Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut Supérieur de l'Aéronautique et de l'Espace

Présentée et soutenue par :
Anthea COMELLINI

le jeudi 28 janvier 2021

Titre :

Rendez-vous autonomes basés vision avec cibles non coopératives

Vision-based navigation for autonomous rendezvous with non-cooperative targets

École doctorale et discipline ou spécialité :

ED AA : Automatique et robotique - Génie industriel et informatique

Unité de recherche :

Institut Clément Ader

Directeur(s) de Thèse :

Mme Christine ESPINOSA (directrice de thèse)
M. Emmanuel ZENOU (co-directeur de thèse)

Jury :

M. Frédéric LERASLE Professeur Université Paul Sabatier - Président
M. Nabil AOUF Professeur University of London - Rapporteur
M. Franco BERNELLI ZAZZERA Professeur Politecnico di Milano
M. Pierluigi DI LIZIA Professeur assistant Politecnico di Milano - Rapporteur
M. Vincent DUBANCHET Ingénieur Thales Alenia Space
M. Olivier DUBOIS-MATRA Ingénieur de recherche Agence spatiale européenne
Mme Christine ESPINOSA Professeure des Universités ISAE-SUPAERO - Directrice de thèse
M. Emmanuel ZENOU Professeur ISAE SUPAERO - Co-directeur de thèse

Abstract

The aim of this thesis is to propose a full vision-based solution to enable autonomous navigation of a chaser spacecraft (S/C) during close-proximity operations in space rendezvous (RDV) with a non-cooperative target using a visible monocular camera.

Autonomous rendezvous is a key capability to answer main challenges in space engineering, such as Active Debris Removal (ADR) and On-Orbit-Servicing (OOS). ADR aims at removing the space debris, in low-Earth-orbit protected region, that are more likely to lead to future collision and feed the Kessler syndrome, thus increasing the risk for operative spacecrafts. OOS includes inspection, maintenance, repair, assembly, refueling and life extension services to orbiting S/C or structures. During an autonomous RDV with a non-cooperative target, i.e., a target that does not assist the chaser in acquisition, tracking and rendezvous operations, the chaser must estimate the target's state on-board autonomously. Autonomous RDV operations require accurate, up-to-date measurements of the relative pose (i.e., position and attitude) of the target, and the combination of camera sensors with tracking algorithms can provide a cost effective solution.

The research has been divided into three main studies: the development of an algorithm enabling the initial pose acquisition (i.e., the determination of the pose without any prior knowledge of the pose of the target at the previous instants), the development of a recursive tracking algorithm (i.e., an algorithm which exploits the information about the state of the target at the previous instant to compute the pose update at the current instant), and the development of a navigation filter integrating the measurements coming from different sensor and/or algorithms, with different rates and delays.

For what concerns the pose acquisition phase, a novel detection algorithm has been developed to enable fast pose initialization. An approach is proposed to fully retrieve the object's pose using a set of invariants and geometric moments (i.e., global features) computed using the silhouette images of the target. Global features synthesize the content of the image in a vector of few descriptors which change values as a function of the target relative pose. A database of global features is pre-computed offline using the target geometrical model in order to cover all the solution space. At run-time, global features are computed on the current acquired image and compared with the database. Different sets of global features have been compared in order to select the more performing, resulting in a robust detection algorithm having a low computational load.

Once an initial estimate of the pose is acquired, a recursive tracking algorithm is initialized. The algorithm relies on the detection and matching of the observed silhouette contours with the 3D geometric model of the target, which is projected into the image frame using the estimated pose at the previous instant. Then, the summation of the distances between each projected model points and the matched image points is written as a non-linear function of the unknown pose parameters. The minimization of this cost function enables the estimation of the pose at the current instant. This algorithm provides fast and very accurate measurements of the relative pose of the target. However, as other recursive trackers, it is prone to divergence. Thus, the detection algorithm is run in parallel to the tacker in order to provide corrected measurements in case of tracker divergences.

The measurements are then integrated into the chaser navigation filter to provide an optimal and robust estimate. Vision-based navigation algorithms provide only pose measurements. However, some RDV operations require the synchronization of chaser motion with target motion, implying the need of knowing also target velocity and rotation rate. For this reason, the navigation function relies on a dynamic filter instead of a kinematic one. Moreover, vision-based measurements can be affected by high latency. Two delay management techniques suitable for the space RDV application have been applied to this problem. The selected methods are the Filter Recalculation method -which always provides an optimal estimation at the expense of a high computational load- and the Larsen's method -which provides a faster solution whose optimality lies on stronger requirements. The performance of the methods has been analyzed in order to allow the selection of the most suitable technique for the RDV problem.

The results of each study have been integrated in order to build a robust navigation solution, taking into account the constraint of the reduced computational resources available on typical space-qualified processors.

Résumé

L'objectif de cette thèse est de proposer une solution complète basée sur la vision pour permettre la navigation autonome d'un vaisseau de poursuite (S/C) lors d'opérations de proximité dans l'espace de rendez-vous (RDV) avec une cible non coopérative en utilisant une caméra monoculaire visible.

Le rendez-vous autonome est une capacité clé pour répondre aux principaux défis de l'ingénierie spatiale, tels que l'enlèvement actif des débris (ADR) et l'entretien en orbite (OOS). L'ADR vise à éliminer les débris spatiaux, dans les régions protégées en orbite basse, qui sont les plus susceptibles d'entraîner des collisions futures et d'alimenter le syndrome de Kessler, augmentant ainsi le risque pour les engins spatiaux opérationnels. L'OOS comprend des services d'inspection, d'entretien, de réparation, d'assemblage, de ravitaillement et de prolongation de la durée de vie des satellites ou structures en orbite. Lors d'un RDV autonome avec une cible non coopérative, c'est-à-dire une cible qui n'aide pas / n'interagit pas le chasseur dans les opérations d'acquisition, de poursuite et de rendez-vous, le chasseur doit estimer l'état de la cible à bord de manière autonome. Les opérations de rendez-vous autonomes nécessitent des mesures précises et actualisées de la pose relative (c'est-à-dire la position et l'attitude de la cible), et la combinaison de capteurs de caméra avec des algorithmes de poursuite peut constituer une solution rentable.

La recherche a été divisée en trois études principales : le développement d'un algorithme permettant l'acquisition de la pose initiale (c'est-à-dire la détermination de la pose sans aucune connaissance préalable de cette pose aux instants précédents), le développement d'un algorithme de poursuite récursif (c'est-à-dire d'un algorithme qui exploite les informations sur l'état de la cible à l'instant précédent pour calculer la mise à jour de la pose à l'instant actuel), et le développement d'un filtre de navigation intégrant les mesures provenant de différents capteurs et/ou algorithmes, avec différents taux et délais.

En ce qui concerne la phase d'acquisition de la pose, un nouvel algorithme de détection a été développé pour permettre une initialisation rapide de la pose. Une approche est proposée pour récupérer entièrement la pose de la cible en utilisant un ensemble d'invariants et de moments géométriques (c'est-à-dire des caractéristiques globales) calculés à partir des images de la silhouette de la cible. Les caractéristiques globales synthétisent le contenu de l'image dans un vecteur de quelques descripteurs qui changent de valeurs en fonction de la pose relative de la cible. Une base de données des caractéristiques globales est pré-calculée hors ligne en utilisant le modèle géométrique de la cible afin de couvrir tout l'espace de la solution. Au moment de l'exécution, les caractéristiques globales sont calculées sur l'image actuelle acquise et comparées avec la base de données. Différents

ensembles de caractéristiques globales ont été comparés afin de sélectionner les plus performants, ce qui a permis d'obtenir un algorithme de détection robuste avec une faible charge de calcul.

Une fois la première estimation de la pose effectuée, un algorithme de suivi récursif est initialisé. L'algorithme repose sur la détection et la correspondance des contours de la silhouette observée avec le modèle géométrique 3D de la cible, qui est projeté dans le cadre de l'image en utilisant la pose estimée à l'instant précédent. Ensuite, la somme des distances entre chaque point du modèle projeté et les points de l'image correspondants est écrite comme une fonction non linéaire des paramètres de pose inconnus. La minimisation de cette fonction de coût permet l'estimation de la pose à l'instant courant. Cet algorithme fournit des mesures rapides et très précises de la pose relative de la cible. Cependant, comme d'autres *trackers* récursifs, il est sujet à des divergences. Ainsi, l'algorithme de détection est exécuté en parallèle avec le *tracker* afin de fournir des mesures corrigées en cas de divergence.

Les mesures sont ensuite intégrées dans le filtre de navigation du chasseur pour fournir une estimation optimale et robuste. Les algorithmes de navigation basés vision ne fournissent que des estimations de pose. Cependant, certaines opérations de rendez-vous nécessitent la synchronisation du mouvement du chasseur avec le mouvement de la cible, ce qui implique la nécessité de connaître également la vitesse linéaire et la vitesse de rotation de la cible. Pour cette raison, la fonction de navigation repose sur un filtre dynamique plutôt que sur un filtre cinématique. De plus, les mesures basées vision peuvent être affectées par une latence élevée. Deux techniques de gestion des retards adaptées au rendez-vous spatial ont été appliquées à ce problème. Les méthodes choisies sont la méthode de recalcul du filtre - qui permet toujours une estimation optimale au détriment d'une charge de calcul élevée - et la méthode de Larsen - qui fournit une solution plus rapide et dont l'optimalité repose sur des exigences plus strictes. Les performances des méthodes ont été analysées afin de permettre la sélection de la technique la plus appropriée pour le problème de la RDV.

Les résultats de chaque étude ont été intégrés afin de construire une solution de navigation robuste, en tenant compte de la contrainte des ressources de calcul réduites disponibles sur les processeurs typiques qualifiés pour le spatial.

Table of Contents

List of Contributions

- Comellini, A., Casu, D., Zenou, E., Dubanchet, V., & Espinosa, C. (2020). Incorporating Delayed and Multirate Measurements in Navigation Filter for Autonomous Space Rendezvous. *Journal of Guidance, Control, and Dynamics*, Vol. 43(6), pp 1164-1172. `https://doi.org/10.2514/1.G005034`

- Comellini, A., Le Ny, J., Zenou, E., Espinosa, C. & Dubanchet, V. Global descriptors for visual pose estimation of a non-cooperative target in space rendezvous. 2nd round of review *IEEE Transaction on Aerospace and Electronic Systems.*

- Comellini, A., Zenou, E., Espinosa, C. & Dubanchet, V. Vision-based navigation for autonomous space rendezvous with non-cooperative targets. Accepted at the *11th International Conference on Information, Intelligence, Systems and Applications* 15–17 July, 2020.

- Comellini, A., Maye, F., Dubanchet, V., Casu, D., Zenou, E., & Espinosa, C. Robust navigation solution for vision-based autonomous rendezvous. *IEEE Aerospace Conference 2021.*

- Bettens A., Comellini A., Zenou E. & Dubanchet V. Pose estimation of a non-cooperative target based on silhouette imagery using Convolutional Neural Networks. Accepted at the *11^{t}h International ESA Conference on Guidance, Navigation & Control Systems*, 20-25 June 2021.

List of Acronyms and Abbreviations

ADR	Active Debris Removal
AOCS	Attitude and Orbit Control System
ASat	Anti Satellite
ATV	Automated Transfer Vehicle
BOL	Begin Of Life
CAD	Computer-Aided Drafting
CAM	Collision Avoidance Maneuver
CNES	Centre National d'Etudes Spatiales
CDEKF	Continuous Discrete Extended Kalman Filter
COPUOS	Committee on the Peaceful Uses of Outer Space
CPU	Central Processing Unit
CV	Computer Vision
CWH	Clohessy-Wiltshire-Hill
DARPA	DefenseAdvanced Research Project Agency
DEKF	Discrete Extended Kalman Filter
DISCOS	Database and Information System Characterising Objects in Space
DOF	Degree of Freedom
DLR	Deutsches zentrum für Luft- und Raumfahrt
EEE	Explicit Edge Extraction
EKF	Extended Kalman Filter
EOL	End Of Life
EROSS	European Robotic Orbital Support Services
ESA	European Space Agency
EVA	Extra Vehicular Activity
FD	Fourier Descriptors
FDIR	Failure Detection, Identification and Recovery
FOV	Field of View
FPGA	Field-Programmable Gate Array
GD	Global Descriptor
GEO	GEostationary Orbit
GM	Geometric Moment
GNC	Guidance Navigation & Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System

GSFC	Goddart Space Flight Center
HEO	High Elliptical Orbit
HST	Hubble Space Telescope
IADC	Inter-Agency Space Debris Coordination Committee
IP	Image Processing
IR	Infrared
ISRO	Indian Space Research Organisation
ISS	International Space Station
JAXA	Japan Aerospace eXploration Agency
KF	Kalman Filter
LIDAR	Light Imaging Detection And Ranging
LEO	Low Earth Orbit
LOF	Local Orbital Frame
LOS	Line Of Sight
LM	Levenberg-Marquardt
MDA	MacDonald, Dettwiler and Associates
MEO	Medium Earth Orbit
MVM	Mission and Vehicle Management
NAC	Narrow Angle camera
NASA	National Aeronautics and Space Administration
NASDA	National Space Development Agency of Japan
N-FRS	N-Fold Rotation Symmetry
NRO	Near Rectilinear Orbit
OOS	On-Orbit Servicing
OST	Outer Space Treaty
RAPiD	Real-time Attitude and Position Determination
RDV	Rendezvous
RF	Reference Frame
RK	Runge Kutta
RSO	Resident Space Object
S/C	Spacecraft
SDM	Space Debris Mitigation
SDR	Space Debris Remediation
SPDM	Special Purpose Dexterous Manipulator
SSN	Space Surveillance Network
SSRMS	Space Station Remote Manipulator System

TAS	Thales Alenia Space
TIR	Thermal Infra Red
TOF	Time-of-Flight
TRL	Technology Readiness Level
VIS	Visible
WAC	Wide Angle Camera
ZM	Zernike Moment

Chapter 0

Synthèse des Travaux (Français)

Contents

Le rendez-vous (RDV) autonome est une capacité clé pour répondre aux principaux défis de l'ingénierie spatiale, tels que l'enlèvement actif des débris (Active Debris Removal, ADR) et l'entretien en orbite (On-Orbit-Servicing, OOS). L'ADR vise à éliminer les débris spatiaux, dans les régions protégées en orbite basse, qui sont les plus susceptibles d'entraîner des collisions futures et d'alimenter le syndrome de Kessler, augmentant ainsi le risque pour les engins spatiaux opérationnels. L'OOS comprend des services d'inspection, d'entretien, de réparation, d'assemblage, de ravitaillement et de prolongation de la durée de vie des satellites ou structures en orbite. Lors d'un RDV autonome avec une cible non coopérative, c'est-à-dire une cible qui n'aide pas le chasseur dans les opérations d'acquisition, de poursuite et de rendez-vous, le chasseur doit estimer l'état de la cible à bord de manière autonome. Les opérations de rendez-vous autonomes nécessitent des mesures précises et actualisées de la pose relative (c'est-à-dire la position et l'attitude de la cible), et la combinaison de capteurs de caméra avec des algorithmes de traitement d'image (IP, Image Processing) et computer vision (CV) peut constituer une solution rentable.

L'objectif de cette thèse est de proposer une solution complète basée sur la vision pour permettre la navigation autonome d'un vaisseau chasseur lors d'opérations de proximité dans l'espace de rendez-vous avec une cible non coopérative en utilisant une caméra monoculaire visible.

La recherche a été divisée en trois études principales: le développement d'un algorithme permettant l'acquisition de la pose initiale (c'est-à-dire la détermination de la pose sans aucune connaissance préalable de cette pose aux instants précédents, Sec.0.3), le développement d'un algorithme de poursuite récursif (c'est-à-dire d'un algorithme qui exploite les informations sur l'état de la cible à l'instant précédent pour calculer la mise à jour de la pose à l'instant actuel, Sec.0.2), et le développement d'un filtre de navigation intégrant les mesures provenant de différents capteurs et/ou algorithmes, avec différents fréquences et délais, Sec.0.1.

Le travail est présenté en partant du filtre de navigation, ce qui permets de introduire aussi les complexes modèles dynamiques qui gouvernent l'environnent spatial.

0.1 Filtre de navigation

La fonction d'un filtre de navigation est d'intégrer les mesures issues des capteurs afin de fournir une estimation optimale et robuste. Les algorithmes de navigation basés vision ne fournissent que des estimations de pose. Cependant, certaines opérations de rendez-vous nécessitent la synchronisation du mouvement du chasseur avec le mouvement de la cible, ce qui implique la nécessité de connaître également la vitesse linéaire et la vitesse de rotation de la cible. Pour cette raison, la fonction de navigation repose sur un filtre dynamique plutôt que sur un filtre cinématique. De plus, les mesures basées vision peuvent être affectées par une latence élevée. Le filtre se trouve donc à devoir gérer deux types de mesures: des mesures dites *lentes*, qui deviennent disponibles pour être intégrées dans le filtre après un délai de traitement et qui ont normalement une fréquence moins élevée que celle du filtre, et des mesures dites *rapides*, qui ont une fréquence plus élevée et un délai de traitement presque négligeable. Deux techniques de gestion des retards adaptées au rendez-vous spatial ont été appliquées à ce problème. Les méthodes choisies sont la méthode de re-calcul du filtre (Filter Recalculation) - qui permet toujours une estimation optimale au détriment d'une charge de calcul élevée - et la méthode de Larsen - qui fournit une solution plus rapide et dont l'optimalité repose sur des exigences plus strictes. Les performances des méthodes ont été analysées afin de permettre la sélection de la technique la plus appropriée pour le problème de la RDV.

On introduit dans cette section la formulation théorique de ces deux techniques. La campagne de validation et la formalisation complète sont détaillés dans le Chapitre 2 de la thèse.

Dans un soucis de clarté, les méthodes sont appliqué à un système linéaire en temps-discret, mais elles seront aussi implémentées sur des modèles non linéaires en Sec.2.3. La représentation d'état d'un système linéaire discret observé par des mesures sans délais, avec bruit d'état w_k et bruit de mesure v_k (w_k et v_k non corrélés), est [Gibbs, 2011]:

$$\begin{cases} x_k = & A_k x_{k-1} + B_k u_k + w_k \\ y_k = & C_k x_k + v_k \end{cases}, \quad \text{with} \quad \mathbb{E}\left[w_k w_j^T\right] = \begin{cases} 0 & k \neq j \\ Q_k & k = j \end{cases}, \quad \mathbb{E}\left[v_k v_j^T\right] = \begin{cases} 0 & k \neq j \\ R_k & k = j \end{cases} \tag{1}$$

Le filtre de Kalman (KF) associé à ce système est brièvement présenté ci dessous (la formalisation complète se trouve en Appx.B.1.2). Le KF se divise en trois étapes:

- La prédiction de l'estimé *a priori* de l'état et de la matrice de covariance d'état ($\hat{x}_{k|k-1}, P_{k|k-1}$, Eq.(2)).
- Calcul du gain optimal K_k qui minimise l'estimé *a posteriori* de la matrice de covariance d'état $P_{k|k}$ (Eq.(3)).
- Mise à jour de l'état et de la matrice de covariance (Eq.(4)).

$$\text{prédiction} \quad \begin{cases} \hat{x}_{k|k-1} = & A_k \hat{x}_{k-1|k-1} + B_k u_k \\ P_{k|k-1} = & A_k P_{k-1|k-1} A_k^T + Q_k \end{cases} \tag{2}$$

$$\text{calcul du gain} \quad K_k = \; P_{k|k-1} C_k^T (C_k P_{k|k-1} C_k^T + R_k)^{-1} \tag{3}$$

$$\text{mise à jour} \quad \begin{cases} \hat{x}_{k|k} = & \hat{x}_{k|k-1} + K_k (y_k - C_k \hat{x}_{k|k-1}) \\ P_{k|k} = & (I - K_k C_k) P_{k|k-1} \end{cases} \tag{4}$$

Quand des mesures affectées par délai sont présentes, le système en Eq.(1) reçoit à l'instant k une mesure qui correspond à l'instant s ($s = k - N_d$, N_d nombre des pas de délai, comme montré en Fig.1), tell que:

$$y_s^* = C_s^* x_s + v_s, \quad \text{with} \quad \mathbb{E}\left[v_s v_j^T\right] = \begin{cases} 0 & s \neq j \\ R_s^* & s = j \end{cases} \tag{5}$$

Dans ce cas Eq.(3) n'est plus optimale et une nouvelle solution qui tient compte de la contribution de y_s^* doit être cherchée pour calculer les estimées optimaux $\hat{x}_{k|k,k^*}$ and $P_{k|k,k^*}$. Généralement, garantir l'optimalité de la solution demande un coût computationnel élévé (i.e., Filter Recalculation method). En raison de ça, des méthodes sub-optimales qui puissent réduire le coût computationnel sont proposées (i.e., Larsen's method).

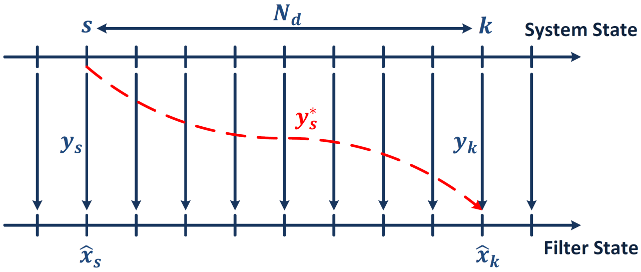


Figure 1: Système avec N_d pas de délais.

0.1.1 Méthode de re-calcul du filtre

La méthode de re-calcul du filtre consiste en retourner à l'instant de temps où la mesuré retardée a été acquise, incorporer cette mesure, et re-calculer la trajectoire entière de l'état jusqu'à l'instant actuel. En faisant ça, toute l'historique de l'estimé sera optimale. L'estimation est faite comme si deux filtres étaient utilisés. Il y a un filtre principal, qui fonctionne à fréquence constante et traite les mesurer sans délais y_k, et un deuxième filtre, qui est activé chaque fois une mesure affectée par un délai y_s^* arrive. Pour permettre au filtre secondaire de re-calculer un estimé optimal en fusionnant la mesure retardée, il est nécessaire que, chaque fois une mesure de ce type est acquise (i.e., quand la caméra capture une image), l'état prédit $\hat{x}_{s|s-1}$ ainsi que la covariance $P_{s|s-1}$ correspondant à cet instant soient stockés en mémoire. De plus, de façon à re-calculer toute l'historique de l'état, aussi les inputs u_{s+i} et les mesures rapides y_{s+i} doivent être enregistré pour tous les instants de temps entre l'échantillon s (i.e., quand l'image a été prise) et l'échantillon $s + N_d$ (quand la mesure devient disponible pour être intégrée dans le filtre), i.e., for $i \in [1, N_d - 1]$.

Les filtres fonctionnent de la manière suivante. A l'instant s, une mesure *lente* est acquise et $\hat{x}_{s|s-1}$, $P_{s|s-1}$ sont sauvegardés. Pour tous les instants $s + i$ ($i \in [1, N_d - 1]$) le filtre principale traite les mesures *rapides* y_{s+i} comme dans un filtre de Kalman classique (Eqs. (2),(3),(4)). Les mesures y_{s+i} et les inputs u_{s+i} sont enregistrés à chaque instant. A l'instant $k = s + N_d$ la mesure lente y_s^* et sa matrice de covariance R_s^* deviennent disponibles et le filtre secondaire est donc activé. Le filtre retourne à l'instant s et calcule l'update optimale en utilisant le vecteur mesure $\tilde{y}_s = [y_s, y_s^*]^T$, où y_s sont les mesures rapides et y^*_s sont les mesures lentes affectées par délai:

$$\begin{cases} \tilde{K}_s = & P_{s|s-1}\tilde{C}_s^T(\tilde{C}_s P_{s|s-1}\tilde{C}_s^T + \tilde{R}_s)^{-1} \\ \hat{x}_{s|s} = & \hat{x}_{s|s-1} + \tilde{K}_s(\tilde{y}_s - \tilde{C}_s\hat{x}_{s|s-1}) \\ P_{s|s} = & (I - \tilde{K}_s\tilde{C}_s)P_{s|s-1} \end{cases} \quad , \text{ with } \quad \tilde{C}_s = \begin{bmatrix} C_s \\ C_s^* \end{bmatrix} \quad , \quad \tilde{R}_s = \begin{bmatrix} R_s & \emptyset \\ \emptyset & R_s^* \end{bmatrix} \tag{6}$$

Les estimés optimales $\hat{x}_{s|s}$ and $P_{s|s}$ sont après propagés par le filtre secondaire de l'instant $s+1$ à l'instant $k = s+N_d$ selon les Eqs.(2),(3),(4). Cela clarifie le besoin de sauvegarder les valeurs des mesures rapides et des inputs. Une fois que la boucle est arrivée à l'instant $k = s+N_d$, le filtre a fourni l'estimé optimal de l'état et de la covariance d'état à l'instant actuel.

Dans le cas où il n'y a pas de mesures rapides, la structure du filtre est simplifiée: l'état et la matrice de covariance vont évoluer en boucle ouverte tant que aucune mesure n'arrive. Quand une mesure lente arrive, le filtre secondaire est activé. Il calcule l'estimé optimale à l'instant s, puis l'état et la matrice de covariance sont re-projetés à l'instant actuel k à travers N_d pas de prédiction.

Cette méthode peut devenir assez chère en terme de coût computationnel à cause du besoin de sauvegarder autant de variables, ainsi que à cause des N_d boucles de Kalman qui doivent être calculée chaque fois une mesure lente arrive. Toutefois, il s'agit de la seule formulation qui fournit un estimé optimal aussi en présence de mesures rapides et avec des systèmes non-linéaires. En outre, cette méthode est applicable à des cas avec plusieurs mesures affectées par des délais différents.

0.1.2 Méthode de Larsen

La méthode de Larsen (autrement dite méthode d'extrapolation) a été proposée en [Larsen et al., 1998] comme amélioration de la méthode de Alexander ([Alexander, 1991]), pour gérer les délais des mesures dans les systèmes linéaires discrets. La méthode consiste en calculer, tout au long le la période de délai, un terme de correction qui doit être ajouté à l'estimé dans l'instant de temps où la mesure devient disponible. La différence entre la méthode d'Alexandre et la méthode de Larsen est que cette dernière ne nécessite pas de connaître, à l'instant s, ni la covariance R^*_s, ni la matrice des mesures C^*_s relatives aux mesures lentes. Ces matrices sont censées devenir disponibles à l'instant k ensemble à la mesure retardée y^*_s. En raison de ça, la méthode de Larsen est appropriée pour des systèmes exploitants des mesures de type IP-CV, car les algorithmes qui fournissent ces mesures traitent la matrice de covariance R^* ensemble à la mesure y^*.

La structure du filtre est la suivante. Comme pour la méthode de re-calcul du filtre, quand une nouvelle mesure lente est acquise à l'instant s, $\hat{x}_{s|s-1}$ et $P_{s|s-1}$ sont stockés. A chaque instant $s+i$ ($i \in [1, N_d - 1]$), la structure classique du KF en Eqs. (2),(3),(4) est appliquée au système en exploitant les mesures rapides. En outre, le terme $M_{s+i} = (I - K_{s+i}C_{s+i})A_{s+i}M_{s+i-1}$ est calculé, où $M_s = I$. A l'instant $k = s + N_d$, y^*_s, R^*_s et C^*_s deviennent disponibles. Le filtre calcule le gain K_k et la mise à jour en accord avec les Eqs.(3),(4) en utilisant la mesure rapide y_k. Le terme de correction finale M^*_k sera donc [Larsen et al., 1998]:

$$M^*_k = M_{s+N_d} = \prod_{i=1}^{N_d}(I - K_{s+i}C_{s+i})A_{s+i} \tag{7}$$

A ce point, la mesure *extrapolée* y^{ext}_k est calculée, de façon à obtenir une représentation de la mesure y^*_s -qui corresponds à l'instant s- à l'instant k:

$$y^{ext}_k = y^*_s - C^*_s \hat{x}_{s|s-1} + C^*_k \hat{x}_{k|k-1} \tag{8}$$

Larsen en [Larsen et al., 1998] démontre comment calculer le gain optimal K^*_k et la mise à jour de l'état correspondant:

$$\begin{cases} K^*_k & = M^*_k P_{s|s-1} C^{*T}_s (C^*_s P_{s|s-1} C^{*T}_s + R^*_s)^{-1} \\ \hat{x}_{k|k,k^*} & = \hat{x}_{k|k} + K^*_k (y^{ext}_k - C^*_k \hat{x}_{k|k}) \\ P_{k|k,k^*} & = P_{k|k} - K^*_k C^*_s P_{k|k} M^{*T}_k \end{cases} \tag{9}$$

où le gain K^*_k est le gain K^*_s (i.e., le gain que aurait été calculé si la mesure y^*_s avait été utilisé à l'instant s) pre-multiplié par le terme de correction de Larsen M^*_k.
En présence de mesures rapides la méthode de Larsen performe de manière sub-optimale: à chaque pas de temps le gain K_{s+i} est calculé en utilisant une matrice de covariance $P_{s+i|s+i-1}$ qui n'est pas optimal car elle ne tient pas compte de la contribution de la mesure lente y^*_s. De toute façon, même si sous-optimale, la méthode requiert simplement deux multiplication de matrices et le stockage de deux variables aux instants où une mesure lente est acquise, sans le besoin de enregistrer les mesures rapides et les inputs. De plus, comme la méthode de re-calcul du filtre, la méthode de Larsen peut être appliqué aussi à de cas avec délais variable. En Sec.2.4 nous présentons une campagne de test qui démontre que la diminution de la performance due à l'approximation introduite par cette méthode est tout à fait acceptable.

0.2 Algorithme de suivi de pose

Dans les application spatiales, les méthodes qui s'appuyant sur la détection des features peuvent ne pas être assez robustes à cause des conditions d'illumination difficiles en environnent spatial, ainsi que à cause des textures particuliers des satellites [Lichter and

Dubowsky, 2004]. D'autre part, il est raisonnable supposer que l'on dispose du modèle géométrique 3D du satellite cible même si ce dernier n'avait pas été conçu pour participer à des rendezvous. Cela nous autorise à chercher une solution de navigation parmi les méthodes *basée modèle*. Cette famille d'algorithmes cherche la pose de la cible qui mieux superpose le modèle 3D a priori avec le satellite dans l'image acquise. Dans le contexte de la vision monoculaire, les techniques qui s'appuient sur l'extraction des arêtes sont très adaptes aux applications spatiales, car les arêtes sont des éléments proéminents facilement détectables en correspondance des gradients d'image élevés.

Un algorithme basé modèle qui s'appuie sur la détection et le suivi des aretes est l'algorithme RAPiD [Harris and Stennett, 1990]. A l'instant k, le modèle 3D est projeté dans le repère image en utilisant la pose estimée à l'instant précèdent ($k-1$). Un algorithme (i.e., algorithme de masquage ou masking) détermine les arêtes du modèle qui sont visibles, et il les échantillonne (voir Appx.D.1) en déterminant un set de *points de contrôle*. Ces points de contrôle sont les points qui vont être "matchés" avec les contours détectes dans l'image prise à l'instant k (en utilisant le filtre de Canny). La correspondance entre un point de contrôle et le point correspondant qui appartient aux contours de l'image se fait en cherchant le long du vecteur normal à l'arête qui contient le point de contrôle (vori Appx.D.2, algotihme de matching). Cette recherche mono-directionnel permet de réduire la dimension de la recherche et de augmenter la vitesse d'exécution de l'algorithme. Pour calculer la nouvelle pose, RAPiD s'appuie sur l'hypothèse que, au premier ordre, des petits changements dans la pose du modèle causent un déplacement des points de contrôle qui est linéaire par rapport aux paramètres de pose. Cela permet de calculer la nouvelle pose en résolvant un problème aux moindres carrés.

Pendant un rendez-vous avec une cible non-coopérative comme un débris spatial qui peut être en fort rotation, la variation de la pose entre deux instants consécutifs peut être trop grande pour permettre la linéarisation du problème, et l'algorithme RAPiD diverge rapidement. Dans cette thèse, nous proposons l'utilisation d'une fonction de coût non-linéaire qui est minimisée en utilisant l'algorithme le Levenberg-Marquardt (LM). De plus, on propose de utiliser comme points de contrôle seulement les points appartenant au contour externe de la silhouette de la cible. Cet approche permet de réduire le nombre de faux correspondances avec les contours de l'image (voir Sec.3.3.3), et est beaucoup plus rapide. Dans la suite on présente d'abord le problème du suivi d'objet avec vision monoculaire (Sec.0.2.1), et on détaille après la construction de la fonction de coût utilisée dans le suivi de pose (Sec.0.2.2). Finalement, en Sec.0.2.3, nous montrons comment intégrer l'algorithme de suivi de pose au filtre de navigation proposée en Sec.0.1. Plus de détails sur l'algorithme proposée et sur ces performances sont fournis dans le Chapitre 3.

