **NN aided with prediction** | 0.24 | 1.0 | 39 | 7.8 |
| **NS aided with prediction** | 0.21 | 1.1 | 16 | 8.7 |

**Table 5.11 - Statistical analysis of performance of the ICP variants in terms of pose estimation accuracy and computational time. The results are computed for two relative orbits around ENVISAT and averaged over 100 simulations.**

By looking at these results, it is clear that all the implemented ICP variants provide comparable accuracy levels since the errors in the relative attitude and position are always kept below 0.3° and around 1 cm, respectively. This result is confirmed also by the other error parameters defined in sub-chapter 5.1. Specifically, the rms of $\delta\varphi_{EA}$ and $\vartheta_{ERR}$ is kept below 0.08° and 0.07°, respectively, independently of the considered ICP variant. In spite of this similar performance, the NS method can be considered more effective than the NN one for pose tracking since it provides a significant saving in terms of computational cost (of about 53 % and 58 %, without and with the aiding of the prediction algorithm, respectively). This can be explained by observing that the NS method allows performing the ICP matching step much faster than the NN one, in spite of a slight increase in the mean number of required iterations. It is also important to notice that the prediction algorithm provides not only the expected acceleration of the ICP convergence (of about 14 % adopting the NS approach) but also a slight increase in the pose estimation accuracy. This latter effect is a consequence of the significant and relatively fast variability of the pose parameters determined by the chaser motion around the target.

For a better understanding of all the statistics shown in Table 5.11, it is necessary to look at the time variation of both the accuracy and the computational time parameters. In this respect, the four implemented ICP variants are compared in terms of $\varphi_{ERR}$ (see Figure 5.16), $|\underline{T}|_{ERR}$ (see Figure 5.17), $t_{CONV}$ (see Figure 5.18) and $N_{IT}$ (see Figure 5.19).

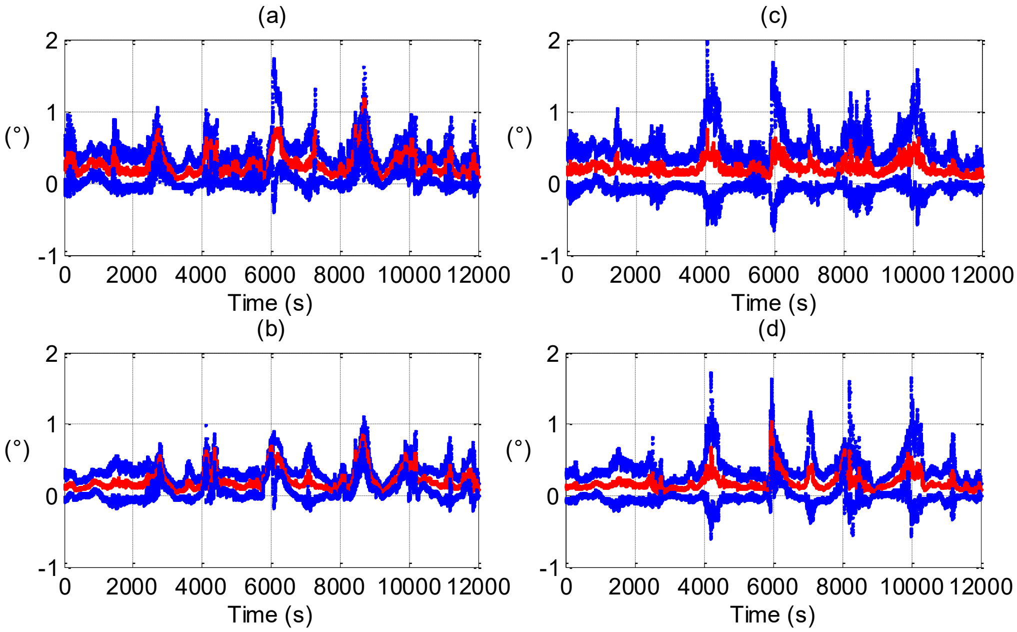


**Figure 5.16 - Time variation of $\varphi_{ERR}$ along the assigned relative trajectory around ENVISAT considering the four implemented ICP variants. (a) NN. (b) NS. (c) NN aided with prediction. (d) NS aided with prediction. The red dotted-line represents the mean value. The blue dotted-lines represent the 3σ bounds. Results are averaged over 100 simulations.**

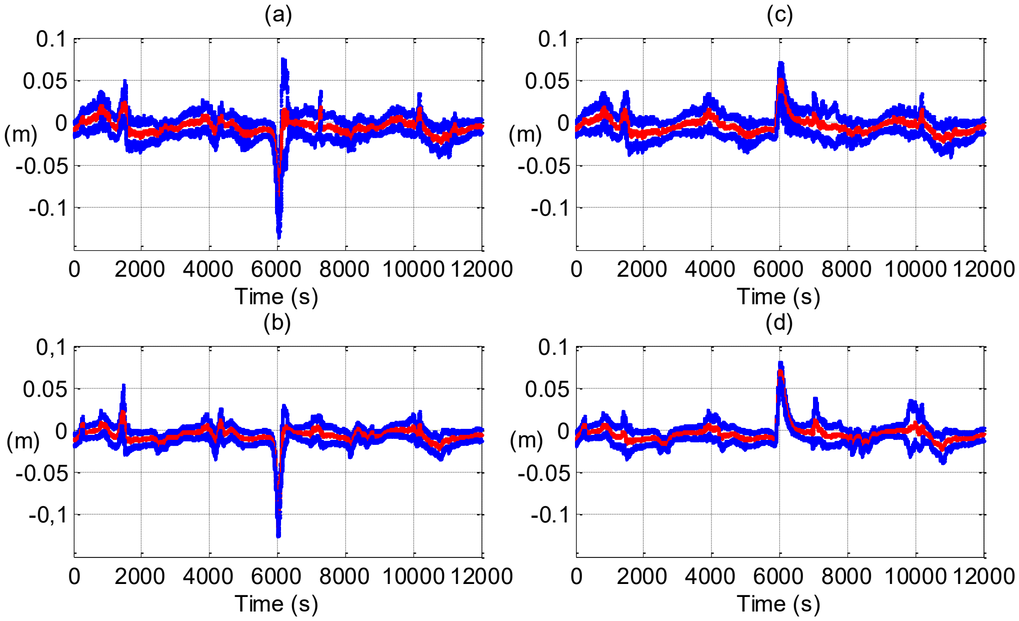


**Figure 5.17 - Time variation of $|\underline{T}|_{ERR}$ along the assigned relative trajectory around ENVISAT considering the four implemented ICP variants. (a) NN. (b) NS. (c) NN aided with prediction. (d) NS aided with prediction. The red dotted-line represents the mean value. The blue dotted-lines represent the 3σ bounds. Results are averaged over 100 simulations.**

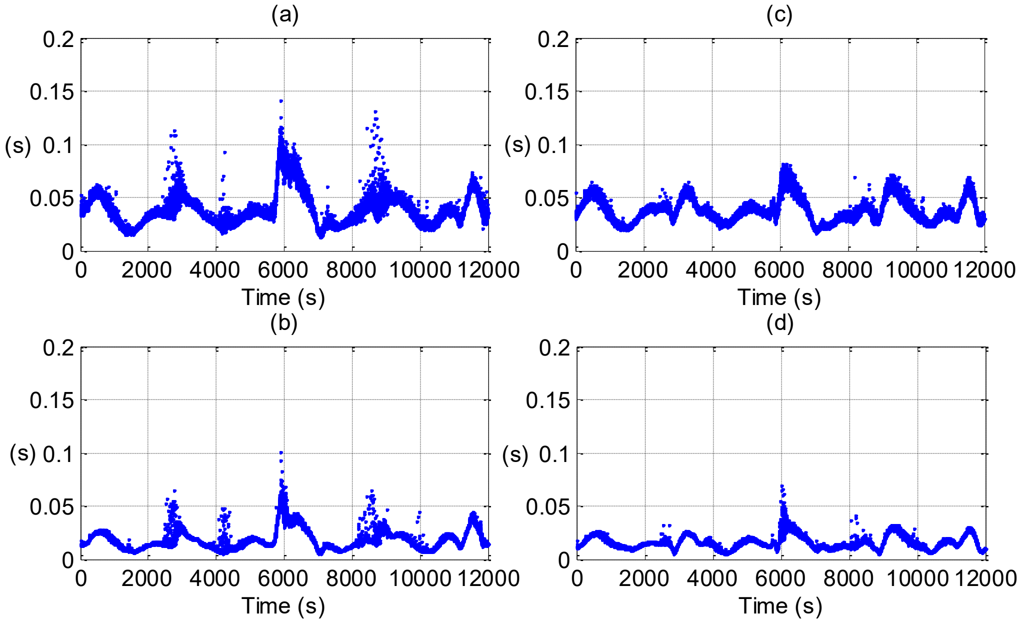


**Figure 5.18 - Time variation of $t_{CONV}$ along the assigned relative trajectory around ENVISAT considering the four implemented ICP variants. (a) NN. (b) NS. (c) NN aided with prediction. (d) NS aided with prediction. Results are averaged over 100 simulations.**

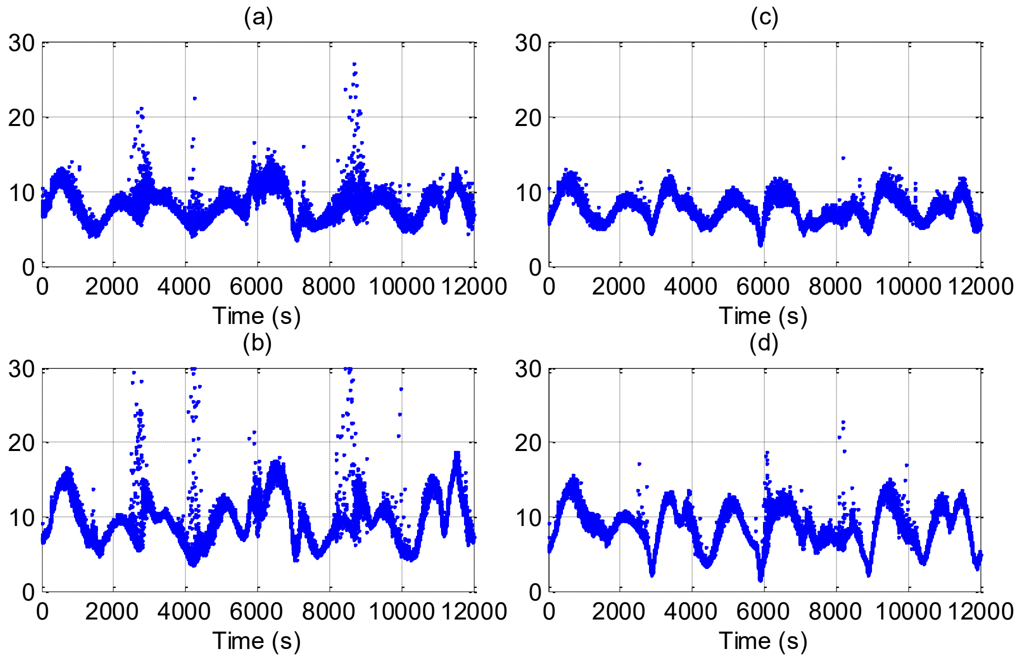


**Figure 5.19 - Time variation of $N_{IT}$ along the assigned relative trajectory around ENVISAT considering the four implemented ICP variants. (a) NN. (b) NS. (c) NN aided with prediction. (d) NS aided with prediction. Results are averaged over 100 simulations.**

Figure 5.16 and Figure 5.17 show the succession of time intervals characterized by completely different behaviors regarding the error in the estimation of the relative pose. Specifically, it is possible to find short periods characterized by a sudden worsening of the accuracy level. In these cases, although all the implemented ICP variants demonstrate capability of avoiding the triggering of instable phenomena, the errors tend to increase (of one order of magnitude with respect to the statistics shown in Table 5.11) both in relative attitude and position up to maximum values of 1.18° and 11 cm, respectively. This phenomenon is also accompanied by a sudden increase in the number of iterations and, consequently, in the computational time as it can be seen in Figure 5.19 and Figure 5.18, respectively. However, also the computational load is kept below a maximum value of 141 ms.

Another interesting aspect which can be noticed by comparing Figure 5.16-a to Figure 5.16-c (or equivalently Figure 5.16-b to Figure 5.16-d), is that the prediction algorithm works as a low-pass filter (indeed, it is by definition a linear Kinematic-only filter) for the time variation of the mean value of $\varphi_{ERR}$. On the other hand, the prediction algorithm also tend to amplify the standard deviation of $\varphi_{ERR}$ in correspondence to the spikes of its mean value. Of course, the same observation can be made by looking at the time variation of $|\underline{T}|_{ERR}$ in Figure 5.17.

Finally, the NS method looks less sensitive to the sensor noise if compared to the NN one. This can be noticed by looking at the time variation of the standard deviation error both in Figure 5.16 and Figure 5.17.

At this point, it is interesting to evaluate the effect of introducing the weighting step in the ICP algorithm, as it is described in sub-chapter 2.5.1. Specifically, this analysis has been done considering the variant of the ICP algorithm which exploits the NN method and the aiding of the prediction algorithm as the pose refinement step. Preliminary numerical simulations, for which the results were averaged over 10 simulations, showed a slight performance worsening due to the fact that the great majority of the sensor/model point pairs exhibited comparable point-to-point distances. Thus, in order to enhance the possible advantage connected to the adoption of a weighting law, ad-hoc simulations are performed considering larger noise levels, i.e. $\sigma_{RANGE}$ is set to 10 cm (instead of 25 mm) and $\sigma_{LOS}$ is set to 0.0028° (instead of 0.0007°). Specifically, the effect of switching on and off the weighting step of the ICP algorithm is analyzed. The related simulation results, averaged over 100 simulations, are in collected in Table 5.12, both in terms of pose estimation accuracy and computational efficiency.

| **ICP weighting step** | **$\varphi_{ERR}$ (°) rms** | **$\|\underline{T}\|_{ERR}$ (cm) rms** | **$t_{CONV}$ (ms) mean** | **$N_{IT}$ mean** |
|---|---|---|---|---|
| **Off** | 0.44 | 1.9 | 56 | 12.1 |
| **On** | 0.42 | 1.9 | 43 | 9.7 |

**Table 5.12 - Effect on ICP performance in terms of pose estimation accuracy and computational efficiency of the weighting step adopting the NN method aided with the prediction algorithm The results are computed over two relative orbits and averaged over 10 simulations.**

Hence, it is possible to state that, when dealing with highly noisy point clouds, the adoption of weights provides faster algorithm convergence (about 22 %) due to a reduced iteration number, while providing also a slight improvement in the estimation of the attitude parameters (about 6 %). However, the most important result is that the ICP algorithm has been proved to be robust against significant increase in the level of LIDAR noise both with and without the introduction of the weighting step.

### 5.3.2 Effect of target visibility

The spikes observed in Figure 5.16 and Figure 5.17, which characterize the time variation of the errors in the estimation of the pose parameters, are clearly related to the variation of the true pose which can lead to particularly unfavorable observation conditions of the target in the sensor FOV. This phenomenon is here analyzed in detail focusing on the results obtained by applying the NS version of the ICP algorithm aided by means of the prediction step.

Figure 5.20 shows the time variation (along the relative trajectory designed around ENVISAT) of $\varphi_{ERR}$ together with the value of the ICP cost function at convergence ($f_{CONV}$), the size of the point cloud detected by the LIDAR ($N_P$), and its distribution around the sensor boresight axis ($D_{BOR}$) as defined in eq. 2.4. Firstly, the comparison between Figure 5.20-a and Figure 5.20-b demonstrates that a direct correlation exist between the accuracy level attained by the ICP algorithm and the value of $f_{CONV}$. Specifically, the higher (lower) $f_{CONV}$ gets, the worse (better) the accuracy level becomes. Secondly, by looking at Figure 5.20-c and Figure 5.20-d, this phenomenon appears to be strictly related to $N_P$ and $D_{BOR}$. In particular, it is clear that ICP performance worsening is caused by unfavorable observation conditions of the target which determine either a significant reduction of $N_P$ or an increased concentration of the measured points around the sensor boresight axis (reduction of $D_{BOR}$) or even the occurrence of both these events. Conversely, observation conditions which give rise to large numbers of measurements coupled with relevant dispersion around the sensor boresight axis (which means better coverage of the target surface), allow the ICP obtaining fine accuracy levels.

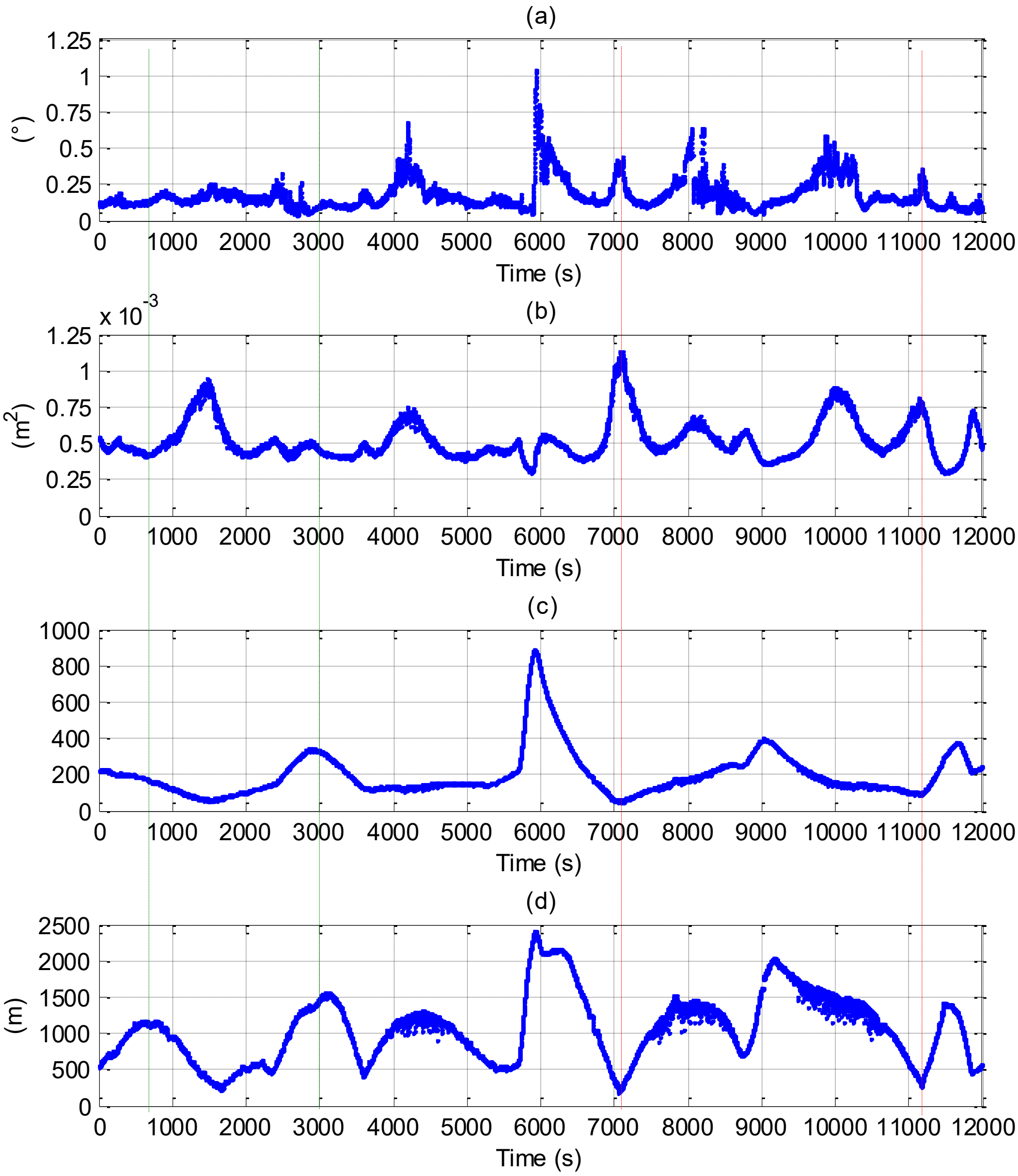


**Figure 5.20 - Effect of the true pose variation on ICP performance adopting the NS method aided with the prediction algorithm. The sub-plots represent the time variation along the two consecutive relative orbits around ENVISAT of $\varphi_{ERR}$ (a), $f_{CONV}$ (b), $N_P$ (c) and $D_{BOR}$ (d). The dashed-red (green) vertical lines identify examples of time instants at which the sets of pose parameters providing unfavorable (favorable) observation conditions.**

Examples of time instants corresponding to favorable (680 s and 3000s) and unfavorable (7100 s and 11100 s) observation conditions are indicated by green and red dashed-lines, respectively, in Figure 5.20. Hence, additional interesting information can be derived by showing the corresponding point clouds. For instance, by looking at the point cloud (composed of 53 points) acquired after 7100 s as depicted in SRF in Figure 5.21, it is clear that a disadvantageous condition ($\varphi_{ERR}$ is 0.39°) occurs if the measured points are mostly concentrated on one single surface of the target. Indeed, this situation can also produce ambiguous pose estimates especially due to the simplified modeling of the target which does not include surface details.