0.2.1 Estimation de pose avec vision monoculaire

L'image en Fig.2 propose une représentation schématique du problème de l'estimation de pose avec vision monoculaire. Nous appelons $X_i^{tg} = [x_i^{tg}, y_i^{tg}, z_i^{tg}]$ les coordonnées exprimées en repère cible (i.e., target, pour simplicité centré au centre de masse COM de la cible) d'un point X_i qui appartient à la cible. Les cordonnées du point X_i exprimés en repère caméra (cam) sont:

$$X_i^{cam} = tr_{cam-tg} + R_{cam-tg} X_i^{tg} \tag{10}$$

Le vecteur $tr_{cam-tg} = [tr^{cam}_{cam-tgt_x}, tr^{cam}_{cam-tgt_y}, tr^{cam}_{cam-tgt_z}]$ est le vecteur de translation qui décrit la position relative de l'origine du repère target (O_{tg}, i.e., le COM de la cible), par rapport à l'origine du repère caméra (O_{cam}), exprimée en repère caméra. Nous allons appeler ce vecteur $tr = [tr_x, tr_y, tr_z]$ par souci de simplicité. La position relative du repère caméra par rapport au repère chasseur est connue par calibration au sol. La

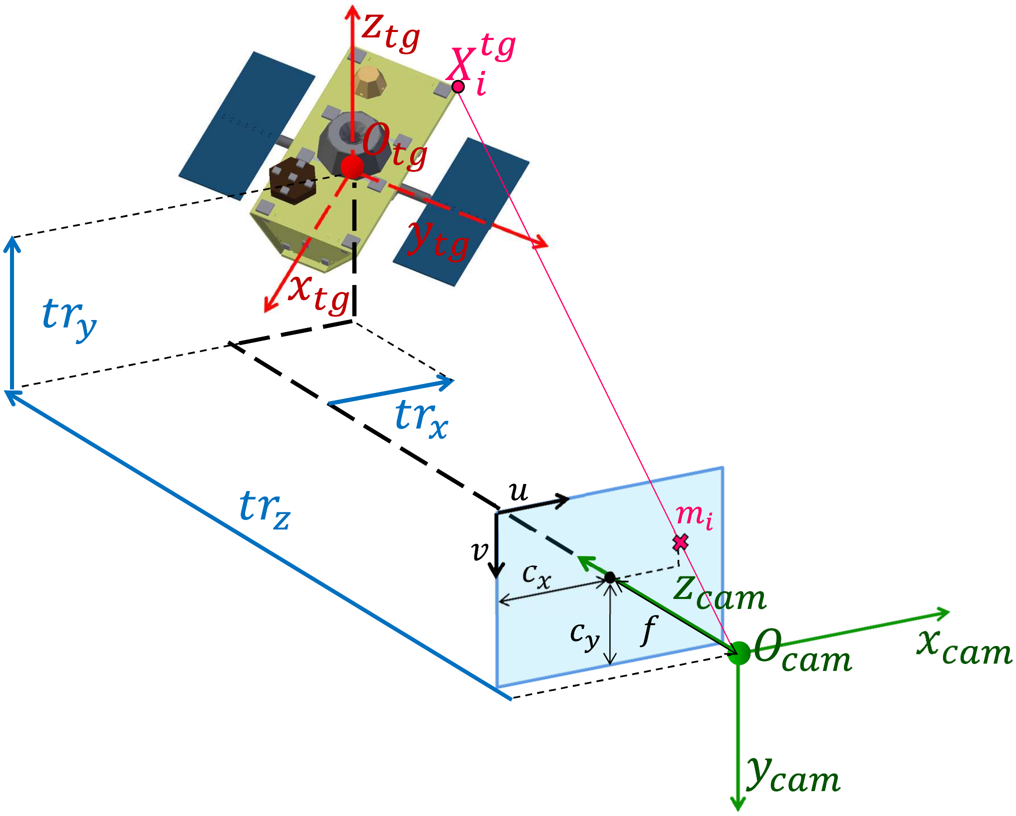


Figure 2: Représentation schématique du problème de l'estimation de pose avec vision monoculaire. $f = f_x = f_y$ longueur focale, (c_x, c_y) point principale de image.

matrice R_{cam-tg} est la matrice de rotation qui décrit l'attitude relative caméra-target, qui peut aussi être représentée avec le quaternion correspondant q_{cam-tg} (voir Appx A.1). Dans ce cas, Eq.(10) devient:

$$X_i^{cam} \quad = tr_{cam-tg} + q_{cam-tg} \otimes X_i^{tg} \otimes q^*_{cam-tg} \tag{11}$$

Pour simplicité, on va renommer R_{cam-tg} en R, et q_{cam-tg} en q. En rappelant le modèle de caméra sténopé (i.e., *pinhole camera*) ayant comme matrice de projection la matrice K telle que:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

et en utilisant la notation introduite en Appx.A.2, où l'on utilise $Q_{rot_j}(X_i^{tg}, q)$ (avec $j = 1:3$) pour indiquer la première, la deuxième et la troisième composantes du vecteur $Q_{rot}(X_i^{tg}, q) = q \otimes X_i^{tg} \otimes q^*$, on obtient les cordonnées en repère image (i.e., le plan uv en Fig.2) du point X_i^{tg}:

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} f_x \dfrac{tr_x + Q_{rot_1}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(X_i^{tg}, q)} + c_x \\ f_y \dfrac{tr_y + Q_{rot_2}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(X_i^{tg}, q)} + c_y \end{bmatrix} \tag{13}$$

0.2.2 Optimisation non linéaire: construction de la fonction de coût

Comme anticipé, pendant un RDV avec une cible non coopérative, la variation de la pose de la cible entre deux prises d'image consécutives peut être trop grande pour permettre la linearization du déplacement. Tout à fait, un débris peut être en rotation à une vitesse qui peut dépasser les 6 deg/s [Bonnal et al., 2013]. Dans cette section nous présentons la formulation analytique de notre méthode d'estimation qui est basée sur l'optimisation d'une fonction non linéaire. Supposons d'avoir une fonction non-linéaire d'ajustement $\hat{y}(p)$ d'un vecteur de N paramètres p, et un ensemble de M points y_i. Les paramètres peuvent être estimés en minimisant la somme des carrés pondérés des erreurs entre les données mesurées y_i et la fonction d'ajustement $\hat{y}_i(p)$. La fonction de coût résultant est:

$$\chi^2(p) \quad = \sum_i^M (y_i - \hat{y}_i(p))^2 = (Y - \hat{Y}(p))^T W (Y - \hat{Y}(p)) \tag{14}$$

où la matrice de poids W correspond à diag$(1/\sigma_i^2)$, avec σ_i l'écart type de l'erreur associée à la mesure y_i. Le set de données mesurées Y est le vecteur composé par les projections des points $[u_i, v_i]^T$ (i.e., les cordonnées mesurées en repère image des points matchés avec les points de contrôle X_i^{tg}) le long de la normale à l'arête contenant les points de contrôle $n_i = [n_{u_i}, n_{v_i}]^T$.

$$y_i = n_{u_i} u_i + n_{v_i} v_i, \quad Y = [y_1, y_2, ..., y_i, ..., y_M]^T \tag{15}$$

La matrice W est définie comme la matrice identité de dimension $M \times M$, car il n'y a pas de moyen de connaître l'erreur associée aux correspondances entre les points de contrôle et les points image $[u_i,\ v_i]$.

Le paramètre de pose p à estimer est la pose complète à l'instant k. Dans le paramètre de pose p, les rotations sont représentés en utilisant le quaternion $q = q_{cam-tg} = [q_0,\ q_1,\ q_2,\ q_3]^T$, car le quaternion a la dimensionnalité plus petite pour une représentation non singulière du group de rotation $SO(3)$ [Markley, 2004]. Le vecteur de paramètres p est donc le vecteur de taille 7×1 égal à $[tr_x,\ tr_y,\ tr_z,\ q_0,\ q_1,\ q_2,\ q_3]$. En rappelant Eq.(13), la fonction d'ajustement devient:

$$\hat{y}_i \ = n_i^T \begin{bmatrix} \hat{u}_i(X_i^{tg}, p) \\ \hat{v}_i(X_i^{tg}, p) \end{bmatrix} = n_{u_i} \left(f_x \frac{tr_x + Q_{rot_1}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(X_i^{tg}, q)} + c_x \right) + n_{v_i} \left(f_y \frac{tr_x + Q_{rot_2}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(X_i^{tg}, q)} + c_y \right) \tag{16}$$

et $\hat{Y}(p) = [\hat{y}_1(p), \hat{y}_2(p), ..., \hat{y}_i(p), ..., \hat{y}_M(p)]^T$. L'optimisation non linéaire est résolue en utilisant l'algorithme de Levenberg-Marquardt (LM) décrite par [Gavin, 2011] et détaillée en Appendix D.4. La matrice jacobienne de la fonction $\hat{Y}(p)$ doit être déterminée afin de calculer à chaque pas l'incrément du paramètre p. Pour chaque point du modèle X_i^{tg}, qui correspond au point mesuré y_i et à la fonction d'ajustement $\hat{y}_i(X_i^{tg}, p)$, la matrice jacobienne $\mathbb{J}_i$ de taille 1×7 est:

$$\mathbb{J}_i = \frac{\partial \hat{y}_i}{\partial p} = \begin{bmatrix} n_{u_i} \dfrac{\partial \hat{u}_i}{\partial tr_x} + n_{v_i} \dfrac{\partial \hat{v}_i}{\partial tr_x}, & \dots, & n_{u_i} \dfrac{\partial \hat{u}_i}{\partial q_3} + n_{v_i} \dfrac{\partial \hat{v}_i}{\partial q_3} \end{bmatrix} \tag{17}$$

Le calcul des éléments de la matrice jacobienne correspondants aux termes de translation est simple, lorsque le calcul des éléments qui correspond aux termes de rotation est plus complexe. En Appendice A.2.3 nous présentons la dérivation de la matrice jacobienne analytique pour la fonction représentant la rotation d'un vecteur $X_i^{tg} = [x_i, y_i, z_i]$ par rapport à un quaternion q. La matrice jacobienne résultante est:

$$\frac{\partial (q \otimes X_i^{tg} \otimes q^*)}{\partial q} = \begin{bmatrix} \dfrac{\partial Q_{rot_1}(X_i^{tg}, q)}{\partial q} \\ \dfrac{\partial Q_{rot_2}(X_i^{tg}, q)}{\partial q} \\ \dfrac{\partial Q_{rot_3}(X_i^{tg}, q)}{\partial q} \end{bmatrix} = \begin{bmatrix} A & D & C & -B \\ B & -C & D & A \\ C & B & -A & D \end{bmatrix} \tag{18}$$

où:

$$\begin{aligned}
A = \ & \frac{\partial Q_{rot_1}}{\partial q_0} = \frac{\partial Q_{rot_2}}{\partial q_3} = -\frac{\partial Q_{rot_3}}{\partial q_2} = && 2(q_0 x_i - q_3 y_i + q_2 z_i) \\
B = \ & \frac{\partial Q_{rot_2}}{\partial q_0} = \frac{\partial Q_{rot_3}}{\partial q_1} = -\frac{\partial Q_{rot_1}}{\partial q_3} = && 2(q_3 x_i + q_0 y_i - q_1 z_i) \\
C = \ & \frac{\partial Q_{rot_1}}{\partial q_2} = \frac{\partial Q_{rot_3}}{\partial q_0} = -\frac{\partial Q_{rot_2}}{\partial q_1} = && 2(-q_2 x_i + q_1 y_i + q_0 z_i) \\
D = \ & \frac{\partial Q_{rot_1}}{\partial q_1} = \frac{\partial Q_{rot_2}}{\partial q_2} = \ \frac{\partial Q_{rot_3}}{\partial q_3} = && 2(q_1 x_i + q_2 y_i + q_3 z_i)
\end{aligned} \tag{19}$$

Les éléments de la matrice jacobienne en Eq.(17), pour la composante $\hat{u}_i$ et pour la composante $\hat{v}_i$, sont:

$$\begin{cases}
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial tr_x} = f_x \dfrac{1}{tr_z + Q_{rot_3}(X_i^{tg}, q)} \\
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial tr_y} = 0 \\
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial tr_z} = -f_x \dfrac{tr_x + Q_{rot_1}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(\Big(X_i^{tg}, q)\Big)^2} \\
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial q_0} = f_x \dfrac{A\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) - C\left(tr_x + Q_{rot_1}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2} \\
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial q_1} = f_x \dfrac{D\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) - B\left(tr_x + Q_{rot_1}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2} \\
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial q_2} = f_x \dfrac{C\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) + A\left(tr_x + Q_{rot_1}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2} \\
\dfrac{\partial \hat{u}_i(X_i^{tg}, p)}{\partial q_3} = f_x \dfrac{-B\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) - D\left(tr_x + Q_{rot_1}(X_i^{tg},)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2}
\end{cases} \tag{20}$$

$$\begin{cases}
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial tr_x} = 0 \\
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial tr_y} = f_y \dfrac{1}{tr_z + Q_{rot_3}(X_i^{tg}, q)} \\
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial tr_z} = -f_y \dfrac{tr_y + Q_{rot_2}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(\Big(X_i^{tg}, q)\Big)^2} \\
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial q_0} = f_y \dfrac{B\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) - C\left(tr_y + Q_{rot_2}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2} \\
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial q_1} = f_y \dfrac{-C\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) - B\left(tr_y + Q_{rot_2}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2} \\
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial q_2} = f_y \dfrac{D\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) + A\left(tr_y + Q_{rot_2}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2} \\
\dfrac{\partial \hat{v}_i(X_i^{tg}, p)}{\partial q_3} = f_y \dfrac{A\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right) - D\left(tr_y + Q_{rot_2}(X_i^{tg}, q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg}, q)\right)^2}
\end{cases} \tag{21}$$

La matrice obtenue est utilisée à chaque itération pour calculer la direction de l'incrément de pose, en partant de l'hypothèse initiale p_0 qui est égale à la pose estimé à l'instant $k-1$. Dans le processus d'estimation, quelque approximation est introduite: à chaque mise à jour du paramètre dans l'algorithme de LM, un nouvel set de points de contrôle devrait être calculé en utilisant l'algorithme de masquage, et des nouvelles correspondances avec les points image devraient être calculées en utilisant l'algorithme de matching. Cependant, ce type de approche augmenterait excessivement le coût computationnel de l'algorithme. En raison de ça, le set de points de contrôle, de points matchés et de vecteurs normaux sont gardés constant dans chaque appel à l'algorithme LM. La méthode d'estimation montre une très bonne performance même en présence de cette approximation, comme montré en Sec.3.4.

0.2.3 Calcul de la matrice de covariance des mesures R

L'un des problèmes principaux des algorithmes de IP-CV c'est de obtenir une bonne caractérisation du bruit associé aux mesures (i.e., de la matrice de covariance du bruit des mesures R). Cette matrice est nécessaire pour intégrer les mesures dans le filtre de Kalman de façon optimale. Toutefois il peut s'avérer très compliqué d'avoir une bonne caractérisation de R car le bruit des mesures dépend de plusieurs facteurs, tels que le bruit intrinsèque du capteur, la distance relative caméra-cible, la vitesse de rotation et translation, la fréquence de capture d'image, les conditions d'illumination, et même la pose relative. Quand on utilise l'algorithme de LM pour estimer un paramètre (dans notre cas, le paramètre p_k, qui correspond -du point de vue du KF- à la pose mesuré à l'instant k), la covariance du paramètre estimé p_k peut être calculée en utilisant la matrice jacobienne et la matrice de poids, selon la formule introduite en [Gavin, 2011]:

$$R = [\mathbb{J}^T W \mathbb{J}]^{-1} . \tag{22}$$

Cependant, comme anticipé, les valeurs de la matrice de poids W ne sont pas connues. En raison de ça, il devient nécessaire d'identifier un nouvel critère pour caractériser la mesure. On a donc décidé d'utiliser comme indicateur du niveau de confiance de la mesure la valeur de la fonction χ^2 évaluée en correspondance du paramètre estimé p_k, divisée par le nombre de dégrées de liberté du problème (i.e., $\nu = M - N + 1$). Cette valeur est nommée *reduced cost function* (fonction de coût réduite) et est aussi un parmi les critères de convergence de l'algorithme de LM.

$$r = \frac{\chi^2(p_k)}{\nu} \tag{23}$$

La valeur de r peut être vue comme une moyenne carrée des erreurs de reprojection, et elle est appelée *résiduel* dans les chapitres suivants. Pour une même géométrie de la cible, pour des paramètres de caméra donnés, et pour une distance relative caméra-cible, le résiduel peut être corrélé avec la matrice de covariance du bruit des mesures (i.e., un résiduel très petit indique que l'erreur de reprojection est très petit et que donc la mesure est très fiable). Cette corrélation est obtenue expérimentalement lors d'une calibration au sol.

0.3 Algorithme de détection de pose

Les algorithmes de suivi récursifs, comme celui décrit en Sec.0.2, peuvent fournir des mesures très précises, mais ils sont aussi sujets à des divergences en présence de minima locaux (voir Sec.3.4.2). En raison de ça, un algorithme de suivi doit être accompagné par une méthode d'estimation de pose par détection, qui puisse permettre l'initialisation de pose et la détection d'erreurs dans l'algorithme de suivi. Dans un algorithme de *détection*, la pose de la cible est calculée sans utiliser aucune information a priori concernant la pose aux instants précèdents. Un nouvel algorithme de détection a été développé pour permettre une initialisation rapide de la pose. Une approche a été proposée pour récupérer entièrement la pose de la cible en utilisant un ensemble d'invariants et de moments géométriques (c'est-à-dire des caractéristiques globales, *global features*) calculés à partir des images de la silhouette de la cible. Les caractéristiques globales synthétisent le contenu de l'image dans un vecteur de quelques descripteurs qui changent de valeurs en fonction de la pose relative de la cible. Une base de données des caractéristiques globales est pré-calculée hors ligne en utilisant le modèle géométrique de la cible afin de couvrir tout l'espace de la solution. Au moment de l'exécution, les caractéristiques globales sont calculées sur l'image actuelle acquise et comparées avec la base de données. Différents ensembles de caractéristiques globales ont été comparés afin de sélectionner les plus performants, ce qui a permis d'obtenir un algorithme de détection robuste avec une faible charge de calcul. On décrit brièvement dans cette section les principes de la méthode, qui sont détaillés sans le Chapitre 4.

L'intérêt d'utiliser des global features comme les descripteurs de Fourier [Wallace and Wintz, 1980, Reeves et al., 1988, Chen and Ho, 1991, Glais and Ayoun, 1994] ou les moments d'image [Dudani et al., 1977, Reeves et al., 1988, Glais and Ayoun, 1994, Breuers, 1999] est que ces features peuvent être construites de façon à être invariantes à la translation, au facteur d'échelle, et surtout à la rotation. Si une forme bidimensionnelle est décrite par ces invariantes, la valeur de la feature ne dépend pas ni de la position du centroïde de la forme (i.e., invariance à la translation), ni de la dimension de la forme (i.e., invariance au facteur d'échelle), ni de la rotation de la forme dans le plan image (i.e., invariance à la rotation). La forme de la silhouette projetée dépend de l'attitude relative et de la position relative de l'objet observé, et la contribution de l'attitude est couplée à la contribution de la position. Cependant, sous l'hypothèse de *weak perspective model* (i.e., modèle de prospective faible) les effets peuvent être découplés au prix d'une acceptable dégradation de la performance (voir Sec.4.5). L'hypothèse de prospective faible est valable quand la profondeur de l'objet le long de l'axe optique est beaucoup plus faible que la distance entre l'objet e la caméra, [Petit, 2013], ou quand le *champ de vue* (field of view, FOV) est relativement petit ou l'objet est placé près du centre du FOV (i.e., $tr^{cam}_{cam-tg_x} \sim 0$ and $tr^{cam}_{cam-tg_y} \sim 0$ en Eq.(10)). Ceci sont des conditions qui se vérifient pendant la phase d'acquisition de pose. Le modèle de perspective faible assume que tous les points de l'objet 3D sont à la même distance z depuis la caméra, sans que des erreurs significatives soient introduit par rapport au modèle complète de caméra sténopé. Sous ces hypothèses, on peut supposer que la distance $tr^{cam}_{cam-tg_z}$ affecte seulement l'échelle de la silhouette projetée, lorsque les composants $tr^{cam}_{cam-tg_x}$ et $tr^{cam}_{cam-tg_y}$ affectent juste la position du centroïde de la silhouette. De cette façon, si des caractéristiques globales invariantes sont utilisées pour décrire la silhouette du S/C à une pose donnée, la valeur de la feature dépendra seulement de l'angle de roulis φ et de l'angle de tangage ϑ de la matrice R_{cam-tg} (Eq.(10)). Au fait, l'angle de lacet ψ affecte seulement la rotation de la forme projetée dans le plan image, comme montré en Fig.3.

Les principes de la méthode proposée sont les suivants. Pendant un processus hors-ligne, un set de vues synthétiques de la cible (appelées images de *training*) est généré pour un nombre discrète de valeur de $(\varphi, \vartheta) \in]-\pi, \pi] \times]-\pi/2, \pi/2]$. Quand on génère la base de données, l'angle de lacet $\psi = \psi_{train}$ est fixé à la valeur de zéro. De plus, la caméra pointe parfaitement vers le COM de la cible, i.e., $tr^{cam}_{cam-tg_y} = tr^{cam}_{cam-tg_y} = 0$, et la distance $tr^{cam}_{cam-tg_z} = d$ est constante pour toute les vues de la base de données. Cette configuration est montrée en Fig.3. Le choix de la distance $d = d_{train}$ peut avoir une influence sur la performance de l'estimation, comme il sera montré en Sec.4.5.2. Pour un couple donné de (φ, ϑ), la position de la caméra, exprimée en repère cible, est $tr^{tg}_{cam-tg} = d_{train} \cdot [\sin\vartheta, \ -\cos\vartheta \sin\varphi, \ -\cos\vartheta \cos\varphi]^T$. Sous l'hypothèse que l'axe optique de la caméra pointe vers le COM de la cible, le lieux des points $tr^{tg}_{cam-tg}(d_{train}, \varphi, \vartheta)$ est la sphère de radius d_{train} centrée dans le COM de la cible. Ainsi, les attitudes relatives utilisées pour générer les images de training peuvent être assignées en sélectionnant N_w points aléatoires sur la sphère. Pour éviter le sur-échantillonage de la zone polaire, qui est obtenu en utilisant un échantillonnage uniforme en φ and ϑ, les points sont assignés

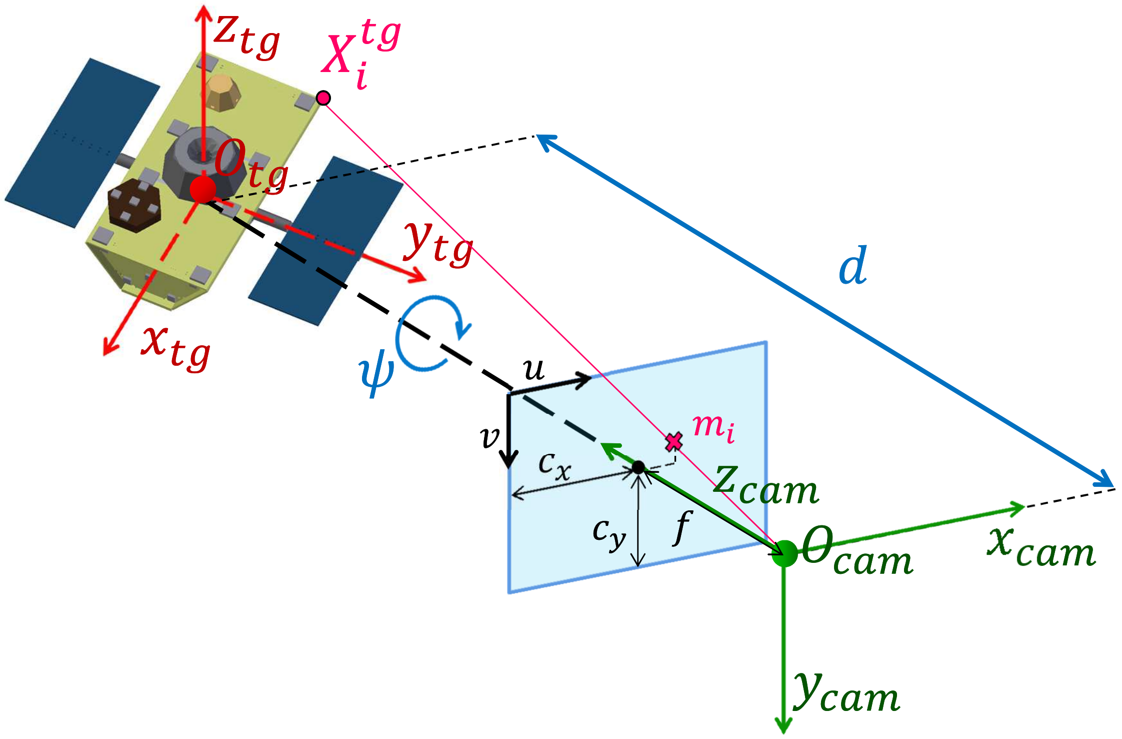


Figure 3: Représentation schématique du problème de l'estimation de pose avec vision monoculaire avec $tr^{cam}_{cam-tg_y} = tr^{cam}_{cam-tg_y} = 0$.

avec l'algorithme décrit en [Kuffner, 2004]

$$\begin{cases} \varphi &= 2\pi \, \mathrm{rand}_{\varphi} - \pi \\ \vartheta &= \mathrm{asin}(1 - 2\mathrm{rand}_{\vartheta}) \end{cases} \tag{24}$$

où rand_{φ} and $\mathrm{rand}_{\vartheta}$ sont deux variables aléatoires indépendantes distribuées uniformément dans l'intervalle $]0,\ 1]$. Il faut noter que, même si un échantillonage aléatoire est conseillé pour faire le benchmark des différents descripteurs globaux, un échantillonage déterministe est préférable pour la solution de navigation nominale. Dans ce cas, le schéma à spirale peut être utilisé pour créer l'échantillonnage uniforme sur la sphère [Koay, 2011].
Une fois que les images correspondantes à les N_w vues, les descripteurs sont calculés pour chaque vue. La dimension du database correspondant est $N_w \times N_f$, avec N_f la dimension du vecteur descripteur. Pendant l'exécution de l'algorithme, quand la caméra acquiert une nouvelle image (appelée ici image de *test*), l'algorithme calcule le vecteur descripteur pour l'image de test et cherche dans la base de donnée le couple de $(\varphi_{meas}, \vartheta_{meas})$ ayant le plus proche vecteur descripteur (en minimisant la distance Euclidienne). Les deux

dégrées de liberté restant, i.e., l'angle de lacet ψ et la distance relative camera-target d, sont estimée en utilisant les moments géométriques (GMs), selon la procédure détaillée ci-dessous. Du point de vue mathématique, les moments sont des projections d'une fonction sur une base polynomiale [Flusser et al., 2009]. Les GMs s'appuient sur l'intuitive base de puissances $p_{pq}(x, y) = x^p y^q$. La définition générale du moment géométrique d'une image est:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y)\, \mathrm{d}x\, \mathrm{d}y\,, \tag{25}$$

où$f(x, y)$ est la fonction égale à l'intensité du pixel ayant coordonnées (x, y). Quand on utilise une image binaire comme une silhouette, $f(x, y)$ est égale à 1 dans la silhouette et à 0 ailleurs. Pour une image binaire, les moments d'ordre inférieure sont associés à des propriétés géométrique de la silhouette observée. Par exemple, m_{00} est la surface, lorsque $m_{10}/m_{00} = x_c$ et $m_{01}/m_{00} = y_c$ sont les coordonnées x et y du centroïde de la silhouette. Les GMs peuvent etre rendus invariant à la translation et au facteur d'échelle très facilement. L'invariance à la translation est obtenue en calculant les moments géométriques *centraux* μ_{pq}, selon:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y)\, \mathrm{d}x\, \mathrm{d}y\,. \tag{26}$$

L'invariance au facteur d'échelle est obtenue en divisant les moments centraux par une puissance du moment d'ordre zéro:

$$\nu_{pq} = \frac{\mu_{pq}}{m_{00}^w}, \quad \text{with} \qquad w = \frac{p + q}{2} + 1\,. \tag{27}$$

Pour une image donnée, l'angle de lacet ψ peut etre calculé en utilisant les moments centraux de deuxième ordre. Il est connu dans des domaines comme l'ingénierie des structures que il est possible de calculer les axes principaux d'inertie d'une section plane avec ses moment du deuxième ordre μ_{20}, μ_{11}, μ_{02}. Pour une forme bidimensionnelle, il est possible de exprimer les moments centraux μ'_{pq} après une rotation de α dans le plan image en utilisant les valeurs des moments μ_{pq} avant la rotation:

$$\mu'_{pq} = \sum_{l_1=0}^{p} \sum_{l_2=0}^{q} r_{11}^{l_1} r_{21}^{l_2} r_{12}^{p-l_1} r_{11}^{q-l_2} \mu_{l_1+l_2, p+q-(l_1+l_2)} \tag{28}$$

où r_{11}, r_{12}, r_{21}, r_{22} sont les éléments de la matrice de rotation qui décrit la rotation dans le plan image:

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{29}$$

Vue cette formula, il est possible de calculer l'inclination $\tilde{\psi}_0$ de l'axe majeur d'inertie de la forme observée selon la procédure décrite ci dessous. Si on applique une rotation de

$\tilde{\psi}_0$ au repère image, et on calcule les moments centraux après rotation, on obtiendra les moments principaux de la section. Les moments de deuxième ordre peuvent être calculés soit en utilisant Eq.(28), soit en utilisant Eq.(25) avec les coordonnées centrées et tournées de $\tilde{\psi}_0$:

$$\mu'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(x\cos(\tilde{\psi}_0) + y\sin(\tilde{\psi}_0)\right)^p \left(-x\sin(\tilde{\psi}_0) + y\cos(\tilde{\psi}_0)\right)^q f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{30}$$

Les moments du deuxième ordre résultants sont:

$$\begin{cases} \mu'_{20} = & \mu_{20}\cos^2(\tilde{\psi}_0) + \mu_{02}\sin^2(\tilde{\psi}_0) + 2\mu_{11}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) \\ \mu'_{02} = & \mu_{20}\sin^2(\tilde{\psi}_0) + \mu_{02}\cos^2(\tilde{\psi}_0) - 2\mu_{11}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) \\ \mu'_{11} = & -\mu_{02}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) - \mu_{20}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) + \mu_{11}\left(\cos^2(\tilde{\psi}_0) - \sin^2(\tilde{\psi}_0)\right) \end{cases} \tag{31}$$

L'angle $\tilde{\psi}_0$ est calculé facilement en imposant que le moment mixte μ_{11} soit égal à zéro, comme dans le cas où les moments sont calculée dans le repère principal d'inertie. Alternativement, il est possible de minimiser μ'_{20}. En utilisant les formules trigonométriques, la troisième équation du système (31) devient:

$$\begin{aligned} \mu'_{11} = & -\frac{1}{2}(\mu_{20} - \mu_{02})\sin(2\tilde{\psi}_0) + \mu_{11}\cos(2\tilde{\psi}_0) = 0 \\ \tilde{\psi}_0 = & \frac{1}{2}\mathrm{atan2}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \end{aligned} \tag{32}$$

Cependant, pour calculer sans ambiguïté l'angle $\psi_0 \in]-\pi, \pi]$, il faut déterminer une direction le long de l'axe d'inertie, i.e., il faut distinguer entre $\psi_0 = \tilde{\psi}_0$ ou $\psi_0 = \tilde{\psi}_0 - \pi$. Pour faire cela, les moments du troisième ordre sont utilisés, car ils changent de signe quand le repère est tourné d'un angle égal à π [Tahri, 2004]. Les moments centraux en repère image tourné d'un angle $\tilde{\psi}_0$ sont donnés par l'Eq.(28), avec $\alpha = \tilde{\psi}_0$ en Eq.(29). Après, pour convention, nous définissons comme *in-plane rotation* (i.e., rotation dans le plan) ψ_0 la direction pour laquelle μ'_{30} après une rotation de ψ_0 est positive, i.e.,

$$\psi_0 = \begin{cases} \tilde{\psi}_0 & \text{if } \mu'_{30} > 0\,, \\ \tilde{\psi}_0 - \pi & \text{if } \mu'_{30} < 0\,. \end{cases} \tag{33}$$

Cette ambiguïté peut être résolue seulement si la silhouette n'est pas symétrique en rotation. Si la silhouette possède une symétrie de type *N-fold rotation symmetry* (N-FRS) (i.e., si elle se répète après une rotation autour de son centroïde d'un angle $2\pi j/N$, pour tous les $j = 1, ..., N$), N solutions sont possibles. Dans ce cas, seulement l'observation de la pose de la cible sur un série d'images consécutives peut permettre de résoudre l'ambiguïté Une fois que $\psi_{0_{test}}$ a été calculé pour l'image actuelle, il est possible de calculer l'angle de lacet ψ_{meas} avec la procédure suivante. Pour la vue de training qui correspond le mieux à la vue actuelle, nous pouvons calculer l'angle $\psi_{0_{train}}$ ($\psi = \psi_{train} = 0$ pour hypothèse). L'angle de lacet mesuré ψ_{meas} sera donc:

$$\psi_{meas} = \psi_{0_{test}} + (\psi_{train} - \psi_{0_{train}}). \tag{34}$$

Finalement, la mesure d_{meas} de la distance relative caméra-cible le long de l'axe optique est obtenue en utilisant le moment d'ordre zéro $m_{00_{test}}$ de l'image actuelle:

$$d_{meas} = \sqrt{m_{00_{train}}/m_{00_{test}}} \cdot d_{train}, \tag{35}$$

où $m_{00_{train}}$ est le moment d'ordre zéro de la vue de training ayant la meilleure correspondance avec la vue de test.

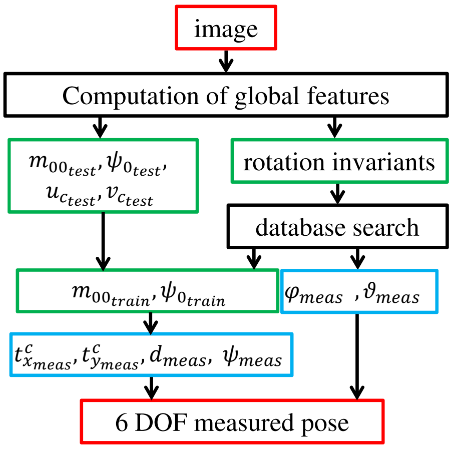


image
Computation of global features
$m_{00_{test}}, \psi_{0_{test}},$
$u_{c_{test}}, v_{c_{test}}$
rotation invariants
database search
$m_{00_{train}}, \psi_{0_{train}}$
φ_{meas} , ϑ_{meas}
$t^c_{x_{meas}}, t^c_{y_{meas}}, d_{meas}, \psi_{meas}$
6 DOF measured pose

Figure 4: Structure de l'algorithme d'estimation de pose proposé.

Pendant la construction du database, l'axe optique de la caméra z_{cam} pointe toujours vers le COM de la cible. Cependant, pendant la phase de acquisition de pose, la caméra sera pointé probablement vers le centroïde de la silhouette observée. Pour des raisons opérationnelles, il est aussi possible que le pointage ne sera pas réalisable, par exemple car il sera nécessaire d'aligner les deux interfaces de docking. Dans ce cas, en rappelant Eq.(13) et en exploitant le fait que $d = \sqrt{(tr^{cam}_{cam-tg_x})^2 + (tr^{cam}_{cam-tg_y})^2 + (tr^{cam}_{cam-tg_z})^2} \sim tr^{cam}_{cam-tg_z}$, les composantes $tr^{cam}_{cam-tg_x}$ et $tr^{cam}_{cam-tg_y}$ peuvent être approximées par

$$\begin{cases} tr^{cam}_{cam-tg_x} & = \dfrac{d}{f}\left(x_{c_{test}} - c_x\right) \\ tr^{cam}_{cam-tg_y} & = \dfrac{d}{f}\left(y_{c_{test}} - c_y\right) \end{cases} \tag{36}$$

où $(x_{c_{test}}, y_{c_{test}})$ sont les coordonnées du centroïde observé. On rappelle que f est la longueur focale de la caméra, et que (c_x, c_y) sont les cordonnées du point principal de la

caméra. Les identités en (36) peuvent être utilisée aussi pour assurer le pointage de la caméra même avant que la pose complète de la cible soit acquise.
En utilisant les invariants rotationnelles, nous pouvons réduire considérablement la dimension du database qui doit être stocké, en augmentant aussi la vitesse d'exécution de l'algorithme. Dans les Sections 4.4.1, 4.4.2, and 4.4.3 nous détaillons la procédure pour calculer les invariantes rotationnelles en utilisant les moments complexes (CMs), les moments de Zernike (ZMs), et les descripteurs de Fourier (FDs). Finalement, dans la Section 4.5, nous comparons les performances des différents set de descripteurs.

0.4 Solution de navigation intégrée

Dans cette section finale (Chapitre 5), les résultats de chaque étude ont été intégrés afin de construire une solution de navigation robuste, en tenant compte de la contrainte des ressources de calcul réduites disponibles sur les processeurs typiques qualifiés pour le spatial.

Chapter 1

Introduction

Contents

A space rendezvous is the series of successive orbital maneuvers which brings an active spacecraft (i.e., the chaser) in the vicinity or in contact with another orbiting object (i.e., the target). Autonomous rendezvous (RDV) are a key capability to answer main challenges in space engineering, such as Active Debris Removal (ADR) and On-Orbit-Servicing (OOS). ADR aims at removing the space debris, in low-Earth-orbit (LEO) protected region, that are more likely to lead to future collision and feed the Kessler syndrome [Kessler et al., 2010], thus increasing the risk for operative spacecraft (S/C). OOS includes inspection, maintenance, repair, assembly, refuelling and life extension services to orbiting S/C or structures [Jasiobedzki et al., 1999], operations that are rarely performed and up till now only by astronauts with a minimal support of robotic system (e.g. ISS, Hubble Space Telescope and MIR Station repairs) and have a very high cost. Nowadays the failure of a single system of a S/C can irrevocably compromise its whole mission. A change in the design of satellites, leading to a modular structure, could allow in-orbit-repair by a servicer S/C. OOS could also have a positive impact on space environment and help reducing the proliferation of space debris, providing life extension services to already orbiting S/C and carrying out End Of Life (EOL) disposal.

Different operational scenarios entail targets of different nature, and consequently also different technical approaches to the RDV operations. The definition and classification of targets in literature has changed through the years as OOS and ADR became a real prospect for space operations. At the beginning of the space rendezvous era, the target vehicles were classified as *cooperative*, *non-cooperative* and *uncooperative* [Polites, 1998]. The target was considered as cooperative if "*stabilized*", which, according to [Polites, 1998], meant that there is a functioning control system acting on it. On the other hand, the target was considered non-cooperative if "*non-stabilized*". The target was considered uncooperative if it was actively evasive (i.e., defense applications, which are out of the scope of this study). Moreover, [Polites, 1998] made a further distinction between an *active* target and a *passive* one. According to this distinction, an active target could be equipped with a radar transponder, laser diode or any other equipment that could actively help the navigation, while a passive target could only be equipped with passive aids such as retro-reflector or visual markers. Thus, the characterization provided by [Polites, 1998] only focused on rendezvous scenarios where the target was originally designed to participate in a rendezvous, without considering the case of rendezvous with objects such as space debris. More recently, [Wertz and Bell, 2003] classified targets as *cooperative*, if equipped with both active or passive aids to help the process, *uncooperative*, if lacking such aids, and *hostile* if actively escaping from the chaser. Therefore the definitions provided by [Wertz and Bell, 2003] do not make any qualitative distinction between a satellite being stabilized or not, and focus more on the distinction between an object originally designed or not to participate in a RDV. However, both the classification made in [Polites, 1998] -which focus on the "*behaviour*" of the satellite- and the classification provided in [Wertz and Bell, 2003] -which focus more on the physical characteristics of the satellite- are necessary to fully define a scenario.

The characteristics of the target deeply affect the RDV mission from multiple point of view, and GNC (Guidance, Control & Navigation) system is one of the chaser's systems that is the mostly affected. As this thesis focuses on the autonomous relative navigation of the chaser in the proximity of the target, we have decided to introduce a precise nomenclature and classification of the target vehicle. In this thesis, we propose to classify targets as *cooperative* or *non-cooperative* and *prepared* or *non-prepared.* According to the definition provided in 2018 by the Consortium for Execution of Rendezvous and Servicing Operations (CONFERS), when the target does not assist the chaser in acquisition, track and rendezvous operations, it is referred to as non-cooperative [CONFERS, 2018], meaning that the chaser has to estimate autonomously on board the target state. The notion of cooperative or non-cooperative therefore relates to target's behavior. On the other hand, the notion of prepared or non-prepared concerns target's design. It indicates whether the satellite was originally conceived to participate in a RDV, and therefore whether it is supplied or not with equipment such as visual markers, navigational aids, docking fixture to help both the tracking and the servicing. Because space debris objects were not conceived to participate in a RDV and are now inoperative, ADR operations will target non-cooperative and non-prepared S/C. On the other hand, OOS ideally targets cooperative and prepared S/C. However, a generation of prepared satellite is not yet in orbit, and a cooperative target can become non-cooperative in case of a system failure. A RDV with a non-cooperative non-prepared target remains the most challenging scenario for chaser's GNC, and it is therefore the case covered in this thesis.

This chapter is structured as follows. Section 1.1 will provide a brief introduction about the operational context in which autonomous rendezvous are going to play a role (i.e., On Orbit Servicing and Active Debris Removal). In Section 1.2 the phases of a space rendezvous are detailed, underlying which are the constraint during each phase and most importantly during the close-proximity phase. A major focus is given to navigation requirements and navigation sensors. The contextualization provided in Sec.1.1 and Sec.1.2 will allow the reader to fully understand the thesis objective and contributions, which are introduced in Section 1.3.

1.1 Operational context

1.1.1 On Orbit Servicing

In the last decade, the need to fully exploit the systems already launched, to provide them cost-effectively support and even to build large structures *in situ* has arisen. The goal is to advance beyond visiting destinations in space with sophisticated systems that allow no room for failure, and beyond bringing at launch everything that is needed by the mission. On-orbit repair, refurbishment or refueling will directly improve overall mission reliability and will help to ensure mission success, while on-orbit assembling and on-orbit depots will open the road for space exploration towards distant destinations.

All these activities can be classified as "On-Orbit Servicing" (OOS) activities, even if lately the acronyms IOS (In-Orbit Servicing) is becoming of common use. An International Space University report ([ISU, 2007]) provides an analysis of the key challenges and opportunities in the field of OOS, incorporating different aspects of the problem and considering politic, economic, and technical aspects of OOS. The report provides the following definition for OOS :

> *"On-Orbit Servicing is a service offered for scientific, security, or commercial reasons that entails an in-space operation on a selected client spacecraft to fulfill one (or more) of the following goals: inspect, move, refuel, repair, recover from launch failure, or add more capability to the systems"*

The spacecraft providing the service is referred to as "servicer" (i.e., the *chaser* according to the rendezvous nomenclature) and the spacecraft receiving the service is referred to as "client" (i.e., the *target* according to the rendezvous nomenclature). Three different categories for OOS as been identified:

- **Inspection Missions**: this kind of mission can provide additional information to analyze and diagnose the cause of a satellite failure, and may be a first step towards repairs or other future servicing missions. Techniques like Infra-red (IR) sensing allow the verification of solar arrays functionality, while visible imagery can detect the presence of debris/meteorites impacts and so on.

- **Manipulation Missions**: during these mission the servicer physically manipulate the client. Manipulation mission include simple refuelling, but also more complex activities such as assembling of large structures or operations where a robotic device performs reparation on the client spacecraft. Substitution of entire module of the client may be possible in a scenario where client spacecrafts are prepared.

- **Maneuvering Missions:** these missions includes re-orbitation of satellites, which may be necessary due to launch failures or even to avoid that the client carries the propellant necessary to reach or maintain its nominal orbit. De-orbitation of satellites at their EOL may be included in this category, but will be discussed better in Sec. 1.1.2.

The advantages of OOS are several. Nowadays the failure of a spacecraft component may cause the loss of the entire mission. Spacecraft manufacturers take a risk-adverse approach and use redundancy, proven technology and long operational lifetimes to attempt to mitigate the risk of failures. This increase the development cost and timelines. Effective OOS will enable the industry to reduce the constraint of fault tolerance, decreasing cost and increasing the technical performance of spacecrafts. OOS will increase the average operational lifetime of satellites.

Nevertheless, to concretely start developing OOS missions, several challenges, both technological and economical, had to be faced. Before discussing these challenges, a brief timeline of the very first OOS mission is given.

1.1.1.1 History of On-Orbit-Servicing

A report from NASA's Goddart Space Flight Center (GSFC) [GSFC, 2010] provides a comprehensive survey about history and achievements of OOS before 2010.

The first OOS activities have been realized by manned crew during Extra Vehicular Activities (EVAs). In 1973, during a series of EVAs, a NASA's crew was able to replace a thermal shield on Skylab (NASA's first space station), completing the first successful on-orbit repair. In 1984, thanks to its modular design, Solar Maximum Mission (SMM,1980-1989) was captured by the Shuttle Challenger, repaired and redeployed, continuing its mission until its reentry in 1989. Always in 1984 two satellites, Palapa B2 and Westar 6 -which had never functioned since their release from the cargo bay of the Shuttle due to propulsion-leak issue- where recovered by a Shuttle mission, returned to Earth, retrieved, resold and relaunched.

However, the biggest example of manned OOS is represented by the Hubble Space Telescope (HST) servicing. HST was specifically designed to support OOS. In the period going from 1993 to 2009, HST underwent five OOS missions.The first OOS (December 1993) allowed the replacement of the solar arrays with a set of improved ones minimizing the thermal sensitivity. In February 1997, a second mission advanced the scientific power of HST by installing two second generation instrument, enabling the study of super-massive black holes and Dark Energy. Also the S/C platform underwent some replacements and repairs. OOS for repair/replacement both on HST platform and payload were repeated in December 1999, March 2002, and May 2009. All the mission were performed by Space Shuttle crews. However, the Columbia tragedy (1^{st} of February 2003) raised awareness about the role of human spaceflight in the space conquest, and pushed forward the development of technologies to perform robotic OOS.

If HST is the biggest example of OOS for repair/refurbishment/replacement missions, the International Space Station (ISS), being the largest artificial satellite ever build, is the demonstration of the utility of on-orbit construction. The size of the structure precluded ground assembly, test and launch as an entire unit. The whole assembly has taken 34 Shuttle and 4 Russian flights to deliver the major ISS elements, and (up to 2010) 143 spacewalks for a total of about 900 hours of EVAs. Moreover, robots have played an important part in ISS construction and maintenance. The Space Station Remote Manipulator System (SSRMS, also known under the name of Canadarm 2), a robotic arm placed on the ISS, enabled berthing operations with the HTV (see Sec. 1.2.3) and pre-installation positioning of new ISS elements. It also provides a platform for EVAs. In the same way, the Special Purpose Dexterous Manipulator (SPDM), is a device having two independent robotic arms, allowing operations which were previously requiring EVAs, such as removal of failed components and installation of space units. SPDM (also known as Dextre), together with Space Shuttle robotic arm (Canadarm), enabled in January 2013 the first refuelling mission on a non-prepared S/C (i.e., Robotic Refuelling Mission). However those devices (Canadarm, Dextre, SSRMS) are operated remotely by man operators (both on ground or on the ISS), while the final goal of OOS is to exclude man from the loop and to design robotic servicers capable to accomplish missions autonomously. Autonomy will improve efficiency, robustness and capabilities, and this becomes of vital importance when communication latency is high and/or communication windows are limited (the robot will be able to perform more tasks without the need of waiting human operator assessment and approval to proceed).

The need of maturing key servicing technologies to perform autonomous missions led to a series of technology demonstration activities. The National Space Development Agency of Japan (NASDA, now known as Japan Aerospace eXploration Agency -JAXA-) realized in November 1997 the first uncrewed automated rendezvous and docking demonstration with cooperative target (ETS-VII mission). In January 2003 and April 2005 the United States Air Force Research Laboratory launched two microsatellites, XSS-10 and XSS-11 to demonstrate technologies for autonomous navigation around target. In particular XSS-11 was capable of realizing autonomous proximity operations with a non-cooperative target (Minotaur first stage), as well as collision avoidance maneuver (CAM). This demonstration raised awareness on the military/ national security implication related to the ability to approach and eventually affect an orbiting satellite. In March 2007 Defense Advanced Research Project Agency (DARPA), in the frame of the Orbital Express mission, launched two system (servicer ASTRO and the prepared client NEXTsat), becoming the first demonstration of successful end-to-end robotic satellite servicing activity. Refuelling and some ORU (orbital replacement unit) activities such as the insertion of a battery where performed, providing the confirmation that the key technologies needed for satellite servicing were mature.

1.1.1.2 Economical challenges

All the missions discussed so far involve big scientific missions (e.g., HST repair), international cooperation projects (e.g., ISS assembly), and Agencies demonstrators. Nevertheless, to push the development of enabling technologies and reduce the costs of servicing mission, the market has to be extended to private satellite manufactures, and to do that the commercial viability of OOS must be assessed. Satellite manufacturers may be suspicious towards the concept of designing for servicing since they consider it harmful to their industry (it would add new complexity to their platforms and could decrease their sales since satellites would last longer, which is actually against the industrial trend of planned obsolescence). Indeed, in a scenario where OOS is a common occurrence, satellite manufacturers would be also the ones who offers servicering, and this will represent another source of income. Nevertheless, the commercial interest in OOS will rise only after the technology and its benefits are proven in an operational context. In fact, even if the viability for hypothetical future market can be demonstrated, private industries may be skeptical about taking the initial risk and development cost. For example, a refuelling spacecraft would need to service multiple costumers in order to be economically viable. Refuelling will give the client satellite an extended life, and the additional profit coming from this life extension must be higher than the cost of refuelling. However, such a viability assessment doesn't take into account the high development cost of technologies that have not yer reached the required TRL (Technology Readiness Level). In the same way, the affirmation of modular design (whose first example is from NASA Multimission Modular Spacecraft design [Falkenhayn, 1988], with SMM being the first spacecraft making use of that design) will allow reparation/replacement of modules in space and reduce the cost of on-ground assembly, integration and testing (AIT), but will need an initial expensive changing of the whole development chain, from conception to manufacturing.

For this reason, public-private-partnerships has been pushed. This model involves the collaboration between one or more government organizations (i.e., the Space Agencies) and private industry: the public organization would incur most of the development cost and initial risks, and scientific objectives could be combined with OOS technology demonstration objectives.

Another problem related to OOS is the need of a standardization: if the servicer is supposed to dock to the client and establish a physical connection, standards need to be developed for what concerns docking interfaces, data connections, power connections, fluid connections and so on (depending on the kind of mission).

Finally, the lack of legal regulations for many OOS applications, such as high-resolution sensing of S/C belonging to different countries or companies, and regulations about liability and responsibility in the case of docking/manipulation, represents a real issue. These aspects are better discussed in [Losekamm et al., 2015].

Viability assessments

During the last decades, several cost effectiveness and viability analysis for commercial OOS have been proposed. Refuelling viability is discussed in [Hibbard, 1996], while [Kreisel, 2003] and [Kreisel, 2002] asses the commercial implication, potential market and impact of OOS. [Long et al., 2007] on the other hand suggests that the viability of robotic OOS depends on a change in the operating practice which will have to shift from expensive, large and reliable satellites to smaller, lighter and cheaper satellite, leading to economic savings in manufacturing and launching costs. More recently [Graham and Kingston, 2015] has assessed the viability of Geostationary Earth Orbits (GEO) servicing missions. Under the hypothesis of the study, refuelling mission and Begin Of Life (BOL) intervention seemed to be the more valuable, while robotic interventions for repairs still have a low viability due to the cost of the technologies needed.

GEO scenarios seem to be promising for several reasons. In January 2002 there were 900 reported objects in GEO of which only almost 28% were controlled, operational satellites [ESA, 2003].The high concentration of satellites, the great potential for failures and the ease of moving between objects in GEO (i.e., the orbits lie on the same planes) made attractive the idea of a multi-satellite repair/refuelling vehicle. For instance, adding a docking interface to a GEO satellite would only slightly increase its weight (the Orbital Express mission used a fully cooperative docking mechanism weighting 32 kg and a refueling mechanism weighing 50 kg, which are less than the 3% of the total weight of an ordinary GEO communication satellite [GSFC, 2010]).

Since GEO satellites cannot be de-orbited on Earth, they are usually moved to a graveyard orbit above the GEO protected region. This means that at launch a percentage of the boarder propellant must be dedicated for this End Of Life (EOL) disposal, increasing S/C weight at launch and therefore launching cost. Moreover, at the moment, there is no practical way to gauge remaining propellant in zero gravity environment, so satellite operators retire S/Cs based on estimation with safe margins. In a scenario where S/Cs rely on a servicer to be moved into the graveyard orbit, the S/C would be able to fully deplete the fuel, thus extending their mission duration.

It is important to underline how the use of a servicer to maneuver a prepared client in EOL disposal (which is a future perspective for satellite that have not yet been launched) is a different scenario with respect to the capture and disposal of a space debris, which is already a Resident Space Object (RSO), is not prepared and may be even in a severe tumbling.

1.1.1.3 On-Orbit-Servicing perspectives

Most of the satellite-servicing activities have already been demonstrated in LEO with humans in the loop: the challenges lie in extending that capability to robots which can autonomously perform these operations in location where communication latency impedes direct ground control (and even monitoring).

Advanced robotics analogous to the task required by a servicer are routinely demonstrated on the ground, and it is their application in Space missions that requires maturation.

In the last decade several design proposal for commercial OOS have been proposed (confirming the rising interest of private companies for OOS), but at the time of writing (i.e., September 2020) only one vehicle has already provided commercial services (i.e., MEV-1). A non-comprehensive list of the on main projects is here provided:

- European Orbital Life Extension Vehicles (**OLEVs**), funded by ESA and Orbital Recovery Ltd (ORL), was designed to operate as an orbital "tugboat," supplying the propulsion, navigation and guidance to keep a telecom satellite in its proper orbital slot for many years. Another application was the rescue of spacecraft that have been placed in a wrong orbit by their launch vehicles, or which have become stranded in an incorrect orbital location during positioning maneuvers. It appears that the mission did not move to completion as very little has been heard from 2010 [Benedict, 2013].

- The German Space Agency, DLR (Deutsches Zentrum für Luft und Raumfahrt), has been developing the Deutsche Orbitale Servicing mission (**DEOS**) robotic spacecraft since 2007 and contracting with various European companies to work on different technical elements. The DEOS project was aimed at the demonstration of technologies for the controlled in-orbit disposal (controlled destructive re-entry in Earth's atmosphere of the berthed spacecrafts) of a defective satellite (tumbling debris). According to planning, DEOS was to be ready for launch in 2018, but the project was canceled after the definition phase (B1) in 2013.

- In the frame of Horizon 2020 (European Framework Programs for Research and Technological Development) activities, **EROSS** program (European Robotic Orbital Support Services) is a project coordinated by Thales Alenia Space France (TAS-F) aimed at demonstrating European solutions for the servicers and the client LEO/GEO satellites, enabling a large range of efficient and safe orbital support services. The project will assess and demonstrate the capability of the on-orbit servicing spacecraft to perform rendezvous, capture, docking, berthing and manipulating of a client satellite provisioned for servicing operations including refuelling and payload transfer/replacement. The project has been followed by EROSS+, which aims at reaching a TRL 6 and demonstrate the developed solution in a complete rendezvous with a prepared target in 2025.

- In 2011 Canadian MDA (which is the designer of the robotic arms Canadarm and Dextre), planned to launch its Space Infrastructure Servicing (**SIS**) vehicle into near geosynchronous orbit, where it would have serviced commercial and government satellites in need of additional fuel, re-positioning or other maintenance. MDA had selected Intelsat General Corporation as its exclusive channel to bring on-orbit services to the US Government. The project came to a standstill and only in 2017, with the creation of the new company Space Infrastructure Service, MDA (now MAXAR) announced an agreement with SES (a world leading satellite-enabled solutions provider, with more than 50 GEO satellites and 12 MEO satellites on orbit) for refuelling and life extension in GEO.

- In January 2018 the Israeli company Effective Space Solution signed a contract with an unnamed customer to perform life extension services for GEO satellites. The company planned to launch two 400 kg servicers named **SPACE DRONE** by the 2020 and two more by the 2022. They targets GEO communication satellites ranging from 1500kg to 4000kg of dry mass to provide them with life extension up to 15 years and post mission disposal. However at the end of June 2020 the company announced that it will be acquired by the U.S. subsidiary of Astroscale (see Sec.1.1.3). The goal of providing life extension services to GEO satellites is maintained, but there will be delays in the development of the missions.

- As a government entity, DARPA aim is to advance technology developments without bringing products services directly to the market. With its Robotic Servicing of Geosynchronous Satellites (**RSGS**), whose projected launch date would be late 2022 or early 2023, will be able on the other hand to provide different servicing, such as high resolution inspection, anomaly resolution, upgrade installation, and repositioning in the GEO belt. The servicer will be capable to service approximately the 90% of the S/C currently in GEO.

- Only one project has currently come to a complete design, launch and docking: on 9 October 2019 **MEV-1** (Mission Extension Vehicle-1), a spacecraft from Northrop Grumman Innovation Systems, was launched. The planned mission of MEV-1 was to reach GEO orbits and link up with the Intelsat IS-901 communication satellite. The rendezvous was planned to happen on a graveyard orbit above GEO region, where IS-901 was placed in 2019 after 18 years of service. MEV-1 was planned to take control of the pointing and orbit-maintenance duties for Intelsat IS-901, replacing the satellite on its GEO slot and performing the first commercial OOS ever done before. The servicing spacecraft was planned to extend IS-901 life by five years and replacing it in the graveyard orbit at the end of the extended life. For each year of MEV services, Intelsat will pay 13 million $. After the nominal mission with IS-901, MEV-1, which has a 15-year design life, could theoretically depart and perform OOS on other satellites. Controlled by the company's satellite operations team, the MEV-1 uses a docking system that attaches to existing features on a customer's non-prepared satellite. On the 25^{th} of February 2020, MEV-1 completed successfully the historic docking with IS-901. MEV-1 and IS-901 became the first commercial satellites to dock in orbit and are the first example of life-extension services in GEO. After the proof of concept of MEV-1, a second servicing vehicle (MEV-2) was launched the 15^{th} of August 2020. It is planned to dock to the Intelsat IS-1002, which, unlike IS-901, is still operative on its GEO slot. Being a carbon copy of MEV-1 from a design point of view, MEV-2 will have therefore to realize a more challenging and risky rendezvous in the populated GEO region.

These examples shows that -even if commercial satellite operator business models have changed drastically (and often) during the past decade- the interest of private companies in OOS has now become -not without some difficulties- a matter of fact, and a new era of achievement has now been opened.

1.1.2 Space Debris problem

The term space debris indicates the collection of all non-operative objects in orbit around the Earth, including spent rocket bodies and inactive satellites, but also fragments generated by destructive events (such as collisions and explosions) and objects released during operations [Letizia, 2016]. This definition of space debris has to be considered an engineering definition, since the legal framework does not provide standards to decide on whether an object constitutes space debris [Popova and Schaus, 2018].
A report from the Inter-Agency Space Debris Coordination Committee (IADC), which met during the 50th Session of the Scientific and Technical Sub-Committee of the United Nations Committee on the Peaceful Uses of Outer Space (COPUOS)(11-22 February 2013), states that the debris situation in low Earth orbit (LEO) may be reaching a catastrophic tipping point. The risk of a scenario where the density of objects in LEO is high enough that collisions between objects will cause a cascade of collisions, with each collision generating space debris thereby increasing the likelihood of further collisions, is now well acknowledged. This cascade event, also known as Kessler Syndrome, first predicted by Donald Kessler from NASA in 1978 [Kessler et al., 2010], will lead to an exponential increase of the number of space debris. This means that the debris flux will increase exponentially with time, even though a zero net input (zero new launches) may be maintained. This phenomena, if not controlled and restrained, will prevent the use of any LEO. Even if the situation in GEO in less critical at the moment, space debris problem must be prevented also in GEO regions, mainly due to the reduced number of

available orbital slots. In order to start monitoring and protect the region of interest, IADC as declared two protected regions:

- LEO_{IADC} region: from 0km to 2000 km of altitude, all the inclinations.
- GEO_{IADC} region: from 35586 km to 35986 km of altitude and from -15^o to +15^o of inclination. This region actually corresponds to the geostationary altitude ±200 km.

1.1.2.1 Space Debris environment

Assessing the actual state of space debris population needs both modeling and observation tools. Objects of size smaller than 5-10 cm cannot be tracked by on-ground observation [Xu et al., 2009] and therefore are not cataloged in the SSN (Space Surveillance Network [Shepherd and Command, 2006]). The actual means of space surveillance include the use of telescopes, radar, and laser (laser ranging for detection and tracking of unknown non-cooperative targets is under demonstration and ESA plans for its own laser ranging tests). In the context of space debris, these means of space surveillance are used to predict reentries, to asses the necessity of performing CAM and to feed propagation models. Nevertheless, it is from small un-trackable objects that comes the highest risk of further collisions and fragmentation. Due to the high velocity of LEO objects (around 7km/s), even the impact with a particles of some centimeters can lead to the destruction of a S/C. Even objects having size lower than 1 mm cause S/C surface degradation. Estimation of this population of fragments was done during Space Shuttle era (i.e., the so-called Long Duration Exposure Facility): the degradation on the surfaces of satellite or solar arrays brought back on Earth allowed an estimation of the debris flux the objects were subjected to. Object from 1mm to 1 cm are likely to cross S/C shielding threshold, penetrate and cause structural damage. Object bigger than 1 cm are likely to cause the complete loss of the mission.

Fig.1.1 and Fig. 1.2 provide a visualization of the evolution of the absolute number of objects orbiting or penetrating LEO (1.1) and GEO (1.2) regions. The objects are classified with respect to their orbital class, being:

LEO Low Eart Orbit, from ground to 2000km of altitude

HEO High Eccentricity Eart Orbit, having the perigee altitude below 31570 km and the apogee altitude above 40002 km (and then crossing both LEO_{IADC} and GEO_{IADC} protected regions).