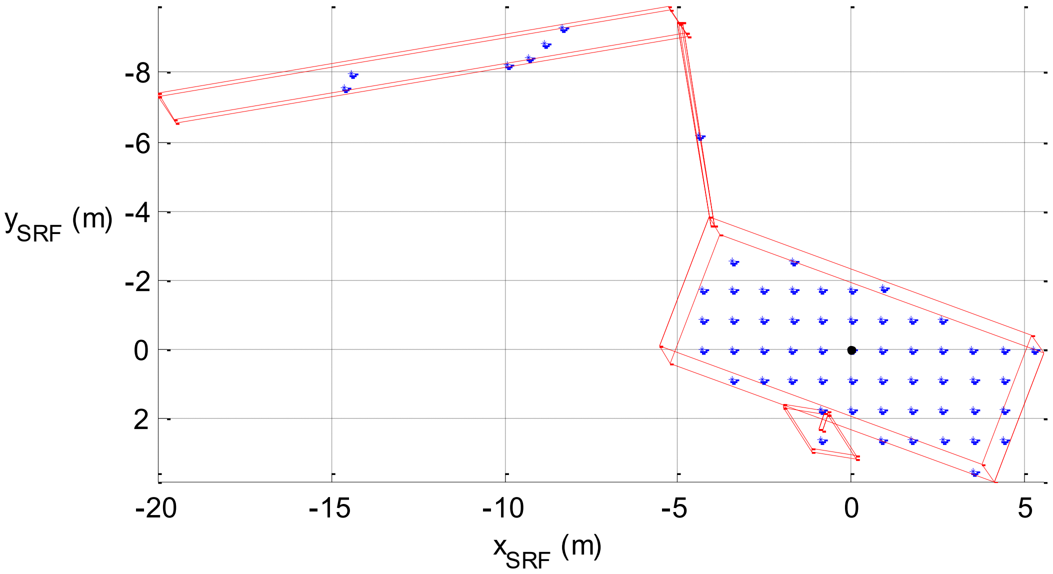


**Figure 5.21 - LIDAR point cloud (blue dots) in SRF after 7100 s.**

On the other hand, after 680 s, the measured point cloud (166 points) covers a large portion of the target and most of its surfaces are in sight. The corresponding point cloud, depicted in Figure 5.22, allows the ICP providing better accuracy ($\varphi_{ERR}$ is 0.14°).

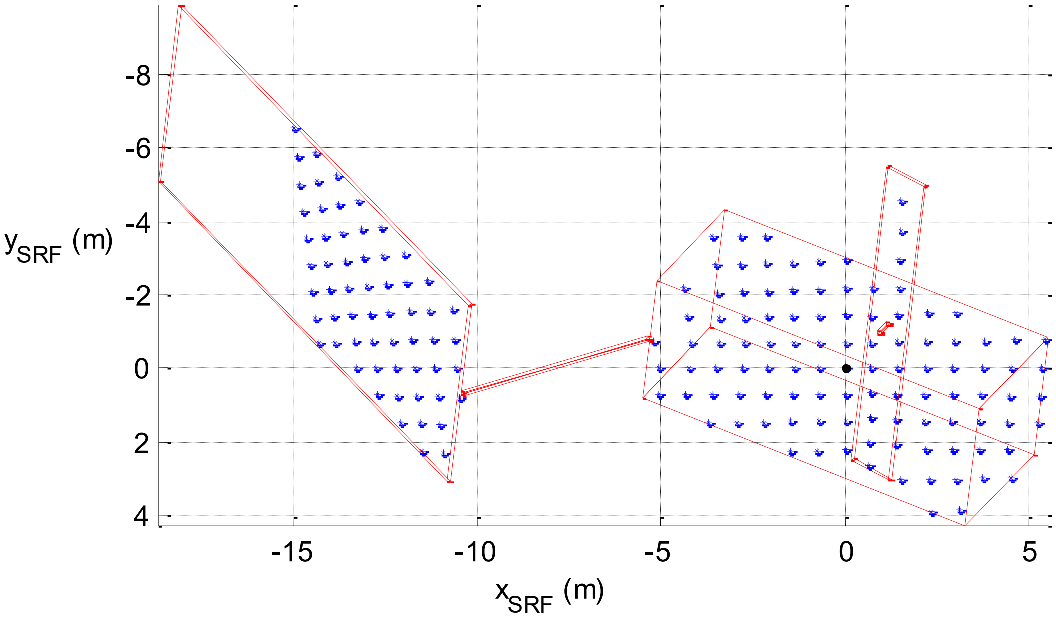


**Figure 5.22 - LIDAR point cloud (blue dots) in SRF after 680 s.**

The discussion presented in sub-chapter 5.2.3 demonstrated that it is possible to improve the level of robustness of the PCA-TM by purposely designing the monitoring trajectory to optimize the observation condition of the target main axis, thanks to the analytical approach introduced in chapter 4. It is now interesting to verify whether the optimal design criterion based on the selection of low values for the out of plane observation angle ($\varphi_{OUT}$), is able to provide advantages also to the performance attained during pose tracking.

To this aim the NS variant of the ICP algorithm aided by means of the prediction step is applied considering ENVISAT and CSM as test cases. Specifically, tracking performance is evaluated considering the relative trajectories around the two targets defined in sub-chapter 5.2.3 and during a time interval of 12000 s (corresponding to two consecutive relative orbits). Results in terms of pose estimation accuracy and computational time are collected in Table 5.13 and Table 5.14 respectively for ENVISAT and CSM. Specifically, the rms of $\varphi_{ERR}$ and $|\underline{T}|_{ERR}$, as well as the mean of $t_{CONV}$ and $N_{IT}$ are computed along the assigned time interval. These quantities are then averaged over 10 simulations (due to the reduced number of simulations the statistical analysis described in sub-chapter 5.1 is not exploited). The same analysis is applied also to the RB. In order to demonstrate consistency of the results against variability of the ICP strategy, the NN variant of the ICP algorithm aided by means of the prediction step is exploited. Simulation results are collected in Table 5.15 using the above-mentioned criteria. It is also worth recalling that for all the targets the trajectories which are compared in terms of tracking performance are designed setting the same minimum range criterion (25 m for ENVISAT, 11 m for CSM, and 7 m for RB).

| **Relative trajectories around ENVISAT** | $\varphi_{OUT}$ (°) | **$\varphi_{ERR}$ (°)<br>rms** | **$\lvert\underline{T}\rvert_{ERR}$ (cm)<br>rms** | **$t_{CONV}$ (ms)<br>mean** | **$N_{IT}$<br>mean** |
|---|---|---|---|---|---|
| **Circular<br>($R$ = 25 m)** | 30 | 0.12 | 0.7 | 8 | 3 |
| **SE$_{\Delta Rmin}$<br>($A_x$ = 21 m, $dxz_{MIN}$ = 13 m)** | 53 | 0.21 | 1.2 | 15 | 7.3 |
| **SE$_{MR}$<br>($A_x$ = 16 m, $dxz_{MIN}$ = 8 m, and $\tilde{z}$ = 25 m)** | 32.6 | 0.18 | 0.9 | 12 | 5.6 |
| **SE$_{MR}$<br>($A_x$ = 24 m, $dxz_{MIN}$ = 22 m, and $\tilde{z}$ = 9 m)** | 69.4 | 0.25 | 1.0 | 166 | 9.2 |

**Table 5.13 - ICP performance in terms of pose estimation accuracy and computational time adopting the NS method aided with the prediction algorithm. The results are computed over the two relative orbits considering four different trajectories around ENVISAT and averaged over 10 simulations.**

| **Relative trajectories around CSM** | $\varphi_{OUT}$ (°) | **$\varphi_{ERR}$ (°)<br>rms** | **$\lvert\underline{T}\rvert_{ERR}$ (cm)<br>rms** | **$t_{CONV}$ (ms)<br>mean** | **$N_{IT}$<br>mean** |
|---|---|---|---|---|---|
| **SE$_{MR}$<br>($A_x$ = 8 m, $dxz_{MIN}$ = 6 m, and $\tilde{z}$ = 14 m)** | 29.7 | 0.30 | 0.6 | 16 | 5.7 |
| **SE$_{MR}$<br>($A_x$ = 10 m, $dxz_{MIN}$ = 8 m, and $\tilde{z}$ = 8 m)** | 51.3 | 0.26 | 1.4 | 15 | 5.8 |

**Table 5.14 - ICP performance in terms of pose estimation accuracy and computational time adopting the NS method aided with the prediction algorithm. The results are computed over the two relative orbits considering two different trajectories around CSM and averaged over 10 simulations.**

| **Relative trajectories around RB** | $\varphi_{OUT}$ (°) | $\varphi_{ERR}$ **(°) rms** | $\lvert\underline{T}\rvert_{ERR}$ **(cm) rms** | $t_{CONV}$ **(ms) mean** | $N_{IT}$ **mean** |
|---|---|---|---|---|---|
| **SE$_{MR}$ ($A_x$ = 5 m, $dxz_{MIN}$ = 4 m, and $\tilde{z}$ = 14 m)** | 19.6 | 0.56 | 4.0 | 6 | 6.0 |
| **SE$_{MR}$ ($A_x$ = 7 m, $dxz_{MIN}$ = 5 m, and $\tilde{z}$ = 7 m)** | 81.9 | 2.96 | 10.2 | 7 | 5.3 |

**Table 5.15 - ICP performance in terms of pose estimation accuracy and computational time adopting the NN method aided with the prediction algorithm. The results are computed over the two relative orbits considering two different trajectories around RB and averaged over 10 simulations**

The results collected from Table 5.13 to Table 5.15 confirm that it is possible to obtain target observation conditions which are favorable also for the implementation of the ICP algorithm, if the relative trajectory is designed by setting $\varphi_{OUT}$ as low as possible. This occurs independently of the target geometry and of the adopted ICP variant. For instance, the simulations performed for ENVISAT (see Table 5.13) show that the circular trajectory, which has $\varphi_{OUT}$ equal to 30°, is characterized by a relative attitude and relative position accuracy of 0.12° and 7 mm, respectively. The result obtained for RB (see Table 5.15) is even more clear. For instance, the value of $\varphi_{ERR}$ goes from about 3° to less than 0.6° by reducing $\varphi_{OUT}$ from 81.9° to 19.6°. On the other hand, in the case of CSM (see Table 5.14), by reducing $\varphi_{OUT}$ from 51.3° to 29.7°, the accuracy improvement is obtained only regarding the rms of $\lvert\underline{T}\rvert_{ERR}$ which reduces from 14 mm to 6 m.

### 5.3.3 ICP performance sensibility to model parameters

This sub-chapter contains the results of the numerical simulations carried out to evaluate how the performance of the two versions of the ICP algorithm obtained by using either the NN or the NS method for the matching step are affected by the variation of the associated settings or modeling parameters. For all these simulations, ENVISAT is selected as test case and pose tracking is performed along the relative trajectory presented in sub-chapter 5.2.1. Moreover, the time statistics of the parameters measuring pose accuracy and computational efficiency are evaluated adopting the statistical analysis presented in sub-chapter 5.1, for which the results are averaged over 10 simulations.

In the case of the NN method, the most important factor which can be modified to determine algorithm performance both in terms of pose estimation accuracy and computational efficiency is the level of discretization of the model point cloud. Ad-hoc simulations are realized by varying the discretization step ($\Delta_P$), i.e. the fixed distance between consecutive model points, which must be assigned for each surface of the target. Specifically, three values of $\Delta_P$ are assigned, i.e. 5 cm, 10 cm and 50 cm thus making the corresponding model point cloud highly dense, intermediate dense and highly sparse, respectively. Simulation results are collected in Table 5.16.

| $\Delta_P$ (cm) | Size of the model point cloud | $\varphi_{ERR}$ (°) rms | $\lvert \underline{T} \rvert_{ERR}$ (cm) rms | $t_{CONV}$ (ms) mean | $N_{IT}$ mean |
|---|---|---|---|---|---|
| **5** | 167426 | 0.28 | 1.3 | 37 | 8.2 |
| **10** | 43020 | 0.42 | 2.2 | 10 | 7.9 |
| **50** | 2046 | 1.17 | 4.5 | 4 | 5.6 |

**Table 5.16 - Statistical analysis of the effect on the ICP algorithm performance of the level of discretization of the model point cloud, in the case of adoption of the NN method. The results are computed over the two relative orbits around ENVISAT and averaged over 10 simulations.**

As expected, the larger the selected $\Delta_P$ is, the more sparse the model point cloud becomes, thus leading to a performance worsening in terms of the errors in both the relative position and attitude parameters, which is mainly caused by the fact that the ICP algorithm is operating on highly sparse point clouds. On the other hand, the computational efficiency tends to improve as $\Delta_P$ is increased. Hence, the selection of

$\Delta_P$ must be done by finding the value which minimizes the computational load provided that specific requirements are satisfied regarding the accuracy level. This is possible since the accuracy level appears to be acceptable, the error in the relative attitude and position are kept below 1.2° and 5 cm, respectively, even if an highly sparse model point cloud is adopted.

In the case of adoption of the NS method, the only tunable parameter, which can be modified to affect the ICP performance, is the distance threshold ($\tau_{NS}$), defined in sub-chapter 2.5, used to verify the consistency to the target geometry of the model points generated by projecting the measurements on the target surfaces. Up to now, all the simulations exploiting the NS method were carried out by setting $\tau_{NS}$ to 10 cm. Hence, ad-hoc numerical simulations are realized to evaluate ICP performance variation considering eight different values of $\tau_{NS}$, from 1 mm to 10 m. Results are collected in Table 5.17.

| **$\tau_{NS}$ (m)** | **$\varphi_{ERR}$ (°)**<br>**rms** | **$\lvert\underline{T}\rvert_{ERR}$ (cm)**<br>**rms** | **$t_{CONV}$ (ms)**<br>**mean** | **$N_{IT}$**<br>**mean** |
|---|---|---|---|---|
| **$10^{-3}$** | 110.51 | 668.8 | 125 | 64.7 |
| **$4\cdot10^{-2}$** | 13.21 | 100.2 | 24 | 12.9 |
| **$5\cdot10^{-2}$** | 3.20 | 22.2 | 19 | 10.4 |
| **$1\cdot10^{-1}$** | 0.24 | 1.2 | 19 | 9.7 |
| **$5\cdot10^{-1}$** | 0.36 | 3.2 | 18 | 9.8 |
| **1** | 0.39 | 3.2 | 19 | 9.9 |
| **10** | 21.53 | 182.0 | 30 | 17.0 |

**Table 5.17 - Statistical analysis of the effect on the ICP algorithm performance of the variation of $\tau_{NS}$ in the case of adoption of the NS method. The results are computed for the two relative orbits around ENVISAT and averaged over 10 simulations. The best configuration is highlighted in red.**

By looking at Table 5.17, it is clear that although the value of $\tau_{NS}$ can be selected within a quite large interval (5 cm, 1 m), the performance of the ICP algorithm tends to become unacceptable outside of it. Moreover, it is possible to state that 10 cm is the best choice both in terms of accuracy level and computational efficiency.

## 5.4 ICP-based autonomous failure detection of pose determination

The correlation, highlighted in Figure 5.20, between the value of the ICP cost function at convergence ($f_{CONV}$) and the accuracy level attained in the estimation of the pose parameters is the key concept on which the autonomous failure detection strategy presented in sub-chapter 2.5 is based. Here, results of numerical simulations described in the previous sub-chapters are used to demonstrate how it is possible to select the threshold ($f_{LIM}$) for the value of $f_{CONV}$ in order to unambiguously identify the success or the failure of the proposed algorithms for uncooperative pose determination. All the simulation results regarding the tracking phase of the pose determination process have never shown the triggering of instability phenomena, leading to a divergence of the error in the estimated relative attitude and position parameters (and consequently of the values reached by $f_{CONV}$). Hence, the attention is focused on the simulation results regarding the application of the algorithms developed for pose acquisition.

Firstly, the analysis of the success rate of the on-line TM over the relative trajectory around ENVISAT, presented in sub-chapter 5.2.1, is considered. By setting $\Delta$ to 30°, the values obtained for the $SR_{TM}$ and the computational time are 71.8 % and 7 s, respectively. Figure 5.23 shows the values of $f_{CONV}$ corresponding to the application of the ICP soon after the on-line TM pose solution is acquired.