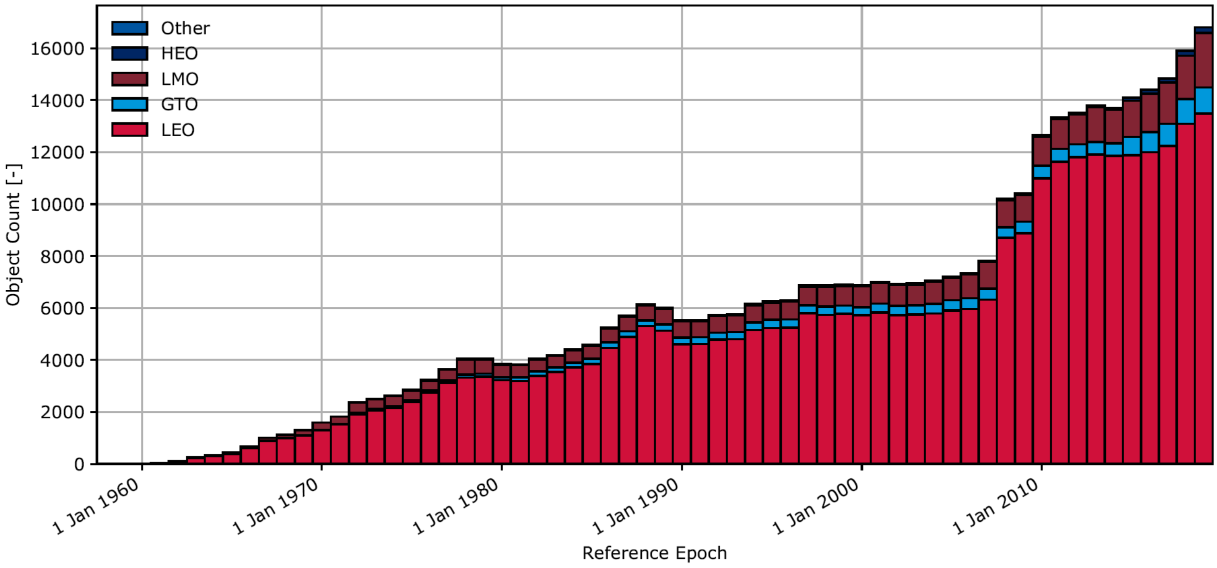


Figure 1.1: Absolute number of objects penetrating LEO_{IADC} protected regions. Images credits: ESA's Annual Space Environment Report, 2019

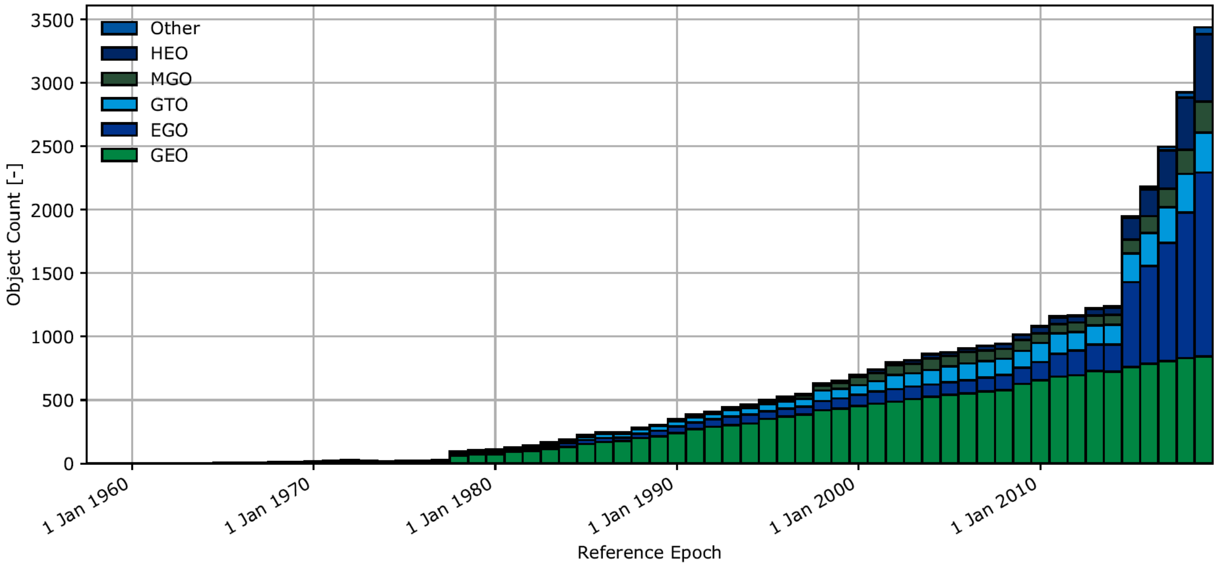


Figure 1.2: Absolute number of objects penetrating GEO_{IADC} protected regions. Images credits: ESA's Annual Space Environment Report, 2019

GTO GEO Transfer Orbit, a transfer obit that has its perigee in LEO_{IADC} region and the apogee near or inside GEO_{IADC} region.

LMO LEO-MEO Crossing Orbit, a orbit having perigee inside LEO_{IADC} protected region and apogee above that region but below 31570 km.

MGO MEO-GEO Crossing Orbit, a orbit having apogee inside GEO_{IADC} protected region and perigee above LEO_{IADC} region and below 31570 km.

EGO Extended Geostationary Orbit, an elliptical orbit having the major semi-axis between 37948 km and 46380 km, eccentricity below 0.25 and a positive inclination below 25^o.

The tracked classes are:

PL Payloads: space object designed to perform a specific function in space excluding launch functionality (e.g. operational satellites).

PM Payload mission related objects: space objects released as **space debris** which served a purpose for the functioning of a payload (e.g. covers for optical instruments, astronaut tools).

PF Payload fragmentation debris, space objects fragmented or unintentionally released from a payload as **space debris** for which their genesis can be traced back to a unique event (e.g. when a payload explodes or when it collides with another object).

PD Payload debris: space objects fragmented or unintentionally released from a payload as **space debris** for which the genesis is unclear, but whose orbital or physical properties enable a correlation with a source.

RB Rocket body: space object designed to perform launch related functionality; This includes the various orbital stages of launch vehicles, but not payloads which release smaller payloads themselves. These objects, once terminated their (short) mission, contribute to the population of **space debris**.

RM Rocket mission related objects: space objects intentionally released as **space debris** which served a purpose for the function of a rocket body. Common examples include shrouds and engines.

RF Rocket fragmentation debris: space objects fragmented or unintentionally released from a rocket body as **space debris** for which their genesis can be traced back to a unique event (e.g. objects created when a launch vehicle explodes).

RB Rocket debris, space objects fragmented or unintentionally released from a rocket body as **space debris** for which the genesis is unclear, but whose orbital or physical properties enable a correlation with a source.

UI Unidentified objects.

The highest amount of space debris is located in LEO region (between 600 km and 1000km of altitude), with a pick at the altitudes around 800 km. For these altitudes, the more populated inclination are the one corresponding to sun-synchronous orbit (97^o-105^o). Below 600 km debris are not likely to "survive" thanks to the natural decay caused by atmospheric drag. ESA publishes every year the "Annual Space Environment Report" providing a photographs of the current situation and evolution of the environment [ESA, 2019a], based on DISCOS Database (Database and Information System Characterising Objects in Space [Klinkrad, 1991]).

The figures underline the exponential growth of orbiting objects during the past decades, even if part of the registered growth has to be attributed to the augmentation of sensing performance. The number of objects isn't the only tracked factor, since both the evolution in the total mass and total surface orbiting are relevant (bigger objects generate a larger amount of fragment in the case of a collision, and the probability of a collision with a debris increases as the surface of the S/C increases). For this reason, also an "equivalent" number, which is obtained multiplying physical property of the object (count, mass, and area) with an equivalence factor, is computed. Table 1.1 shows then number of newly added objects in 2018: Being the class of PL the only one not appertaining to the population of space debris, the extent of the problem becomes evident.

	PL	PF	PD	PM	RB	RF	RD	RM	UI	Total
both regions, absolute	2	0	0	0	6	0	0	3	46	57
LEO_{IADC}, absolute	337	57	17	22	72	31	19	95	562	1252
LEO_{IADC}, equivalent	369	57	17	21	45	8	10	87	88	703
GEO_{IADC}, absolute	29	0	0	4	9	44	2	3	495	586
GEO_{IADC}, equivalent	25	0	0	2	0	3	0	0	33	64
none, absolute	28	0	0	1	15	395	2	2	270	713

Table 1.1: Absolute and equivalent number of newly added object intersecting with the protected regions. Table credits: ESA's Annual Space Environment Report, 2019

Fig. 1.3 shows the monthly number of objects tracked by the SNN, from the 60's to 2016. Two pick in the population of fragmentation debris can be found in 2007 and 2009. The pick of 2007 is due to Chinese anti-satellite missile test and consequent destruction of Chinese weather satellite Fengyun-1C. The pick of 2009 is due to the collision between the operative Iridium-33 satellite and the inoperative Kosmos-2251, whit the following decrease due to the atmospheric reentry of a percentage of the resulting fragment. These two single events have increased by the 50% the risk of further collision. Currently, almost 2/3 of the CAM performed on operative satellites are aimed avoiding fragments due to these two single events. The event of 2009 was a real game changer for the community since it was the first collision involving an operative satellite, thus raising awareness on the real risk coming from space debris in LEO.

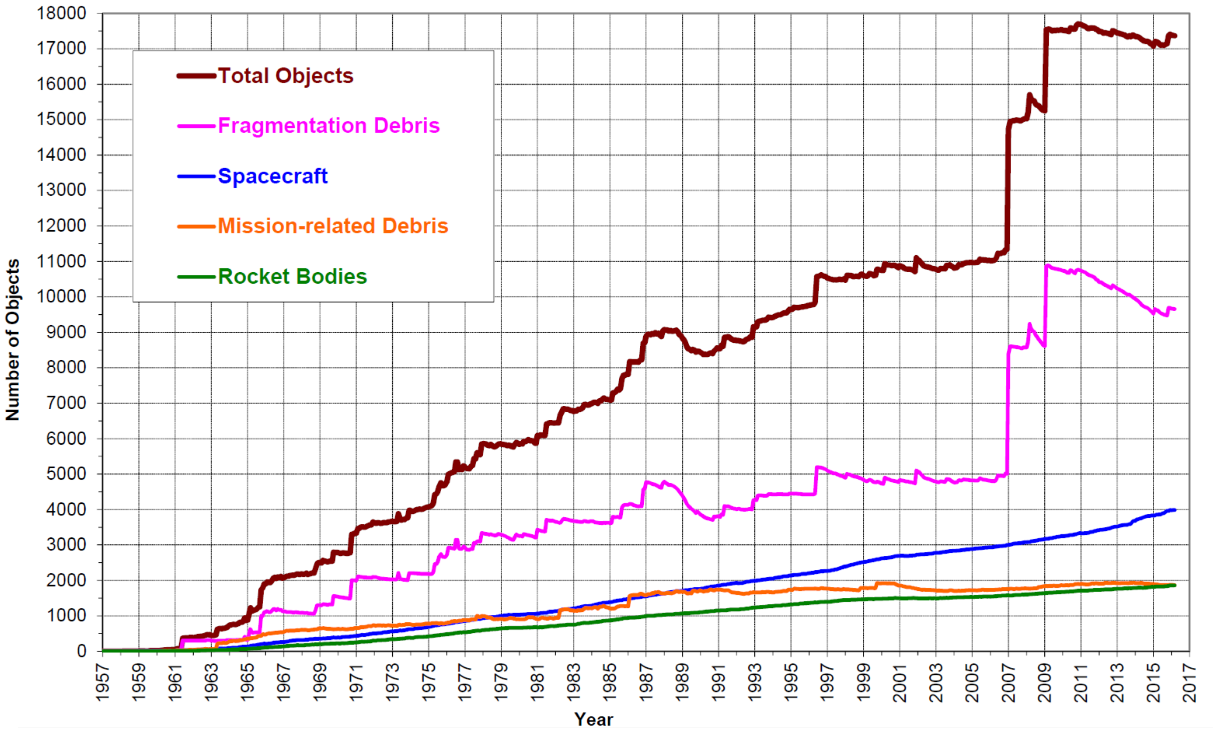


Figure 1.3: Monthly Number of Objects in Earth Orbit by Object Type
Images credits: NASA - Orbital Debris Quarterly News - April 2016 - vol. 20 (1-2)

For the non trackable populations, as well as for the prediction for the next years, complex mathematical and statistical models are exploited (i.e ESA's MASTER 2009 model, [Flegel et al., 2009]). Modeling for prediction implies the propagation of the actual population for a long period of time, with the inclusion of new objects from future launches and eventual explosions, and considering the possibility of future fragmentation due to collision. Models has consider different scenarios, such as the possibility of implementing mitigation guidelines (Sec.1.1.2.2), "business as usual" (no mitigation measures applied), and even the future advent of the mega-constellations. The study in [Liou et al., 2013], starting from an initial population provided by MASTER 2009 model, compares the prevision in LEO_{IADC} region for the next 200 years provided by six different model. The considered model are ASI (Italian Space Agency) model SDM (Space Debris Mitigation), ESA's model DELTA (Debris Environment Long-Term Analysis), ISRO's (Indian Space Research Organization) model KSCPROP, NASA's model LEGEND (LEO-to-GEO Environment Debris model), JAXA's model LEODEEM, and UK Space Agency model DAMAGE (Debris Analysis and Monitoring Architecture for the Geosynchronous Environment) which has now been extended also to LEO region. All the models provide consistent results: even with a 90% compliance of the commonly-adopted mitigation measures (see Sec. 1.1.2.2) and no future explosion, the simulated LEO debris population will increase by an average of approximately 30 % in the next 200 years, with catastrophic collisions expected to occur every 5 to 9 years. The conclusion drawn from [Liou et al., 2013] is that active debris removal should be considered to stabilize the future LEO environment.

To contrast the space debris problem, space operators have to act in two directions:

SDM Space Debris Mitigation: prevention of in-orbit explosion or collision and end of life disposal.

SDR Space Debris Remediation: removal of the biggest space debris in the protected regions.

Generally, SDM involves all those measures that must be taken during the design of the satellite, and therefore can be applied only for satellite that are not yet in orbit. On the other side, SDR is an extreme measure that has to target the RSO which are not compliant with SDM guidelines and represent a real danger for operational satellite.

1.1.2.2 Space Debris Mitigation

The most effective short term means of reducing the space debris growth rate is to prevent in-orbit explosions (e.g., via passivation of space objects at the end of their operational life) or collisions (e.g., via CAM while the objects are still active). On orbit breakup events do not depend only on collision with existing fragments, but also on explosions (e.g., caused by electrical overcharging, battery explosion, and propulsion issues). Accidental breakup must be avoided both during the operational life of the S/C and after its EOL. The sets of suggested measures to adopt after S/C EOL include also the clearance of the protected regions (i.e., EOL disposal, which is considered as the most effective long-term means of stabilizing the space debris environment at a safe level). According to [ESA, 2019a], most internationally accepted space debris mitigation measures can be traced back to the following objectives:

- The limitation of space debris released during normal operations.
- The minimization of the potential for on-orbit break-ups (an average of 8 non deliberate fragmentation occur each year)
- Post mission disposal to clear protected regions

- Prevention of on-orbit collisions (An average of 1/2 CAM are performed each year accepting a fatality risk below 10^{-4})

ESA's mitigation guidelines are collected in ESA's Standardization Steering Board (ESSB standard) Space Debris Mitigation Compliance Verification Guidelines [ESA, 2015a]. The set of procedures to reduce post-EOL breakup risk are referred to as "passivation" ([Anselmo and Pardini, 2008]), and may include discharge of batteries, disconnection of solar arrays from batteries, depletion burns to empty propellant tanks, and so on.

For what concerns the clearance of the protected region, both Earth reentry or disposal to a graveyard orbit are accepted. Earth reentry, due to the required ΔV, is likely only for LEO satellites. Satellites have to reenter Earth within 25 years after the EOL, with an on-ground fatality risk lower than 10^{-4}. Reentry can be obtained by placing the satellite in an orbit where natural orbital decay due to atmospheric drag allows the reentry within the prescribed 25 years, but only if the on-ground fatality risk of an uncontrolled reentry -which can be assessed with tools such as ESA's DRAMA [Martin et al., 2005]- is below 10^{-4}. Otherwise, a controlled reentry targeting uninhabited area must be performed. An alternative to the controlled reentry - which is more expensive both in terms of required ΔV and on a system level (e.g., more performing AOCS)- is to "design for demise" [Kelley, 2012]. Design for demise refers to the "intentional design of space system hardware so that it will completely burn up – or "ablate" – during uncontrolled atmospheric reentry as a means of post-mission disposal". Of course this kind of design leads to an increase of S/C production cost, which, depending on the mission, may be higher than the cost of a controlled reentry. [Heinrich et al., 2015] summarizes current state of art on both techniques in order to provide guidelines and recommendations to help in the trade-off for this issue. EOL disposal and efficient CAM will become more and more important with the advent of mega-constellations which will dramatically increase the number of objects in LEO, with more than 1000 satellites planned for OneWeb constellation, almost 3000 planned for Boeing constellation, 4600 satellite planned for Samsung constellation and about 1000 planned for Google Constellation [Larbi et al., 2017]. Alone, SpaceX Starlink constellation could swell by more than 30 000 satellites, with more than 12 000 already approved and 775 S/C already orbiting by September 2020. Nevertheless, SpaceX decided to reduce the impact on LEO environment of the first set of 1440 planned satellites by reducing their altitude from the initially planned 1100km to 550km.

For what concerns GEO clearance, GEO satellite should be disposed on an orbit having an eccentricity lower than 0.003 and having a perigee altitude sufficiently above the geostationary altitude, such that long-term perturbation forces will not cause the spacecraft to enter the GEO Protected Region within 100 years. A first assessment of the altitude of the disposal orbit can be done using the following formula, where C_r is the dimensionless solar radiation pressure coefficient and A/m is the ratio of the cross-section area to dry mass of the space system:

$$h_{disposal} = h_{GEO} + 235 + 10^3 C_r A/m \tag{1.1}$$

A deeper insight of SDM in geosynchronous orbit is given in [Anselmo and Pardini, 2008]. LEO and GEO clearance requirements don't involve only LEO/GEO satellites, but also those S/C which are likely to enter the region after their EOL (e.g. missions in the unstable Sun-Earth Lagrangian point L2, which should be disposed into heliocentric orbits with no revisit closer than 1.5x10^6 km from Earth within 100 years).
Even if some passive deorbitation device is now under study (e.g., drag-augmentation devices [E Roberts and Harkness, 2007], and solar pressure augmentation devices [Lücking et al., 2013]), at the moment the only way to clear the protected regions is to perform propelled maneuvers (except for those LEO satellites whose orbit will naturally decay within the prescribed 25 years). As anticipated in Sec. 1.1.1, the possibility of relying on a servicer S/C to clear the protected regions will augment the operational lifetime of the client. Indirectly, this will also contribute to a reduction of space debris population growth since it will avoid early replacement of S/C. It becomes clear how OOS can positively contribute to SDM. Currently, SDM guidelines suggest particulars design to reduce the risk of breakup, debris release, and so on. The maturation of OOS technology may lead to the definition of new mitigation guidelines for S/C design, such as the possibility to include visual aids and docking or grapple fixture to help RDV operations: OOS could then provide both planned EOL disposals and emergency disposals in case of client system failure.

1.1.2.3 Space Debris Remediation

Space Debris Remediation aims at removing the already existing debris that are more likely to lead to future collision and feed the Kessler syndrome. Many works have assessed the impact of ADR on the stability on environment, and more precisely LEO environment, which presents the most dangerous situation in a short term scenario. In [Liou and Johnson, 2009, Liou et al., 2010], prior to Iridium-Kosmos collision, NASA proposes a sensitivity study to illustrate and quantify the effectiveness of various remediation options. The problem of selecting the best candidates for the removal is underlined and a selection criterion is established, targeting the debris having a maximum value of $R(t) = P(t) \times m$ where $P(t)$ is the probability of collision and m is the mass of the object. The environment simulations, carried out with LEGEND model, rely on assumption such as the percentage of compliance with SDM requirements of orbiting S/Cs, the solar activity, and the launch traffic. The results of the study lead to the conclusion that ADR must be implemented together with SDM to keep the environment stable, with a removal of mass equal to 5–10 large objects per year from regions with high object densities and long orbital lifetimes starting from the 2020. After the Iridium-Kosmos collision, a new study from NASA [Liou, 2011] was presented in order to take into account the new evolution of LEO environment. The collision had changed the "initial condition environment" of the simulations and has reignited the interest in performing ADR. The study also underline the priority of realizing ADR in LEO environment, since in GEO and MEO environment only few accidental collision between objects bigger than 10 cm where predicted in the next 200 years. The work also underlines how the TRL to allow safe ADR operation has not yet been reached and that the hypothetical start of ADR operation may be post-posed to 2060, leading to 7 more collisions with respect to an early implementation in 2020. In the same way, ESA's internal studies show that continuous removal actions starting in 2060 would be 25% less effective in comparison to an immediate start [ESA, 2015b]. The study in [Liou, 2011] also focuses in defining the LEO areas in which the highest values for $R(t)$ are found, resulting in altitudes between $600 - 1000$ km and inclinations in the ranges $70^o - 75^o$, $80^o - 85^o$, $95^o - 100^o$, with almost the average of the mass in the retrograde regions represented by rocket upper stages. According to [Liou, 2011], rocked bodies may be the first targets of ADR demonstration due to their simple shapes and the fact that they do not carry any sensitive instrument so that agreement for international cooperation may be achieved easily. A deeper study assessing the breakup severity on operational satellites aimed at defining a severity index for existing debris can be found in [Letizia et al., 2016]. This severity index can help to define priority target for ADR. A study from the CNES (Centre Nationale d'Etudes Spatiales) investigate the main issues related to ADR [Bonnal et al., 2013]. The study underlines the need of ADR to be compliant also with SDM requirements, which translates in the need of performing controlled reentry when deorbiting debris of size bigger than 1000 kg, which are likely to survive the atmospheric reentry and represent a risk for the population on Earth. A second issue is related to the problem of realizing RDV operation with a satellite of another launching state and will be investigated in Sec. 1.1.2.4. Moreover, the question of how to define the operational scenario in an efficient way arises: for example, a chaser can be designed to target a single debris, to capture it and to deorbit together with it, or to target more satellites and deliver a deorbiting kit to each one of them. Finally, the realization of an automated mechanical interfacing with a non prepared target which may be in tumbling is a difficult task, and the TRL of the possible solution remain rather low. Technologies such as soft mechanical interface (e.g., nets, hooks, clamps, and harpoons) or even contact-less solution (e.g. Ion beam [Bombardelli and Pelaez, 2011] for de-tumbling and removal) are investigated but still needs to reach the required level of maturity. The study in [Wormnes et al., 2013] proposes a survey of the technologies that can enables debris capture. Of course ADR implementation needs the development of complex GNC functions, which will be deeply investigated in Sec. 1.2.

Indeed, the challenges are numerous but the development of adequate SDR mechanisms at an international and national level should not be postponed in order to ensure long term space access.

Some works (e.g., [GSFC, 2010, Larbi et al., 2017]) include ADR withing the OOS activities, with [Larbi et al., 2017] assessing the possibility of using CubeSat to deorbit satellites of the future mega-constellations. Although an EOL disposal is indeed a form of OOS, main differences lie between performing a planned EOL disposal on a S/C which was designed for this scope, and performing ADR on the already existing debris. These differences have major impacts on the needed technologies, navigation algorithms, proximity operations that will be discussedin Sec.1.1.3. Moreover, SDR (which aims at reducing the number of existing debris with ADR) and SDM (which aims at reducing the production of new debris with CAM, passivation, shielding, EOL disposal and so on) are considered as two separate contexts. For all these reasons, we will refer to ADR and to OOS as two different but related issues.

1.1.2.4 Legal framework for SDM and SDR

As mentioned, the challenges that have to be faced to implement SDM and SDR are not only technological and economical, but also legals. In order to understand the existing legal framework of SDM and SDR, a brief history of space laws and regulation will be presented.

In 1958 UN General Assembly set up the Committee on the Peaceful Uses of Outer Space (UNCOPUOS), with the aim of governing the exploration and use of space for the benefit of all humanity.. The Committee was tasked with reviewing international cooperation in peaceful uses of outer space, studying space-related activities that could be undertaken by the United Nations, encouraging space research programs, and studying legal problems arising from the exploration of outer space. The work of the Committee lead to the creation of the five treaties and five principles of outer space:

- Treaty on Principles Governing the Activities of States in the Exploration and Use of Outer Space, Including the Moon and Other Celestial Bodies [UN, 1967b], which opened for signature in January 1967, and it entered into force in October 1967. As for January 2019, it is ratified by 98 States and signed but non ratified by 23 States.

- Agreement on the Rescue of Astronauts, the Return of Astronauts and the Return of Objects Launched into Outer Space [UN, 1967a], which opened for signature in 1968 and entered into force in December 1968. As for January 2019, it is ratified by 98 States and signed but non ratified by 23 States.
- Convention on International Liability for Damage Caused by Space Objects [UN, 1972], which opened for signature in 1972 entered into force in September 1972. As for January 2019, it is ratified by 96 States and signed but non ratified by 19 States.
- Convention on Registration of Objects Launched into Outer Space, [UN, 1974], which opened for signature on 14 January 1975 and entered into force on 15 September 1976. As for January 2019, it is ratified by 98 States and signed but non ratified by 23 States.
- Agreement Governing the Activities of States on the Moon and Other Celestial Bodies, [UN, 1979], which opened for signature on 1984 but entered into force only in June 1984 as the fifth involved country ratified the convention. This convention has not been ratified by all the Countries involved (or that plans to be involved) in human space flight (i.e., the United States, the majority of the member states of the European Space Agency, Russia, People's Republic of China and Japan), so it is considered to have a very low relevancy in international law.

The Outer Space Treaty (OST) can be considered as the constitution of space law: it contains the basic principles for space activities, provides the basis for the next four treaties, and has gained significant support. OST is considered to contain principles of customary international law, which bind not only state parties to the treaty but also non-signatories [Popova and Schaus, 2018]. OST enshrines freedoms in relation to human space activities: freedom to explore outer space, the freedom to use outer space, the freedom of scientific investigation, the freedom of access to all areas of celestial bodies.

[Popova and Schaus, 2018] proposes a detailed analysis of how these treats affects the legal framework underlying SDM and SDR. OTS does not mention the Freedom to access outer space, which is actually supposed as a pre-condition. Accessibility should be preserved non only as a short-term perspective, but on a long-term basis, and such a condition is maintained only if the space environment is stable. Sustainability is an indispensable condition for the usability of outer space, therefore it is required that the use of outer space by the current generation takes place on the basis of responsibility towards future generations. Even though the legal framework provides some general direction for co-operation between the users of outer space, concrete laws on how to ensure sustainability need to be formulated in more detail. In fact, the treaties on space law neither expressly prohibit the creation of space debris nor impose an obligation on states and their space actors to remove space objects from orbit. During its 62nd session in June 2019, the COPUOS adopted a preamble and 21 guidelines for the long-term sustainability of outer space activities. These provide guidance on policy and regulatory framework for space activities, safety of space operations, international cooperation, capacity-building and awareness, and scientific and technical research and development. The guidelines have been endorsed by the UN General Assembly in November 2019, but remains legally non-binding.

On the technical level, both mitigation and remediation concepts have been developed in order to facilitate the protection of near-Earth space from space debris aiming to maintain the conduct of space activities indefinitely in the future and therefore to guarantee the freedom to use outer space. In the following paragraphs more details on the legal frameworks of SDM and SDR are provided.

SDM legal framework

Since the 80', no new international treaty on space have been developed, neither the existing ones have been revised. States are now increasingly focusing on non-legally binding documents, such as technical guidelines. The technical guidelines for space debris mitigation were introduced in 2002 by the IADC, an organization consisting of 12 national space agencies and ESA. IADC SDM guidelines [IADC, 2002] were written by scientist and engineers and not by governments, and were adopted in 2007 also by the UNCOPUOS. SDM guidelines depict environmentally relevant technical measures for future missions that will ensure the protection of the outer space environment from space debris. Guidelines on Space Debris Mitigation, are applicable to the mission planning and to the operations of newly designed spacecraft and orbital stages and, if possible, to existing ones. These guidelines are voluntary and include different measures (e.g. sharing information on space objects and orbital events, conjunction assessment during all orbital phases of controlled flight, criteria and procedures for the active removal of space objects and for the intentional destruction of space objects, risks associated with the uncontrolled re-entry of space objects and so on). As these instruments are not legally binding, they do not create rules of international law, the violation or non-observation of which would give rise to an international responsibility of states for creating or for not mitigating space debris. Compliance with SDM measures is only of a voluntary nature and cannot be legally enforced. Nevertheless, internal regulation of spaces agencies can contribute to the implementation

of SDM. For example ESA Space Debris Mitigation Policy for Agency Projects (2014) is applicable to the procurement of all ESA space systems and all operation under the responsibility of ESA. Moreover SDM guidelines can serve as a model for the development of national space laws (e.g France) which impose concrete obligations for implementing mitigation measures on private space actors. Even if few States adopt SDM measures as laws, this can deeply affects de general framework, due to the concept of *launching State*. A launching State is any State (strictly speaking: any State party to the Outer Space Treaty of 1967 or the Liability Convention of 1972) that "launches or procures the launching of an object into outer space", and any State "from whose territory or facility an object is launched". Any space object can thus be 'tied' to at least one State, and quite often to more [ESA, 2017]. Given a certain payload (e.g. a communication satellite), its launching States will be the State owning the satellite and paying for the launch, the State providing the launcher and the State from which the launcher lifts off. According to the Liability Convention, launching States are "jointly liable" of any damaged caused by their space object, where damage means "loss of life, personal injury or other impairment of health, or loss of or damage to property of States or of persons, natural or juridical, or property of international intergovernmental organization [UN, 1972]". Liability is *relative* (culpability must be proven) for damages occurred in outer space and *absolute* for damage occurred in air and ground. This means that even the State that procure the launch has an interest in verifying that the carried payload is compliant with the current regulation. It becomes clear how important is that a state such as France, which is a launching State for all the mission relying on a Ariane launcher and lifting off from Kourou, has adopted SDM guidelines as a state law. France, whose Space Law imposes the reentry of LEO satellite within 25 years after the EOL with a fatality risk lower than 10^{-4}, may ask, to the State demanding the launch, to be compliant with the SDM requirements.
Currently IADC SDM guidelines are under discussion at COPUOS to become international laws, but still have to overcome the reservation of some states (emerging states are reticent towards taking expensive technical measure to help preserving an environment whose actual pollution wasn't caused by them).

SDR legal framework

If the legal framework of SDM is complex, the one for SDR is even more. From a legal point of view, neither is space debris defined nor its production prohibited, nor are the mitigation and remediation of space debris considered in the binding law. Thus, the creation and the non-removal of space debris is not recognized to be an unlawful act. The legal framework does not provide standards to decide on whether an object constitutes space debris, but also the legal regime for space activities does not define what space debris is (e.g., does it depend on S/C controllability or on functionality?). The Registration Convention provide that space objects have to be registered, but no information concerning the functionality or the current status of the space object must be provided, meaning the register cannot be used to asses the eligibility for removal of a registered object. Moreover, as anticipated in 1.1.2.3, it is not yet clear how a substantial risk should be defined so as to decide which fragments should be removed first.

As there is no legal obligation for states to remove their objects, another relevant question is how to gain authorization to remove in cases where, for example, the state or registry neither consents to undertake the removal not does it provide authorization to a third party due to security concerns. Also, the specific liability regime for space activities pose problems when it comes to ADR operations: if a third party undertakes an ADR operation on a object, and during this operation some damage is caused, also the launching States of the object to be removed are liable. Moreover at the moment no change or transfer of ownership or control of space objects is foreseen in space law treaties. Thus, in the case of an accident that occurred during an ADR mission on a transferred satellite, the original launching state will be held liable for any potential damage, although it might have no possibility to control the satellite. This lack of regularization interest also OOS (e.g for EOL disposal). Furthermore, ADR systems (as well as OOS systems) entail a capability which is not restricted only to space debris and they could be used, if such an intent is given, for the removal or inspection of any satellite. This dual characteristic, both civil and military, makes ADR (and OOS) a sensitive capability.

One legal avenue to incorporate SDR mechanisms in the existing legal framework could be to follow the same steps made to push national adherence to SDM guidelines (i.e. implementing SDR and OOS measures nationally as part of authorization or licensing requirements, or through national legislation). Moreover, space agencies can contribute to the development and implementation of SDR by defining policies which are applicable to the procurement of all space systems and operation under agencies responsibility. Such a move from ESA will ensure accordance also to SDR and OOS mechanisms for European projects).

1.1.3 From OOS to ADR and vice-versa: future perspectives

In the past five years, many things have changed. In December 2018 ESA announced the end of e.Deorbit mission, the ESA's Clean Space initiative began in 2013 aiming at capturing and removing ENVISAT, an Earth observation satellite failed in 2012 [ESA, 2018]. The project has helped identifying the novel technologies that would need to be developed, from the capture mechanism to the GNC functions. The mission stopped after the B1 phase, namely due to a lack of a commercial market for space debris removal and the lack of funding. As ESA declared the end of e.Deorbit as an ADR mission, the reborn of e.Deorbit as a servicing vehicle was announced. In the summer 2018 ESA issued a ‘request for information’ to industries with the mandate to take down one ESA-owned defunct satellite while demonstrating in-orbit servicing. This signifies the fact that the synergies between ADR and OOS is now acknowledged, with ADR being the boundary case: technologies enabling safe ADR would enable OOS too, while the reverse is not true [ESA, 2019b]. The interconnection between ADR and OOS is well underlined by the new project of ESA approved the 27-28 of November 2019, which has been called ADRIOS (Active Debris Removal/In-Orbit Servicing). The project aims at the development of GNC technologies and RDV/capture methods which will be applied to the mission ClearSpace-1 that will realize the first demonstration of ADR. The mission will target VESPA (Vega Secondary Payload Adapter), the upper stage of the VEGA launcher that was left by ESA on a orbit between 660 km and 800 km in 2013. This upper stage, which is relatively small (100 kg) seems to be an easy target due to its simple shape and its rugged construction, represents a good start to demonstrate enabling technologies that will pave the road to the capture of multiple and/or biggest object (e.g. the 8-tons ENVISAT).

In the meantime, the Japanese Astroscale has targeted the market for end of life services in LEO. In the end of the 2019 Astroscale has entered the AIT phase of ELSA-d mission. The mission, originally scheduled to launch in 2020, will demonstrate technologies for target search, target inspection, docking and removal with a prepared spacecraft, both in tumbling and non- tumbling conditions. The aim of the mission is to develop a standard approach for EOL disposal of LEO satellites in mega constellations. The procedure consists in attaching a standard docking mechanism on the client satellites before launch. At end of life, this mechanism will allow efficient capture by a magnetic capture system installed on the servicer, which will dock with two or possibly more clients and subsequently deorbit them.

Meanwhile, the market has started to change and satellite servicing has become an actual market. In 2018, only about 10 GEO satellites orders have been placed. GEO operators are uncertain of their 15 years business model and are therefore hesitating in purchasing replacement satellites. Life extension servicing allows the operators to defer fleet replacement decision making, allowing the market and technology uncertainties to mature. The life extension servicers will in some way also contribute to the SDM, reducing the number of decommissioned satellites. In the meantime, the rise of the constellations has pointed out the need of a change in the manufacturing, which has now to meet the requirement of delivering satellites to costumers in a shorter period of time. The answer can be given by flexible satellites exploiting modularity and software defined programmable payloads. Together with modularity, also standardization is now promoted. These changes in design are paving the way for future OOS replacement/repairs by creating a catchment area for future servicers. Constellations open the OOS market to LEO orbit, where servicers will provide payloads installation, inspection and EOL disposal for those S/C unable to perform it autonomously.

1.2 Space Rendezvous: phases, constraints, and solutions

This section provides to the reader a general overview of the major issues related to the development of an autonomous rendezvous, with a special focus on the navigation problem. All the information provided in this section is retrieved from the book "Automated Rendezvous and Docking of Spacecraft" [Fehse, 2003], of Wigbert Fehse. The book is a world-wide recognized milestone in the subject of autonomous space rendezvous and includes all the expertise gained by ESA during the pre-development phase of the Hermes-Columbus rendezvous program in 1989-1993 (Hermes was meant to be the European Space Shuttle, and was planned to dock with the Columbus orbiting modulus; however the project never came to light and Columbus became a modulus of the ISS), and the pre-development phase of the ATV program in 1994-1998 (i.e., the Automated Transfer Vehicle, the European cargo ship in charge of supplying the ISS within 2008 and 2015). The book only covers cases involving cooperative and prepared targets, since at the time of writing ADR and OOS were not already in the road-map of the space agencies. However, the book provides a compendium on all the issues related to automated rendezvous.

1.2.1 The phases of a rendezvous mission

A rendezvous mission is conventionally divided into five phases, i.e., the launch, the phasing, the far range rendezvous, the close range rendezvous, and the mating.

Launch

The launch phase does not have any particular difference with respect to the launch phase of any other mission: the launch window is univocally determined by the moment in which the launch site is passing through the orbital plane in which the chaser is meant to be delivered by the launcher. A given orbital plane is intersected twice a day by a given launching site, but usually launches in easterly direction are preferred in order to exploit the tangential velocity component of the Earth rotation, which is almost 436 meters par second. The targeted orbital plane should correspond to the target orbital plane since a change of plane maneuver performed by the chaser once in orbit would mean an unacceptable propellant consumption. In the scenario of a chaser servicing more than one client, it would be preferable to target spacecrafts on the same (or close) orbital plane. At the end of the launch phase the chaser should be ideally placed on the same orbital plane of the target, on a slightly lower orbit, at an arbitrary phase angle behind the target (depending on the launching window and the target orbital parameters at the launch date). This position can be reached after launch with an orbit raising maneuver performed by the chaser's own mean of propulsion.

Phasing

The aim of this phase is to reduce the orbital angular phase between the target and the chaser. During this phase some adjustment of the orbital plane are performed, both to correct launch injection error and compensate the target plane slow variation due to the orbital perturbations. This phase is considered to be ended when an *initial aim point* is acquired, or when a set of margins for position and velocities at a given range (i.e., the *trajectory gate* or *entry gate*) is achieved. From this moment on, the navigation switches from absolute to relative. Being based on absolute navigation (e.g., target and chaser telemetries based on on-ground observation, GNSS measurements), the phasing is controlled from ground control. The location of the aim point changes with the mission and depends on many factor. Usually it is placed slightly below and behind the target, as the natural drift will get the chaser close to the target without any propelled maneuver. If the entry gate is used instead of the aim point, the apogee of the chaser is raised to the altitude of the target orbit, while the perigee is raised successively with a series of maneuvers that decrease the phasing rate, until the pre-established conditions in position and velocity are met. The accuracy on the chaser position required, at the end of this phase, for safe operations is typically in the order of a few hundreds of meters in altitude and a few kilometers in the orbital direction. This accuracy can be achieved with maneuvers executed in open-loop (e.g., by Homann transfer) based on absolute navigation.

Far range rendezvous operations

This phases is also referred to as *homing*, in analogy with the terminology used in aeronautics. As anticipated, far range operations rely on relative navigation. In terms of relative ranges, this phase usually starts at a few tens of kilometers from the target and ends at few kilometers from the target. The tasks of this phase are the acquisition of the target orbit, the reduction of the approach velocity and the synchronization of the mission timeline. This means that the phase may include some time-flexible elements in order to synchronize the mission timeline with external events, such as Sun illumination, communication window, crew operation timeline (in a rendezvous with a manned target spacecraft). Time flexible elements may include faster or slower phasing rate as for the phasing phase. Usually, synchronization is obtained by placing the chaser at the end of the phase on a hold point on the V-bar (i.e., a point on the target orbit where the chaser can stay indefinitely at nominal zero relative velocity with respect to the target without propelled maneuvers).

Close range rendezvous operations

This phase is usually divided into two sub-phases, the *closing*, which place the chaser on a final approach corridor, and the *final approach phase*, which leads to the mating conditions. This phase can be initialized only if the out-of-plane errors have been corrected to the same accuracy of the in-plane-errors, and the mission timeline has been synchronized. During the closing, the approach trajectory should be designed such that the incapacity to execute a thrust maneuver does not leave the vehicle on a trajectory which eventually leads to collision with the target (i.e., *passive trajectory protection*). The trajectory should be designed taking into account the need of re-placing the chaser on a hold point if contingencies arise. The objective of the final approach on the other hand is to achieve docking or berthing conditions in terms of positions, velocities, relative attitude and angular rates. For this reason, starting from a distance of almost $40-20$ m, the chaser must estimate also the relative attitude and rotation rate of the target. Trajectories in final approach are closed loop controlled straight-lines (if controlled by automated on-board system) or quasi straight-lines realized by a multitude of small hoops (if man-controlled). Straight-lines (or quasi straight-lines) are chosen because of the relatively small field of view of the sensors used in this phases, and to keep aligned the docking interfaces. In this final phase, the measurement accuracy required before docking is approximately of 1% of the range, few centimeters on lateral position, about 1 cm/s of axial and lateral rates, about 1 degrees for relative attitude and about 0.1 deg/s for rotation rates. This requirement can be relaxed if instead of docking a berthing is foreseen (i.e., as a rule of the thumb, they can be a factor of 5 higher than the ones required for docking). The exception is in the angular rate, which should be a factor of 5 lower than the ones required for the docking. These values are reported in the following section.

This phase is obviously very delicate due to the closeness of the chaser and target spacecrafts. Usually, a cone-shaped (half cone angle of 10-15 deg) approach corridor is defined, and the chaser trajectory must remain within the boundary of the cone. If the chaser exits the cone, stop, retreat, or collision avoidance commands (i.e., *active trajectory protection*) can be issued.

During this phase, the chaser thruster plume on the target can have major effects, such as the deviation of its orbit or attitude due to the force exerted on the target vehicle, the overheating of parts of the target surfaces, and the contamination of sensitive elements (e.g., optical sensors) on the target surface by the combustion products. To avoid these problems, usually the final braking burn is performed at a residual distance from the target, and then the chaser move at constant and slow speed up till the docking. In the case of berthing, the chaser-target relative distance at the end of the maneuver will be high enough to avoid the consequences related to the exhaust plume.

The final approach phase is the phase under study in this thesis.

Mating

The mating phase can be realized through docking or berthing. In the case of docking, the chaser GNC system controls the vehicle state parameters required to connect the docking interfaces. In the case of berthing, the chaser GNC system delivers the vehicle at nominally zero relative velocities and angular rates to a meeting point, where a manipulator, located either on the target or chaser vehicle, grapples it, transfers it to the final position and inserts it into the interfaces of the relevant target berthing port. In the case of ADR, also the capture should be considered as a particular case of the mating phase.

Departure

The phase of departure is the phase in which, once that the mechanical link between chaser and target is released, the chaser moves on a non-returning trajectory until it reaches a sufficiently safe distance with respect to the target, where it can perform large thrust maneuvers to de-orbit or reach another target. As for the final approach, issues such as the thruster plume should be taken into account. In the case of EOL disposal, the chaser will leave target orbit together with the target.

1.2.2 Sensors for rendezvous navigation

During far and close rendezvous, an increasingly accurate knowledge of the relative chaser-target position, orientation and velocities is necessary, with the most strict requirements to be met at capture.

In docking, the driving factor is the useful reception range of the docking mechanism in terms of lateral and angular misalignments. Depending on the type of docking port and vehicle characteristic, the typical ranges are:

- approach velocity: $0.03 - 0.3$ m/s
- lateral alignment: $0.05 - 0.2$ m
- lateral velocity: $0.01 - 0.05$ m/s
- angular misalignment: $1 - 5$ deg
- angular rate: $0.05 - 0.25$ deg/s

For berthing, the requirements are less strict in positioning and more constraining for what concerns relative velocities, with:

- positioning error (along each axis): $0.1 - 0.5$ m
- residual velocities: < 0.01 m/s
- angular misalignment: ≤ 10 deg
- angular rate: ≤ 0.1 deg/s

These docking and berthing constraints concern the *real* relative state. The errors in positioning and residual velocities arise not only from navigation errors, but from multiple sources including the presence of orbital disturbances, control errors (i.e., difference from the commanded control forces/torques and the actuated ones) and so on. For this reason, the precision required on the estimated state is higher that the required precision in positioning and residual velocities. During phases controlled in closed-loop (i.e., the final approach phase, which is the phase considered in this thesis), measurement accuracy should be a factor from 2 to 5 times lower than the desired final accuracy.

The sensor used to obtain measurements of the target state during rendezvous operations can be divided into three main families, i.e., radio frequency sensors, satellite navigation systems, and optical sensors.

1.2.2.1 Radio frequency sensors

Radio frequency sensors can provide range, range-rate, line-of-sight (LOS), and even attitude measurements.

The principle of range and range-rate measurement is very simple: a sources on the chaser emits radio-waves and then measures the time-of-flight or alternatively the phase-shift of the received signal (after reflection on the target).

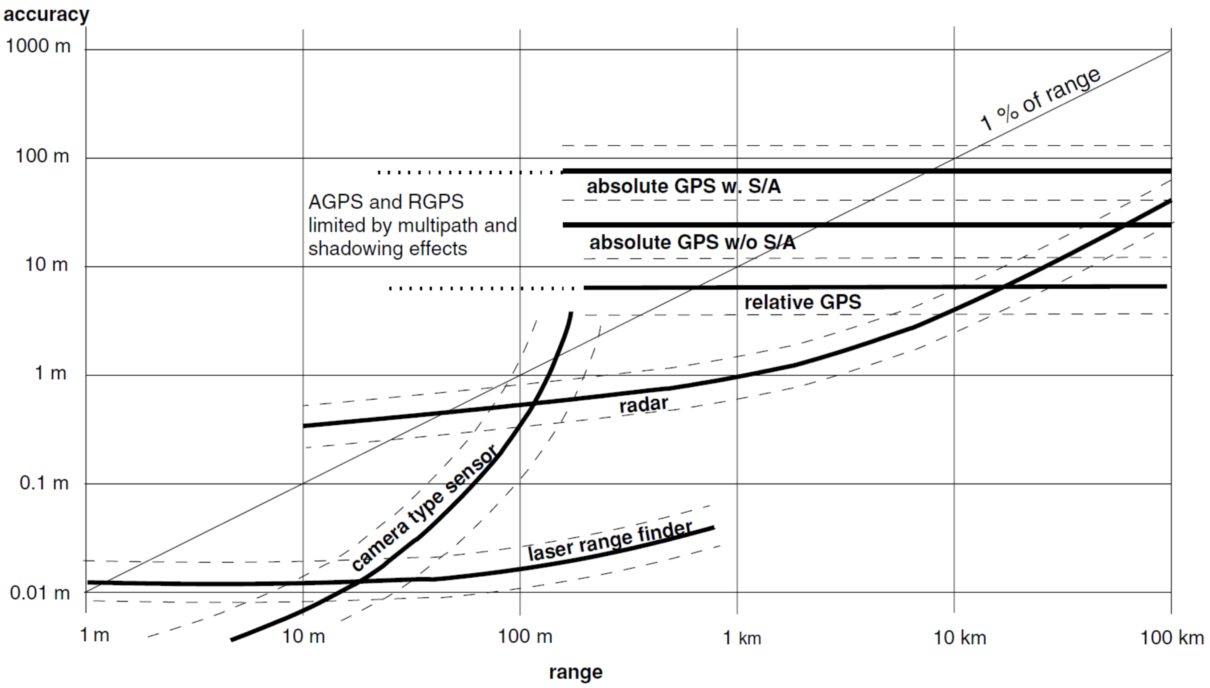


Figure 1.4: Typical operational ranges and measurement accuracies of rendezvous sensors. Images credits: Automated Rendezvous and Docking of Spacecraft, Wigbert Fehse, 2003, Chap.5, Pg. 125.

Radio frequency sensors can also provide LOS measurements and attitude measurements. The use of directional antennas (narrow beam antennas) mounted on a two axis gimbal system allows moving the antenna in order to detect the maximum amplitude of the returning signal and the corresponding LOS. Attitude measurements on the other hand can be obtained only with a cooperative-prepared target on which a rotating pattern is mounted.

The operational range of radio frequency systems is limited by the lowest power of the returning signal that can be detected on the receiver (chaser) side. The maximal operational range is typically of the order of 100 km. The presence on the target side of a transponder that amplifies the received signal before re-transmitting it towards the chaser increases the operational range of the radio frequency sensor. However, this implies that the target is both cooperative and prepared.

In general, radio frequency systems are characterized by a high mass and power consumption, and are used only in the far and middle range rendezvous phases, as they cannot provide the required performance at shorter range. In the early development of automatic on-board rendezvous systems, when no alternatives were available, all sensor design was based on radio frequency techniques. An example is the Russian system *Kurs*, which was (and is still) used for RDV navigation of the Soyuz and Progress vehicles with the

MIR and the ISS (both cooperative and prepared targets). However, as a result of the development of new techniques and technologies, in many cases either measurement performance or mass and power consumption are nowadays more advantageous with other sensor principles, which are, in the far and medium range, satellite navigation systems, and, in the short range, optical Sensors.

1.2.2.2 Satellite navigation systems

The principles of satellite navigation systems (now known under the acronym GNSS, for Global Navigation Satellite System) are well known. The system is composed by three segment, the *space segment*, the *ground segment*, and the *user segment*. The user location is determined as the intersection of three spheres (centered in three satellites belonging to the space segment) and the time-bias resulting from un-synchronization of the user clock can be corrected with a fourth measurement. GNSS positioning system ensures very accurate measurements of the absolute position of a S/C equipped of a receiver, and is therefore used for absolute navigation of the chaser. However, satellite navigation enables relative measurements of position and velocities between chaser and target if both vehicles are equipped with appropriate navigation receivers and if the raw data of one of the receivers can be transmitted to the other vehicle. Such a system is known under the name of Relative GPS. Relative navigation can be performed over relatively large distances between the vehicles (from 50 km or more) with an accuracy of the order of 10 m for position and 0.05 m/s for velocity, provided that a communication link is available. Thus, such approach is not applicable to ADR missions and generally to RDV with non-cooperative targets. The range and range-rate measurement accuracy achieved by satellite navigation techniques is independent of the actual range between chaser and target. The overall expenditure on power and mass is, for navigation systems based on satellite navigation receivers, considerably lower than the one required by radio frequency systems for the same operational range.

1.2.2.3 Optical sensors

Optical sensors are the optimal sensors for the close proximity operations (see Fig.1.4), with an accuracy that increases as the distance chaser-target decreases, which is in line with the RDV navigation requirements.

When Fehse published its book 20 years ago, he considered within the family of optical sensors only scanning laser range finders and camera sensor, and stated that both sensors required corner-cube reflectors to be mounted on the target. However, he also acknowledged that, with the advance of image recognition techniques and other new technologies,

it could be expected that in future a larger variety of sensor principles would have been both available and suitable for automated rendezvous.
This has eventually happened, and we are now in a situation not only where more sensors solution are available, but also where new image processing and computer vision techniques enable the estimation of the pose of an non-prepared target. The sensors that are currently considered for rendezvous are optical cameras in the visible range (both in monocular and stereo configuration), optical camera in the infra-red ranges, and LIDAR. LIDAR is the acronym of Light Imaging, Detection and Ranging. It is a laser scanning technique that consists in illuminating the target with a laser light and measuring the reflected pulses with a sensor, providing a digital 3D representation of the target. It can provide depth information starting from a distance of 20-10 km according to ([Miravet et al., 2008]), even if [Christian and Cryan, 2013] documents LIDAR utilization up to 5 km range for space applications. LIDAR can be classified in three major classes with respect to their functioning mechanism [Christian and Cryan, 2013]:

- **Scanning LIDARs:** a narrow laser beam is swept over all the sensor Field Of View (FOV), and the return signal is collected by a single detector. Laser direction is changed by a set of mirror/lenses and other devices, thus allowing high precision, very high resolution point clouds and target tracking. Since there is only a sensor, scanning LIDAR is quite simple to calibrate, but it is a potential source of hardware failure since it has moving parts. One of the major drawback of this kind of LIDAR is that it takes a finite amount of time to scan the whole FOV and this can result in motion blur if the object in the scene undergoes substantial relative motion.

- **Detector Array LIDARs:** the entire FOV is illuminated by a broad laser pulse at once, while the laser Time-Of-Flight (TOF) is measured at pixel level on a detector array, mainly as a conventional camera does. Detector Array LIDARs are more difficult to calibrate with respect to Scanning LIDARs since they rely on multiple detectors, but they are also more reliable since they don't have any moving part. The density of the 3D cloud of points is lower than the one obtained with Scanning LIDARs since the size of the detector arrays must be limited. Within the Detector Array LIDARs there are Flash LIDARs, which send out a laser pulse and measure the TOF when the return pulse is detected at each pixel, and Continuous Wave LIDARs, which on the other hand modulate the intensity of the laser and measure the phase difference between emitted and received signal. Continuous Wave technique for the measurement of the TOF can be applied also to Scanning LIDAR. The limitation of this technique (Continuous Wave) is the phase integer ambiguity problem: the maximal measurable range r_{max} is limited by the modulation frequency f_{mod}

according to the formula: $r_{max} = c/2f_{mod}$, where c is the speed of light. Considering a classical LIDAR operating at 15 MHz, the range is limited to about 10 m. For this reason, Flash LIDARs, which can operate in a range from many kilometers to docking are preferred to Continuous Wave LIDARs. Within Flash LIDARs, Detector Array Flash LIDARs can be preferred to Scanning Flash LIDARs since the whole scene is illuminated at once, so that no noticeable motion blur is observed in Detector Array LIDAR images (if the object is moving at a speed substantially lower than the speed of light).

- **Spatial Light Modulator LIDARs:** This class of sensor is still under development and is years away from being space-qualified. The Spatial Light Modulator sequentially illuminates subsets of the scene with a sequence of known patterns, and then the time history of the laser return is measured by a single detector. The 3D points cloud is found using Compressed Sensing Algorithms. Even if this sensor seems promising since it has no moving parts and only a single detector, it must rely on approximations of the scene geometry.

A Scanning LIDAR has been used on the Space Shuttle to help docking with the MIR Space Station, the Hubble Space Telescope and the ISS, as well as on ESA's ATV. This latter was capable to provide range and line-of-sight measurement from 300 m to docking, and relative position and attitude from 30 m, at the rate of 1 Hz. Another Scanning LIDAR has flown on the ATV and the HTV (i.e., the Japanese transfer vehicle), operating from 1.5 km to docking. A Flash Detector Array LIDAR was selected by SpaceX for the Dragon vehicle in order to perform proximity operations and capture, with a range that didn't exceed 1 km. This kind of LIDAR was selected also for the Orion Multi-Purpose Crew Vehicle (NASA), with a FOV that can be set either to 12 deg (for long range) or to 20 deg (for short range), and a maximal range of 5 km. A LIDAR device has been successfully employed also on a satellite platform, the micro satellite XSS-11 of Lockheed Martin and the Air Force Research Laboratory, for a 23 months mission [Allen et al., 2008], demonstrating the capability of target full pose acquisition and tracking from a distance of 400 m.

However, LIDARs have a relatively high mass and are very power-hungry instruments that can affect the power budget of the chaser. For this reason, passive sensor such as visible and infrared camera are considered. A single camera (i.e., also referred to as *monocular vision*) provides a bi-dimensional information. Thus, when looking at an unknown object, it will be possible to determine its relative position only up to an unknown scale factor, i.e., it would not be possible to understand whether the object is big and far, or small and close. For this reason, monocular vision is said to be *bearing-only*. This additional degree of freedom can be solved if the observed object is known (i.e., *model-based* pose estimation, see Sec.3.2). The use of two monocular cameras is referred to as *stereo-vision* and allows to get range measurements even for unknown targets. In fact, the difference between the position of a given feature at a given instant in the two images frames provides an additional relation that allows determining the scale factor and measuring the range. However, the accuracy of stereo-vision systems is inversely proportional to the distance (i.e., the *baseline*) between the cameras.

Visible cameras operate in the visible range (i.e., 380-740 nm), and for this reason are more sensitive to the illumination conditions and are likely to be coupled with artificial illumination devices to ensures the navigation even in the absence of sun enlightenment. On the other hand, thermal infra-red (TIR) cameras operate in the range of 8-16 μm and therefore provide images that are independent on the illumination conditions (even if the sun-direction still plays a role in heating the S/C surfaces). However, TIR cameras are usually more expensive and with a lower resolution than VIS cameras. It is claimed in [Opromolla et al., 2017b] that attitude measurements are not obtainable using only TIR images due to the low resolution of TIR camera. However low level fusion of TIR images with VIS images (and possibly also Ultra-Violet images [ESA, 2016]) can provide enhanced images with high resolution and high robustness towards adverse illumination conditions. The range of operation of camera sensors depends on their FOV. A very narrow FOV enables position estimation starting from some hundreds of meters, and attitude estimation from about 100 m, depending on the size of the target. However, small FOVs require higher pointing precision and become unusable at shorter distances. The trend is to associate a Narrow Angle Camera (NAC, with FOV around 5 deg) with Wide Angle Camera (WAC, with FOV around $30-40$ deg) to cover all the close proximity phases of rendezvous.

1.2.3 Where are we now?

In Sec.1.1.1.3 we have anticipated that, at the moment of writing, the Mission Extension Vehicle (MEV-1) manufactured by Northrop Grumman and launched the 9 October 2019 is the first commercial satellite servicing spacecraft ever built, and the first to perform an automated rendezvous with a non-cooperative and non-prepared target.

Globally, the MEV relies on existing technologies. The spacecraft is based on Northrop Grumman's GEOStar design, and it is equipped with an electrical propulsion system (i.e., ion thrusters for orbit raising and orbit keeping), but it also has a small tank of liquid hydrazine fuel on-board, mainly for higher-impulse maneuvers during the docking sequence. The MEV-1 also uses a docking structure based on devices developed for the Space Shuttle servicing flights to the Hubble Space Telescope. The navigation sensors installed on the MEV-1 are similar to the rendezvous aids used on the Northrop Grumman's Cygnus space station supply ship. The MEV-1 is equipped with **visible and infrared cameras**, along with a **scanning LIDAR** [Spaceflight Now, 2020].

All the rendezvous operations have been carried out in the so-called *supervised autonomy.* The nominal planned close proximity operations are detailed in the following. As the MEV-1 reaches a hold point, named *far hold*, placed at 80 m above the target (i.e., on the + R-bar side), it waits for ground command to proceed. Then, MEV-1 moves forward till a *near hold* point placed at a relative distance of 20 m, and waits for another clearance from ground. After this clearance, it proceeds towards the last way-point, placed about 1 m from the target. These approaches are performed very slowly and last some hours. After the final go-aheads (both from the client ground control and the servicer ground control), the MEV-1 deploys a stinger into the nozzle of the apogee engine of Intelsat, pulling the two satellites together [Spaceflight Now, 2020].

However, before attempting the actual docking, the MEV-1 has performed several *rehearsal* approach. In fact, MEV-1 arrived in the vicinity of IS-901 on February 5^{th}, and until February 24^{th} it did a number of approaches, allowing ground control calibrating sensors and algorithms, and testing out the procedures [Spaceflight Now, 2020].

From this paragraph, three important points must be underlined:

1) The rendezvous has been carried out in supervized autonomy, which means that man is still in the loop to provide go/no-go clearance.

2) Before the final approach maneuver and docking, a series of approaches (lasted 3 weeks) has been performed to allow sensor and algorithms calibration. Even if it is not mentioned in the source, it is very likely that this calibration has been performed on-ground from streamed images of IS-901 acquired by the MEV-1 cameras, and not autonomously by the MEV-1 embedded algorithms. This underline one of the major problem related to autonomous navigation for space RDV, which is the lack of extended databases of images acquired in space with whom testing the algorithms. Algorithms are tested and calibrated on-ground with synthetic images of the target, which means that they are going to need a re-calibration when running with real space imagery. At the moment, it is very unlikely to think of a vision-based autonomous rendezvous that can be performed without this "rehearsal" phase.

3) The MEV-1 is equipped with visible, TIR, and LIDAR cameras. Even if the tendency of the literature works is to provide innovative solutions which can work with a single sensor, it is also true that a multisensor configuration ensures an higher ro-

bustness and redundancy of information. The configuration chosen for the MEV-1 provides robustness towards bad illumination conditions (with the thermal infrared camera and the LIDAR), and ensures a good knowledge of the relative range thanks to the LIDAR. Moreover, each one of the sensor can potentially enable the navigation even if the other two are unavailable, ensuring redundancy.

1.3 Thesis contributions and contents

As anticipated in the previous sections, this thesis will focus on the relative autonomous navigation of the chaser in the final approach phase of a rendezvous with a non-cooperative and non-prepared target. The navigation solution should provide relative position estimation from a distance of about 100 m, and attitude estimation from a distance of about 40-20 m from the target. The relative position and relative attitude of the target with respect to the chaser are referred to as relative *pose* of the target.

In Sec.1.2.2 it has been explained how optical sensors are the best candidate for this close proximity phase. Navigation that exploits measurements coming from optical sensors is referred to as *vision-based* navigation. In Par.1.2.2.3 different sensors have been introduced, namely the VIS and TIR cameras (both in monocular and stereo configuration), and the LIDAR. In Sec.1.2.3 we have shown how the first ever automated commercial OOS vehicle was equipped with all the optical sensor previously discussed. Multiple sensors configuration was chosen to ensure redundancy and increase the reliability of the mission. LIDARs, VIS cameras, and TIR cameras are in some way complementary, providing different kinds of information which needs different kind of processing algorithms. LIDAR, in particular, provide 3D information independent from illumination conditions and which can be easily translated into pose measurements. From the 3D points-cloud provided by LIDARs, the pose can be initialized with template matching and Principal Component Analysis (PCA,[Wold et al., 1987]) techniques [Jasiobedzki et al., 2005, Taati and Greenspan, 2005], and tracked with methods such as the Iterative Closest Point (ICP) [Opromolla et al., 2015a, Opromolla et al., 2015b, Opromolla et al., 2017a, Liu et al., 2016], which are also suitable for stereo configurations [Jasiobedski et al., 2001]. However, LIDARs are power hungry devices with a relatively high mass. For this reason, systems based solely on monocular cameras are recently gaining popularity as cost effective solutions ensuring rapid pose estimation under low power and mass requirements. Moreover, monocular pose estimation methods can provide also range measurements when observing a known target, which is the case of the majority of ADR and OOS scenarios. This makes monocular configuration preferable also to stereo configuration, as stereo configuration implies an higher mass and a lower operational range limited by the size of the chaser platform. Moreover, even with stereo configuration, it would be necessary to have some backup monocular-based navigation algorithm to still ensure the realization of the mission if one of the two cameras becomes unavailable.

For the above reasons, this thesis will focus on the development of a navigation solution for close-proximity operations based on the exploitation of a monocular visible camera. This does not exclude a priori the possibility of coupling the VIS camera with a TIR camera to perform low-level fusion and provide to the navigation algorithms an enhanced image. The objective is to develop a robust solution which takes into account also the reduced computational resources available on space enabled processors.

The problem of relative navigation can be divided into two major issues. The first issue is to translate the information provided by the sensor (i.e., the image) into a pose measurement. The families of algorithms that deal with this problem are known under the name of Image Processing (IP) and Computer Vision (CV) techniques. This issue can be divided into two sub-topics. On the one hand, there is the so-called frame-by-frame *tracking* problem. In the tracking problem, the IP-CV algorithms provide a measurement of the current observed pose exploiting the knowledge of the estimated pose at the previous instant. On the other hand, there is the so-called *pose acquisition* (or initialization) problem, where the IP-CV algorithms must provide a measurement of the current observed pose without any prior information about the state of the target (i.e., this condition is also referred to as *lost-in-space*). These algorithms are referred to as pose estimation by *detection* algorithms. The algorithms that ensure robust tracking are usually very different from the algorithms providing a robust pose initialization, and for this reason these two aspects will be covered in two different chapters.

The second issue is to integrate the obtained measurements into the Navigation function of the chaser, where different measurements from multiple sensor and/or algorithms can converge. The objective of the Navigation function is therefore to obtain, from the measurement it receives, a robust estimate of the observed state.

This latter issue is discussed in Chapter 2. It has been placed before the chapters dealing with the IP-CV problem because it allows us to introduce concepts that are necessary at the understanding for the following chapters. In Chapter 3 the problem of pose tracking will be addressed, while Chapter 4 will deal with the problem of pose estimation by detection. Finally, in Chapter 5 the solutions developed in each chapter are integrated and tested in a RDV scenario.

The main contributions of the thesis are summarized below:

Chapter 2: Navigation Function

This Chapter has two main contributions: it provides a detailed modelization of the relative dynamics within the estimation filter, and it proposes a comparison of two delay management techniques suitable for the space rendezvous applications. In fact, IP-CV algorithms usually have high latency, implying that the chaser navigation filter has to fuse delayed and multi-rate measurements. The selected methods are the Filter Recalculation method -which always provides an optimal estimation at the expense of a high computational load- and the Larsen's method -which provides a faster solution whose optimality lies on stronger requirements. The application of these techniques to the space rendezvous problem is discussed and formalized. Finally the methods are compared in a Monte-Carlo campaign, aimed at demonstrating whether the loss of performance of Larsen's method due to its sub-optimality still enables target state robust tracking.

Chapter 3: Frame-by-frame Tracking

In this Chapter, the theory underlying monocular model-based tracking for pose estimation is recalled and an innovative tracking algorithm is formally developed and implemented. The algorithm is based on the detection and matching of the observed target silhouette contours with the 3D geometric model of the target. This algorithm provides fast and very accurate measurements of the relative pose of the target. The algorithm is then coupled with the dynamic Kalman Filter introduced in Chap.2. The performance of the algorithm is tested in different scenarios with high fidelity synthetic images.

Chapter 4: Pose Estimation by Detection

The Chapter proposes a method based on global descriptors to estimate the pose of a known object using a monocular camera, in the context of space rendezvous between an autonomous spacecraft and a non-cooperative target. The method estimates the pose by detection, i.e., it require no prior information about the pose of the observed object, making it suitable for initial pose acquisition and for the monitoring of faults in other on-board estimators. An approach is presented to fully retrieve the object's pose using a pre-computed set of invariants and geometric moments. Three classes of global invariant features are analyzed, based on complex moments, Zernike moments and Fourier descriptors. The robustness of the different invariants is tested under various conditions and their performance is discussed and compared. The method offers a fast and robust solution for pose estimation by detection, with a lower computational complexity than other space-qualified pose acquisition algorithms.