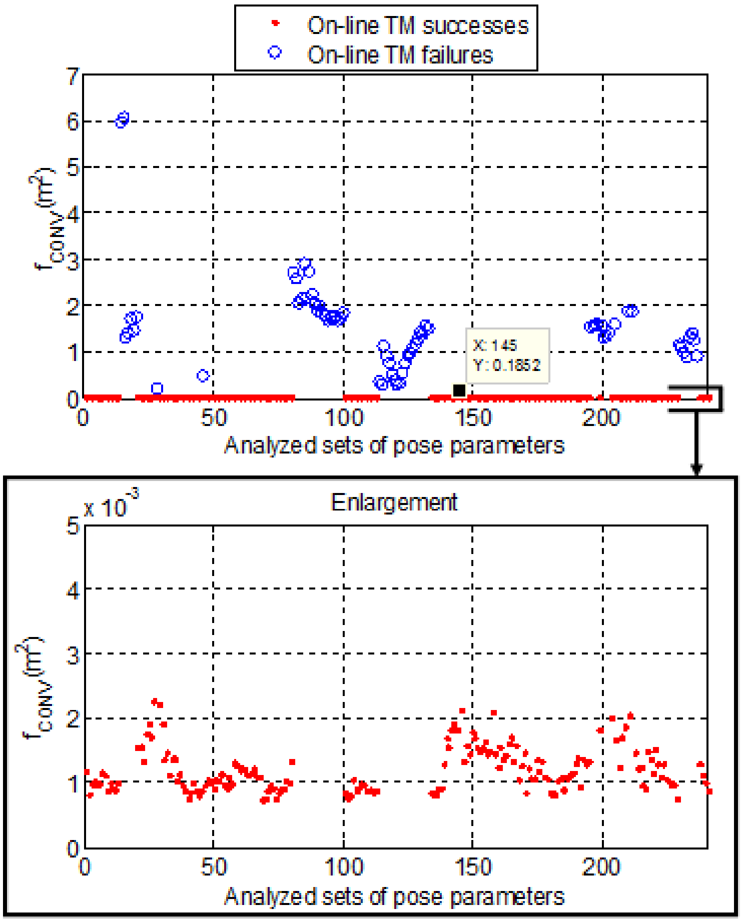


**Figure 5.23 - (Up) Values of $f_{CONV}$ corresponding to the 241 sets of pose parameters considered along the two consecutive orbits around ENVISAT. (Down) Enlargement of the same graph. The red dots and blue circles correspond to the successes and failures, respectively, of the on-line TM. The maximum value of $f_{CONV}$ which corresponds to algorithm failure is highlighted.**

It is clear that a difference of at least two orders of magnitude exists between the values of $f_{CONV}$ corresponding to the algorithm's success and failure. Specifically, in the former case $f_{CONV}$ is of the order of few mm$^2$, while in the latter case it varies from more than 10 cm$^2$ up to a few m$^2$.

At this point, it is necessary to extend this analysis considering a wide interval of relative distances and the entire 3-DOF space of the relative attitude parameters. To this aim, the effect of the selection of $f_{LIM}$ on the conditioned probabilities of declaring success or failure of pose acquisition is analyzed by referring to the PCA-TM simulation results. In the following, $P_{SS}$ ($P_{FF}$) is the probability of declaring success (failure) given success (failure), while $P_{SF}$ ($P_{FS}$) is the probability of declaring success (failure) given failure (success). Figure 5.24 shows the behavior of these conditioned probabilities as a function of $f_{LIM}$. Specifically, it is obtained from the results of the simulations done for ENVISAT, setting $\Delta$ to 30° and $R$ to 20 m.

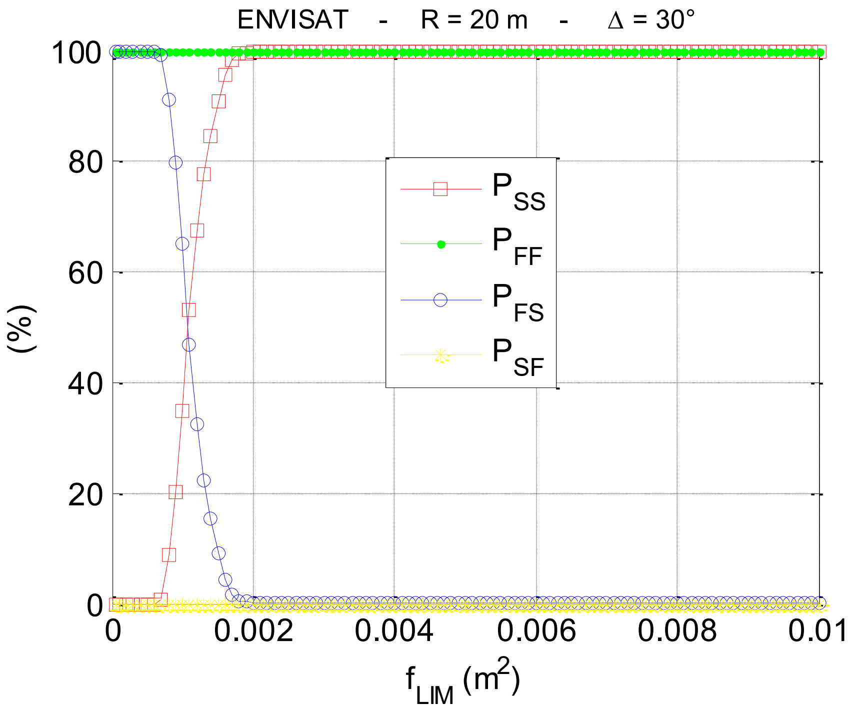


**Figure 5.24 - Variation of the conditioned probability of declaring success or failure of the PCA-TM algorithm as a function of $f_{LIM}$.**

It is clear that a very low value of $f_{LIM}$ (in the order of few cm$^2$) could be selected as an adequate and safe threshold to unambiguously identify successes and failures of pose acquisition. In fact, for any value of $f_{LIM}$ above 2 mm$^2$, the $P_{SS}$ and the $P_{FF}$ are close to 100 %, while simultaneously the $P_{SF}$ and the $P_{FS}$ are close to 0 %. For instance, having set $f_{LIM}$ to 5 cm$^2$, it is possible to determine the absolute probability of correctly declaring success and failure ($P_{OK}$). This quantity is computed from the results of the numerical simulations done for ENVISAT and CSM considering different values of the relative range and setting $\Delta$ to 30°. Results are collected in Table 5.18, including, for any value of $P_{OK}$, the corresponding value of $SR_{PCA\text{-}TM}$.

| ***R*** **(m)** | **ENVISAT** | | **CSM** | |
|---|---|---|---|---|
| | **SR$_{PCA\text{-}TM}$ (%)** | ***$P_{OK}$*** **(%)** | **SR$_{PCA\text{-}TM}$ (%)** | ***$P_{OK}$*** **(%)** |
| **10** | / | / | 84.4 | 97.0 |
| **20** | 95.7 | 99.6 | 70.4 | 78.0 |
| **30** | 89.5 | 99.4 | 54.7 | 62.4 |
| **40** | 68.2 | 99.8 | 46.9 | 57.7 |
| **50** | 73.9 | 99.1 | / | / |
| **60** | 79.4 | 98.4 | / | / |
| **70** | 70.6 | 98.6 | / | / |
| **80** | 46.2 | 94.8 | / | / |

**Table 5.18 - Summary of the values of $P_{OK}$ obtained setting $f_{LIM}$ to 5 cm$^2$. The corresponding values achieved for the SR$_{PCA\text{-}TM}$ are also included.**

These simulations demonstrate that the proposed approach works extremely well when operating on large and asymmetric targets, e.g. ENVISAT. In fact, the absolute probability of correctly declaring success and failure is higher than 94 % even at a relative range of 80 m where the SR$_{PCA\text{-}TM}$ is extremely low. On the other hand, in the case of CSM, the reliability of the autonomous failure detection strategy is assured only at very close ranges, when partial views allow to solve the ambiguity caused by the symmetry of the target, or if the LIDAR resolution is high enough to obtain more information from the measured point clouds.

Finally, by looking at the time variation of $f_{CONV}$ in Figure 5.20-b, it is worth outlining that the above-mentioned value of $f_{LIM}$ could be successfully exploited also when applying the autonomous failure detection step during pose tracking.

# Chapter 6 - LIDAR-based localization and mapping of UAVs

SLAM is the real-time process by which a mobile ground/aerial robot, moving within an unknown environment, computes its own trajectory (localization) while simultaneously building a map of that environment (mapping). A fundamental feature of any SLAM algorithm is that this map must be globally consistent, meaning that the robot must be able to recognize previously visited locations (i.e. loop closure detection) and to integrate the subsequent constraints within the existing map [131]. Although the SLAM problem can be considered solved from the theoretical point of view [18], and several improvements have been carried out in the last two decades [19], there are still many open issues especially regarding its real-time implementation on board MAVs.

The SLAM problem can be dealt with by exploiting EO sensors, which can be active and passive, or other ranging systems, like RADAR. As regards this latter technological solution, the main related issue is that RADAR are still too heavy and power consuming, and may require significant data processing efforts, to be installed on board any kind of MAV. However, it is worth mentioning that great amount of research efforts are currently in progress toward the integration of compact radar systems on board MAVs [132]. Hence, by focusing on EO sensors, a distinction can be made between visual SLAM algorithms, which rely on monocular or stereovision passive cameras, and LIDAR ones which use 2D or 3D laser scanners.

Visual SLAM algorithms can be classified into two different categories, namely the loosely-coupled and tightly-coupled approaches, depending on the way visual and inertial data are integrated. In the former case, the visual data are processed, separately and (typically) at different frequency with respect to the inertial ones, by a "pose estimator" block in order to compute the position and attitude parameters of the considered platform, which are then sent within the navigation filter [133]. On the other hand, the tightly-coupled approach involves the adoption of image processing techniques to identify robust features from the acquired visual data. Each of these features provides two additional equations, defining their motion within the image plane, to the navigation filter (for instance an EKF) [134].

In this work LIDAR-based localization and mapping algorithms are addressed. In fact, LIDAR are characterized by low sensibility to variable ambient light conditions, thus providing a higher degree of autonomy with respect to passive sensors.

This chapter is relative to the 6-months activity carried out at Cranfield University (UK) in the frame of Program STAR - Linea 2 - financially supported by UniNA and Compagnia di San Paolo. It is organized as follows.

Sub-chapter 6.1 is a short review of most recent solutions developed for localization and mapping using laser scanner measurements. Specific attention is addressed to the issue of implementing SLAM on board MAV and some recent solutions are listed.

Sub-chapter 6.2 describes in detail the innovative techniques developed for localization and mapping to be performed on board of a MAV. They are based on the integration of laser scan data provided by a two-dimensional (2D) LIDAR, and inertial measurements, i.e. vehicle's acceleration, angular velocity and attitude given by a low cost IMU.

Sub-chapter 6.3 describes the experimental setup designed to acquire real-data over which algorithm performance are assessed, as well as the related test area.

Finally, sub-chapter 6.4 contains the results obtained by running off-line the developed algorithms over the acquired data.

## 6.1 Literature review

LIDAR-based techniques for the localization step of a SLAM algorithm can be generally classified into probabilistic and scan matching approaches [135].

Probabilistic techniques (which have been most commonly adopted over the years especially for indoor applications) deal with the problem of matching detected landmarks between consecutive datasets by using the concept of maximum likelihood. An example of probabilistic method for local SLAM with detection and tracking of moving objects from a moving vehicle in dynamic outdoor environments can be found in [136]. On the other hand, matching methods are typically conceived to estimate the vehicle trajectory incrementally, i.e. by comparing subsequently acquired datasets. However, in some cases the measured dataset relative to a certain time instant is directly matched to the updating map, as this procedure can reduce the risk of ambiguities, which is particularly relevant to vehicles moving at high speed in outdoor environment. Scan matching approaches can be further divided into three different categories, namely feature-to-feature, point-to-feature and point-to-point techniques.

The selection of the localization technique affects also the choice of the approach adopted for the mapping step. For instance, the main advantage of using features for localization is that they can also be used to build the map, thus reducing the related amount of information to store. Also, the search for the correspondences is faster, although the contribution of the feature extraction algorithm must be considered, for an overall comparison of the computational time performance. To be precise, three different methods basically are applicable to perform mapping [135]. In addition to the raw data representation, which is extremely simple to implement but can lead to a huge amount of data to be stored, the remaining approaches (mostly adopted in the literature) are the occupancy grid methods and the feature-based ones. Occupancy grids techniques aim at organizing the environment where the vehicle is moving as a grid, which can be 2D or 3D. Each time a new dataset is available the probability that each grid location is occupied by an obstacle is computed. On the other hand, feature-based methods represent the environment as a set of predefined features which need to be extracted from the acquired datasets. Occupancy grid solutions have two main advantages. Firstly, they are conceptually much more tailored to deal with unstructured environments, in which reliable features (corners, lines or planes) are difficult to detect. Secondly, they can provide dense as well as sparse representation of the observed environment, depending on its complexity, by properly setting the resolution of the grid. At the same time, the main drawback is that the independence assumption between cells during updates may lead to inconsistent maps. On the other hand, feature-based algorithms for mapping lead to compact representations of the observed environment, with a high speed of execution and little memory consumption. Also, this representation is less easily prone to generate inconsistencies, as it implicitly considers correlation between cells. The main drawbacks are the possible loss of information due to the sparse (discontinuous) representation of the environment, as well as the difficulties intrinsic to the feature extraction and the data association processes.

Some interesting LIDAR-based solutions recently proposed for localization and mapping are discussed in the following.

One scan matching method developed to overcome the computational load issue of point-to-point correspondence determination is the Polar Scan Matching (PSM) algorithm. It basically associates points having the same bearing in two successive scans, taking into account the polar nature of the local reference frame of many 2D laser scanners [137]. Algorithm's performance is evaluated considering both simulated data and real scans acquired by a Sick LMS 200 laser scanner installed on a ground robot. With regards to the real scans, firstly, static acquisitions are taken at four different locations for each element of a list of indoor environments for which the ground truth is available. Secondly, the PSM is applied to all the possible pairs of scans relative to the environment, thus computing the variation of orientation and the displacement. Results show that this technique works better in cluttered indoor environment, where the scan matching error is of cm order in position and sub-degree in attitude, while it has problems in corridor environments where the error tends to diverge. Additional tests are realized by implementing the PSM technique within a Kalman filter-based SLAM algorithm, which demonstrate capability to obtain global accuracy level which is better than the one provided by the odometry of the robot used for the tests [138].

Another scan matching method, which is based on the HT, namely the Hough Scan Matching (HSM), is proposed in [139]. This technique allows comparing dense scans without the need of extracting features. Specifically, it is based on the definition of a spectrum function which allows to translate the matching problem to the Hough domain whose invariance properties allows to compute separately the orientation and translation pose parameters. Algorithm's performance is evaluated performing simulations on real scan data of different environments taken from the Internet, and adopting models of four different range sensors (i.e. SICK LMS, SICK PLS, Hokuyo PB9-01, and the edge-extracting pipeline of an omnidirectional camera). The algorithm performs slightly better in an office-like environment than in an unstructured one. Specifically, in the former case, sub-degree and cm level accuracies are respectively obtained in heading and translation, while in the latter case, the heading error is of a few degrees and the translation error is around 10 cm.

One scan matching method based on cross correlation of two LIDAR scans can be found in [140]. The algorithm looks for the rigid-body transformation which maximize a correlation function in a probabilistic way. It is worth outlining that unlike point-to-point or point-to-feature matching techniques (e.g. the Iterative Closest Point, ICP, or the Iterative Closest Line, ICL) which perform a local search, the correlative scan matching aims at finding the global maximum of the correlation function over a large space of plausible transformations which must be provided a priori (e.g. by means of wheel/visual odometry). The performance of different kinds of correlative approaches proposed in [140] is evaluated on simulated data obtained from a map generated from real LIDAR data. Also, the performance of these correlative techniques is compared to modified implementation of the ICP and ICL algorithms, both in terms of pose estimation accuracy and computational load. The correlative approach is more computational time demanding than ICP and ICL and provides comparable accuracy level in case of small initialization error. However, it is proven to be more robust toward larger errors in the required initialization.

A LIDAR based approach to perform SLAM in 3D outdoor environments can be found in [141]. It performs localization by means of a modified ICP algorithm (point-to-point scan matching approach), which is designed to compensate the orientation errors due to erroneous point correspondence determinations. The peculiar aspect of this work is that 3D feature points are extracted from the raw data and identified by purposely defined descriptors in order to compute faster and more robust point-to-point correspondences. The mapping step of this SLAM approach relies on a raw data representation. Experimental tests are carried out installing a 2D laser scanner (LMS291, SICK, Germany) on board an unmanned ground robot. The laser scanner is mounted on a DC motor which provides a tilt angle measured by an encoder in order to obtain 3D scans. The absolute roll and pitch angles of the robot are given by an IMU, while the yaw angle and motion increments are get by means of a wheel encoder and the IMU. The results of the experimental tests show that the algorithm is able to provide a faster convergence with respect to the basic ICP implementation, and also to reduce the vertical drift in the estimated trajectory.

A flexible and scalable 2D SLAM approach usable for a multitude of scenarios involving unmanned ground and marine vehicles is proposed in [142]. The mapping step is based on an occupancy grid map [143] concept, while localization is performed by means of a scan matching approach which is based on optimization of the alignment of beam endpoints with the map learnt so far. The main advantage of this approach is that it does not require the necessity to search for point correspondences. The full state vector is estimated by a 3D EKF which exchanges information with the 2D SLAM algorithm (although they run at different frequencies). Specifically, the EKF provides the pose information (once projected on the xy plane) necessary to initialize the scan matcher, while the pose computed by SLAM is used as a measurement update for the filter. The reliability of this approach is tested in various scenario by installing an Hokuyo UTM-30LX 2D laser scanner on board unmanned ground/marine vehicles. No information about the quantitative accuracy of the approach is provided.

The works analyzed up to now show several techniques capable of providing successful localization and mapping capabilities to autonomous ground/marine vehicles moving within large scale outdoor and indoor environments. However, when trying to apply these algorithms to enable MAVs with SLAM capabilities, more complex technical challenges arise with respect to ground/marine vehicles. Specifically, three main issues must be taken into account. Firstly, the flying robot's propeller can generate a limited thrust to remain airborne and this significantly restrains the amount of payload available for sensing and computation. Secondly, MAVs have a very complex and fast dynamics thus having to rely on complex filtering techniques to get reliable state estimation. Finally, MAVs can hover at different heights and so they move along 3D trajectories. A more detailed description of these technical challenges can be found in [20].

An example of design of a quadrotor helicopter capable of performing fully autonomous exploration within an indoor unknown environment is given by [20]. Specifically, an Hokuyo UTM-30LX 2D laser scanner is installed on the AscTec Hummingbird from Ascending Technologies GmBH to perform localization by applying the correlative scan matching approach [140]. The relative motion parameters computed from scan matching are then sent to a standard EKF which provides the entire state vector solution. Experimental tests show that after 1 min flight, the average distance error was less than 1.5 cm, while the average velocity error was 0.02 m/s, with a standard deviation of 0.025 m/s. The proposed localization algorithm is also integrated to the GMapping technique [144] to perform 2D SLAM. This SLAM solution is proven to be reliable for autonomous exploration of large indoor environment. The only problem is the computational load since the algorithm runs on-line but it takes 1 s or 2 s to process incoming laser scans. This latter aspect could cause too large delays to be able to safely control the MAV.

Droeschel at al. [145] propose a local multi-resolution occupancy grid map method to perform SLAM on board a MAV by exploiting 3D LIDAR scans. Specifically, they use an Hokuyo UTM-30LX-EW 2D laser scanner which is rotated around the axis perpendicular to its scan plane by means of a Dynamixel MX-28 servo actuator in order to produce 3D scans. It is worth outlining that visual odometry from a stereocamera system is used in order to compensate vehicles motion (during the sensor rotation) for the 3D scan assembly. The main characteristic of the proposed approach is that the occupancy grid map has higher resolution closer to the vehicle, and conversely lower one by moving farther from it. This solution allows to reduce the computational effort of the SLAM algorithm while simultaneously keeping the same accuracy level. A scan matching technique based on a probabilistic approach (maximizing occupancy likelihood) which registers the acquired scans to the map built so far in order to estimate the MAV's motion is also presented. The performance of this approach is compared to the classical ICP and the Generalized ICP (GICP) algorithms by means of experimental tests in which the ground-truth pose information are given by an indoor motion capture (MoCap) system. Results demonstrates that the algorithm estimates the trajectory more accurately with faster runtime than the compared techniques. An hexarotor MAV equipped with redundant sensing payload (two rotating Hokuyo UST-20LX laser scanners, three stereo camera pairs, and one IMU) is presented in [146]. This hardware architecture aims at providing omnidirectional environment perception as well as 6-DOF pose estimation capabilities. The 2D laser scanner are rotated to produce 3D scans and the same SLAM algorithms as in [145] are tested to the data acquired by means of this platform.

Another technique using LIDAR scans for localization and mapping in real time is presented in [147]. The adopted sensor is again an Hokuyo UTM-30LX EW laser scanner mounted on a motor capable of providing rotations around its boresight axis (which are measured by an encoder) in order to produce 3D scans. The LIDAR odometry algorithm looks for two different types of 3D feature points (i.e. edge and planar points) in two consecutive 3D scans, and it finds the feature correspondences by adopting purposely defined metrics. Finally, it estimates the rigid transformations which best aligns the two sets of corresponding feature points by exploiting the Levenberg-Marquand algorithm [148]. The mapping algorithm is similar to the odometry ones, in the way feature points are extracted and the pose parameters are updated, but it runs at lower frequency (i.e. 1 Hz instead of 10 Hz). Experimental tests show that these techniques are able to estimate the trajectory of autonomous robot with an accuracy of about 1 % and 3 % of the travelled path, respectively indoor and outdoor. Other tests are aimed at comparing the accuracy levels of three approaches, i.e. LIDAR odometry alone, IMU alone, LIDAR odometry and IMU. Results show that the most accurate trajectory estimations in terms of percentage of the travelled path are obtained with the third approach. A more recent paper [149] presents an integration of visual and LIDAR odometry for localization. Specifically, visual odometry runs at 60 Hz, LIDAR odometry is used for refinement of the localization solution and it runs at 10 Hz, and finally the mapping algorithm runs at 1 Hz. This approach is aimed at minimizing the trajectory drift and providing more robust maps. Experimental tests show that this approach provides an error on the estimated trajectory which is kept below 1% of the travelled path. Of course the major drawback of this method is that it can only tolerate temporary light outrage, but it is not suitable in the case of continuous darkness.

An ICP-based SLAM algorithm is provided by [150]. It is conceived to be used on board of a quadrotor MAV purposely designed to fly in unknown indoor environments for inspection purposes. The main original approach of the proposed version of the ICP algorithm relies in a modified approach to the ICP matching step aiming at improving the computational efficiency. The mapping step is performed on the basis of a raw data representation.

In spite of the promising results achieved by the techniques mentioned in this short review, the problem of performing in real time LIDAR SLAM on board a MAV is still an open issue. In this context, the objective of the research activity is the development of algorithms capable of estimating the vehicle's trajectory with high accuracy as well as of simultaneously providing highly accurate maps by exploiting limiting processing and data storage resources. Moreover, the level of robustness against variability of the environment and of the lighting conditions must be improved.

## 6.2 Localization and mapping based on LIDAR-inertial integration

The pose of the MAV is described in the following by a set of 6 parameters representing the position and the attitude of its body reference frame (BRF) with respect to a local inertial reference frame, i.e. the East-North-Up (ENU). Specifically, $\underline{T}_A$ is the 3D position vector of the MAV with respect to the ENU and expressed in the ENU, while a 321 sequence of Euler angles, i.e. heading ($\gamma_A$), pitch ($\beta_A$), and roll ($\alpha_A$), or equivalently the unit quaternion ($q_A$), is used to represent the attitude of the BRF with respect to the ENU. For the sake of clarity, two assumptions are made. Firstly, the MAV's BRF is considered coincident and aligned with the LIDAR reference frame (LRF), which has the x-axis in the boresight direction, the z-axis perpendicular to the scan plane (pointed upward), and the y-axis oriented to obtain a right-handed reference frame. Secondly, the ENU origin is assumed to be coincident with the initial position of the MAV. However, the proposed approach can be easily extended to a more general case.

### 6.2.1 LIDAR/Inertial Odometry algorithm

Localization is carried out by a hybrid LIDAR/Inertial Odometry algorithm (L/I-O). Specifically, it is one point-to-point ICP-based scan matching algorithm. The pose of the considered flying vehicle is tracked by integrating attitude information from an IMU with position information obtained by recursively registering two consecutive scans provided by a 2D LIDAR. Of course, the initial pose of the MAV must be fully known at the start of the trajectory, in order to be compliant with the concept of odometry [131].

Generally speaking, 2D scan matching is the problem of registering two sets of 2D data, i.e. a reference scan and a current scan, by looking, in the pose search space, for the optimal rotation and translation, i.e. the ones that provide the best alignment by minimizing a purposely defined error function [139]. Typically, the reference scan represents the environment in which the vehicle moves and it can be a pre-built map or a previous scan, while the current scan is the measurement dataset provided by the available range sensor at the time of interest.

The L/I-O algorithm extracts the attitude directly from an IMU at high frequency (about 90 Hz), while the position is estimated at lower frequency (up to 36 Hz) using a LIDAR odometry technique based on a point-to-point scan matching algorithm. A block diagram to describe this system architecture is provided in Figure 6.1.

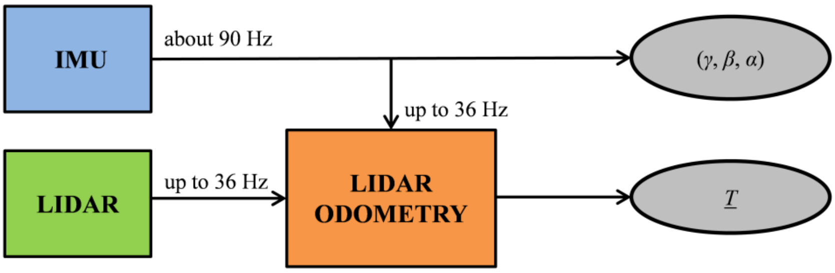


**Figure 6.1 -Block diagram describing the system architecture for localization.**

The scan matching algorithm is described in detail as follows. Let $t_k$ and $t_{k+1}$ be two successive time instants at which two LIDAR scans are acquired and let $\underline{\underline{P}}^{LRF}(t_k)$ and $\underline{\underline{P}}^{LRF}(t_{k+1})$ be the corresponding point clouds in LRF. Firstly, a coordinate transformation is applied to convert these point clouds from LRF to the Vehicle Reference Frame (VRF), which is a new frame aligned to ENU with the origin at the current position of the MAV (and so coincident to the LRF origin). This is done as shown in eq. (6.1),

$$\begin{aligned} \underline{\underline{P}}^{VRF}(t_k) &= \underline{\underline{R}}_{VRFtoLRF}(\gamma_{A,k}, \beta_{A,k}, \alpha_{A,k})^T \, \underline{\underline{P}}^{LRF}(t_k) \\ \underline{\underline{P}}^{VRF}(t_{k+1}) &= \underline{\underline{R}}_{VRFtoLRF}(\gamma_{A,k+1}, \beta_{A,k+1}, \alpha_{A,k+1})^T \, \underline{\underline{P}}^{LRF}(t_{k+1}) \end{aligned} \tag{6.1}$$

where $\underline{\underline{R}}_{VRFtoLRF}$ is the rotation matrix representing the attitude of LRF with respect to VRF. Secondly, a customized version of the ICP algorithm is applied to find the best estimate of the rotation and translation necessary to align $\underline{\underline{P}}^{VRF}(t_{k+1})$, i.e. the current scan, to $\underline{\underline{P}}^{VRF}(t_k)$, i.e. the reference scan. Specifically, the algorithm provides in output an estimate of the variation of the Euler angles ($\Delta\alpha_A$, $\Delta\beta_A$, $\Delta\gamma_A$) and of the position vector ($\underline{\Delta T}_A$) between the two VRFs, occurred during the time interval from $t_k$ to $t_{k+1}$.

This ICP algorithm is similar to the one described in sub-chapter 2.5. Specifically, it is characterized by a sequence of steps, i.e. initialization, matching, outliers rejection, selection and minimization of an error metric function, which are iteratively repeated until a convergence criterion is met. Each time a new scan is available, the ICP algorithm is initialized by setting to zero all the previously defined parameters, i.e. $\Delta\alpha_A$, $\Delta\beta_A$, $\Delta\gamma_A$ and $\underline{\Delta T}_A$. With regards to the matching step, which mainly determines the algorithm's computational load and the accuracy level, the NN approach is adopted, meaning that each point in the current scan is associated to the closest one in the reference scan according to the Euclidean metric. In addition, the reference scan is pre-processed to build a K-D tree [84], in order to accelerate the NN search. The outliers rejection step is introduced to compensate for wrong point associations that may arise if there is a poor overlap between the scenes observed in two LIDAR acquisitions, meaning that a large number of measurements in the actual scan may not have real correspondences in the reference scan. Indeed, this phenomenon can lead to significant error in the ICP pose solution. Hence, if $\underline{d}$ is the set of distances between correspondent points, the reference/actual scan matches characterized by a relative distance which is outside the interval defined by the mean and mode values of $\underline{d}$ are considered as outliers. As regards the error metric function (*f*), it is selected as the mean squared distance of corresponding points between the two scans, as shown in eq. (6.2).

$$f(q_A) = \frac{1}{N(t_{k+1})} \sum_{i=1}^{N(t_{k+1})} \left| \underline{P}_i^{VRF}(t_k) - \underline{\underline{R}}_{(VRF(t_k)toVRF(t_{k+1}))}(q_A)^T (\underline{P}_i^{VRF}(t_{k+1}) + \underline{\Delta T}_A) \right|^2 \qquad (6.2)$$

Also in this case, the ICP cost function is minimized by using the closed form solution proposed by Horn [85]. In (2), $\underline{P}_i^{VRF}(t_{k+1})$ and $\underline{P}_i^{VRF}(t_k)$ are respectively the i^th point of the actual scan and the corresponding one in the reference scan, $N(t_{k+1})$ is the number of points in the current scan, $\underline{\underline{R}}_{(VRF(tk)toVRF(tk+1))}$ is the rotation matrix representing the attitude variation of the VRF between the time instants of the two scans. Once $q_A$ is estimated, the corresponding values of $\Delta\alpha_A$, $\Delta\beta_A$, and $\Delta\gamma_A$ can be extracted, and, finally, $\underline{\Delta T}_A$ is computed as the difference between the centroid of the two scans, as given by eq. (6.3).

$$\underline{\Delta T}_A = \underline{\underline{R}}_{(VRF(t_k)toVRF(t_{k+1}))}(q_A) Mean\{\underline{\underline{P}}^{VRF}(t_k)\} - Mean\{\underline{\underline{P}}^{VRF}(t_{k+1})\} \qquad (6.3)$$

At this point, the value of *f* can be updated and the procedure is repeated until the variation between two successive iterations goes below a threshold (e.g. $10^{-6}$ m$^2$). Once the criterion to stop iteration is met, the output of eq. (6.3) can be used to update the MAV's position vector using eq. (6.4).

$$\underline{T}_{A,t_{k+1}} = \underline{T}_{A,t_k} + \Delta\underline{T}_A \tag{6.4}$$

The logic behind the proposed L/I-O algorithm is summarized by the flow diagram shown in Figure 6.2.

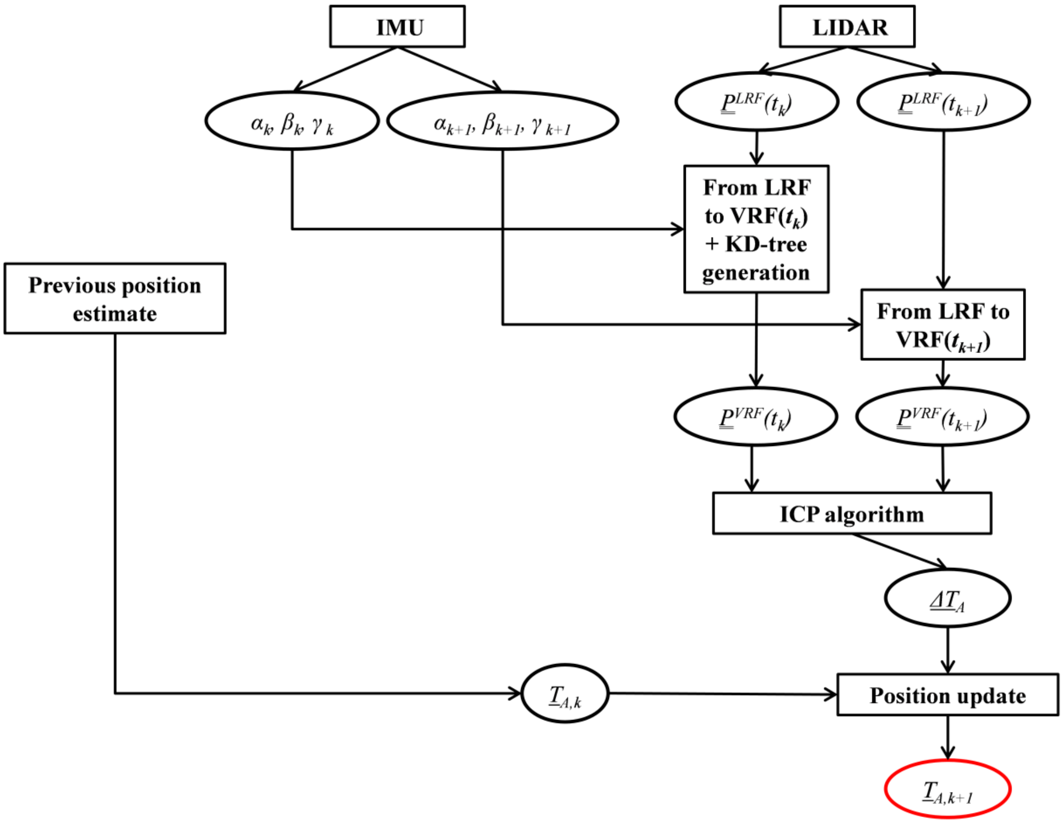

IMU
LIDAR
$\alpha_k, \beta_k, \gamma_k$
$\alpha_{k+1}, \beta_{k+1}, \gamma_{k+1}$
$\underline{P}^{LRF}(t_k)$
$\underline{P}^{LRF}(t_{k+1})$
From LRF to VRF($t_k$) + KD-tree generation
Previous position estimate
From LRF to VRF($t_{k+1}$)
$\underline{P}^{VRF}(t_k)$
$\underline{P}^{VRF}(t_{k+1})$
ICP algorithm
$\Delta\underline{T}_A$
$\underline{T}_{A,k}$
Position update
$\underline{T}_{A,k+1}$

**Figure 6.2 -Flow diagram describing the proposed L/I-O algorithm for localization. The red circle contains the final output.**

Also, the level of robustness of the solution provided by the presented ICP algorithm can be augmented by means of the autonomous failure detection strategy defined in sub-chapter 2.5. If the value of the cost function at convergence ($f_{CONV}$) is larger than one pre-defined threshold ($f_{LIM}$), the MAV's position is updated by integrating the inertial acceleration provided by the IMU, according to eq. (6.5)

$$\underline{T}_{A,t_{k+1}} = \underline{T}_{A,t_k} + \underline{v}(t_{k+1} - t_k) + \frac{1}{2}\underline{a}_{t_k}(t_{k+1} - t_k)^2\} - Mean\{\underline{\underline{P}}^{VRF}(t_{k+1})\} \quad (6.5)$$

where $\underline{v}$ and $\underline{a}$ represent velocity and acceleration of the MAV, respectively. These latter quantities are converted in ENU thanks to the attitude estimates.

### 6.2.2 Line-based mapping

The feature-based approach is preferred to the occupancy grid method since it leads to compact representations of the environment with a high speed of execution and little memory requirement, thus being more compliant with MAVs' limitation both in terms of weight and power consumption. Specifically, the attention is focused on line-based mapping algorithms, as this activity is preliminary interested to SLAM applications in structured 2D indoor environments.

Before entering the details of the proposed approach, it is necessary to review the most important methods available in the literature for line or edge detection, which is the core function within line-based algorithms for mapping. A complete review of the main existing algorithms for edge detection from 2D range data can be found in [151]. Specifically, the most used techniques in the literature are compared in terms of speed complexity, correctness and precision by means of experimental tests on real scan data collected in a large scale office environment. A line extraction algorithm can be divided in three different steps respectively aiming at finding the number of lines in the scan, identifying which points correspond to which line, converting those points into the representation chosen storage (line fitting). While this latter goal is typically performed by exploiting least-squares methods [152-154], the most common solutions for the former two problems are listed hereunder.

- ➢ The split and merge algorithm, used for example in [152] and [154], recursively searches for lines starting from the entire scan. If a line is not extracted, the scan is split by two and the search is repeated. This method is quite common due to its easy implementation.

- The line regression algorithm, used for example in [153], transforms the line extraction problem into a search problem in the line parameter space by adopting a concept similar to the Hough transform. This method is quite complex to implement.
- The incremental algorithm tries to incrementally build lines by moving along the scan with a fixed point step.
- The RANSAC algorithm [33] has the advantage to be more versatile since it can be used to detect different kind of features once the feature-model is provided. The Hough transform method has only been quite popular [155]. However, since it applies to 2D images, it requires a pre-processing step, namely the rasterization process, to transform the 2D scan into an image, thus incrementing the overall computational load.
- The Expectation-Maximization (EM) algorithm is a probabilistic method whose main problem is the fact that it needs a proper selection of the initialization in order not to be staked by local minima.