Chapter 5: Integrated Solution

In this Chapter the algorithms described in the previous Chapters are integrated in order to provide an integrated solution enabling the autonomous navigation from a relative distance of 100 m from the target. The integrated solution is tested in different scenarios.

Chapter 2

Navigation Function

Contents

The task of the Navigation function is to provide the Control and the Guidance functions with the information about the current state of the vehicle and the environment surrounding. This function is implemented as a digital filter processing inputs obtained from different sensors or via communication links from external sources in order to obtain the best estimate. Filters relying on representative dynamic modes allow propagating the observed state even when sensor information is only intermittently available. In the navigation function of autonomous robotic systems, the algorithms generally used for the estimation belong to the family of the Kalman Filters (KF). The Kalman Filter is an optimal observer in the sense that it minimizes the sum of the estimation error variances for all the state vector components. In Appendix B the theory underlying the Kalman theory is recalled, and the filter formulations and conventions that are going to be used in the current chapter are detailed.

Vision-based navigation relies on the use of optical sensors coupled with image processing and computer vision algorithms to obtain a measurement of target relative pose. These algorithms usually have high latency, implying that chaser navigation filter must be able to fuse delayed and multi-rate measurements. This chapter provides a detailed modelization of the vision-based space rendezvous dynamics estimation problem, and it proposes a comparison of two delay management techniques suitable for this application. The selected methods are the Filter Recalculation method -which always provides an optimal estimation but has a high computational load- and the Larsen's method -which provides a fast solution that is optimal only under certain conditions. The application of these techniques to the space rendezvous problem is discussed and formalized. The methods are then compared in a Monte Carlo campaign, aimed at demonstrating whether the loss of performance of Larsen's method due to its sub-optimality still enables target robust tracking.

2.1 Introduction

As anticipated, IP-CV tracking algorithms can have a relatively high latency. This results in a delay between the time instant of data acquisition (i.e., when the image is captured by the camera) and the time instant when the processed measurements are available and ready to be fused into the navigation filter. The navigation filter will therefore need to merge infrequent and delayed measurements of the relative target state. Since the tracking can be provided by additional sensors (e.g., visible camera, infrared camera, LIDAR) and algorithms (e.g., contour-based, 3-D points cloud based, marker-based algorithms) with different computational burden and latency, the filter must be able to fuse multi-rate measurements as accurately as possible. While slow measurements are available after a certain delay, fast measurements (i.e., referred to as *interim* measurements) are available at a higher rate and processed almost instantaneously. Interim measurements have therefore to be fused within the delay period in order to provide the best estimate.

2.1.1 Delay management techniques in space applications

The problem of delay management in space applications and more precisely in space RDV scenarios has been sporadically assessed [Zhou et al., 2012, Rems et al., 2017, Benninghoff et al., 2014]. These works propose delay management techniques for the estimation of the chaser-target relative translational dynamics -which is described by the Clohessy-Wiltshire-Hill (CWH) equations [Fehse, 2003, Curtis, 2013]- and are suitable only for linear systems where no interim measurements are considered. No solutions have already been presented for the estimation of target rotational dynamics, which is, in addition, represented by a non-linear model. The current recommended best practice for S/C attitude estimation is the M-EKF (Multiplicative-Extended Kalman Filter, [Markley, 2003]), where both attitude quaternion and rotation rate are measured and no consistent delay affects the measurements. However vision-based navigation algorithms usually provide only pose measurements, without velocity nor rate, and these measurements can be affected by a substantial delay. Some RDV operations may require the synchronization of chaser motion with target motion, implying the need of knowing also target velocity and rotation rate (which are not directly measured). In the case of high rotation rates typical of a tumbling object, a kinematic filter relying only on attitude measurements is not capable to converge to the right values of rate. Nevertheless these variables can be estimated using a dynamic filter instead of a kinematic one. For this reason, the target rotational state estimation will be formulated as an Additive Extended Kalman Filter (A-EKF). Finally, this filter must be capable of managing multi-rate and delayed measurements.

2.1.2 Delay management techniques in literature

Outside from space engineering, in areas such as the automation industry, there is a vast literature on Kalman Filtering techniques with delayed, infrequent and multi-rate measurements. In [Gopalakrishnan et al., 2011] methods are classified into two main families: “state augmentation approaches” and methods which fuse the measurements on arrival. State augmentation methods rely on augmenting the current state with appropriate past information required to fuse the delayed measurements. These methods, such as the *fixed-lag smoothing* [Simon, 2006, Gelb, 1974], provide optimal estimates and can be extended to other filters (e.g., Particle Filter, Unscented Kalman Filter), but are suitable only for fixed-delay measurements. Moreover one major drawback is the fact that the size of the system increases as the delay increases, leading thus to a proportional increase of the computational load. State augmentation methods can be therefore very suitable for applications where delays consist in small number of samples ([Hsiao et al., 1996, Kaszkurewicz and Bhaya, 1996]) or applications where the computational burden is not an issue, such as industrial process control. As a consequence they appear to be inapplicable to space RDV due to the limited on-board computational resources of S/C. On the other side, methods that fuse the delayed measurements on arrival can handle large and variable delays with a reasonably low computational load, while granting the optimality of the estimation under certain intervals and conditions. These methods can incorporate interim measurements while merging the slowest ones, which makes them suitable for domain such as autonomous navigation. For these applications, fast measurements coming from fast sensors need to be fused during the delay period, which is usually associated to the processing of slow measurements coming from vision-based estimation ([Larsen et al., 1998]).

Within this family of delay management techniques, one method always provides an optimal estimate even in presence of interim measurements: the *Filter Recalculation method*. This method has been used in chemical process control industry ([Prasad et al., 2002, Li et al., 2004]). It can be applied also to non-linear systems without any loss of optimality, except for the one naturally introduced by the Extended Kalman Filter (EKF), at the expense of a high computational load. On the other side, the method that provides the best trade-off between optimality and computational burden is the *Larsen's method* [Larsen et al., 1998]. This method has been theorized for linear systems but can be extended also to non-linear systems with the introduction of certain approximations that will be further developed in the sequel (Sec. 2.3.2.2).

The aim of this chapter is to offer a comparison of these two delay management techniques, which have never been applied to the vision-based autonomous rendezvous navigation problem in space, and especially to the estimation of the highly non-linear target relative rotational dynamics. In particular, we will analyze the loss of performance of Larsen's method with respect to Filter Recalculation method when the former is in suboptimal conditions. It will allow to assess whether this technique is suitable for vision based RDV with non-cooperative targets. The chapter is structured as follows: in Sec.2.2 the techniques are formalized, underlining their differences in optimality and computational burden. In Sec.2.3 the implementation of these techniques is described towards the vision-based autonomous RDV problem, and in Sec.2.4 the performance of the filters is investigated under different sources of uncertainties (e.g., state initialization, knowledge of target inertia matrix, knowledge of chaser control forces, measurement noise). In Sec.2.5 the conclusions are drawn and an overview of the future developments is presented.

2.2 Filter Equations

In this section the formulation of the delay management techniques is presented. The methods are implemented on a linear time-discrete system for sake of clarity but will be extended to a non-linear continuous system in Sec.2.3. A linear discrete system observed by non-delayed measurements, where the process noise w_k and the measurement noise v_k are assumed to be zero-mean Gaussian white-noise processes (i.e. the errors are not correlated forward or backward), can be put in state-space form as follows [Gibbs, 2011]:

$$\left\{\begin{array}{rl} x_k = & A_k x_{k-1} + B_k u_k + w_k \\ y_k = & C_k x_k + v_k \end{array}\right. , \quad \text{with} \quad \mathbb{E}\Big[w_k w_j^T\Big] = \left\{\begin{array}{cc} 0 & k \neq j \\ Q_k & k = j \end{array}\right. , \quad \mathbb{E}\Big[v_k v_j^T\Big] = \left\{\begin{array}{cc} 0 & k \neq j \\ R_k & k = j \end{array}\right. \tag{2.1}$$

The associated KF formulation is briefly recalled here for sake of clarity, and fully developed in Appendix B.1.2. The KF is divided in the following steps:

- Prediction of the *a priori* estimate of the state and the state error covariance matrix ($\hat{x}_{k|k-1}, P_{k|k-1}$, Eq.(2.2)).
- Computation of the optimal Kalman gain K_k which minimize the *a posteriori* estimate of the state error covariance $P_{k|k}$ (Eq.(2.3)).
- Update of state and covariance matrix (Eq.(2.4)).

$$\text{prediction} \quad \left\{\begin{array}{rl} \hat{x}_{k|k-1} = & A_k \hat{x}_{k-1|k-1} + B_k u_k \\ P_{k|k-1} = & A_k P_{k-1|k-1} A_k^T + Q_k \end{array}\right. \tag{2.2}$$

$$\text{gain computation} \quad K_k = \quad P_{k|k-1} C_k^T (C_k P_{k|k-1} C_k^T + R_k)^{-1} \tag{2.3}$$

$$\text{update} \quad \left\{\begin{array}{rl} \hat{x}_{k|k} = & \hat{x}_{k|k-1} + K_k (y_k - C_k \hat{x}_{k|k-1}) \\ P_{k|k} = & (I - K_k C_k) P_{k|k-1} \end{array}\right. \tag{2.4}$$

When delayed measurements are presents, at instant k the system in Eq.(2.1) receives a delayed measurement corresponding to time instant s ($s = k - N_d$, N_d number of delay samples, as shown in Fig.2.1), such that:

$$y_s^* = C_s^* x_s + v_s, \quad \text{with} \quad \mathbb{E}\Big[v_s v_j^T\Big] = \left\{\begin{array}{cc} 0 & s \neq j \\ R_s^* & s = j \end{array}\right. \tag{2.5}$$

In such a case, Eq.(2.3) is no more optimal and a new solution has to be found in order to compute the best estimates $\hat{x}_{k|k,k^*}$ and $P_{k|k,k^*}$ which take into account the contribution of y_s^*. Usually the optimality of the solution implies a high computational load (i.e., Filter Recalculation method), so that some methods propose sub-optimal solutions which can reduce the computation time (i.e., Larsen's method).

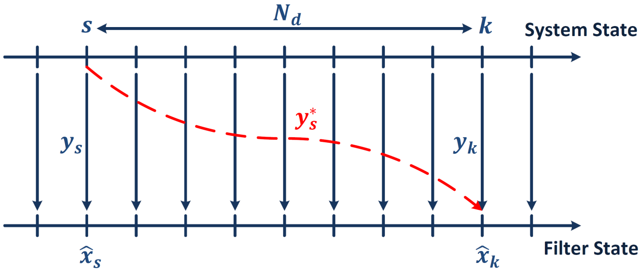


Figure 2.1: System with a N_d-saples delayed output

2.2.1 Filter Recalculation method

The Filter Recalculation method consists in going back to the time step when the delayed measurement was taken, incorporating the measurement and recomputing the entire trajectory of the state until the current step. In so doing, the whole estimate time history will be optimal. The estimation is made as if two filters were employed: a principal one, which operates at constant rate by processing fast measurements y_k, and a second one, which is activated any time a delayed (i.e., slow and infrequent) measurement y_s^* arrives. In order to let the secondary filter recalculate an optimal estimate by fusing the delayed measurement and reconstructing the state through the N_d delay samples, any time a slow measurement is acquired (i.e. when a camera captures an image), the predicted state $\hat{x}_{s|s-1}$ and the predicted covariance $P_{s|s-1}$ corresponding to that time step have to be stored. Moreover, in order to update the whole time history, also the input u_{s+i} and the fast measurements y_{s+i} must be stored for any time step going from the sample s (when the slow measurement is taken) to the sample $s+N_d$ (when the slow measurement become available after the processing delay), i.e., for $i \in [1, N_d - 1]$.

The filters operate as follows. At instant s a slow measurement is taken and $\hat{x}_{s|s-1}$, $P_{s|s-1}$ are stored. For all the time steps $s + i$ ($i \in [1, N_d - 1]$) the principal filter processes fast measurements y_{s+i} as in a classical KF (Eqs. (2.2),(2.3),(2.4)), and measurements y_{s+i} and inputs u_{s+i} are stored at each of these steps. At $k = s + N_d$ the slow measurement y_s^* and its corresponding covariance matrix R_s^* become available and the secondary filter is activated. The filter goes back to instant s and computes the optimal update using the full measurements vector $\tilde{y}_s = [y_s, y_s^*]^T$, where y_s are the fast measurements and y_s^* the slow ones:

$$\begin{cases} \tilde{K}_s = & P_{s|s-1}\tilde{C}_s^T(\tilde{C}_s P_{s|s-1}\tilde{C}_s^T + \tilde{R}_s)^{-1} \\ \hat{x}_{s|s} = & \hat{x}_{s|s-1} + \tilde{K}_s(\tilde{y}_s - \tilde{C}_s\hat{x}_{s|s-1}) \\ P_{s|s} = & (I - \tilde{K}_s\tilde{C}_s)P_{s|s-1} \end{cases} \quad , \quad \text{with} \quad \tilde{C}_s = \begin{bmatrix} C_s \\ C_s^* \end{bmatrix} \quad , \quad \tilde{R}_s = \begin{bmatrix} R_s & \emptyset \\ \emptyset & R_s^* \end{bmatrix} \tag{2.6}$$

The optimal estimates $\hat{x}_{s|s}$ and $P_{s|s}$ are then propagated by the secondary filter from instant $s+1$ to instant $k = s + N_d$ according to Eqs.(2.2),(2.3),(2.4), thus clarifying the need of storing the values of interim measurements and inputs. Once the loop has reached instant $k = s + N_d$, the filter has provided an optimal estimate of the current state and of the state error covariance matrix.

In the case where no interim measurements are present, the structure of the filter will be simplified: the state and the covariance matrix will evolve in open loop (only prediction step with no update) as long as no delayed measurement arrives. At the arrival of a delayed measurement, the secondary filter computes the optimal estimate at instant s and then state and covariance are re-projected to the current instant k through N_d prediction steps. This method can become quite expensive in terms of computational burden due to the need of storing variables and the N_d Kalman loops to be performed any time a delayed measurement arrives. Nevertheless, it is the only formulation which provides an optimal estimate even in presence of interim measurements and also for non-linear systems. More precisely, in the case of non-linear systems, the loss of optimality is introduced by the use of the EKF, which is intrinsically sub-optimal due to the linearization of state and measurement equations, and not by the use of the Filter Recalculation method. Finally, the Filter Recalculation method can be easily extended to variable delays.

2.2.2 Larsen's method

The Larsen's (or extrapolation) method was proposed in [Larsen et al., 1998] as an improvement of Alexander's method ([Alexander, 1991]) for delay management in discrete linear system. These methods rely on the computation, throughout the delay period, of a correction term to add to the filter estimate when the delayed measurement becomes available. The main difference between Alexander's and Larsen's methods is that the latter does not need to know, at instant s, neither the covariance matrix R_s^* nor the measurement sensitivity matrix C_s^* of the slow measurement. These matrices are supposed to become available at instant k together with the delayed measurement y_s^*. Larsen's method is therefore suitable for systems relying on measurements processed by IP-CV algorithms, as these algorithms usually process R^* together with y^*.

The structure of the filter is the following. As for the Filter Recalculation method, at time s, when a new slow measurement is acquired, $\hat{x}_{s|s-1}$ and $P_{s|s-1}$ are stored. At each instant $s+i$ ($i \in [1, N_d - 1]$) the classic KF structure in Eqs. (2.2),(2.3),(2.4) is applied using fast measurements. Moreover, the term $M_{s+i} = (I - K_{s+i}C_{s+i})A_{s+i}M_{s+i-1}$ is computed, with $M_s = I$. At time instant $k = s + N_d$, y_s^*, R_s^* and C_s^* become available. The filter firstly computes the Kalman gain K_k and the updates as in Eqs.(2.3),(2.4) using fast measurements y_k. The final correction term M_k^* will be [Larsen et al., 1998]:

$$M_k^* = M_{s+N_d} = \prod_{i=1}^{N_d} (I - K_{s+i}C_{s+i})A_{s+i} \tag{2.7}$$

Then an extrapolated measurement y_k^{ext} is computed to derive a representation of measurement y_s^*, which is referred to instant s, at instant k:

$$y_k^{ext} = y_s^* - C_s^* \hat{x}_{s|s-1} + C_k^* \hat{x}_{k|k-1} \tag{2.8}$$

In [Larsen et al., 1998] Larsen provides the demonstration of the computation of the optimal gain K_k^* and the resulting state and covariance updates, which are:

$$\begin{cases} K_k^* & = M_k^* P_{s|s-1} C_s^{*T} (C_s^* P_{s|s-1} C_s^{*T} + R_s^*)^{-1} \\ \hat{x}_{k|k,k^*} & = \hat{x}_{k|k} + K_k^* (y_k^{ext} - C_k^* \hat{x}_{k|k}) \\ P_{k|k,k^*} & = P_{k|k} - K_k^* C_s^* P_{k|k} M_k^{*T} \end{cases} \tag{2.9}$$

where the Kalman gain K_k^* is actually the Kalman gain K_s^* (i.e., the gain that would have been computed if the measurement y_s^* had become available at instant s) pre-multiplied by the Larsen correction term M_k^*.

In the presence of interim measurements this method performs sub-optimally: at each interim step, the gain K_{s+i} is computed using a covariance matrix $P_{s+i|s+i-1}$ that is not optimal because it has not yet taken into account the contribution of y_s^*. Even if in this case the method performs sub-optimally, it requires only two matrix multiplications at each time instant and the storage of two variables any time a slow measurement is acquired, as in the case without interim measurements. Moreover there is no need of storing interim measurements and inputs. As the Filter Recalculation method, Larsen's method can be extended to variable delays which are not known *a priori.*

2.2.3 No interim measurements case

In the absence of interim measurements and for a linear system, Filter Recalculation and Larsen's methods give the same estimates. The theoretical demonstration of this result is an original contribution of this study and is provided in the current section. Let us assume that the delayed measurement y_s^* and its corresponding sensitive matrix and noise covariance matrix are available at instant s. The update at instant s is then given by:

$$\begin{cases} \hat{x}_{s|s} = & \hat{x}_{s|s-1} + K_s^*(y_s^* - C_s^*\hat{x}_{s|s-1}) \\ P_{s|s} = & (I - K_s^*C_s^*)P_{s|s-1} \end{cases} \quad \text{with} \quad K_s^* = P_{s|s-1}C_s^{*T}(C_s^*P_{s|s-1}C_s^{*T} + R_s^*)^{-1} \tag{2.10}$$

For all the time steps $k = s + i$ $(i \in [1, N_d - 1])$ no interim measurements are processed, such that state and error covariance matrix evolve in open-loop:

$$\begin{cases} \hat{x}_{k|k} = \hat{x}_{k|k-1} = A_k\hat{x}_{k-1|k-1} + B_k u_k \\ P_{k|k} = P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k \end{cases} \tag{2.11}$$

After N_d loops, at $k = s + N_d$, the estimated state $\hat{x}_{s+N_d|s+N_d}$ will be:

$$\hat{x}_{s+N_d|s+N_d} = \left(\prod_{i=1}^{N_d} A_{s+i}\right)\hat{x}_{s|s} + \sum_{i=1}^{N_d}\left(\prod_{j=i+1}^{N_d} A_{s+j}\right)B_{s+i}u_{s+i} \tag{2.12}$$

Recalling the expression of $\hat{x}_{s|s}$ in Eq.(2.10) and calling z_s^* the residual $y_s^* - C_s^*\hat{x}_{s|s-1}$, Eq.(2.12) becomes:

$$\hat{x}_{s+N_d|s+N_d} = \left\{\left(\prod_{i=1}^{N_d} A_{s+i}\right)\hat{x}_{s|s-1} + \left[\sum_{i=1}^{N_d}\left(\prod_{j=i+1}^{N_d} A_{s+j}\right)B_{s+i}u_{s+i}\right]\right\} + \left(\prod_{i=1}^{N_d} A_{s+i}\right)K_s^* z_s^* \tag{2.13}$$

where the expression inside the curly brackets corresponds to the open-loop evolution of the predicted state $\hat{x}_{s|s-1}$ from s to $s + N_d$, and the term $\prod_{i=1}^{N_d} A_{s+i}$ is the correction term M^* of Eq.(2.7) in the absence of interim measurements, and, as a consequence, in the absence of interim Kalman gains. It therefore appears that using the delayed measurement at instant s and then propagating the *a posteriori* estimate $\hat{x}_{s|s}$ (i.e., Filter Recalculation method) is equivalent to propagate the *a priori* estimate (prediction) $\hat{x}_{s|s-1}$ and to perform the update at $k = s + N_d$ by pre-multiplying the update $K_s^* z_s^*$ by a correction factor, which is Larsen's correction factor. It becomes also clear that Larsen's method exploits the superposition property of linear systems to project in the "future" the update $K_s^* z_s^*$, which can be seen as a Δx_s that is propagated through the same transformation of $\hat{x}_{s|s-1}$.

The same demonstration can be done for the state error covariance matrix. At $k = s+N_d$, the estimated covariance $P_{s+N_d|s+N_d}$ will be:

$$\begin{aligned} P_{s+N_d|s+N_d} = & \left(\prod_{i=1}^{N_d} A_{s+i}\right) P_{s|s} \left(\prod_{i=1}^{N_d} A_{s+i}^T\right) + \sum_{i=1}^{N_d}\left(\prod_{j=i+1}^{N_d} A_{s+j}\right) Q_{s+i} \left(\prod_{i=1}^{N_d} A_{s+i}^T\right) \\ = & \left\{\left(\prod_{i=1}^{N_d} A_{s+i}\right) P_{s|s-1} \left(\prod_{i=1}^{N_d} A_{s+i}^T\right) + \left[\sum_{i=1}^{N_d}\left(\prod_{j=i+1}^{N_d} A_{s+j}\right) Q_{s+i} \left(\prod_{i=1}^{N_d} A_{s+i}^T\right)\right]\right\} - \left(\prod_{i=1}^{N_d} A_{s+i}\right) K_s^* C_s^* P_{s|s-1} \left(\prod_{i=1}^{N_d} A_{s+i}\right) \end{aligned} \tag{2.14}$$

where the term in the curly brackets is the open loop evolution of $P_{s|s-1}$ and the term outside the curly brackets is equal to Larsen covariance update in Eq.(2.9).

This demonstration shows how, in the particular case of the absence of interim measurements (and therefore in the absence of interim updates of state and state error covariance matrix) the Filter Recalculation method -which goes back at instant s, computes the update at s and predicts the state along the delay interval up to the current instant-corresponds to Larsen's method -which exploits the linearity of the system to compute at the current time an update that takes into account the propagation occurred from s to $s + N_d$. In such a case, Larsen's method is preferable since it always requires a lower amount of computation, and a lower storage for values of N_d above a certain threshold, as shown in Sec.2.4.4. Filter Recalculation method should be selected only if the whole optimal time history of the estimate from s to $s + N_d$ needs to be known.

These considerations are valid only for linear systems. Non-linear systems, as it will be shown in section 2.3, cannot exploit the superposition property and force to consider the use of the approximated transition matrix instead of A_k. This hypothesis will introduce approximations also in Larsen's state and covariance update.

2.3 Application to the space rendezvous problem

In this section, the dynamics underlying a space RDV is presented first, before applying the methods discussed in Sec.2.2 to this specific dynamic system.

When expressed at its Center-of-Mass (CoM), the motion of a S/C can be decoupled between the translational motion of its CoM and its rotational motion. Therefore the navigation filters for translational and rotational dynamics will be completely decoupled and the estimation of these two motions will be studied as two different and decoupled problems. Actually, when the relative translational dynamics is modeled according to the CWH equations, a small coupling between translational and rotational motion exist, but can be neglected as long as the chaser is controlled with respect to a reference frame that originates in its CoM [Fehse, 2003]. In this thesis, rotations are described using quaternions, according to Hamilton convention [Sola, 2015] (i.e., $q = q_0 + q_1 i + q_2 j + q_3 k$, where the first element q_0 is the scalar part). A vector x is rotated from a reference frame b (x^b) to a reference frame a (x^a) according to : $x^a = q_{a-b} \otimes x^b \otimes q^*_{a-b}$, with q^* the quaternion conjugate of q and $\otimes$ the quaternion product. In Appendix.A.1 more details about the employed convention are given.

2.3.1 Translational dynamics in space rendezvous

For what concerns the translational dynamics, the scenario of the rendezvous allows the introduction of very powerful hypotheses, such as the fact that the relative distance chaser-target is much smaller than the distance Earth-target. This allows to introduce some simplifications to derive the dynamic model which describes the relative motion of the chaser with respect to the target. This model, in the case of a circular orbit, can be put under the form of a system of linear differential equations called Hill's equations [Fehse, 2003]:

$$\begin{cases} \ddot{x} - 3\omega^2 x - 2\omega\dot{y} &= \dfrac{F_x}{m_c} \\ \ddot{y} + 2\omega\dot{x} &= \dfrac{F_y}{m_c} \\ \ddot{z} + \omega^2 z &= \dfrac{F_z}{m_c} \end{cases} \tag{2.15}$$

where $F_{x,y,z}$ are the control forces acting on the chaser CoM, m_c is the chaser's mass and ω is the target's orbit angular rate. The relative position is expressed in target LOF (Local Orbital Frame), according to the convention used in [Curtis, 2013] (x axis directed as the radial that goes from Earth to the target, z axis directed as the angular momentum of target orbit and y axis that completes the right-handed trihedron). In Appendix C.1 the derivation of this reference frame with respect to the Inertial Earth Centered frame is provided. The homogeneous solution of the system in Eq.(2.15), which leads to the analytical computation of the transition matrix $\Phi(t)$ of the time-continuous linear state-space representation corresponding to Eq.(2.15), is known under the name of Clohessy-Wiltshire equations. The CWH transition matrix is valid only for circular orbits, even if the analytical computation of a transition matrix valid for elliptical orbits is described in [Yamanaka and Ankersen, 2002], which includes the CWH equations as a special case for zero eccentricity orbits. A comprehensive survey and comparison of linear and non-linear models for spacecraft relative motion can be found in [Sullivan et al., 2017]. In Appendix C.2 the derivation of the relative dynamics model according to the CWH equations is fully detailed.

Given the transition matrix $\Phi(t)$ of the time-continuous linear state-space representation corresponding to Eq.(2.15), the time-discrete system can be expressed according to:

$$\begin{aligned} A_k &= e^{AT} = \Phi(T) \\ B_k &= \int_0^T e^{AT} B dt \end{aligned} \tag{2.16}$$

where A, B are the time-continuous state matrices, T is the time step size of the time-discrete system and A_k, B_k the time-discrete state matrices. The demonstration of this correspondence is provided in Appendix B.1.3. The computation of A_k and B_k is carried out in Appendix C.2.2 and results in:

$$A_k = \left[\begin{array}{cccccc} 4-3cos(\omega T) & 0 & \dfrac{\sin(\omega T)}{\omega} & \dfrac{2}{\omega}(1-cos(\omega T)) & & \\ 6\,(\sin(\omega T)-\omega T) & 1 & \dfrac{2}{\omega}(\cos(\omega T)-1) & \dfrac{1}{\omega}(4\sin(\omega T)-3\omega T) & \multicolumn{2}{c}{\emptyset_{4\text{x}2}} \\ 3\omega\sin(\omega T) & 0 & \cos(\omega T) & 2\sin(\omega T) & & \\ 6\omega(\cos(\omega T)-1) & 0 & -2\sin(\omega T) & (4\cos(\omega T)-3) & & \\ & & & & \cos(\omega T) & \dfrac{1}{\omega}\sin(\omega T) \\ \multicolumn{4}{c}{\emptyset_{2\text{x}4}} & -\omega\sin(\omega T) & \cos(\omega T) \end{array}\right]$$

$$B_k = \frac{1}{m_c}\left[\begin{array}{ccc} \dfrac{1}{\omega^2}(1-\cos(\omega T)) & \dfrac{2}{\omega^2}(\omega T-\sin(\omega T)) & \\ \dfrac{2}{\omega^2}(\sin(\omega T)-\omega T) & \dfrac{4}{\omega^2}(1-\cos(\omega T))-\dfrac{3}{2}T^2 & \\ \dfrac{\sin(\omega T)}{\omega} & \dfrac{2}{\omega}(1-\cos(\omega T)) & \emptyset_{4\text{x}1} \\ \dfrac{2}{\omega}(\cos(\omega T)-1) & \dfrac{4}{\omega}\sin(\omega T)-3T & \\ & & \dfrac{1}{\omega^2}(1-\cos(\omega T)) \\ \multicolumn{2}{c}{\emptyset_{2\text{x}2}} & \dfrac{\sin(\omega T)}{\omega} \end{array}\right] \tag{2.17}$$

The system can be therefore written in the form of Eq.(2.1) and both Larsen's and Filter Recalculation methods can be easily implemented and will provide the same results in case of no interim measurements.

In the case of vision-based navigation, the CV algorithms will compute a relative target position in camera reference frame. Assuming that the rotation quaternion from camera to chaser reference frame is known a priori by on-ground calibration, it will be necessary to know at each instant the quaternion q_{ch-LOF}, in order to rotate the measurements from chaser *ch* reference frame into target LOF coordinates before using them in the KF based on Eq.(2.17). This quaternion is needed also to rotate chaser control forces, which are the input of the dynamic system. q_{ch-LOF} can be computed by the quaternion product between the conjugate of chaser absolute attitude quaternion, denoted q_{i-ch} with i the inertial reference frame, and q_{i-LOF}, which is the rotation quaternion from inertial reference frame to target LOF. Eq.(2.18) shows the relation between a vector expressed in *ch* reference frame and the same vector expressed in LOF frame:

$$\begin{cases} q_{ch-LOF} & = q^*_{i-ch} \otimes q_{i-LOF} \\ x^{LOF} & = q^*_{ch-LOF} \otimes x^{ch} \otimes q_{ch-LOF} \end{cases} \tag{2.18}$$

Note the chaser inertial attitude q_{i-ch} is actually estimated by the usual AOCS (Attitude and Orbit Control System) usually based on a traditional M-EKF. Besides, the q_{i-LOF} is related to target orbital parameters, and therefore to its absolute velocity and position. In a RDV scenario q_{i-LOF} (and target absolute position and velocity) can be easily initialized by on-ground telemetries. This allows the initialization of the KF, which is fed by the vision-based measurements. From that moment on, at each time step, the KF-estimated relative pose will be added to the absolute chaser position (i.e., whose estimation can rely on GNSS and accelerometers measurements), resulting in an absolute translational target state from which an updated estimation of q_{i-LOF} can be derived.

2.3.2 Rotational dynamics in space rendezvous

As anticipated in Sec.2.1, certain close proximity operations require the knowledge of the complete rotational state of the target (i.e., attitude quaternion q_{i-tg} and rotation rate ω_{i-tg}^{tg}). The estimation of the absolute rotational dynamics of the chaser is not considered in this work and it will be assumed that both chaser attitude and rotation rate are known under low level of uncertainties (i.e., 0.1 deg as a typical performance of classical sensor configuration such as star-tracker+gyroscope).

As mentioned early, the current best practice for chaser rotational state estimation through a M-EKF [Trawny and Roumeliotis, 2005] cannot be applied directly to the target estimation. In fact the majority of the IP-CV algorithms only provides the relative chaser-target attitude measurement but not necessarily the rotation rate. Therefore, the target angular momentum equation must be added to the state equations in order to allow also the estimation of the rotation rate. In Appendix C.3 the derivation of the rotational dynamics model for a rigid body in space is detailed. The target rotational dynamics prediction modeled is given by the system already derived in Eq.(C.87):

$$\begin{cases} \dot{q}_{i-tg} & = \dfrac{1}{2} q_{i-tg} \otimes \begin{bmatrix} 0 \\ \omega_{i-tg}^{tg} \end{bmatrix} \\ \dot{\omega}_{i-tg}^{tg} & = -I_{tg}^{-1} \left(\omega_{i-tg}^{tg} \times I_{tg} \omega_{i-tg}^{tg} \right) \end{cases} \tag{2.19}$$

where I_{tg} is the inertia matrix of the target at its CoM. The orbital disturbance torques that affects the target dynamics will be modeled in the system as process noises. This second order system is formulated as an A-EKF, in order to ensure observability of the rotation rate from attitude measurements. Moreover, the application of Larsen's method to the M-EKF is not straightforward due to the presence of a multiplicative update. The M-EKF formulation was developed in order to avoid the ill-conditioning problems that could appear in the state error covariance matrix when the quaternion normalization is forced ([Markley and Crassidis, 2014]). In fact quaternion attitude representation, which is the lowest dimensional non-singular representation of the rotation group $SO(3)$, has an additional degree of freedom which is constrained by the normalization constraint. The M-EKF, instead of an additive update of a classic EKF, computes a quaternion error and multiplies it to the predicted quaternion, obtaining a unit quaternion estimate. An in-depth study is carried out in [Carmi and Oshman, 2007] to understand whether or not such a constraint leads to an ill-conditioned P matrix. The work provides the mathematical demonstration that the quaternion estimation error covariance matrix does lead to ill-conditioning problem as its trace tends to zero, but also that this property is not inherited by its corresponding second order Taylor approximation, as the one computed by the A-EKF. This theoretical demonstration explains how several works relying on an A-EKF attitude estimation (e.g., such as [Bar-Itzhack and Oshman, 1985]) have never shown ill-conditioning problems. Moreover [Carmi and Oshman, 2007] has provided a practical demonstration that the covariance matrix computed by the A-EKF is well conditioned even in static and noise-free attitude estimation problem, contrary to predictions made in literature (e.g., [Zanetti and Bishop, 2006, Pittelkau, 2003]). This A-EKF estimation filter has been used in the sequel, and it is illustated through the tests in Sec.2.3 how this well-posedness brought numerically stable results.