For any of these methods, a clusterization algorithm is proposed in [151] aimed at reducing the computational load. Clusters which contain a number of points below a threshold are deleted. The main result of the performance comparison between these algorithms is that the Split-and-merge and the Incremental approach are the most suitable for SLAM since they are fast enough to ensure real-time implementation, they tend to provide a very low number of false positive, and the accuracy level is comparable (although lower) to the one provided by more accurate but much more computationally demanding probabilistic approaches (e.g. RANSAC or EM).

The proposed line-based mapping technique which processes the 3D data provided by the 2D laser scanner, has three basic steps. Firstly, the updated position solution is used to translate the measured point cloud from VRF to ENU. Secondly, the dataset must be processed to detect lines which are then stored in the updating map. While different approaches to data clusterization and line identification exist, line fitting is typically performed by means of least-squares methods. However, they can be computationally slow especially when dealing large number of points in the acquired dataset. Hence, the main purpose of the proposed line detection method is to find a solution for line-fitting which is as accurate and reliable as the least-squares methods, while simultaneously having better computational efficiency. The algorithm's operation can be divided into three steps.

The first step is the clusterization process. Unlike traditional approaches, it is characterized by two hierarchical levels. Firstly, the scan is subdivided into separate clusters by looking for those locations, indicated as break-points, at which the inter-point distance (i.e. the distance between consecutive points in the scan) is larger than a specific threshold (*DTh*). This part of the algorithm is similar to the clusterization process proposed in [151], where the radial distance between consecutive points is compared to a threshold. However, the proposed method foresees a second level of clusterization, meaning that additional break-points are identified within each cluster, by exploiting the polar structure of the point cloud acquired by the 2D laser scanner. Basically, both the $x$ and $y$ coordinates of the measured points can be considered function of the scan angle $\theta_{LID}$. Hence, the sub-cluster search aims at looking for those locations $i$ at which either eq. (6.6) or eq. (6.7) is satisfied. These two conditions, where $N$ is the number of points in the cluster, compare the local value of the $x$ and $y$ derivatives with respect to $\theta_{LID}$, to the sum of their mean and standard deviation computed over the cluster. At the end of this process, all the clusters with less than 10 points are neglected.

$$\left\{ \frac{dx(\theta_i)}{d\theta_{LID}} > \left[ \frac{1}{N} \sum_{i=1}^{N} \frac{dx(\theta_{LIDi})}{d\theta_{LID}} + \frac{1}{N-1} \sum_{i=1}^{N} \left( \frac{dx(\theta_{LIDi})}{d\theta_{LID}} - \frac{1}{N} \sum_{i=1}^{N} \frac{dx(\theta_{LIDi})}{d\theta_{LID}} \right)^2 \right] \right\} \quad (6.6)$$

$$\left\{ \frac{dy(\theta_i)}{d\theta_{LID}} > \left[ \frac{1}{N} \sum_{i=1}^{N} \frac{dy(\theta_{LIDi})}{d\theta_{LID}} + \frac{1}{N-1} \sum_{i=1}^{N} \left( \frac{dy(\theta_{LIDi})}{d\theta_{LID}} - \frac{1}{N} \sum_{i=1}^{N} \frac{dy(\theta_{LIDi})}{d\theta_{LID}} \right)^2 \right] \right\} \quad (6.7$$

An example of application of this hierarchical clusterization process to real data acquired by a 2D laser scanner is shown in Figure 6.3 (*DTh* is set to 0.1 m).

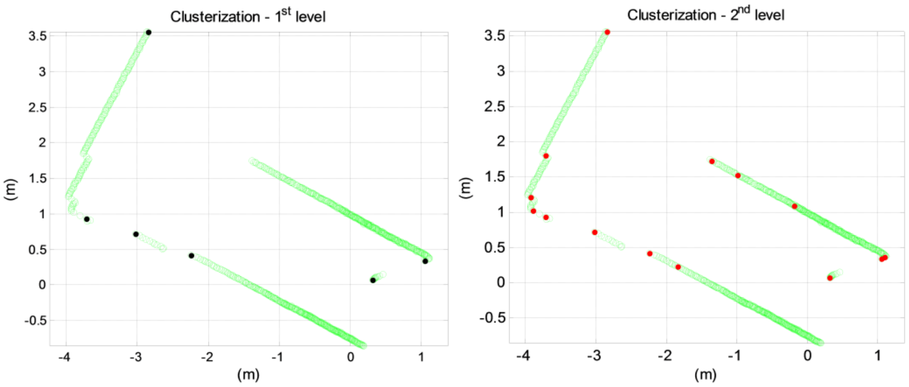


**Figure 6.3 -Clusterization step of the line-based mapping algorithm. Black and red dots identify the first element of each cluster, respectively after the 1st and the 2nd level of the process.**

After clusterization, the identification step can start. Unlike the techniques listed in [151], this approach relies on the PCA. Specifically, the PCA is applied by assigning to each cluster the ratio between the maximum and minimum eigenvalues (*r*) associated with its covariance matrix. Hence, if *r* is larger than a fixed threshold (*ETh*), a cluster is considered as line feature and its direction is given by the eigenvector corresponding to the maximum eigenvalue (whose components are $\lambda_x$, $\lambda_y$ and $\lambda_z$). Specifically, this line passes through the centroid of the cluster and it is oriented as the eigenvector corresponding to the maximum eigenvalue of the covariance matrix. Figure 6.4 is obtained by applying the PCA-based algorithm for line fitting (*ETh* is set to 500) to the clusters identified in Figure 6.3.

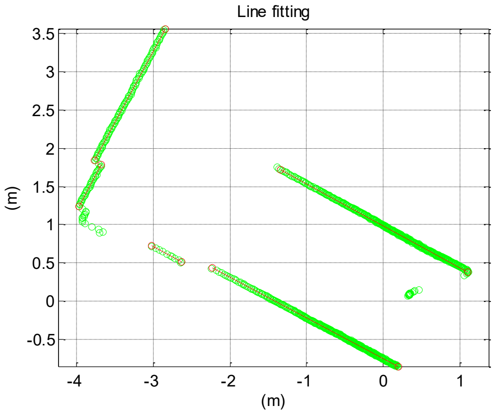


**Figure 6.4 - PCA-based line fitting.**

Finally, the line detection algorithm foresees a storage step aimed at assigning to each detected line a list of 8 parameters (4 of which are independent) to be stored in memory. Firstly, $\alpha_L$ is the angle associated to the line direction, according to eq. (6.8).

$$\alpha_L = \tan^{-1}\left(\frac{\lambda_y}{\lambda_x}\right) \tag{6.8}$$

Secondly, $x_C$ and $y_C$ are the coordinates of the centroid of the cluster, while $x_{E1}$ and $y_{E1}$, as well as $x_{E2}$ and $y_{E2}$, are the coordinates of the two ends of the line segments. They are found by projecting the first and the last elements of the cluster on the edge direction according to eq. (6.9),

$$\begin{aligned}\begin{bmatrix} x_{E1} \\ y_{E1} \end{bmatrix} &= \begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix} \rho_{E1} + \begin{bmatrix} x_C \\ y_C \end{bmatrix} \\ \begin{bmatrix} x_{E2} \\ y_{E2} \end{bmatrix} &= \begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix} \rho_{E2} + \begin{bmatrix} x_C \\ y_C \end{bmatrix}\end{aligned} \tag{6.9}$$

where $\rho_{E1}$ and $\rho_{E2}$, i.e. the distances of the two ends from the line segment centroid, can be computed using eq. (6.10).

$$\begin{aligned}\rho_{E1} &= \sqrt{(x_{E1} - x_C)^2 + (y_{E1} - y_C)^2}\left(\frac{\begin{bmatrix} x_{E1} \\ y_{E1} \end{bmatrix} - \begin{bmatrix} x_C \\ y_C \end{bmatrix}}{\sqrt{(x_{E1} - x_C)^2 + (y_{E1} - y_C)^2}} \cdot \begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix}\right) \\ \rho_{E2} &= \sqrt{(x_{E2} - x_C)^2 + (y_{E2} - y_C)^2}\left(\frac{\begin{bmatrix} x_{E2} \\ y_{E2} \end{bmatrix} - \begin{bmatrix} x_C \\ y_C \end{bmatrix}}{\sqrt{(x_{E2} - x_C)^2 + (y_{E2} - y_C)^2}} \cdot \begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix}\right)\end{aligned} \tag{6.10}$$

Finally, $d_L$ is the distance of the line segment from the origin of the reference frame in which it is represented, and it can be computed using eq. (6.11).

$$d_L = \frac{-y_C + \tan(\alpha_L)x_C}{\sqrt{1 + \tan^2(\alpha_L)}} \tag{6.11}$$

Since the sub-cluster search could generate break-points within a real edge (due to the LIDAR measurement noise), an intermediate merging step is implemented. It allows merging two consecutive clusters if the corresponding lines satisfy the condition defined by eq. (12),

$$\{(\alpha_{L1}-\alpha_{L2})<\alpha_T\}\cap\left\{\begin{array}{l}[((\alpha_{L1}-\alpha_{CC})<\alpha_T)\cap((\alpha_{L2}-\alpha_{CC})<\alpha_T)] \\ \cup \\ [((\alpha_{L1}-(180-\alpha_{CC}))<\alpha_T)\cap((\alpha_{L2}-(180-\alpha_{CC}))<\alpha_T)]\end{array}\right\} \tag{6.12}$$

where $\alpha_{CC}$ is the orientation of the direction of the segment which links the two centroids, and $\alpha_T$ is a very small angular threshold (e.g. 0.05°). This means that the line detection algorithm is applied to the aggregate cluster.

In order to prove the advantages of this PCA-based line fitting with respect to classical Least-Squares (LS in the following) method, ad-hoc numerical simulations are conceived and realized. Firstly, a set of $n$ points is defined in the 2D space. This points are randomly distributed along a direction identified by a fixed angular coefficient ($m_l$), and their coordinates ($x^i$ and $y^i$) are given by eq. (6.13),

$$\begin{array}{l} x^i = x_S i \\ y^i = m_l x^i + \nu_\sigma^{\ i} \end{array}, i = 1...n \tag{6.13}$$

where $x_S$ is the fixed step between points along the x-axis, and $\nu_\sigma^{\ i}$ is the i<sup>th</sup> extraction from a normal distribution with zero mean and standard deviation equal to $\sigma$. Secondly, both the PCA and LS methods are applied obtaining the corresponding lines, each one identified by an angular coefficient ($m_{PCA}$ and $m_{LS}$), and a constant term ($n_{PCA}$ and $n_{LS}$). Finally, the line fitting accuracy is evaluated as the mean squared distance of the assigned points from the estimated line ($Err_{PCA}$ and $Err_{LS}$), according to eq. (6.14).

$$\begin{array}{l} Err_{PCA} = \frac{1}{n}\sum_{i=1}^{n}(y^i - m_{PCA}x^i - n_{PCA})^2 \\ Err_{LS} = \frac{1}{n}\sum_{i=1}^{n}(y^i - m_{LS}x^i - n_{LS})^2 \end{array} \tag{6.14}$$

Results are averaged over 10000 Monte-Carlo simulations and are shown in Table 6.1, in terms of line fitting accuracy and computational load. With regards to the simulation inputs, $m_l$ is set to 5, $x_S$ is set to 4 cm, and $\nu_\sigma$ is equal to 5 cm.

| ***n*** | **PCA time saving (%)** | **Err$_{PCA}$ (m$^2$) rms** | **Err$_{LS}$ (m$^2$) rms** | **Err$_{PCA}$ - Err$_{LS}$ (%)** |
|---|---|---|---|---|
| **25** | 83 | $2.4 \cdot 10^{-3}$ | $2.4 \cdot 10^{-3}$ | $1.2 \cdot 10^{-1}$ |
| **50** | 83 | $2.4 \cdot 10^{-3}$ | $2.4 \cdot 10^{-3}$ | $2.9 \cdot 10^{-2}$ |
| **100** | 83 | $2.5 \cdot 10^{-3}$ | $2.5 \cdot 10^{-3}$ | $7.1 \cdot 10^{-3}$ |
| **250** | 83 | $2.5 \cdot 10^{-3}$ | $2.5 \cdot 10^{-3}$ | $1.1 \cdot 10^{-3}$ |
| **500** | 82 | $2.5 \cdot 10^{-3}$ | $2.5 \cdot 10^{-3}$ | $3 \cdot 10^{-4}$ |
| **1000** | 80 | $2.5 \cdot 10^{-3}$ | $2.5 \cdot 10^{-3}$ | $1 \cdot 10^{-4}$ |
| **10000** | 57 | $2.5 \cdot 10^{-3}$ | $2.5 \cdot 10^{-3}$ | $<10^{-4}$ |

**Table 6.1 - Comparison between PCA and LS methods for line fitting.**

This analysis proves that the PCA-based line fitting is almost twice faster than the LS approach while being able to provide the same level of accuracy.

Once the line detection phase is completed, the storage step of the mapping method must be carried out. Specifically, each candidate has to be compared to every line in the map in terms of two parameters, i.e. $\alpha_L$ and $d_L$. If a correspondence is not found, the candidate line becomes a new element of the map. On the other hand, if the candidate represents a visualization of the same feature of the environment from a different position, the two lines must be merged. This is done by projecting the two ends of the candidate line on the direction of the corresponding one in the map, and by comparing these projections to the pre-existing ends. The updated ends are the ones that determine the maximum length of the updated line in the map.

## 6.3 Experimental setup and test scenario

In order to assess the performance of the algorithms developed for localization and mapping, an experimental setup and a test area are prepared for data recording and testing.

The setup, shown in Figure 6.5 is composed of the items listed hereunder.

- One 2D LIDAR, i.e. the UTM-30LX-EW produced by Hokuyo.
- One autopilot, i.e. the Pixhawk produced by 3DRobotics.
- One embedded board, i.e. the Nitrogen6X, produced by Boundary-Devices.
- One battery and two voltage regulators.

The latter components are needed since the battery has to power both the LIDAR (at 12 V) and the Nitrogen board (5 V). The selection of the LIDAR is mainly due to its limited size and weight with respect to similar sensors. Thus, it can be easily integrated in the architecture of a small MAV. The specifications of the UTM-30LX-EW can be found in [156], while a complete characterization can be found in [157].

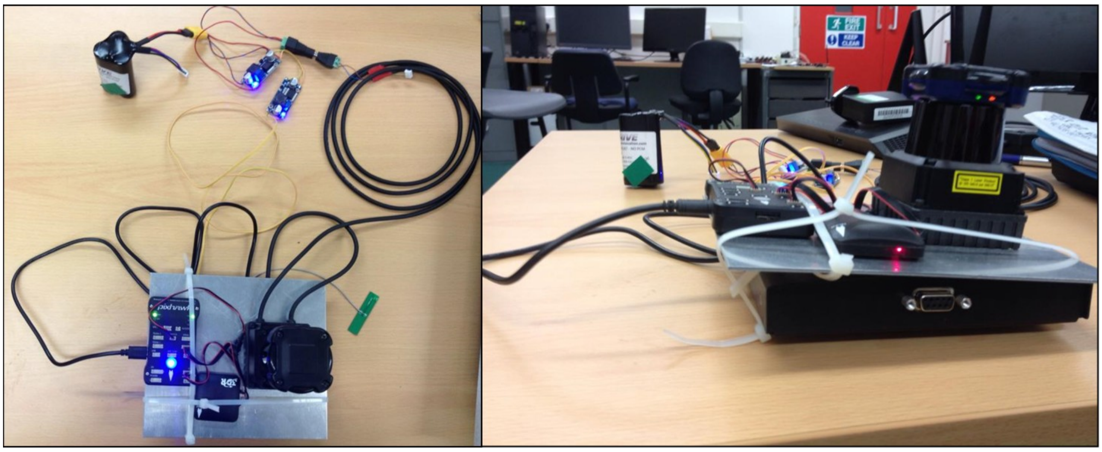


**Figure 6.5 - Assembled experimental setup for LIDAR/Inertial data recording. (Left) Top view. (Right) Front view.**

The Nitrogen board is used to register data from both the Pixhawk (using USB connection) and the LIDAR (using Ethernet connection) by exploiting the corresponding nodes of the Robot Operating System (ROS) [158], namely the *mavros* and the *urg_node,* respectively. In this way the IMU data from the Pixhawk and the range data from the LIDAR can be simultaneously recorded, together with their timestamps, within the same bag-file. This makes it possible to run the proposed algorithms offline in MATLAB environment by directly reading from the generated bag-files. Since IMU data are collected at higher update rate than the LIDAR data, the attitude parameters corresponding to the LIDAR timestamps are obtained through linear interpolation. The test area, imaged in Figure 6.6, is a 2D maze in which the experimental setup is carried by hand. Since the ground truth is not available, the localization error cannot be determined point by point along the estimated trajectory. Hence, once the indoor scenario is selected, the localization accuracy level is evaluated by comparing the length of the estimated travelled path with the length of a reference trajectory. This latter quantity (9 m) is computed by taking measurements from a single point Laser Range Finder (BOSCH DLR130 Distance Measurer).

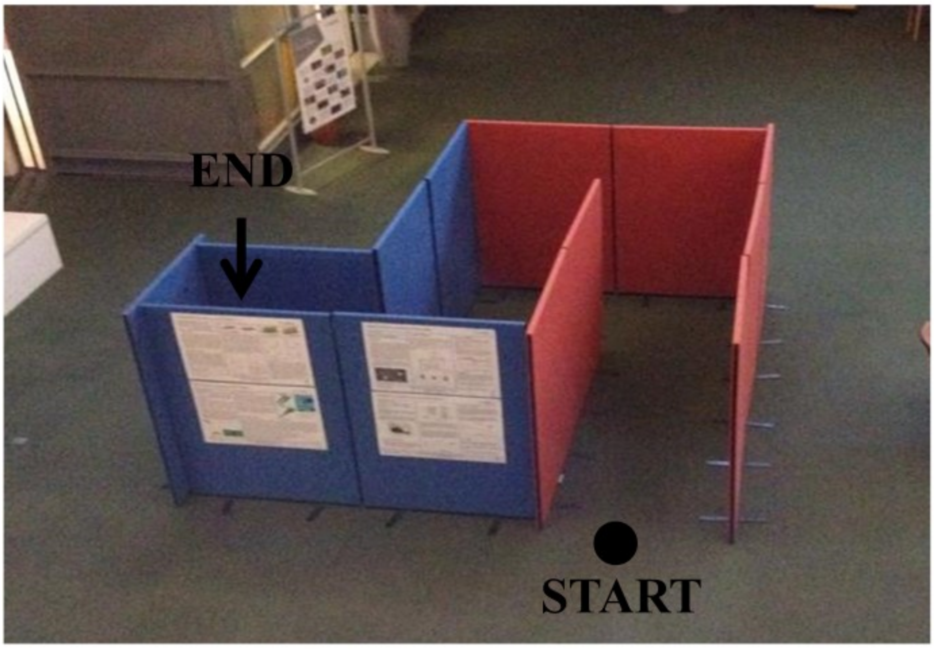


**Figure 6.6 - Test area on the right, with indication of the start and ending points of the travelled path.**

## 6.4 Localization and mapping results

Firstly, Several runs of the L/I-O algorithm are realized to evaluate the effect on performance of its tuning parameters, which are defined hereinafter. The Range Limit (RL) is the value of distance over which LIDAR measurements are disregarded from the acquired scan. The Angular Resolution (AR) is the angle between two consecutive LIDAR measurements which are not deleted from the acquired scan (minimum value for AR is 0.25°). The Odometry rate (OR) is the frequency at which the L/I-O algorithm is applied (maximum value for OR is 36 Hz which is the LIDAR measurement rate). On the other hand, other parameters of the algorithm are kept constant and are listed hereinafter. The ICP maximum number of iteration is set to 30. The minimum value for the time derivative of the ICP cost function at convergence is set to $10^{-6}$ m$^2$. The value of $f_{LIM}$ is set to 0.5 m$^2$.

Firstly, the effect of RL is evaluated by considering different values (from 60 m to 3 m) while keeping the AR (0.25°) and the OR (6 Hz) fixed. Results in Table 6.2 show that the RL should always be set below 30 m. This is not highlighted by the error on the estimated length of the overall trajectory ($L_{EST}$), which is almost the same for any value of RL, but by looking at the sum of $f_{CONV}$ during the test ($f_{SUM}$) and at its mean ($f_{MEAN}$). Indeed, these parameters represent an index of how well two consecutive scans are aligned by the algorithm, meaning that the lower their value is, the larger the accuracy of localization becomes. When RL is 60 m, $f_{SUM}$ and $f_{MEAN}$ reach the values of 8.8 m$^2$ and 0.149 m$^2$, respectively, which are one order of magnitude larger than for the other runs. This degradation in performance is explained by the fact that LIDAR measurements longer than 30 m are not reliable and can cause wrong point-to-point matching by the ICP thus compromising its operation. Indeed, when the laser intensity reflected back at the detector is below an internal threshold of the sensor (this happens, for instance, when the laser shot passes through a window), a value of range around 60 m is the output. Below 30 m, a reduction of RL gives advantages in terms of computational load (less number of points to be matched by the ICP routine) and localization accuracy. However, if RL is too low (3 m), it may produce an increase in the error in $L_{EST}$, which shows that it is convenient to neglect part of the scan to reduce the computing time, provided that it does not cause excessive loss of information. It is finally worth mentioning that the attained very low values of $f_{SUM}$ and $f_{MEAN}$ do not disprove the previous statements since they are caused by the significant reduction in the amount of points analyzed in the scan.

| RL (m) | $f_{CONV} > f_{LIM}$ (%) | $f_{SUM}$ (m$^2$) | $f_{MEAN}$ (m$^2$) | Mean comp. time (s) | $L_{EST}$ (m) | Error on $L_{EST}$ (%) |
|---|---|---|---|---|---|---|
| **60** | 10 | 8.799 | 0.149 | 0.134 | 8.16 | 9 |
| **30** | 7 | 1.202 | 0.020 | 0.136 | 8.20 | 9 |
| **15** | 5 | 0.973 | 0.017 | 0.149 | 8.30 | 8 |
| **11** | 5 | 0.984 | 0.017 | 0.130 | 8.44 | 6 |
| **7** | 2 | 0.282 | 0.005 | 0.160 | 8.50 | 6 |
| **3** | 0 | 0.171 | 0.003 | 0.159 | 8.02 | 11 |

**Table 6.2 - Performance analysis of the LIDAR/Inertial-Odometry algorithm. Effect of RL (AR is 0.25° and OR is 6 Hz).**

Secondly, the effect of AR is analyzed considering three values (0.25°, 0.5° e 1°) while keeping fixed the RL (11 m) and the OR (6 Hz). Results in Table 6.3 show that it is convenient to change AR from 0.25° to 0.5° since it causes a faster ICP solution (about 30 %) while ensuring the same accuracy level. It is also possible to state that a further reduction of the resolution (AR set to 1°) is not advisable, not only because it generates a performance worsening but also because it compromises the applicability of the line extraction algorithm for mapping (the analyzed scans become too sparse to obtain robust line features).

| **AR (°)** | $f_{CONV} > f_{LIM}$ **(%)** | $f_{SUM}$ **(m$^2$)** | $f_{MEAN}$ **(m$^2$)** | **Mean comp. time (s)** | $L_{EST}$ **(m)** | **Error on** $L_{EST}$ **(%)** |
|---|---|---|---|---|---|---|
| **0.25** | 5 | 0.984 | 0.017 | 0.130 | 8.44 | 6 |
| **0.5** | 3 | 0.947 | 0.016 | 0.094 | 8.48 | 6 |
| **1** | 7 | 2.603 | 0.044 | 0.063 | 8.40 | 7 |

**Table 6.3 - Performance analysis of the LIDAR/Inertial-Odometry algorithm. Effect of AR (RL is 11 m and OR is 6 Hz).**

Thirdly, the effect of OR is analyzed considering four values (36 Hz, 18 Hz, 6 Hz, 3 Hz) while keeping fixed the RL (11 m) and the AR (0.5°). Results in Table 6.4 show that low values of OR (3 Hz and 6 Hz) provide better performance than by applying the localization algorithm at larger frequencies (18 Hz and 36 Hz). This is because the lower rate of execution reduces the propagation of the error, which is bonded to the concept of odometry. Since pose variation is computed by comparing two successive datasets without considering the history of the trajectory, there is no way to correct any mistake committed during the application of the algorithm.

| **OR (Hz)** | $f_{CONV} > f_{LIM}$ **(%)** | $f_{SUM}$ **(m$^2$)** | $f_{MEAN}$ **(m$^2$)** | **Mean comp. time (s)** | $L_{EST}$ **(m)** | **Error on** $L_{EST}$ **(%)** |
|---|---|---|---|---|---|---|
| **36** | 1 | 2.751 | 0.008 | 0.059 | 7.21 | 20 |
| **18** | 1 | 1.682 | 0.010 | 0.077 | 7.93 | 12 |
| **6** | 3 | 0.947 | 0.016 | 0.094 | 8.48 | 6 |
| **3** | 13 | 1.357 | 0.047 | 0.118 | 8.49 | 6 |

**Table 6.4 - Performance analysis of the LIDAR/Inertial-Odometry algorithm. Effect of OR (RL is 11 m and AR is 0.5°).**

This same principle can be used to understand why the accuracy of the algorithm proposed for localization is also affected by the velocity at which the experimental system is moved along the same path. Table 6.5 contains the results obtained for the L/I-O algorithm with the same tuning parameters (RL set to 11 m, AR set to 0.5° and OR set to 6 Hz) applied to two different datasets, respectively, which were recorded by moving along the assigned trajectory within the test area but with different velocities

| **Time length (s)** | **LIDAR / IMU measurements** | $f_{SUM}$ **(m$^2$)** | $f_{MEAN}$ **(m$^2$)** | **Mean comp. time (s)** | $L_{EST}$ **(m)** | **Error on** $L_{EST}$ **(%)** |
|---|---|---|---|---|---|---|
| **26.36** | 961 / 2441 | 1.180 | 0.007 | 0.013 | 7.79 | 13 |
| **9.72** | 355 / 903 | 0.947 | 0.016 | 0.018 | 8.48 | 6 |

**Table 6.5 - Performance analysis of the LIDAR/Inertial-Odometry algorithm. Effect of motion velocity (RL is 11 m, AR is 0.5°, and OR is 6 Hz).**

It is clear that, moving faster within the test area limits the error propagation of the odometry approach.

Finally, an example of application of the proposed localization and mapping algorithms is shown in Figure 6.7. It is worth outlining that mapping can be carried out at different rate with respect to localization, e.g. in this case the mapping rate is 3 Hz while the localization one is 6 Hz. First of all, it is important to outline that the reference trajectory, depicted in green in Figure 6.7, is an approximation of the real travelled trajectory and it is used to determine the error in $L_{EST}$. by looking at Figure 6.7, it is possible to state that the line-based mapping technique is able to provide a sparse but accurate representation of the travelled environment. Specifically, all the edges of the real map are accurately extracted in terms of length, location and inclination in the ENU. However, an exception is given by the two lines identified by blue circles at their ends since they have the same inclination as the corresponding real edges but are displaced of some centimeters from the real lines. This can be justified as a consequence of the error propagated in the estimated trajectory along the travelled path. Future work will be aimed at solving this issue by improving the accuracy of the proposed localization and mapping algorithms. This can be done by exploiting the information given by the map itself, in real time, to correct the solution given by the odometry algorithm.

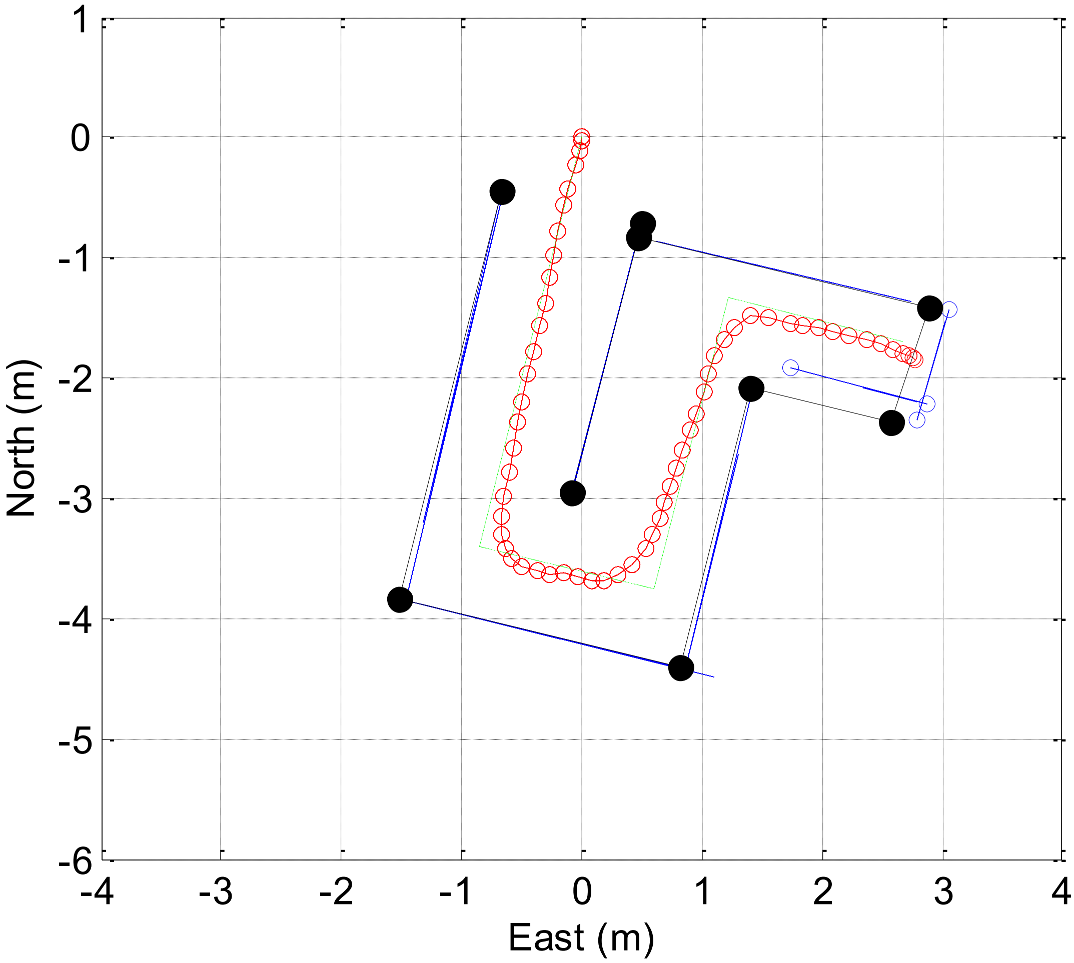


**Figure 6.7 -Example of application of localization and mapping algorithm to the analyzed test case: reference trajectory (dashed-dot green); estimated trajectory red; vertices of the real map (black dots);real map (black lines);estimated map (blue lines).**

# Chapter 7 - Conclusion and future work

The research activity presented in this thesis was addressed to the development and performance assessment of innovative techniques for advanced applications regarding the autonomous navigation of aerospace platforms.

Autonomous relative navigation of multi-satellite systems

In the framework of space activities like on-orbit servicing and active debris removal an original architecture was proposed to carry out the process of pose determination of an uncooperative target in relative motion in close-proximity with respect to a servicing spacecraft (chaser). A trade-off study, aimed at evaluating advantages and drawbacks of the related potential technological solutions, led to the selection of the LIDAR as the sensor responsible of relative navigation. Hence, this architecture was composed of original 3D model-based algorithms using LIDAR measurements (point clouds), specifically tailored to the different steps of the process, namely acquisition and tracking. The global aim was to provide solutions ensuring high accuracy levels in the estimation of the relative position and attitude parameters, while simultaneously optimizing computational efficiency and data storage issues as well as enhancing robustness of the overall architecture against variable observation conditions and potential algorithms' failures.

With regards to the pose acquisition step, different algorithms were designed which combine the template matching concept with the centroiding approach, and, in one case, also with the principal component analysis to restrain the search for the pose solution to an extremely limited portion of the relative state space, thus being able to reduce the computational load with respect to traditional approaches. Another crucial original aspect, common to all the presented techniques, was the possibility to perform the template matching search within a database built dynamically (on-line) thus allowing relevant saving in the amount of on-board data storage. With regards to the pose tracking step, a customized version of the iterative closest point algorithm was developed, which exploits different model-sensor point matching strategies at different stages during the tracking phase, and is augmented by a prediction technique to accelerate algorithm convergence.

In order to cope with the extremely restricted amount of open-access data regarding uncooperative relative motion on orbit, a software environment was developed to assess pose determination algorithms' performance by means of numerical simulations. The simulator proved to be a powerful tool to realistically reproduce the target-chaser relative dynamics, typical of close-proximity maneuvers, and the operation of a 3D spaceborne LIDAR.

The 3D PCA-based on-line template matching algorithm outperformed the other techniques conceived in this thesis for pose acquisition both in terms of success rate and computational efficiency. It was proved able to provide very high success rates (up to 96 %) considering targets of different shape and size. This result was achieved over a much wider portion of the relative state space than what was done for other techniques representing the state of the art in the context of uncooperative pose determination. In fact, the entire relative attitude space coupled with a wide interval of target-chaser relative distances (from 10 m to 80 m) were selected in order to cover the overall space of pose solutions which is of interest in the context of target monitoring scenarios. Also, the computational time was kept always around 1 s, thus being compliant with the necessity to provide the tracking algorithm with an initial guess of the pose parameters fast enough to avoid losing track of the target. Algorithm's robustness was improved by adopting two solutions. Firstly, a method to establish the applicability of the algorithm during operation, based on the analysis of the shape of the measured point cloud was introduced. Secondly, one additional step to ensure safe transition from acquisition to tracking by solving any possible ambiguity in the pose solution was included in the pose determination process. Both, these strategies were proved to work properly in any simulated scenario, by respectively identifying a-priori potential failures and solving pose ambiguities. The 3D PCA-based on-line template matching algorithm was specifically envisaged to estimate the initial relative position and attitude of targets characterized by a principal direction, as it is mostly verified for both operating spacecraft and large debris. If this condition is not satisfied, the other developed algorithms, namely the on-line template matching and the on-line fast template matching, can be applied, instead. Indeed, they were able to provide high success rates (up to 97 %, but lower on average over the analyzed region of the relative state space) in spite of a larger but still acceptable computational time (around 10 s). Also, it was possible to state that the on-line fast template matching is preferable to the basic version of the algorithm, as it improves the computational efficiency while keeping constant the success rate, only at close range or if the database is adequately sampled so that enough information is available to solve potential pose ambiguities.

Numerical simulations regarding the tracking process demonstrated the capability of the proposed implementation of the iterative closest point algorithm to attain sub-degree and centimeter accuracy level in the estimation of the relative attitude and position, respectively, again considering different targets. This result was achieved performing the pose determination process along target-chaser relative trajectories designed for monitoring scenarios, which ensured large variability of pose conditions. Sensibility of the tracking performance to the adoption of different variants of the implemented algorithm as well as to changes in the model parameters was tested by means of ad-hoc simulations. They allowed demonstrating that the adoption of the normal shooting method in the matching step aided by the prediction algorithm is the best performing both in terms of pose estimation accuracy and computational load, thus being the most suitable approach when the stationary state of the tracking process is reached. Moreover, the optimal configuration of the model parameters was also found.

It is crucial to outline that all these results were obtained in spite of the extremely relevant sparseness of the considered datasets (from 50 to 500 points, on average, reducing range), obtained assigning low resolution (1°) to the simulated LIDAR with respect to the achievable performance (up to 0.01°). This helped limiting the computational burden, on one side, but also allowed testing the pose determination process against much more challenging conditions, on the other side. It is also worth outlining that the proposed techniques were designed to extract the pose by observing the global structure of the target. Hence, they were not conceived to be applied at very close distances (below 10 m/20 m from the target depending on its size). Indeed, during final approach and docking maneuvers, pose determination (both cooperative and uncooperative) is preferably performed relying on the analysis of local structures. However, in the analyzed region of the relative state space, the proposed methods demonstrated the capability to work properly even in case of relevant occlusion of the target shape. The robustness of the overall architecture was further enhanced by means of an autonomous failure detection strategy applied to both the acquisition and the tracking algorithms. For instance, this method provided an absolute probability of correctly identifying successes and failures of pose acquisition above 99.8 % considering a large asymmetric-shaped debris (ENVISAT) as test case. An additional major result accomplished by this research activity was the introduction of an original analytical approach to design innovative safe trajectories for target monitoring. Specifically, the core principle was given by the possibility of estimating the differences between the initial mean orbit parameters of the chaser and the target which univocally identify the relative trajectory by assign high level requirements regarding its stability, size and shape. Results from ad-hoc numerical simulations demonstrated that this method can be used to design trajectory which provide optimal observation condition to the sensor responsible of relative navigation both regarding the acquisition and tracking step of the pose determination process. In particular, it was demonstrated that for targets whose rotational dynamics is gravity gradient stabilized, which is typical of several uncontrolled space objects having a minimum inertia axis, optimal observability for pose determination can be attained by setting the out-of-plane observation angle as low as possible, of course provided that it is compatible to other geometrical and stability constraints.