The time-discrete measurement equation must be added to the time-continuous dynamic system in Eq.(2.19). The measured quantity is the relative chaser-target attitude quaternion ($y_k = q_{ch-tg}$), which must be expressed as a function of the state vector q_{i-tg}. Moreover q_{ch-tg} also depends on the absolute chaser quaternion q_{i-ch}, which is supposed to be previously estimated by a classic M-EKF. Exploiting the quaternion properties it is possible to write the quaternion product in matrix form:

$$q_{a-b} \otimes q_{b-c} = \Sigma(q_{a-b}) q_{b-c} \quad , \qquad \text{with} \quad \Sigma(q) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{2.20}$$

and therefore:

$$q_{ch-tg} = q^*_{i-ch} \otimes q_{i-tg} = \Sigma(q^*_{i-ch})\, q_{i-tg} \tag{2.21}$$

The time-discrete output equation $y_k = C_k x_k + v_k$ is therefore linear in the variable q_{i-tg}, with:

$$C_k = \begin{bmatrix} \Sigma(q^*_{i-ch}) & \emptyset_{4\text{x}3} \end{bmatrix} , \qquad x_k = \begin{bmatrix} q_{i-tg} \\ \omega^{tg}_{i-tg} \end{bmatrix} \tag{2.22}$$

Since q_{i-ch} is the result of an estimation process, the covariance R_k associated to q_{ch-tg} should take into account also the uncertainties introduced by $\Sigma(q^*_{i-ch})$ through the computation of the composed variance of the function $q_{ch-tg} = f(q_{i-ch}, q_{i-tg})$ according to the *propagation of uncertainties* rules. Since this work focuses on the characterization of the intrinsic performance of the delay management techniques, the true q_{i-ch} will be used during the performance analysis not to introduce a cascading coupling between the covariances of the chaser and the target states.

2.3.2.1 Filter Recalculation method implementation

Since the system dynamics is given by the continuous process in Eq.(2.19) while the measurement presented in Eq.(2.21) is a discrete process, the CD-EKF (Continuous Discrete Kalman Filter [Frogerais et al., 2012, Kulikov and Kulikova, 2014]) structure will be exploited in order to propagate target rotational state and covariance matrix. The derivation of the CD-EKF equations is carried out in Appendix B.2. Both state and covariance prediction equations can be written in the following time-continuous form:

$$\begin{cases} \dfrac{dx}{dt} = & f(t,x) \\ \dfrac{dP}{dt} = & \dfrac{\partial f(t,x)}{\partial x}P(t) + \left(\dfrac{\partial f(t,x)}{\partial x}\right)^T P(t) + G(t)Q(t)G(t)^T \end{cases} \tag{2.23}$$

where the second equation is a differential Lyaponouv equation derived using the approximated state equation $\frac{dx}{dt} \sim \frac{\partial f(t,x)}{\partial x}dx + G(t)w(t)$, with $G(t)$ the input matrix of the process noise [Crassidis and Junkins, 2011]. The jacobian $F(x) = \frac{\partial f(t,x)}{\partial x}$ has an analytical formulation which is derived in Appendix C.3.2, and its evaluation does not require a lot of computational resources. The prediction step, both for the state and the covariance matrix P, is computed by numerical integration from t_{k-1} to t_k of the ordinary differential equations in Eq.(2.23). The numerical integration is done using the explicit fourth order Runge-Kutta (RK) method (see Appendix B.2). Numerical integration procedures may provide solution of P that are not necessarily positive semi-definite matrix, which is in contradiction to the intrinsic properties of the state error covariance matrix. The numerical integration of the class of coupled differential equations in Eq.(2.23) is investigated in [Mazzoni, 2008], which suggests procedures that ensure stable solutions and guarantee positive semi-definite covariance matrices P. Nevertheless, for the rotational dynamics estimation problem, it has not been experienced any issues, neither with the stability of the solution, nor with the properties of the covariance matrix. With a filter run frequency of 10 Hz, a single sub-step for the fourth order RK integration is necessary. The so computed $x(t_k)$ and $P(t_k)$ are the prediction of the state ($\hat{x}_{k|k-1}$) and of the covariance matrix ($P_{k|k-1}$). From this moment on, the CD-EKF will follow the steps of a classical Discrete Kalman Filter to compute the gain K_k, the covariance update and the state update using the discrete measurement y_k.

When the delayed measurement y_s^* arrives, the update at time s is computed using the stored $\hat{x}_{s|s-1}$, $P_{s|s-1}$, and q_{i-ch_s} (that is needed to compute matrix C_s^*). No inputs u_s need to be stored since the disturbance torques acting on the target are not modeled and affect the dynamics as process noises. Then, if no interim measurements are present, $\hat{x}_{s|s}$ and $P_{s|s}$ are propagated through the delay interval using N_d RK integration cycles. On the other side, if interim measurements have to be processed, q_{i-ch} has to be stored for any time step going from s to $s+N_d$, and then the filter implementation follows the steps explained in Sec.2.2.1.

2.3.2.2 Larsen's method implementation

As described in Sec.2.2.2, Larsen's method has been expressly designed for linear system. The correction term M^* requires the knowledge of the transition matrix $\Phi_k = A_k$, implying that, for a non-linear system, an approximation must be computed. Here a second order RK approximation will be used [Carpenter and D'Souza, 2018]:

$$\begin{aligned} \Phi_k &= I + T\,F_+ + \frac{T^2}{2}\,F_\times \\ \text{with} & \begin{cases} F_\times = & F(\hat{x}_{k-1|k-1})F(\hat{x}_{k|k-1}) \\ F_+ = & \dfrac{F(\hat{x}_{k-1|k-1}) + F(\hat{x}_{k|k-1})}{2} \end{cases} \end{aligned} \tag{2.24}$$

where $F(\hat{x}_{k-1|k-1})$ and $F(\hat{x}_{k|k-1})$ are the Jacobian $\frac{\partial f}{\partial x}$ of the state equation f, evaluated respectively in the estimated state at $k-1$ and the predicted state at k, and T is the time stem size. The prediction of the covariance matrix can be done using the discrete Lyaponouv equation presented in Eq.(2.2), adapted for a non-linear system:

$$P_{k|k-1} = \Phi_k P_{k-1|k-1} \Phi_k^T + Q_k \tag{2.25}$$

In fact, as discussed in [Carpenter and D'Souza, 2018], both methods in Eq.(2.23) and Eq.(2.25) can be used to compute the prediction of the state error covariance matrix for non-linear systems. The choice of using the discrete Lyaponouv equation instead of the differential Lyaponouv equation as for the CD-EKF is done in order to save computational resources. Indeed, the matrix Φ_k needs to be computed to implement Larsen's method, and Q_k is computed as follows [Gibbs, 2011] :

$$\begin{aligned} Q_k &= \int_{t_{k-1}}^{t_k} \Phi(t_k,\tau)G(\tau)Q(\tau)G^T(\tau)\Phi^T(t_k,\tau)d\tau \\ &= \int_{t_{k-1}}^{t_k} \left[I + (T-\tau)\,F_+ + \frac{(T-\tau)^2}{2}\,F_\times\right] G(t_k)Q(t_k)G^T(t_k) \left[I + (T-\tau)\,F_+ + \frac{(T-\tau)^2}{2}\,F_\times\right]^T d\tau \end{aligned} \tag{2.26}$$

where $Q(t)$ and $G(t)$ have been considered constant along the interval (which is a good assumption since matrix G depends on target inertia matrix and disturbance torques have a slow dynamics) and equal to $Q(t_k)$ and $G(t_k)$. On the other hand, $\Phi(t_k,\tau)$ has been substituted by the second order approximation of Eq.(2.26). The resulting Q_k is:

$$\begin{aligned} Q_k = \;& Q_G T + \left(F_+ Q_G + Q_G F_+^T\right)\frac{T^2}{2} + \left(F_+ Q_G F_+^T + \frac{1}{2}(F_\times Q_G + Q_G F_\times^T)\right)\frac{T^3}{3} + ... \\ & ... + \left(F_+ Q_G F_\times^T + F_\times Q_G F_+^T\right)\frac{T^4}{8} + F_\times Q_G F_\times^T \frac{T^5}{20} \end{aligned} \tag{2.27}$$

where $Q_G = G(t_k)Q(t_k)G(t_k)^T$. The expression of course can be truncated at lower orders if needed, namely for small values of T.

2.4 Simulations and Performance analysis

The Kalman Filters have been implemented in a full GNC RDV simulator developed in Simulink and have been tested in a Monte Carlo (MC) Campaign. The generated true S/C dynamics takes into account all the principal sources of disturbance in LEO environment, such as atmospheric disturbances, solar pressure and $J2$ corrective term, according to the models described in Appendix C.1. The translational dynamics is generated integrating Gauss planetary equations (which take into account also chaser thrusters acceleration), while the rotational dynamics is generated integrating the angular momentum equation (which takes into account also control torques applied to the chaser). Since the main goal, at this stage, is to compare the "intrinsic" performance of the filtering techniques, the true state is used to compute guidance and control profiles.

The filters are tested first with simulated measurements that are generated by adding a Gaussian noise to the true relative state. It is necessary to go through this stage in order to test the performance of the filters under Kalman optimality hypothesis before coupling it with IP-CV measurements. Measurements resulting from IP-CV algorithms are affected by noise which depends on many factors, such as the intrinsic noise of the sensor, the relative distance camera-target, the relative rotation rate and velocity, the camera capture rate, the illumination conditions and even the target relative pose itself. Some of these factors can be taken into account thanks to the knowledge of camera parameters and the estimated dynamics, but other factors cannot be modeled on-line in real time (i.e. the coupling between illumination source direction with a certain attitude and reflective materials). This makes it very difficult to have a representative model of the covariance matrix R which is valid in any condition. The delay management techniques must therefore demonstrate robustness with respect to uncertainties in the knowledge of matrix R.

In all the simulated scenarios, we will assume that the relative position and the relative attitude quaternion measurements are acquired at a rate of 1 Hz and become available for the filter after a delay of 1 second, which corresponds to $N_d = 10$ assuming that the navigation filter operates at 10 Hz. These are all reasonable values taking into account the typical latency of an IP-CV algorithm and navigation filter run frequency with typical space processing capabilities. A particular focus is given to the estimation of the rotational dynamics, which will be tested firstly in the case without interim measurements and then in the presence of a set of interim measurements. The latter case is the most critical for Larsen's method, since sub-optimality arises both from the linearization of the state transition matrix and from the presence of interim measurements, as explained in Sec.2.2.2. All the simulation scenario that are presented are tested over 200 Monte Carlo runs on a 500 seconds simulation. For all the simulations, the state error covariance matrix P is initialized as the identity matrix.

2.4.1 Performance of the translational dynamics estimation with delayed measurements

The simulated scenario is the following: the target is on a circular orbit in LEO at an altitude of 765 km and the chaser is on an orbit having the same orbital parameters as target orbit, except for the altitude, which is lower. The initial conditions of the relative target-chaser position in LOF are: $x_0 = [-50,\ 0,\ 0,]^T$ m. The chaser is subjected to a continuous profile of thrust in order to perform an *R-bar* maneuver (see Appx.C.2.3) that makes the chaser intercept the target after 500 seconds, at a relative speed of 10 cm/s. We have already precised that the distribution of the measurements around the true state could not be Gaussian and that the standard deviation of the measurements tends to decrease as the camera approaches the target. Moreover the uncertainty of the measured relative position will be higher in the direction of camera optical axis since the depth of the observed object cannot be directly measured (i.e., monocular vision is bearing-only and the scale factor can be determined only by model-based tracking methods). If the chaser is performing an *R-bar* maneuver and it is pointing the target, as in the simulated scenario, the optical axis of the camera corresponds to the direction of target LOF x axis. Therefore the measurement component along x direction is generated with an higher standard deviation with respect to the other components.
Four MC scenarios have been selected in order to test the performance of the filtering techniques under uncertainties in the knowledge of the true covariance of the measurements R and the knowledge of the chaser applied acceleration (i.e., which is the input of CWH state-space representation). The latter may be due to an error in the knowledge of chaser mass or to a difference between the commanded and the true thrust. All the simulations have a relative position initialization error uniformly distributed in the interval $[-10m, +10m]$ for x component and $[-5m, +5m]$ for y and z components. The relative velocity estimate is always initialized to be 0 m/s along each direction.

Table 2.1 summarizes the different conditions tested in each MC scenario. In cases *T.A* and *T.B* the standard deviation of the generated measurements noise is fixed to a constant value of 2 m for the x component and 1 m for y and z components. These values

Table 2.1: Definition of the MC scenarios for the translational dynamics

	Position initialization error	Control thrust F knowledge error	Simulated measurement noise
T.A	$\Delta x \in [-10m, +10m]$ $\Delta y, \Delta z \in [-5m, +5m]$	$\Delta F = 0$	$\sigma_x = 2m$ $\sigma_y,\ \sigma_z = 1m$
T.B	$\Delta x \in [-10m, +10m]$ $\Delta y, \Delta z \in [-5m, +5m]$	$\Delta F \in [-0.25F, 0.25F]$	$\sigma_x = 2m$ $\sigma_y,\ \sigma_z = 1m$
T.C	$\Delta x \in [-10m, +10m]$ $\Delta y, \Delta z \in [-5m, +5m]$	$\Delta F = 0$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = [2, 1, 1]m$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$
T.D	$\Delta x \in [-10m, +10m]$ $\Delta y, \Delta z \in [-5m, +5m]$	$\Delta F \in [-0.25F, 0.25F]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = [2, 1, 1]m$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$

of σ are quite representative for a distance around 50 m, but are overestimated for shorter distances (i.e., the second half of the simulations), since camera sensors are characterized by an increase of measurement accuracy with decreasing range [Fehse, 2003].

Cases $T.C$ and $T.D$ test the sensitivity of filter performance under the presence of random variations in the standard deviation of the measurement noise, in order to assess the robustness to a variable image processing covariance. The standard deviation will be equal to $\sigma_0 + \Delta\sigma$, where $\Delta\sigma$ is a uniformly distributed variable that varies at any instant in the interval $[-0.8\sigma_0, +0.8\sigma_0]$ and σ_0 corresponds to the values defined for cases $T.A$ and $T.B$. Such a noise is more similar to the one that will characterize the IP-CV measurements. The filter is unaware of the measurement noise variation, therefore both cases $T.C$ and $T.D$ will have the nominal tuning respectively of $T.A$ and $T.B$.

In cases $T.A$ and $T.C$, chaser acceleration is supposed to be known and only the sensitivity to uncertainties on the initials condition is tested. In cases $T.B$ and $T.D$, an uniformly distributed uncertainty in the interval of $[-25\%, +25\%]$ is added to the knowledge of chaser control accelerations. In such a case the difference between the truly applied acceleration and the one seen by the KF affects the system as a process noise. In order to grant the convergence of the filter in these cases, the diagonal terms of matrix Q are increased of two order of magnitude with respect to case $T.A$.

Tables 2.2 and 2.3 show the steady-state performance of the filter. The reported values, averaged over 200 MC runs, are:

- σ_m: the standard deviation of the measurement noise generated by the simulator.
- σ_e: the standard deviation of the estimation error (Absolute Knowledge Error AKE, according to [Ott et al., 2011] standard)
- $(1 - \sigma_e/\sigma_m)$: an index (in percentage) of the noise attenuation introduced by the

Table 2.2: Performance with known measurement noise

	Case T.A			Case T.B		
	σ_m	σ_e	$(1-\sigma_e/\sigma_m)$	σ_m	σ_e	$(1-\sigma_e/\sigma_m)$
x [m]	1.999	0.053	97.33%	1.999	0.145	92.77%
y [m]	1.000	0.033	96.73%	1.000	0.099	90.22%
z [m]	1.000	0.025	97.47%	1.000	0.070	93.03%
$\dot{x}$ [m/s]	-	$0.232{\cdot}10^{-3}$	-	-	$0.182{\cdot}10^{-2}$	-
$\dot{y}$ [m/s]	-	$0.203{\cdot}10^{-3}$	-	-	$0.173{\cdot}10^{-2}$	-
$\dot{z}$ [m/s]	-	$0.103{\cdot}10^{-3}$	-	-	$0.088{\cdot}10^{-2}$	-

Table 2.3: Performance with variable measurement noise

	Case T.C			Case T.D		
	σ_m	σ_e	$(1-\sigma_e/\sigma_m)$	σ_m	σ_e	$(1-\sigma_e/\sigma_m)$
x [m]	1.924	0.052	97.29 %	1.921	0.138	92.82%
y [m]	0.985	0.032	96.71%	0.967	0.095	90.22 %
z [m]	0.965	0.024	97.46%	0.989	0.069	93.01 %
$\dot{x}$ [m/s]	-	$0.225{\cdot}10^{-3}$	-	-	$0.173{\cdot}10^{-2}$	-
$\dot{y}$ [m/s]	-	$0.201{\cdot}10^{-3}$	-	-	$0.168{\cdot}10^{-2}$	-
$\dot{z}$ [m/s]	-	$0.099{\cdot}10^{-3}$	-	-	$0.087{\cdot}10^{-2}$	-

filter.

This latter performance index is presented only for those variables which are measured (the relative position for what concerns the translational dynamics). As no interim measurements are present, both Filter Recalculation method and Larsen's method provide the same results, so there is no distinction in the performance of the estimation. Comparing $T.B$ to the correspondent case with known control acceleration $T.A$, and $T.D$ to the correspondent case with known control acceleration $T.C$, it is possible to see a slight performance degradation. Indeed, the calibration of Q used in cases $T.A$ and $T.C$ provides very high level of attenuation (around 97%) that is paid by a loss of robustness with respect to uncertainties in the knowledge of the control accelerations, which could even make the filter diverge. For the cases having variable measurement noise ($T.C$ and $T.D$), the averaged standard deviation of the measurement noise σ_m is slightly decreased and therefore also the standard deviation of the AKE σ_e. The attenuation index is slightly degraded from case $T.A$ to case $T.C$, while it is almost the same for cases $T.B$ and $T.D$, where the uncertainty on the process noise is dominant with respect to the uncertainty on the measurement noise.

2.4.2 Performance of the rotational dynamics estimation with delayed measurements

The simulated rotational dynamics is the following: the chaser is rotating very slowly around the $LOF-z$ axis in order to ensure target pointing during the *R-bar* maneuver (a rotation of about 360 deg within one orbital period), while the target is rotating under the effect of its initial conditions and of the orbital disturbances. Four scenarios have been analyzed, case $R.A$, case $R.B$, case $R.C$ and case $R.D$, whose different conditions are summarized in Table 2.4. The initial attitude quaternions q_{i-tg} and q_{i-ch} are equal to the initial q_{i-LOF} quaternion, while the initial target rotation rate is equal to 1 deg/s around each body axis for cases $R.A$, $R.B$, and $R.C$, and equal to 3 deg/s around each body axis for case $R.D$. These last rotation rates are representative of drifting S/C rotation rates, and remain particularly challenging for an IP algorithm running on a space processor.

Table 2.4: Definition of the MC scenarios for the rotational dynamics

	Attitude initialization error	Target's true initial rotation rate	I_{tg} knowledge error	Simulated measurement noise
R.A	$\Delta\theta \in [-40deg, +40deg]$	$\omega_{i-tg}^{tg} = [1\,,1\,,1\,]^T deg/s$	$\Delta I_{tg} = 0$	$\sigma = 4deg$
R.B	$\Delta\theta = 0deg$	$\omega_{i-tg}^{tg} = [1\,,1\,,1\,]^T deg/s$	$\Delta I_{tg} \in [-0.5I_{tg_{ii}}, 0.5I_{tg_{ii}}]$	$\sigma = 4deg$
R.C	$\Delta\theta \in [-20deg, +20deg]$	$\omega_{i-tg}^{tg} = [1\,,1\,,1\,]^T deg/s$	$\Delta I_{tg} \in [-0.2I_{tg_{ii}}, 0.2I_{tg_{ii}}]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = 2deg$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$
R.D	$\Delta\theta \in [-20deg, +20deg]$	$\omega_{i-tg}^{tg} = [3\,,3\,,3\,]^T deg/s$	$\Delta I_{tg} \in [-0.2I_{tg_{ii}}, 0.2I_{tg_{ii}}]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = 2deg$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$

In cases $R.A$ and $R.B$ the standard deviation of the measurements noise (i.e., represented in Euler attitude angles) is equal to 4 deg, which is an overestimation of the typical noise standard deviation of IP-CV measurements. In cases $R.C$ and $R.D$ the measurements noise has standard deviation equal to $\sigma_0 + \Delta\sigma$, with $\sigma_0 = 2$ deg and $\Delta\sigma$ a uniformly distributed variable that varies at any instant in the interval $[-0.8\sigma_0, +0.8\sigma_0]$. Since the filter is unaware of the measurement noise variation, both $R.C$ and $R.D$ have the nominal tuning for a constant $\sigma = 2$ deg.

Case $R.A$ tests the performance of the methods under uncertainties in the initialization of the state of the filter. Target rotation rate estimation is always initialized at 0 deg/s while target attitude quaternion estimation is initialized by adding a random error $\Delta\theta$ expressed in Euler angles to the true state. The initialization error Euler angles are uniformly distributed in the interval $[-40\ deg,\ 40\ deg]$.

Case $R.B$ tests the filters performance under the presence of uncertainties in the knowledge of target inertia matrix I_{tg} and no error in the state initialization. The inertia matrix used in the filter is obtained by adding to each diagonal term $I_{tg_{ii}}$ of the true inertia matrix a value α, where α is a uniformly distributed variable in the interval $[-0.5I_{tg_{ii}}, +0.5I_{tg_{ii}}]$. Cases *R.C* and *R.D* reproduce scenarios with uncertainties values closer to the ones encountered in a real vision-based RDV in space. The initialization error Euler angles in both cases is in the interval $[-20\ deg,\ 20\ deg]$, which could be the convergence interval of a classical model-based recursive tracking algorithm [Lepetit et al., 2005]. Actually it is even more probable that the KF will be initialized with a lower initialization error since it is preferable to have the IP-CV tracking algorithms converge on their own before starting to feed the navigation filter with these measurements. The uncertainty on the target inertia diagonal terms is uniformly distributed in the interval $[-0.2I_{tg_{ii}}, +0.2I_{tg_{ii}}]$. An uncertainty of 20% in the knowledge of the inertia matrix is quite probable for RDV where the industrial model of the target is supposed to be known but still there could be uncertainties on the amount of remaining propellant or on the degradation of the S/C. The only difference between case *R.C* and *R.D* is that in the first one the true target initial rotation rate is equal to 1 deg/s around each body axis, while in the second one is equal to 3 deg/s around each body axis. In all the four scenarios, the estimated rotation rate is initialized at 0 deg/s.

Table 2.5 shows the steady-state performance of the filters for each one of the scenarios described, averaged on 200 Monte Carlo runs over a simulation of 500s. The performance of the two delay management techniques is compared to the performance of a classic CD-EKF (Eq.(2.23)) processing the infrequent measurements without delay. For what concerns the attitude estimation, the estimation error θ is given in the axis-angle representation, which provides a scalar representation of the error:

$$\begin{aligned} \delta q &= q^*_{i-tg_{true}} \otimes q_{i-tg_{est}} \\ \theta &= 2\left|\operatorname{atan}\left(\sqrt{\delta q_1^2 + \delta q_2^2 + \delta q_3^2}/\delta q_0\right)\right| \end{aligned} \tag{2.28}$$

The reported values in Table 2.5, averaged on 200 MC runs, are:

- σ_m: the root mean square of the measurement noise generated by the simulator.
- σ_{ND}: the root mean square of the AKE for the filter with infrequent non-delayed measurements.
- σ_R: the root mean square of the AKE for the Filter Recalculation method.
- σ_L: the root mean square of the AKE for the Larsen's method.

As for the translational dynamics, also the attenuation index for each filter is shown. For the rotation rate components, the root mean square corresponds to the standard

Table 2.5: Performance of the rotational dynamics estimation

	Case R.A						
	σ_m	σ_{ND}	σ_R	σ_L	$(1-\sigma_{ND}/\sigma_m)$	$(1-\sigma_R/\sigma_m)$	$(1-\sigma_L/\sigma_m)$
θ $[deg]$	6.919	1.730	1.750	1.787	75.00%	74.71%	74.17%
ω_x $[deg/s]$	-	0.0131	0.0131	0.0128	-	-	-
ω_y $[deg/s]$	-	0.0120	0.0121	0.0124	-	-	-
ω_z $[deg/s]$	-	0.0191	0.0189	0.0187	-	-	-
	Case R.B						
	σ_m	σ_{ND}	σ_R	σ_L	$(1-\sigma_{ND}/\sigma_m)$	$(1-\sigma_R/\sigma_m)$	$(1-\sigma_L/\sigma_m)$
θ $[deg]$	6.919	1.778	1.808	1.841	74.30%	73.87%	73.39 %
ω_x $[deg/s]$	-	0.0130	0.0130	0.0128	-	-	-
ω_y $[deg/s]$	-	0.0124	0.0125	0.0123	-	-	-
ω_z $[deg/s]$	-	0.0203	0.0202	0.0201	-	-	-
	Case R.C						
	σ_m	σ_{ND}	σ_R	σ_L	$(1-\sigma_{ND}/\sigma_m)$	$(1-\sigma_R/\sigma_m)$	$(1-\sigma_L/\sigma_m)$
θ $[deg]$	3.402	0.907	0.920	0.939	73.35%	72.97%	72.39%
ω_x $[deg/s]$	-	0.0075	0.0075	0.0073	-	-	-
ω_y $[deg/s]$	-	0.0076	0.0077	0.0076	-	-	-
ω_z $[deg/s]$	-	0.0010	0.0099	0.0098	-	-	-
	Case R.D						
	σ_m	σ_{ND}	σ_R	σ_L	$(1-\sigma_{ND}/\sigma_m)$	$(1-\sigma_R/\sigma_m)$	$(1-\sigma_L/\sigma_m)$
θ $[deg]$	3,460	1.025	1.071	1.096	70.38%	69.06%	68.31%
ω_x $[deg/s]$	-	0.0044	0.0045	0.0046	-	-	-
ω_y $[deg/s]$	-	0.0069	0.0070	0.0079	-	-	-
ω_z $[deg/s]$	-	0.0072	0.0072	0.0075	-	-	-

deviation of the estimation since the mean of the estimate is zero. No attenuation index is computed for this state since the rotation rate is not measured. The results confirm the expectations: the performance of the attitude estimation is in all the cases better for the filter without delay, followed by Filter Recalculation method and then by Larsen's method. On the other side the performance of the estimation of the angular rate is almost comparable for all the filters. In general the filter shows better performance under uncertainties in the initialization ($R.A$) with respect to uncertainties in the knowledge of the inertia matrix ($R.B$) -which actually corresponds to uncertainties in the knowledge of the prediction model. The performance degrades as the rotation rate increases (case $R.D$ is the only one having an attenuation index lower than 70%). In any case, it is worth highlighting that the attenuation index of Larsen's method is only one percent below the one of Filter Recalculation method. Also in the transient phase the Filter Recalculation method shows very slightly better performance than the Larsen's method. It is only in the simulations with interim measurements that the transient performance of the Filter Recalculation method is remarkably better than the one of Larsen's method, as we will show in the next paragraph.

2.4.3 Performance of the rotational dynamics estimation with delayed and interim measurements

We have analyzed the performance of the filters in two more cases, case $RI.C$ and $RI.D$, adding also a set of interim (fast) measurements of the attitude quaternion. This could be the case of measurements coming from marker-based methods or from the use of a trained convolutional neural network, since both methods have a relatively low latency. The measurements are generated with a rate equal to the filter run frequency (10 Hz). They are affected by a Gaussian noise having standard deviation equal to $\sigma_0+\Delta\sigma$, with $\Delta\sigma$ a uniformly distributed variable that varies at any instant in the interval $[-0.8\sigma_0, +0.8\sigma_0]$. Slow delayed measurements have $\sigma_0 = 2$ deg and fast measurements have $\sigma_0 = 4$ deg. The characteristics of the simulated scenarios are summarized in Table 2.6.

Table 2.6: Definition of the MC scenarios for the rotational dynamics with interim measurements

	Attitude initialization error	Target's true initial rotation rate	I_{tg} knowledge error	Slow measurements noise	Fast measurements noise
RI.C	$\Delta\theta \in [-20deg, +20deg]$	$\omega_{i-tg}^{tg} = [1\,,1\,,1\,]^T deg/s$	$\Delta I_{tg} \in [-0.2I_{tg_{ii}}, 0.2I_{tg_{ii}}]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = 2deg$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = 4deg$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$
RI.D	$\Delta\theta \in [-20deg, +20deg]$	$\omega_{i-tg}^{tg} = [3\,,3\,,3\,]^T deg/s$	$\Delta I_{tg} \in [-0.2I_{tg_{ii}}, 0.2I_{tg_{ii}}]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = 2deg$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$	$\sigma = \sigma_0 + \Delta\sigma,\ \sigma_0 = 4deg$ $\Delta\sigma \in [-0.8\sigma_0, 0.8\sigma_0]$

Case $RI.C$ reproduces the conditions of case $R.C$ for what concerns the initialization error, the uncertainties on the knowledge of target inertia matrix, and the true rotation rate, while $RI.D$ reproduces the same conditions of $R.D$. Table 2.7 shows the performance of Filter Recalculation and Larsen's method. The root mean square of the averaged measurement error and of the attenuation index are not presented since the measurements are characterized by different noises. The root mean square of the averaged estimation errors of each case can be directly compared to the values of the corresponding case in Table 2.5, taking into account that the only difference is the presence of high frequency attitude measurements with higher standard deviation. The performance of both methods is highly increased with respect to the cases without interim measurements since, within a delay interval, the state does not evolve in open loop but it continues being corrected by the fast measurements. The steady-state performance of the Filter Recalculation method is only slightly better than Larsen's one, and it is only during the transient phase that Filter Recalculation method shows a remarkably better performance, with lower overshoots, but with a comparable convergence time. This degradation of Larsen's method performance can be explained as follows: in the case without interim measurements Larsen's method was optimal (i.e., to the extent that it was linearizing the propagation of the update); in the case with interim measurements Larsen's method becomes non-optimal also with respect to Kalman theory, since -as explained in Sec.2.2.2- the correction term M^* is computed using Kalman gains that do not take into account the contribution of the delayed measurement.

The performance of the filters in the transient phase can be observed in Fig.2.2 and Fig.2.3, which show the first 200 seconds of a single-run in cases $R.C$, $RI.C$ (Fig.2.2), and $R.D$, $RI.D$ (Fig.2.3), with the same filter state initialization (i.e., an error on Euler attitude angles of [+20 deg, −10 deg, +15 deg]) and the same uncertainty on the diagonal terms of target inertia matrix (i.e., 20%). In the figures it can be noticed how the filters exploiting interim measurements converge faster than filters with only infrequent delayed measurements. Moreover it can be noticed how the Filter Recalculation method has performance comparable to Larsen's method in the cases without interim measurements ($R.C$ and $R.D$), but is more performing in the case with interim measurements ($RI.C$ and $RI.D$), showing much lower error overshoots: 4.4 deg versus 15.2 deg for case $RI.C$, and 5.6 deg versus 13.4 deg for case $RI.D$. Actually, in case $RI.C$ Larsen's method shows a higher overshoot than the corresponding no-interim measurements case $R.C$ (15.2 deg versus 11.8 deg), even if it rapidly converges to a lower error. The curves representing the angle error for the no-interim measurements cases show the peculiarity of appearing piece-wise linear: this is due to the fact that the estimate evolves in open loop as long as a new measurement arrives and only every N_d steps the state is corrected by the Kalman update. The same behavior cannot be observed in cases $RI.C$ and $RI.D$ since a fast measurement is processed at each time step.