The future of this activity lies in two potential advances. Firstly, algorithms' effectiveness can be tested over real data acquired thanks to a purposely designed experimental setup including sensors able to measure point clouds, like 3D LIDAR or depth cameras. Secondly, the solution of pose determination can be integrated into a navigation filter.

Autonomous navigation of Unmanned Aerial Vehicles

Regarding the problem of autonomous safe navigation of UAVs flying in complex environments, like GPS-challenging or GPS-denied, a research activity was carried out in collaboration with the Cranfield University (UK). The focus was on the development of original solutions to the problems of localization and mapping for micro-UAVs taking their related limitations in terms of payload and processing resources into account. The LIDAR technology was selected as it provides the best compromise considering operating distances, weight and power requirements, and achievable level of autonomy, among all the potential ranging sensors. The proposed techniques were both based on the low-level integration (not inside a navigation filter) of LIDAR measurements and inertial data. Localization was entrusted to an odometry algorithm based on scan matching which implements a customized version of the iterative closest point algorithm and foresees an autonomous failure detection strategy. Mapping was performed adopting a line-based representation of the environment. The most important aspect of the original line-based mapping algorithm was relative to the adoption of the principal component analysis to carry out line fitting, instead of using state-of-the-art least-squares methods, thus obtaining a time saving of about 80 %, while keeping constant the accuracy level. Localization and mapping algorithm performance were evaluated by means of off-line runs over real data acquired by means of a purposely conceived experimental setup carried by hand within a 2D test area. They demonstrated capability to estimate the vehicle's trajectory and the map of the environment with centimeter-level accuracy. Also, effectiveness of the autonomous failure detection of localization was demonstrated.

Further work will be aimed at using information from the updating map to improve localization accuracy and at comparing the proposed solution to the results provided by means of an extended Kalman filter algorithm. Indeed, this comparison will show advantages and drawbacks of the proposed method with respect to more consolidated state-of-the-art solutions.

# References

[1] A.M. Long, M.G. Richards, and D.E. Hastings, “On-Orbit Servicing: A New Value Proposition for Satellite Design and Operation,” AIAA Journal of Spacecraft and Rockets, vol. 44 (4), pp. 964-976, 2007.

[2] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” Progress in Aerospace Sciences, vol. 68, pp. 1-26, 2014.

[3] X. Clerc, and I. Retat, “Astrium Vision on Space Debris Removal,” Proc. of the 63rd International Astronautical Congress, Naples, Italy, Oct. 2012.

[4] C. Bonnal, J.M. Ruault, and M.C. Desjean, “Active debris removal: Recent progress and current trends,” Acta Astronautica, vol. 85, pp. 51- 60, 2013.

[5] J.C. Liou, D.T. Hall, P.H. Krisko and J.N. Opiela, “LEGEND–A three-dimensional LEO-to-GEO debris evolutionary model,” Advances in Space Research, vol. 34 (5), pp. 981-986, 2004.

[6] J.L. Liou, “An active debris removal parametric study for LEO environment remediation,” Advances in Space Research, vol. 47 (11), pp. 1865-1876, 2011.

[7] D.J. Kessler and B.G. Cour-Palais, “Collision frequency of artificial satellites: Creation of a debris belt,” Journal of Geophysical Research, vol. 83, pp. 2637-2646, 1978.

[8] F. Kennedy, “Orbital Express: accomplishments and lessons learned,” Advances in the Astronautical Sciences, vol. 131, pp. 575-586, 2008.

[9] P. Bodin, R. Noteborn, R. Larsson, T. Karlsson, S. D'Amico, J.S. Ardaens, M. Delpech, and J.C. Berges, “PRISMA Formation Flying Demonstrator: Overview and Conclusions from the Nominal Mission,” Advances in the Astronautical Sciences, vol. 144, pp. 441-460, 2012.

[10] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, “On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations,” AIAA SpaceOps 2010 Conference, Huntsville, AL, USA, Apr. 2010.

[11] M. Wieser, R. Haarmann, G. Hausmann, J.C. Meyer, S. Jäkel, M. Lavagna, and R. Biesbroek, “e.Deorbit Mission: OHB Debris Removal Concepts,” Proc. of the 13th Symposium on Advanced Space Technologies in Robotics and Automation, Noordwijk, The Netherlands, May 2015.

[12] D.A Smith, C. Martin, M. Kassebom, H.Petersen, A. Shaw, B. Skidmore, D. Smith, H. Stokes and A. Willig, "A mission to preserve the geostationary region." Advances in Space Research, vol. 34 (5), pp. 1214-1218, 2004.

[13] F. Kendoul, “Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems,” Journal of Field Robotics, vol. 29 (2), pp. 315-378, 2012.

[14] R. L. Greenspan, “GPS and inertial integration,” in *Global Positioning System: Theory and applications. Volume 2*, ch. 7, pp. 187-220, 1996.

[15] J. Farrell, “Aided navigation: GPS with high rate sensors,” McGraw-Hill, Inc., 2008.

[16] K. Jong-Hyuk, S. Sukkarieh, and S. Wishart, “Real-time navigation, guidance, and control of a UAV using low-cost sensors,” Field and Service Robotics, Springer Berlin Heidelberg, 2006.

[17] J. Wendel, O. Meister, C. Schlaile, and G.F. Trommer, “An integrated GPS/MEMS-IMU navigation system for an autonomous helicopter,” Aerospace Science and Technology, vol. 10 (6), pp. 527-533, 2006.

[18] H. Durrant-Whyte, and T. Bailey, “Simultaneous localization and mapping: part I,” IEEE Robotics & Automation Magazine, vol. 13 (2), pp. 99-110, 2006.

[19] T. Bailey, and H. Durrant-Whyte, “Simultaneous localization and mapping (SLAM): Part II,” IEEE Robotics & Automation Magazine, vol. 13 (3), pp. 108-117, 2006.

[20] A. Bachrach, R. He, and N. Roy, “Autonomous flight in unknown indoor environments,” International Journal of Micro Air Vehicles, vol. 1 (4), pp. 217-228, 2009.

[21] W. Fehse, “Automated rendezvous and docking of spacecraft,” Cambridge University Press, 2003.

[22] A. Renga, M. Grassi, and U. Tancredi, “Relative navigation in LEO by carrier-phase differential GPS with intersatellite ranging augmentation,” International Journal of Aerospace Engineering, 2013.

[23] S. D'Amico, J.S. Ardaens, and S. De Florio, “Autonomous formation flying based on GPS—PRISMA flight results,” Acta Astronautica, vol. 82 (1), pp. 69-79, 2013.

[24] T. Weismuller, and M. Leinz, “GN&C Technology Demonstrated by the Orbital Express Autonomous Rendezvous and Capture Sensor System,” 29th Annual American Astronautical Society Guidance and Control Conference, Feb. 2006.

[25] H. Benninghoff, T. Tzschichholz, T. Boge, and G. Gaias, “Far Range Image Processing Method For Autonomous Tracking of an Uncooperative Target,” Proc. of the

12th Symposium on Advanced Space Technologies in Robotics and Automation, Noordwjik, The Netherlands, May 2013.

[26] J.M. Kelsey, J. Byrne, M. Cosgrove, S. Seereeram, and R.K. Mehra, "Vision-based relative pose estimation for autonomous rendezvous and docking," Proc. of the 2006 IEEE Aerospace Conference, Big Sky, MT, USA, Mar. 2006.

[27] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Active vision-based pose estimation of an uncooperative target," Proc. of the 2nd IAA Conference on Dynamics and Control of Space Systems, Roma, Italy, Mar. 2014.

[28] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Spaceborne LIDAR-based system for pose determination of uncooperative targets," Proc. of the 1st IEEE International Workshop on Metrology for Aerospace, Benevento, Italy, May 2014.

[29] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "LIDAR-based autonomous pose determination for a large space debris," Proc. of the 65th International Astronautical Congress, Toronto, Canada, Sept.-Oct. 2014.

[30] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Performance Evaluation of 3D Model-based Techniques for Autonomous Pose Initialization and Tracking," AIAA SciTech 2015, Kissimmee, FL, USA, Jan. 2015.

[31] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, "Uncooperative pose estimation with a LIDAR-based system," Acta Astronautica, vol. 110, pp.287-297, 2015.

[32] R.Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE Journal of Robotics and Automation, vol. 3 (4), pp. 323-344, 1987.

[33] 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 (6), pp. 381-395, 1981.

[34] Y. Wu, and Z. Hu, "PnP Problem Revisited," Journal of Mathematical Imaging and Vision, vol. 24 (1), pp. 131–141, 2006.

[35] M. Benn, and J.L. Jørgensen, "Short Range Pose and Position Determination of Spacecraft using a μ-Advanced Stellar Compass," Proc. of the 3rd International Symposium on Formation Flying, Missions and Technologies, Noordwijk, The Netherlands, Apr. 2008.

[36] M.A. Abidi, and T. Chandra, "A new efficient and direct solution for pose estimation using quadrangular targets: Algorithm and evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17 (5), pp. 534-538, 1995.

[37] M.D. Shuster, and S.D. Oh, “Three-axis attitude determination from vector observations,” AIAA Journal of Guidance and Control, vol.4 (1), pp.70-77, 1981.

[38] G. Fasano, A. Accardo, and M. Grassi, “A Stereo-vision Based System for Autonomous Navigation of an In-orbit Servicing Platform,” AIAA Infotech 2009, Seattle, WA, USA, 2009.

[39] J. Gang-wu , J. Ting , and W. Jing, “A framework of relative navigation system for non-cooperative target using dual--CCD,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 37, pp. 751-756, 2008.

[40] C. Doignon, “An Introduction to Model-Based Pose Estimation and 3-D Tracking Techniques,” in *Scene Reconstruction, Pose Estimation and Tracking*, ch. 20, pp. 359-382, 2007.

[41] C. Reinbacher, M. Ruther, and H, Bischof, “Pose Estimation of Known Objects by Efficient Silhouette Matching,” Proc. of the IEEE 20th International Conference on Pattern Recognition, pp. 1080-1083, Istanbul, Turkey, Aug. 2010.

[42] M. Greenspan, L. Shang, and P. Jasiobedzki, “Efficient tracking with the bounded hough transform,” Proc. of the 2004 IEEE Computer Society Conference, on Computer Vision and Pattern Recognition, vol. 1, pp. I-520 - I-527, Washington, DC, USA, Jun. 2004.

[43] D.G. Lowe, “Object Recognition from Local Scale-Invariant Features,” Proc. of the 7th IEEE international conference on Computer Vision, vol. 2, pp. 1150-1157, Sept. 1999.

[44] T. Drummond, and R. Cipolla, “Real-time visual tracking of complex structures,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24 (7), pp. 932-946, 2002.

[45] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, “Real-time markerless tracking for augmented reality: the virtual visual servoing framework,” IEEE Transactions on Visualization and Computer Graphics, vol. 12 (4), pp. 615-628, 2006.

[46] C. Liu, and W. Hu, “Relative pose estimation for cylinder-shaped spacecrafts using single image,” IEEE Transactions on Aerospace and Electronics Systems, vol. 50 (4), pp. 3036-3056, 2014.

[47] P. David, D.F. DeMenthon, R. Duraiswami, and H. Samet, “SoftPOSIT: Simultaneous pose and correspondence determination,” International Journal of Computer Vision, vol. 59 (3), pp. 259-284, 2004.

[48] P. David, D.F. DeMenthon, R. Duraiswami, and H. Samet, “Simultaneous pose and correspondence determination using line features,” Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 424-431, Madison, WI, USA, Jun. 2003.

[49] S. Gold, and A. Rangarajan, “A graduated assignment algorithm for graph matching,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18 (4), pp. 377-388, 1996.

[50] S. Gold, A. Rangarajan, C.P. Lu, S. Pappu, and E. Mjolsness, “New algorithms for 2d and 3d point matching: pose estimation and correspondence,” Pattern Recognition, vol. 31 (8), pp. 1019-1031, 1998.

[51] D.F. DeMenthon, and P. Davis, “Model-based object pose in 25 lines of code,” International Journal of Computer Vision, vol. 15 (1-2), pp. 123-141, 1995.

[52] P.J. Besl, and N.D. McKay, “Method for registration of 3-D shapes,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, pp. 239–256, 1992.

[53] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “A Model-Based 3D Template Matching Technique for Pose Acquisition of an Uncooperative Space Object,” Sensors, vol. 15 (3), pp. 6360-6382, 2015.

[54] R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Large space debris pose acquisition in close-proximity operations,” Proc. of the 2nd IEEE International Workshop on Metrology for Aerospace, Benevento, Italy, Jun. 2015.

[55] M. Greenspan, and P. Jasiobedzki, “Pose determination of a free-flying satellite,” Proc. of the 2002 International Conference on Imaging Science, Systems, and Technology, Las Vegas, NV, USA, Jun. 2002.

[56] S. Ruel, T. Luu, M. Anctil, and S. Gagnon, “Target localization from 3d data for on-orbit autonomous rendezvous and docking,” Proc. of the 2008 IEEE Aerospace Conference, Big Sky, MT, USA, Mar. 2008.

[57] T.F. Cootes, G.J. Edwards, and C.J. Taylor, “Active Appearance Models,” IEEE Transactions on pattern Analysis and Machine Intelligence, vol. 23 (6), pp. 681-685, 2001.

[58] P. Mittrapiyanuruk, G.N. DeSouza, and A.C. Kak, “Calculating the 3d-pose of rigid-objects using active appearance models,” Proc. of the IEEE International Conference on Robotics and Automation, pp. 5147-5152, New Orleans, LA, USA, Apr. 2004.

[59] M.J. Black, and A.D. Jepson, “Eigentracking: Robust matching and tracking of articulated objects using a view-based representation,” International Journal of Computer Vision, vol. 26 (1), pp. 63-84, 1998.

[60] R.C. Gonzalez, and R.E. Woods, “Digital Image Processing,” 2nd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2002.

[61] N.N. Dawoud, B.B. Samir, and J. Janier, “Fast Template Matching Method Based Optimized Sum of Absolute Difference Algorithm for Face Localization,” International Journal of Computer Applications, vol. 18, pp. 30–34, 2011.

[62] K. Briechle, and U.D. Hanebeck, “Template matching using fast normalized cross correlation,” Proc. SPIE 2001, 4387, doi:10.1117/12.421129.

[63] D.M. Gavrila, and V. Philomin, “Real-time object detection using distance transforms,” Proc. of the IEEE International Conference on Intelligent Vehicles, pp. 274–279,Stuttgart, Germany, Oct. 1998.

[64] R.O. Duda, and P.E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” Communications of the ACM, vol. 15 (1), pp. 11-15, 1972.

[65] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” Chemometrics and intelligent laboratory systems, vol. 2 (1), pp. 37-52, 1987.

[66] K. Kanani, A. Petit, E. Marchand, T. Chabot, and B. Gerber, “Vision Based Navigation for Debris Removal Missions,” Proc. of the 63rd International Astronautical Congress, Naples, Italy, Oct. 2012.

[67] A. Petit, E. Marchand, and K. Kanani, “Vision-based space autonomous rendezvous: A case study,” Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 619-624, San Francisco, CA, USA, Sept. 2011.

[68] A. Petit, E. Marchand, K. Kanani, “Vision-based Detection and Tracking for Space Navigation in a Rendezvous Context,” Proc. of the International Symposium on Artificial Intelligence, Robotics and Automation in Space, Turin, Italy, Sept. 2012.

[69] S. D’Amico, M. Benn, and J. L. Jørgensen, “Pose estimation of an uncooperative spacecraft from actual space imagery,” International Journal of Space Science and Engineering, vol. 2 (2), pp. 171-189, 2014.

[70] D.G. Lowe, “Three Dimensional Object Recognition from Single Two Dimensional Images,” Artificial Intelligence, vol. 31, pp 355-395, 1985.

[71] J.G. Semple, and G.T. Kneebone, “Algebraic Projective Geometry,” London: Clarendon Press, 1952, ch. IV and V, pp. 72–129.

[72] A. Cropp, P. Palmer, and C.I. Underwood, “Pose estimation of target satellite for proximity operations,” Proc. of the 14th Annual AIAA/USU Conference on Small Satellites, Paper no. SSC00-III-7, Logan, UT, USA, Aug. 2000.

[73] M.A. Rodrigàlvarez, D.E. Olmos, and J.G. Fernàndez, “IP-based pose estimation for space debris removal: implementation and results,” Proc. of the 65th International Astronautical Congress, Toronto, Canada, Sept.-Oct. 2014.

[74] C. Lee, and R.M. Haralick, “Statistical estimation for exterior orientation from line-to-line correspondences”, Image and Vision Computing, vol. 14, pp. 379-388, 1996.

[75] T.V. Peters, D. E. Olmos, A. Pellacani, M. A. Rodrigálvarez, M. Lavagna, F. Ferrari, P. Attina, G. Parissenti, and A. Cropp, “The COBRA IRIDES Experiment,” Proc. of the 65th International Astronautical Congress, Toronto, Canada, Sept.-Oct. 2014.

[76] P. Jasiobedzki, S. Se, T. Pan, M. Umasuthan, and M. Greenspan, “Autonomous Satellite Rendezvous and Docking Using LIDAR and Model Based Vision, “ Proc. of SPIE Space borne Sensor II, vol. 5798, Jun. 2005.

[77] J. Sommer, and I. Ahrns, “GNC for a Rendezvous in Space with an Uncooperative Target,” 5th International Conference on Spacecraft Formation Flying Missions and Technologies, Munich, Germany, May 2013.

[78] www.neptec.com

[79] S. Ruel, C. English, M. Anctil, and P. Church, “3DLASSO: Real-time pose estimation from 3D data for autonomous satellite servicing,” Proc. of the 2005 International Symposium on Artificial Intelligence for Robotics and Automation in Space, Munich, Germany, Sept. 2005.

[80] http://www.teledyneoptech.com/index.php/product/optech-ilris/

[81] P. Jasiobedzki, M. Greenspan, G. Roth, H. Ng, and N. Witcomb, “Video-based system for satellite proximity operations,” Proc. of the 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation, Noordwijk, The Netherlands, Nov. 2002.

[82] T. Tzschichholz, T. Boge, and K. Schilling, “Relative pose estimation of satellites using PMD-/CCD-sensor data fusion,” Acta Astronautica, vol. 109, pp. 25-33, 2015.

[83] S. Rusinkiewicz, and M. Levoy, “Efficient variants of the ICP algorithm,” Proc. of the 3rd IEEE International Conference on 3-D Digital Imaging and Modeling, pp. 145-152, Quebec City, Canada, Jun. 2001.

[84] J.K. Friedman, J. Bentely, and R.A. Finke, “An Algorithm for Finding Best Matches in Logarithmic Expected Time”, ACM Transactions on Mathematical Software, vol. 3 (3), pp. 209-226, 1977.

[85] B.K. Horn, “Closed-form solution of absolute orientation using unit quaternions,” Journal of the Optical Society of America, vol. 4 (4), pp. 629-642, 1987.

[86] F. Blais, “Review of 20 Years of Range Sensor Development,” Journal of Electronic Imaging, vol. 13, pp. 231-240, 2004.

[87] S. Crosby, and S.H. Kang, "Object Identification in 3D Flash Lidar Images," Journal of Pattern Recognition Research, vol. 6 (2), pp. 193-200, 2011.

[88] J.A. Christian, and S. Cryan, “A Survey of LIDAR Technology and its Use in Spacecraft relative Navigation,” AIAA Guidance, Navigation, and Control Conference, vol. 7061, San Diego, CA, USA, Aug. 2008.

[89] A. Kovács, “Scanning Strategies for Imaging Arrays,” SPIE Astronomical Telescopes and Instrumentation. International Society for Optics and Photonics, 2008.

[90] R.M. Willett, F.M. Roummel, and J.M. Nichols, “Compressed sensing for practical optical imaging systems: a tutorial,” Optical Engineering, vol. 50 (7), 2011.

[91] J.A. Beraldin, F. Blais, M. Rioux, L. Cournoyer, D. Laurin, and S.G. MacLean, “Eye-safe digital 3-D sensing for space applications,” Optical Engineering, vol. 39 (1), pp. 196-211, 2000.

[92] C. English, S. Zhu, C. Smith, S. Ruel, and I. Christie, “Tridar: A hybrid sensor for exploiting the complementary nature of triangulation and LIDAR technologies,” Proc. of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Sept. 2008.

[93] J. A. Beraldin, F. Blais, M. Rioux, L. Cournoyer, D. G. Laurin, and S. G. MacLean, “Short and medium range 3D sensing for space applications,” Proc. of SPIE Visual Information Processing VI, vol. 3074, pp. 29-46, Jul. 1997.

[94] D. G. Laurin, J. A. Beraldin, F. Blais, M. Rioux, and L. Cournoyer, “A three dimensional tracking and imaging laser scanner for space operations,” Proc. of SPIE Laser radar technology and applications IV, vol. 3707, pp. 278-289, May 1999.

[95] T. D. Cole, “NEAR Laser Rangefinder: A Tool for the Mapping and Topologic Study of Asteroid 433 Eros,” Johns Hopkins APL Technical Digest, vol. 19 (2), pp. 142-157, 1998.

[96] A. F. Cheng, O. Barnouin-Jha, L. Prockter, M.T. Zuber, G. Neumann, D. E. Smith, J. Garvin, M. Robinson, J. Veverka, and P. Thomas, “Small-scale topography of 433 Eros from laser altimetry and imaging,” Icarus, vol. 155 (1), pp. 51-74, 2002.

[97] T. Kasai, M. Oda, and T. Suzuki, “Results of the ETS-7 mission – rendezvous docking and space robotics experiments,” Proc. of the 5th international symposium on

artificial intelligence, robotics and automation in space, Noordwijk, The Netherlands, ESA SP-440, pp 299–306, Jun. 1999.

[98] M. Mokuno, I. Kawano, and T. Suzuki, “In-Orbit Demonstration of Rendezvous Laser Radar for unmanned autonomous rendezvous docking,” IEEE Transactions on Aerospace and Electronic Systems, vol. 40 (2), pp. 617–626, Jul. 2004.

[99] C.L. Smithpeter, R.O. Nellums, S.M. Lebien, G. Studor, and G. James, “LADAR measurements of the international space station,” Proc. of SPIE Laser radar technology and applications VI, vol. 4377, pp. 65-72, Sept. 2001.

[100] C. Samson, C. English, A. Deslauriers, I. Christie, F. Blais, and F. Ferrie, “Neptec 3D Laser Camera System: From space mission STS-105 to terrestrial applications,” Proc. of the 2002 ASTRO Conference, Ottawa, Canada, Nov. 2002.

[101] C.C. Liebe, A. Abramovici, R.K. Bartman, R.L. Bunker, J. Chapsky, C.C. Chu, D. Clouse, J.W. Dillon, B. Hausmann, H. Hemmati, R.P. Komfeld, C. Kwa, S. Mobasser, M. Newell, C. Padgett, W. T. Roberts, G. Spiers, Z. Warfield, and M. Wright, “Laser Radar for Spacecraft Guidance Applications,” Proc. of 2003 IEEE Aerospace Conference, vol. 6, pp. 2647-2662, Big Sky, MT, USA, Mar. 2003.

[102] R.P. Kornfeld, R.L. Bunker, G.C. Cucullu, J.C. Essmiller, F.Y. Hadaegh, C.C. Liebe, C.W. Padgett, and E.C. Wong, “New millennium ST6 autonomous rendezvous experiment (ARX),” Proc. of the 2003 IEEE Aerospace Conference, vol. 1, pp. 369-380, Big Sky, MT, USA, Mar. 2003.

[103] www.jena-optronik.de

[104] E. Dupuis, Y. Gonthier, J.C. Piedbœuf, “An overview of Canadian space robotics activities,” Field Robotics, pp. 35-52, 2012.

[105] http://www.advancedscientificconcepts.com

[106] A.S. Glassner, “An introduction to ray tracing,” Elsevier, 1989.

[107] G. Euliss, A. Christiansen, and R. Athale, “Analysis of laser-ranging technology for sense and avoid operation of unmanned aircraft systems: the tradeoff between resolution and power,” SPIE Defense and Security Symposium, vol. 6962, pp. 696208-1-10, 2008.

[108] R.D. Richmond, and S.E. Cain, “Direct Detection LADAR Systems,” SPIE Press, Bellingham, WA, USA, 2010.

[109] F.E. Nicodemus, J.C. Richmond, J.J. Hsia, I.W. Ginsberg, and T. Limperis, “Geometrical considerations and nomenclature for reflectance,” National Bureau of Standards, vol. NBS MN-160, Washington, DC, USA, p. 52, 1977.

[110] J.W. Goodman, “Statistical optics,” John Wiley & Sons, Hoboken, NJ, USA, 2015.

[111] S. Kim, I. Lee, and Y.J. Kwon, “Simulation of a geiger-mode imaging LADAR system for performance assessment,” Sensors, vol. 13 (7), pp. 8461-8489, 2013.

[112] L. Ansalone, “A Search Algorithm for Stochastic Optimization in Initial Orbit Determination,” PhD Thesis, University of Rome "La Sapienza", 2014.

[113] http://rredc.nrel.gov/solar/spectra/am1.5/ASTMG173/ASTMG173.html

[114] www.hamamatsu.com.

[115] www.thorlabs.com

[116] ENVISAT Mission & System Summary, www.earth.esa.int/support-docs.

[117] Y.L. Desnos, C. Buck, J. Guijarro, J.L. Suchail, R. Torresand, E. Attema, “ASAR-Envisat's advanced synthetic aperture radar,” ESA Bulletin, vol.102, pp. 91–100, 2000.

[118] A. Torre, P. Capece, “COSMO-SkyMed: the advanced SAR instrument,” Proc. of the IEEE 5th International Conference on Recent Advances in Space Technologies, pp. 865-868, Istanbul, Turkey, Jun. 2011.

[119] E. Ferrando, F. Rizzi, D. Hazan, E. Scrozafava, Y. Combet, F. Geyer, C. Flores, R. Campesato, G. Gabetta, N. Fatemi, P. Sharps, B. Clevenger, “The COSMO-SkyMed solar array,” Proc. of the IEEE 4th World Conference on Photovoltaic Energy Conversion, vol. 2, pp. 1990-1993, Waikoloa, HI, USA, May 2006.

[120] http://www.spacelaunchreport.com/kosmos.html

[121] D.G. Gilmore, “Spacecraft Thermal Control Handbook,” The Aerospace Press, USA, 2002.

[122] Appendix1toESAITTAO/1-7477/13/NL/MH, Statement of Work: Advanced GNC for Debris Removal (AGADR), 2013.

[123] S. Croomes, “Overview of the DART mishap investigation results,” NASA Report, 2006, pp. 1-10.

[124] B. Naasz, “Safety ellipse motion with coarse sun angle optimization,” Proc. of the NASA-GSFC Flight Mechanics Symposium, Oct. 2005.

[125] D.E. Gaylor, and B.W. Barbee, “Algorithms for safe spacecraft proximity operations,” AAS/AIAA Spaceflight Mechanics Meeting, Jan. 2007.

[126] R. Larsson, J. Mueller, S. Thomas, B. Jakobsson, and P. Bodin, “Orbit constellation safety on the PRISMA in-orbit formation flying testbed,” Proc. of the 3rd International Symposium on Formation Flying, Missions and Technologies, ESA, Apr. 2008.

[127] C. Sabol, R. Burns, and C.A. McLaughlin, “Satellite formation flying design and evolution,” Journal of Spacecraft and Rockets, vol. 38 (2), pp. 270-278, 2001.

[128] G. Fasano, M. D’Errico, “Modeling orbital relative motion to enable formation design from application requirements,” Celestial Mechanics and Dynamical Astronomy, vol. 105 (1-3), pp. 113-139, 2009.

[129] M. Schmidt, D. Salt, “Herschel Pointing Accuracy Improvement,” Proc. of the 12th International Conference on Space Operations, Stockholm, Sweden, Jun. 2012.

[130] T. Finch, “Incremental calculation of weighted mean and variance,” University of Cambridge Computing Service, 2009.

[131] D. Scaramuzza, and F. Fraundorfer, “Visual odometry [tutorial] ,” IEEE Robotics & Automation Magazine, vol. 18 (4), pp. 80-92, 2011.

[132] A.F. Scannapieco, A. Renga, and A. Moccia, “Preliminary Study of a Millimeter Wave FMCW InSAR for UAS Indoor Navigation,” Sensors, vol. 15 (2), pp. 2309-2335, 2015.

[133] S. Weiss, M.W. Achtelik, S. Lynen, M.C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, “Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium,” Journal of Field Robotics, vol. 30 (5), pp. 803-831, 2013.

[134] G. Chowdhary, E.N. Johnson, D. Magree, A. Wu, and A. Shein, “GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft,” Journal of Field Robotics, vol. 30 (3), pp. 415-438, 2013.

[135] Z. Lu, Z. Hu, and K. Uchimura, “SLAM estimation in dynamic outdoor environments: A review,” Intelligent Robotics and Applications, Springer Berlin Heidelberg, pp. 255-267, 2009.

[136] T.D. Vu, J. Burlet, and O. Aycard, “Mapping of environment, detection and tracking of moving objects using occupancy grids,” Intelligent Vehicles Symposium, pp. 684-689, 2008.

[137] A. Diosi, and L. Kleeman, “Laser scan matching in polar coordinates with application to SLAM,” Proc. of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3317-3322, Edmonton, Canada, Aug. 2005.

[138] M. Bosse, P. Newman, J. Leonard, and S. Teller, “Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework,” The International Journal of Robotics Research, vol. 23 (12), pp. 1113-1139, 2004.

[139] A. Censi, L. Iocchi, and G. Grisetti, “Scan matching in the Hough domain,” Proc. of the 2005 IEEE International Conference on Robotics and Automation, pp. 2739-2744, Barcelona, Spain, Apr. 2005.

[140] E.B. Olson, "Real-time correlative scan matching," Proc. of the 2009 IEEE International Conference on Robotics and Automation, pp. 4387-4393, Kobe, Japan, May 2009.

[141] Y.J. Lee, J.B. Song, and J.H. Choi, "Performance improvement of iterative closest point-based outdoor SLAM by rotation invariant descriptors of salient regions," Journal of intelligent & robotic systems, vol. 71 (3-4), pp. 349-360, 2013.

[142] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," Proc. of the 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, pp. 155-160, Kyoto, Japan, Nov. 2011.

[143] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics," MIT Press,2008.

[144] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," IEEE Transactions on Robotics, Vol. 23 (1), pp. 34-46, 2007.

[145] D. Droeschel, J. Stuckler, and S. Behnke, "Local multi-resolution representation for 6D motion estimation and mapping with a continuously rotating 3D laser scanner," Proc. of the IEEE International Conference on Robotics and Automation, pp. 5221-5226, 2014.

[146] M. Beul, N. Krombach, Y. Zhong, D. Droeschel, M. Nieuwenhuisen, and S. Behnke, "A High-performance MAV for Autonomous Navigation in Complex 3D Environments," Proc. of the 2015 International Conference on Unmanned Aircraft Systems, pp. 1241-1250, Denver, CO, USA, 2015.

[147] J. Zhang, and S. Singh, "LOAM: Lidar odometry and mapping in real-time," Proc. of the 2014 Robotics: Science and Systems Conference, Berkeley, CA, USA, Jun. 2014.

[148] H. Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems," Department of Civil and Environmental Engineering, Duke University, pp. 1-15, 2011.

[149] J. Zhang, and S. Singh, "Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast," Proc. of the 2015 International Conference on Unmanned Aircraft Systems, pp. 2174-2181, Denver, CO, USA, 2015.

[150] S. Winkvist, "Low computational SLAM for an autonomous indoor aerial inspection vehicle," Doctoral dissertation, University of Warwick, 2013.

[151] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics," Proc. of

the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1929-1934, Edmonton, Canada, Aug. 2005.

[152] L. Zhang, and B.K. Ghosh, “Line segment based map building and localization using 2D laser rangefinder,” Proc. of the IEEE International Conference on Robotics and Automation ICRA'00, vol. 3, pp. 2538-2543, San Francisco, CA, USA, Apr. 2000.

[153] K.O. Arras, and R. Siegwart, “Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building,” Proc. of the Symposium on Intelligent Systems and Advanced Manufacturing, Pittsburgh, PA, USA, Oct. 1997.

[154] C. Berger, “Toward rich geometric map for SLAM: Online Detection of Planes in 2D LIDAR,” Journal of Automation Mobile Robotics and Intelligent Systems, vol. 7, pp. 36-41, 2013.

[155] Y. Li, and E.B. Olson, “Extracting general-purpose features from LIDAR data,” Proc. of the 2010 IEEE International Conference on Robotics and Automation, pp. 1388-1393. Anchorage, AK, USA, May 2010.

[156] https://www.hokuyo-aut.jp/02sensor/07scanner/download/products/utm-30lx-ew/.

[157] P. Demski, M. Mikulski, and R. Koteras, “Characterization of hokuyo utm-30lx laser range finder for an autonomous mobile robot,” Advanced Technologies for Intelligent Systems of National Border Security. Springer Berlin Heidelberg, pp. 143-153, 2013

[158] M. Quigley, K. Conley, B.P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A.Y. Ng, “ROS: an open-source Robot Operating System,” ICRA Workshop on Open Source Software, 2009.

# List of publications

## Journal papers

- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Uncooperative pose estimation with a LIDAR-based system,” Acta Astronautica, vol. 110, pp.287-297, 2015.
- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “A Model-Based 3D Template Matching Technique for Pose Acquisition of an Uncooperative Space Object,” Sensors, vol. 15 (3), pp. 6360-6382, 2015.
- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Pose Estimation for Spacecraft Relative Navigation Using Model-based Algorithms,” IEEE Transactions on Aerospace and Electronic Systems, under final review.
- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Optimal design of relative trajectories for in-orbit proximity operations,” Acta Astronautica, under review.

## Conference papers

- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Active vision-based pose estimation of an uncooperative target,” Proc. of the 2nd IAA Conference on Dynamics and Control of Space Systems, Roma, Italy, Mar. 2014 (published in the Advances in Astronautical Sciences, vol. 153, pp. 1-15, 2015).
- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Spaceborne LIDAR-based system for pose determination of uncooperative targets,” Proc. of the 1st IEEE International Workshop on Metrology for Aerospace, Benevento, Italy, May 2014.
- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “LIDAR-based autonomous pose determination for a large space debris,” Proc. of the 65th International Astronautical Congress, Toronto, Canada, Sept.-Oct. 2104.

- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Performance Evaluation of 3D Model-based Techniques for Autonomous Pose Initialization and Tracking,” Proc. of the AIAA SciTech 2015, Kissimmee, USA, Jan. 2015.
- R. Opromolla, G. Fasano, G. Rufino, and M. Grassi, “Large space debris pose acquisition in close-proximity operations,” Proc. of the 2$^{nd}$ IEEE International Workshop on Metrology for Aerospace, Benevento, Italy, Jun. 2015.
- R. Opromolla, A. Savvaris, G. Fasano, G. Rufino, and M. Grassi, “LIDAR-inertial integration for UAV localization and mapping in complex environments,” submitted to The 2016 International Conference on Unmanned Aircraft Systems, Arlington, VA, USA, Jun. 2016.

# Acknowledgements

Here I would like to spend few words to sincerely thank all the people who accompanied me through this amazing experience.

First, I would like to thank my family and in particular my father and my mother who always supported my decisions and encouraged me during these years.

I would like to express immense gratitude to my supervisors Prof. Michele Grassi, Prof. Giancarmine Fasano and Prof. Giancarlo Rufino who constantly guided me and taught me how to be a good researcher. Thanks Michele for giving me the chance of working on such an interesting research topic, and, mostly, for being a wise mentor, and a nice and kind person. Thanks Giancarmine for being not only an inspiring teacher, who transmitted to me his knowledge, his enthusiasm and his passion about research, but also a loyal friend. Thanks Giancarlo for always giving me valuable suggestions and for constantly transmitting me the importance of patience, calm and precision.

I am also grateful to Prof. Antonio Moccia, Prof. Domenico Accardo, Prof. Alfredo Renga, Prof. Marco D'Errico and Prof. Urbano Tancredi. Each of you taught me something precious during my young academic career. Special thanks go to my colleagues (friends!) Antonio (Scanna), Amedeo (sunshine), Anna, Daniela, Flavia, Rita and Salvatore. I am happy and I have always felt honored and proud of being part of our Aerospace Systems Team.

I would like to thank Prof. Al Savvaris for giving me the opportunity to have a great experience at Cranfield University and for his guide during the research activity I carried out there. Thanks also to all the people of the Autonomous Systems and Control Group for the interesting discussions and for the support. A special mention goes to all the Italian friends I met and to our soccer team (Go Maccabi!).

I am thankful to Carlo Greco and Francesco Avallone for all the funny moments we had and for the great support we gave to each other, first as undergraduate, then as PhD students, and especially for sharing our daily road trip to the University.

Thanks to my precious lifelong friends Tommaso, Domenico, Carlone and Andrea. It does not count how far we go from each other, I have always been able to count on you.

Thanks to Felice and Elisa for making me feel like an older brother since the first day we met.

Last but most importantly, I would like to deeply thank my girlfriend Carla. Your simple presence has always given me the strength, the enthusiasm and the happiness I needed to overcome any difficulty. I really do not know how you could tolerate me during these years. Thanks for everything!