Table 2.7: Performance of the rotational dynamics estimation with interim measurements

	Case RI.C		Case RI.D	
	σ_R	σ_L	σ_R	σ_L
θ $[deg]$	0.667	0.690	0.736	0.790
ω_x $[deg/s]$	0.0077	0.0077	0.0137	0.0140
ω_y $[deg/s]$	0.0101	0.0101	0.0200	0.0212
ω_z $[deg/s]$	0.0113	0.0114	0.0281	0.0293

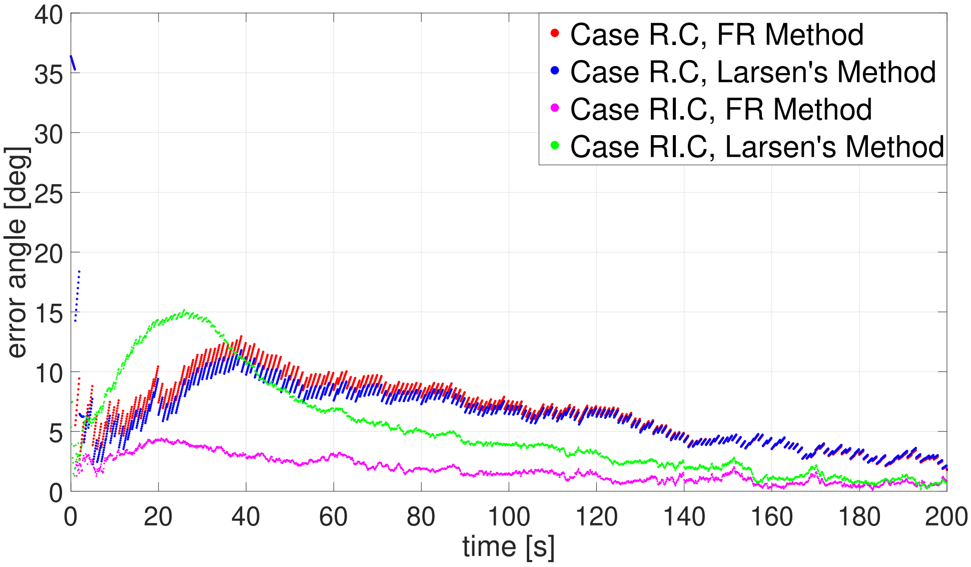


Figure 2.2: Transient phase estimation error in a single run for cases $R.C$ and $RI.C$

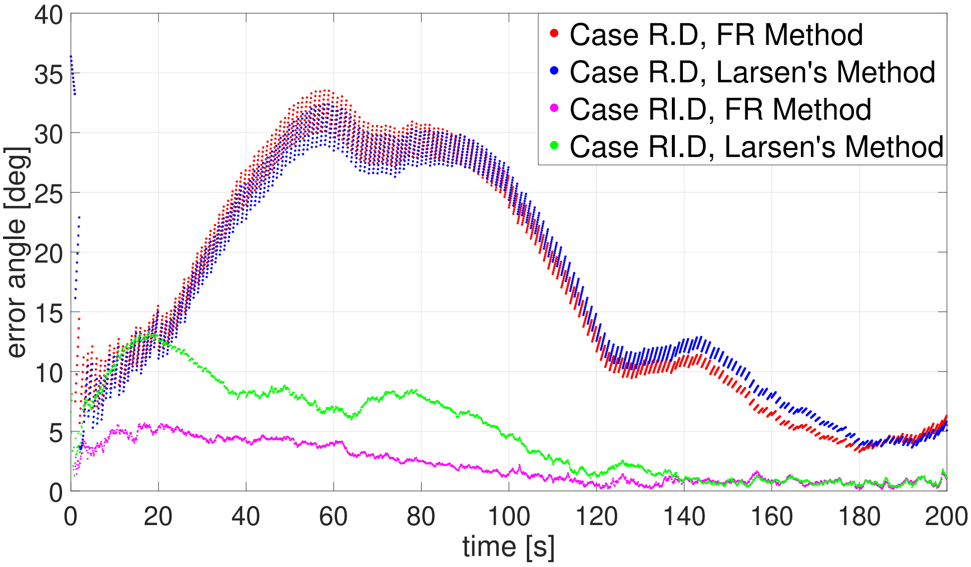


Figure 2.3: Transient phase estimation error in a single run for cases $R.D$ and $RI.D$

2.4.4 Execution time and needed storage

As already mentioned, the choice of the best filter is a trade-off between performance and computational load. Both Larsen's and Filter Recalculation methods have almost the same latency for the time steps in which no delayed measurements arrive, while, at the arrival of a delayed measurement, Filter Recalculation method has to completely

recompute the estimate through the delay period. This means that the latency -averaged on a full simulation- of the Filter Recalculation method will be almost two times the latency of Larsen's method in a case where the slow measurement acquisition frequency is equal to the reciprocal of the delay period multiplied by the filter run frequency.
Another important aspect is the amount of data that need to be stored for each method, with respect to a classical KF without delayed measurements. Table 2.8 shows the amount of "double" to be stored within a delay period (from s to $s + N_d$) for each method, under the following hypothesis:

- $N_d = 10$: number of delay samples of the delayed measurements
- $f = 10$ Hz: navigation filter run frequency
- $f/N_d = 1$ Hz: frequency of the delayed measurements (any time a delayed measurement becomes available for the filter, another one is acquired and starts being processed by the IP-CV algorithms)
- m_{int}: size of the interim measurements vector y_k.

Table 2.8: Amount of *double* to be stored within a delay period by both estimation filters

	Rotational Dynamics		Translational Dynamics	
	Recalculation	Larsen	Recalculation	Larsen
$\hat{x}_{s\|s-1}$	7x1	7x1	6x1	6x1
$P_{s\|s-1}$	7x7	7x7	6x6	6x6
q_{i-ch_k}	4xN_d	4x1 (q_{i-ch_s})	-	-
y_k	m_{int}xN_d	-	m_{int}xN_d	-
u_k	-	-	3xN_d	-
M_k^*	-	7x7	-	6x6
	56+(4 +m_{int}) N_d	109	42+(3+m_{int})N_d	78

Larsen's method, due to the need of propagating through each time instant the 7x7 matrix M^*, requires to allocate a higher memory with respect to a classical KF, but the required space does not depend on the number of the delay samples nor on the size of the interim measurements vector. On the other side, the Filter Recalculation method, which for small values of delay and small size of interim measurements vector requires to store a lower amount of data, rapidly increases its storage burden as N_d or m_{int} increase. Figure 2.4 shows the required number of *double* to be stored as a function of the number of delay samples N_d, for different sizes m_{int} of the interim measurements vector. In the case with no interim measurements ($m_{int} = 0$), Filter Recalculation method becomes worse (in terms of needed storage) than Larsen's methods for $N_d > 13$, while for the case with $m_{int} = 4$ (e.g., the one treated in Sec. 2.4.3) Filter Recalculation method becomes worse than Larsen's methods for $N_d > 6$. The same considerations on the needed storage can be done for the translational dynamics: in the case without interim measurements Filter Recalculation method becomes worse than Larsen's method for $N_d > 12$, while, for the case with $m_{int} = 3$, Filter Recalculation method becomes worse than Larsen's methods for $N_d > 6$. Therefore, even if it requires less operations than the Recalculation method, Larsen's method needs to store a rather high amount of data, which is almost comparable to the one needed by the Filter Recalculation method for values of $N_d \sim 10$ (which is reasonable value taking into account latency of IP-CV algorithms and the typical navigation filter run frequency). Nevertheless, the main advantage of Larsen's method concerns the computational burden, which is equally distributed over every time step (the update of the correction term M_k^*, which requires only two matrix multiplications for every time step). On the other side Filter Recalculation method concentrates the whole computational load in the time instant in which the delayed measurements become available. In these time steps, the computational load of the Filter Recalculation method is multiplied by a factor equal to N_d and might be incompatible with the critical applications run by the on-board computer for this particular time step.

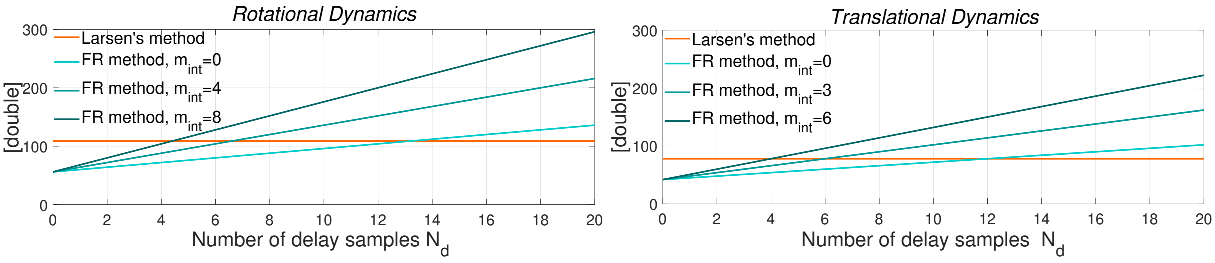


Figure 2.4: Amount of *double* to be stored as a function of the delay samples

2.5 Conclusion

The problem of incorporating delayed and multi-rate measurements in a navigation filter for the estimation of the dynamics of a non-cooperative target has been assessed. A dynamic filter for the estimation of full target rotational and translational state exploiting relative pose measurement has been formalized. Two delay management techniques have been compared: Larsen's method, which provides a fast but sub-optimal solution, and Filter Recalculation method, which always provides the optimal estimate but has a higher computational load. The Monte Carlo validation campaign has shown that

Larsen's method performance is comparable to Filter Recalculation method performance. The latter shows remarkably better performance only in the transient phase of simulations exploiting interim measurements but at the expense of a higher computational and storage need. When a delayed measurement arrives, Filter Recalculation method computational load is multiplied by a factor equal to the number of delay samples, which might be incompatible with the critical applications run by the on-board computer for this particular time step. This suggests that, in applications where the on-board resources are limited, Larsen's method is preferable since it provides a faster estimation without any significant degradation of the steady-state performance.

In this work we have discussed the case with two set of measurements: a set of fast and instantaneous measurements and a set of slow and delayed measurements. The methods discussed in this chapter can be extended, with some modifications, to cases with multiple set of measurements having variable delays and rates.

The following step will be to test the filtering techniques with real vision-based measurements. The simulator described in Sec.2.4 and developed with the collaboration of Thales Alenia Space integrates a high fidelity image generator that allows to generate realistic S/C images in order to implement IP-CV tracking algorithm in the loop. Therefore, the developed navigation function will be coupled and tested with a monocular model-based tracking algorithm, with a special focus on the evaluation and update of the noise covariance matrix R as the chaser gets closer to the target. This analysis is carried out in Chapter 3.

Chapter 3

Frame-by-frame Tracking

Contents

3.1 Introduction

Autonomous rendezvous navigation algorithms require accurate, up-to-date measurements of the relative pose (i.e., position and attitude) of the target. Inexpensive camera sensors have a small form factor -so that they are easily integrated to the S/C without affecting its design- and a low power budget (unlike LIDARs [Kelsey et al., 2006, Sharma et al., 2018b]). For this reason, the coupling of camera sensors with image processing (IP) and computer vision (CV) algorithms can provide a cost effective solution. In this chapter we propose a vision based navigation algorithm that allows the 6-degrees-of-freedom (DOF) pose estimation of a non-cooperative non-prepared target using a single visible monocular camera. Indeed, the use of monocular vision provides advantages with respect to stereo-camera configurations, because single cameras have a lower complexity and a much larger operational range which is not limited by the size of the satellite platform.

3.2 Monocular model-based tracking

Image-based pose estimation can be classified into *non-model-based* and *model-based* techniques [Kelsey et al., 2006]. Non-model-based techniques do not assume any *a priori* knowledge of the tracked object's shape, texture and other visual attributes. These methods rely on visual features (e.g., SURF, SIFT, ORB) which are ideally recognizable and distinguishable from one image to another, regardless of the pose of the camera, the lighting or the spectral conditions in the image. For space applications, methods that depend upon visual features detection are computationally expensive and yield poor results[Sharma et al., 2016] since phenomena such as occlusions, harsh lighting, and reflective materials can make reliable detection and correspondence impossible [Lichter and Dubowsky, 2004]. On the other hand, model-based techniques take advantage of *a priori* knowledge of the object whose pose is to be estimated. This knowledge can be in the form of "fiducials" (or "markers"), or in the form of a 3D geometrical description of the object (which could incorporate also textures, reflectance and other visual attributes). Fiducials are features expressly designed in such a way (shape/colour) that are easily detected and identified with an *ad hoc* method [Lepetit et al., 2005] (e.g., ArUco, ARToolkit, AprilTags). The use of fiducials implies that the target is originally designed to be easily trackable, and therefore it is prepared (see Sec.1). On the other hand, the 3D model of a satellite is exploitable even if the S/C was not supposed to participate in RDV operation. 3D model-based tracking algorithms are therefore more general than fiducial points based ones, and this is why they are the best candidates for this study. The model matching process computes a cost function describing how well the 3D model, in a specific pose or configuration, matches the observed data in image [Jasiobedzki et al., 1999]. In the case of a monocular sensor (i.e., a single camera), the matching will be performed between the 2D projection of the 3D model and the image. Within the many 3D model-based techniques for monocular camera, the more suitable for space applications are the ones relying on edge extraction and tracking, since edges are strong features easily detectable in correspondence of high image gradients. These methods are computationally efficient and naturally stable to lighting changes even for specular materials [Lepetit et al., 2005]. A well-known model-based algorithm relying on edge tracking is RAPiD (Real-time Attitude and Position Determination) algorithm. RAPiD was first theorised in 1990 by C. Harris and C. Stennett in [Harris and Stennett, 1990] and it was one of the first monocular 3D tracker to successfully run in real-time due to its low computational complexity. At instant k, the 3D *a priori* model is projected in the image frame using the pose parameters estimated at instant $k-1$. Visible edges are selected and sampled in order to determine a set of "control points" that will be used in the optimization process. At the same time, edges are extracted on the greyscale image captured at the instant k, resulting in a binary image. Then the control points are associated to the observed points on the image. The matching is carried out by searching along the vector normal to the edge that contains the control point. This mono-directional search reduces the matching search-space from bi-dimensional to one-dimensional, thus allowing fast tracking. To compute the pose correction, RAPiD method relies on the fact that, at first order, small changes in the object pose will cause a displacement of the control points in the image frame which is linear in the pose parameters. This linearity enables to determine the variation of pose through the solution of a simple linear least square problem.

The first use of RAPiD for vision-based autonomous space rendezvous dates from the year 2006 [Kelsey et al., 2006]. The study underlines why simple cameras should be preferred with respect to LIDAR and other scanning and ranging sensor which could provide good relative position accuracy at distances up to several kilometers as well as relative orientation cues at shorter distance, but are typically very expensive and power hungry, and often characterized by a small field of view (FOV). The algorithm in [Kelsey et al., 2006] relies on the iterative recursive least squares method (IRLS) to reject outliers proposed in [Drummond and Cipolla, 2002] and includes a Kalman enhancement (as suggested by [Evans, 1990]) in order to improve the robustness of the estimation. The application of a non-linear version of RAPiD in space rendezvous has been assessed also in [Petit et al., 2012, Petit et al., 2011, Petit et al., 2013], where a graphic process units (GPU) is used in order to render not only geometrical edges but also texture discontinuities of the *a priori* model. Other RAPID-like methods have been proposed in [Lourakis and Zabulis, 2017, Lentaris et al., 2019]. All the cited works couple the pose estimation algorithm with a linear Kalman filter (KF) which propagates a simple kinematic model. However, in the case of high rotation rates typical of a tumbling object, a simple kinematic filter doesn't allow to estimate the rotation rate of the target, which needs to be known to perform some RDV operations requiring the synchronization of chaser motion with target motion. In this chapter, a robust tracking algorithm, based on a non-linear version of RAPID method is proposed. The method does not need any GPU augmentation and it is compatible with typical space processing capabilities. Then, the algorithm is coupled with the navigation filter described in Chapter 2. This coupling enables the estimation of the full target rotational and translational state exploiting only relative pose measurement affected by delay. The chapter is structured as follows. In Sec.3.3 the implementation of the proposed tracking algorithm is presented: Sec.3.3.1 recalls the theory underlying RAPID algorithm, while Sec.3.3.2 and Sec.3.3.3 describe in details the proposed IP-CV algorithm; in Sec.3.3.4 the coupling with the dynamic filter is discussed. In Sec.3.4 the performance of the proposed solution is investigated, while in Sec.3.5 we introduce a discussion about the real-time validation on the robotic test bench. In Sec.3.6 the conclusions are drawn.

3.3 Implementation of the methods

In this section the theory underlying RAPiD and our algorithm will be presented. Both the methods rely on the same steps for the edge extraction, the projection of the *a priori* model, and the matching, but differ in the construction of the cost function. The edge extraction on the greyscale image captured at instant k relies on the Canny edge detector, whose output is a binary image which is white in correspondence of edges and black elsewhere. For the projection of the model, a "masking algorithm", which behaves almost like a classic z-buffer algorithm, has been developed. The algorithm projects into the image frame the *a priori* 3D model of the target using the last available estimate of the pose, and determines a set of control points belonging to the visible geometrical edges of the target. More details on the algorithm logic and development are given in Appendix D.1. In order to enable the matching procedure, for each control point the two-component normal vector (orthogonal to the edge containing the point) is computed. For each control point, the corresponding 3D coordinates in target frame are stored because they will be used in the optimization process. Finally, the matching of a control point with an observed point in the binary image is found by moving, from the projected control point, along the projected normal vector until a value equal to 1 is found. Of course this can produce a discrete number of false matching, which is one of the main drawbacks of the mono-directional search that underlies the method fastness. In order to reduce the number of false matching, a maximal acceptable distance from the control point to the corresponding matched point d_{toll} is introduced. If no match is found at a distance lower than d_{toll}, the control point is discarded. This value of tolerance depends on many factors such as the time span between two time step, the relative rotational and translational target rate, the relative distance camera-target, and the image size. High values of d_{toll} increase the probability of false matching, and above a certain value of d_{toll} the hypothesis of small pose variation underlying the matching procedure is no more valid. However, it is important not to underestimate the value of d_{toll} since this would prevent the recovering of the tracker from small divergence. More details about the matching algorithm are provided in Appendix D.2. Once that the set of matched points is determined, the cost function can be built.

3.3.1 Linear optimization with RAPiD

The analytical formalisation of RAPiD proposed in [Lepetit et al., 2005] is now briefly recalled. This is necessary to help the reader to better understand the method proposed in Sec.3.3.2.

Figure 3.1 provides a schematic representation of the monocular pose estimation problem. Let $X_i^{tg} = [x_i^{tg}, y_i^{tg}, z_i^{tg}]$ be the coordinates of a point X_i belonging to the target object expressed in the target RF (tg), which we assume for simplicity to be centered at target COM. The coordinates of X_i expressed in the camera (cam) RF are:

$$X_i^{cam} = tr_{cam-tg} + R_{cam-tg} X_i^{tg} \tag{3.1}$$

Figure 3.1 The vector $tr_{cam-tg} = [tr^{cam}_{cam-tgt_x}, tr^{cam}_{cam-tgt_y}, tr^{cam}_{cam-tgt_z}]$ is the translation vector that describe the relative position of the center of target RF (O_{tg}, i.e., target's COM), with respect to center of camera RF (O_{cam}), expressed in camera RF. We will refer to it as $tr = [tr_x, tr_y, tr_z]$ for sake of simplicity. The relative position of the camera RF with respect to the chaser RF should be known from on-ground calibration. The rotation matrix R_{cam-tg} is the rotation matrix that describes the relative attitude camera-target, which can be equivalently expressed with the corresponding quaternion q_{cam-tg} (see Appendix A.1). For sake of lightness of the notation, we will refer to tr_{cam-tg} as to tr and to R_{cam-tg} as to R. After a small time interval, the coordinates of X expressed in camera frame can be written as a function of the former pose parameters (i.e., tr and R), and the pose increment $\delta tr = [\delta tr_x, \delta tr_y, \delta tr_z]$ and ΔR:

$$X_i^{cam} \quad = tr + \delta tr + \Delta R \cdot R \, X_i^{tg} \tag{3.2}$$

RAPiD relies on the hypothesis that the variation of pose within two subsequent frames is small so that the rotation matrix can be linearised. From Eq.(A.11) the derivative of the rotation matrix is know and it is therefore possible to write:

$$R(t + \delta t) = R(t) + \left.\frac{\mathrm{d}R}{\mathrm{d}t}\right|_t \delta t \, , \quad \text{with} \quad \frac{\mathrm{d}R}{\mathrm{d}t} = [\omega]_\times \tag{3.3}$$

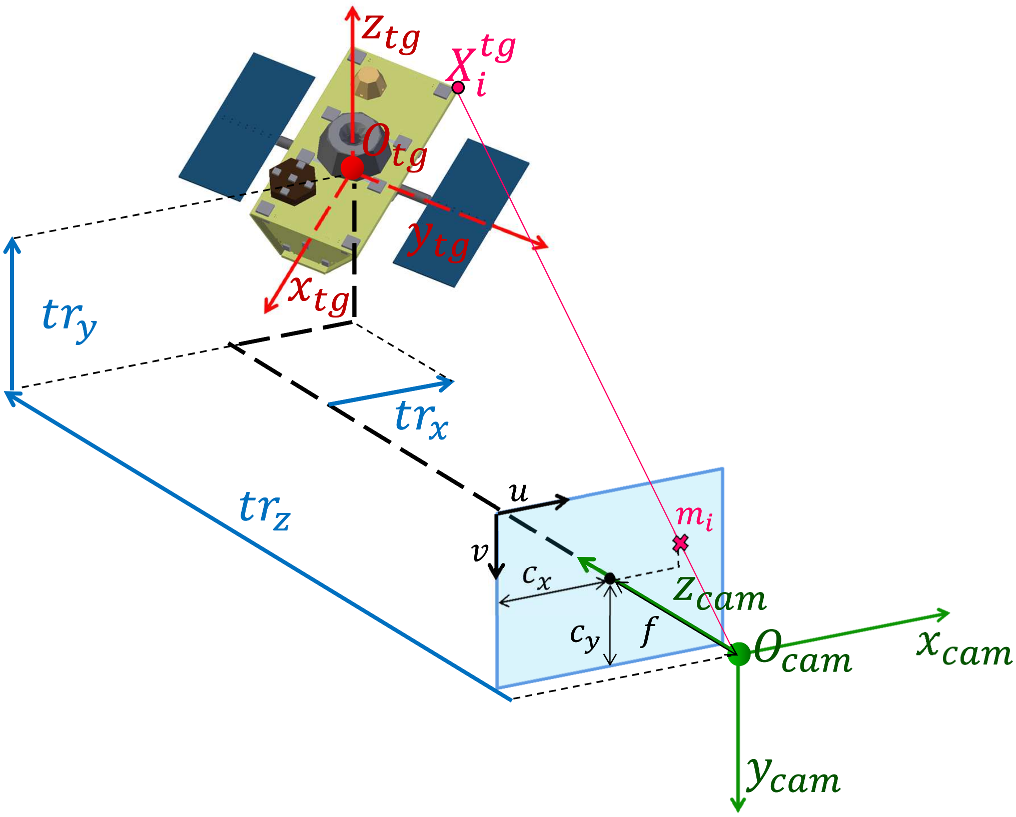


Figure 3.1: Schematic representation of the pose estimation problem. In the figure, we have the focal length of the camera $f = f_x = f_y$, and (c_x, c_y) the principal point of the image

where $\omega = \omega_{cam-tg}^{cam}$ and

$$[\omega]_\times \delta t = [\delta\theta]_\times \quad \text{with}[\delta\theta]_\times = \begin{bmatrix} 0 & -\delta\theta_z & \delta\theta_y \\ \delta\theta_z & 0 & -\delta\theta_x \\ -\delta\theta_y & \delta\theta_x & 0 \end{bmatrix} \tag{3.4}$$

Then Eq.(3.2) becomes:

$$\begin{aligned} X_i^{cam}(t+\delta t) &= tr + \delta tr + (I_{3\times3} + [\delta\theta]_\times) R\, X_i^{tg} \\ X_i^{cam}(t+\delta t) &= X^{cam}(t) + \delta tr + [\delta\theta]_\times R\, X_i^{tg} \end{aligned} \tag{3.5}$$

Reminding the classical pin-hole calibrated and undistorded camera model having projection matrix

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.6}$$

and denoting the product $R\,X_i^{tg} = [R_{x_i},\, R_{y_i},\, R_{z_i}]$, the projection $m_i(t)$ of the point $X_i^{cam}(t)$ in the image plane (i.e., uv plane in Fig.3.1) is given by:

$$m_i(t) = m_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \begin{bmatrix} f_x \dfrac{tr_x + R_{x_i}}{tr_z + R_{z_i}} + c_x \\ f_y \dfrac{tr_y + R_{y_i}}{tr_z + R_{z_i}} + c_y \end{bmatrix} \tag{3.7}$$

while $m_i(t + \delta t)$ is given by:

$$m_i(t+\delta t) = m_i' = \begin{bmatrix} u_i' \\ v_i' \end{bmatrix} = \begin{bmatrix} f_x \dfrac{tr_x + R_{x_i} + \delta tr_x - \delta\theta_z R_{y_i} + \delta\theta_y R_{z_i}}{tr_z + R_{z_i} + \delta tr_z - \delta\theta_y R_{x_i} + \delta\theta_x R_{y_i}} + c_x \\ \\ f_y \dfrac{tr_y + R_{y_i} + \delta tr_y - \delta\theta_x R_{z_i} + \delta\theta_z R_{x_i}}{tr_z + R_{z_i} + \delta tr_z - \delta\theta_y R_{x_i} + \delta\theta_x R_{y_i}} + c_y \end{bmatrix} \tag{3.8}$$

Since the components of δtr and $\delta\theta$ are assumed to be infinitesimal, u' and v' can be manipulated in order to retain only terms up to first order:

$$u_i' = f_x \frac{tr_x + R_{x_i} + (\delta tr_x - \delta\theta_z R_{y_i} + \delta\theta_y R_{z_i})}{tr_z + R_{z_i} + (\delta tr_z - \delta\theta_y R_{x_i} + \delta\theta_x R_{y_i})} + c_x = f_x \frac{a+y}{b+x} + c_x \qquad \text{with } x, y \to 0 \tag{3.9}$$

For $x \to 0$, the fraction can be approximated to:

$$\frac{a+y}{b+x} \sim \left.\frac{a+y}{b+x}\right|_{x=0} - x \left.\frac{(a+y)}{(b+x)^2}\right|_{x=0} = \frac{a+y}{b} - \frac{ax+xy}{b^2} = \frac{a}{b} + \frac{1}{b}\left(y - \frac{a}{b}x + \mathcal{O}(x^2)\right) \tag{3.10}$$

resulting in:

$$u_i' = u_i + f_x \frac{1}{tr_z + R_{z_i}} \left(\delta tr_x - \delta\theta_z R_{y_i} + \delta\theta_y R_{z_i} - u(\delta tr_z - \delta\theta_y R_{x_i} + \delta\theta_x R_{y_i})\right) \tag{3.11}$$

The same linearization procedure can be applied on the expression of v', leading to:

$$v_i' = v_i + f_y \frac{1}{tr_z + R_{z_i}} \left(\delta tr_y - \delta\theta_x R_{z_i} + \delta\theta_z R_{x_i} - v(\delta tr_z - \delta\theta_y R_{x_i} + \delta\theta_x R_{y_i})\right) \tag{3.12}$$

The projected point m_i' can therefore be written as a function of the point m_i plus the product of a matrix $C_i = C(tr, R, X_i^{tg})$ and a vector $\delta p = [\delta tr_x,\, \delta tr_y,\, \delta tr_z,\, \delta\theta_x,\, \delta\theta_y,\, \delta\theta_z]^T$:

$$m_i' = m_i + C_i\,\delta p \tag{3.13}$$

assuming for simplicity that $f_x = f_y = f$, the expression of C_i is:

$$C_i = f \begin{bmatrix} \dfrac{1}{tr_z + R_{z_i}} & 0 & -\dfrac{tr_x + R_{x_i}}{(tr_z + R_{z_i})^2} & -R_{y_i}\dfrac{tr_x + R_{x_i}}{(tr_z + R_{z_i})^2} & \dfrac{R_{z_i}}{tr_z + R_{z_i}} + R_{x_i}\dfrac{tr_x + R_{x_i}}{(tr_z + R_{z_i})^2}, & -\dfrac{R_{y_i}}{tr_z + R_{z_i}} \\ 0 & \dfrac{1}{tr_z + R_{z_i}} & -\dfrac{tr_y + R_{y_i}}{(tr_z + R_{z_i})^2} & \dfrac{-R_{z_i}}{tr_z + R_{z_i}} - R_{y_i}\dfrac{tr_y + R_{y_i}}{(tr_z + R_{z_i})^2} & R_{x_i}\dfrac{tr_y + R_{y_i}}{(tr_z + R_{z_i})^2}, & \dfrac{R_{x_i}}{tr_z + R_{z_i}} \end{bmatrix} \tag{3.14}$$

i.e., a 2x6 coefficient matrix which is function of the known pose (tr, R) at the current instant t, and of the coordinates of the point X_i in target R.F.

From an operative point of view, this means that at the time step k, the projection m'_i of the known point X_i in the image frame can be written as the sum of m_i, which is the projection of the point X_i in the image frame using the known pose parameters at the time step $k-1$, and $C_i \delta p$, which is the product between a known coefficient matrix depending on the pose at the instant $k-1$ and the coordinates of the point X_i^{tg}, and the unknown parameters δp. In Appendix D we provide a physical interpretation of the coefficient matrix C_i.

At the time instant k, an image of the target is acquired and the edges are extracted. The control point X_i are projected into the image frame using the known pose parameters at the instant $k-1$, and are matched to the image points through the monodirectional search previously described. The variation of pose δp can be estimated minimizing the cost function $\chi(\delta p) = \sum_i^M d_i^2$. The cost function is the summation of the square distances between each projected control point m_i (with $i = 1 : M$) and the corresponding matched point in the image frame m_i^{meas}, projected on the vector n_i. The vector n_i is the vector normal to the edge to which the control point belongs. Note that m_i^{meas} is the point on the captured image at the current instant that has been matched to m_i through the mono-directional search along the normal $n_i = [n_{u_i}, n_{v_i}]^T$.

$$d_i^2 \quad = \left[n_i^T \left(m'_i - m_i^{meas}\right)\right]^2 = \left[n_i^T \left(m_i - m_i^{meas}\right) + n_i^T C_i \delta p\right]^2 \tag{3.15}$$

Note that the quantity $l_i^{meas} = n_i^T \left(m_i - m_i^{meas}\right)$, that we name *perpendicular distance*, is a known scalar depending on the pose parameters at the instant $k-1$ and on the measured point. Denoting A_i the 1×6 vector equal to $n_i^T C_i$, the cost function becomes:

$$\chi(\delta p) = \sum_i^M (A_i \delta p - l_i^{meas})^2 \tag{3.16}$$

and the estimated parameter δp is:

$$\delta p \quad = \arg\min_{\delta p} \textstyle\sum_i^M (A_i \delta p - l_i^{meas})^2 \tag{3.17}$$

Eq.(3.17) corresponds to the linear least-square problem $A \delta p = l$, where A is the $M \times 6$ matrix whose i line is equal to A_i, and l is the constant term vector whose i element is equal to l_i. The estimated parameter δp is given by:

$$\delta p = \left(A^T A\right)^{-1} A \, l \tag{3.18}$$

The update of the pose, i.e., the pose measured by the tracking algorithm at the time instant k, is therefore:

$$\begin{cases} tr_k & = tr_{k-1} + \delta tr \\ R_k & = (I_{3\times3} + [\delta\theta]_\times) R_{k-1} \end{cases} \tag{3.19}$$

or alternatively, naming q the relative attitude quaternion q_{cam-tg}:

$$\begin{cases} q_k & = \delta q \otimes q_{k-1} \\ \delta q & = \dfrac{1}{\sqrt{1+\delta\theta_x^2+\delta\theta_y^2+\delta\theta_z^2}} \begin{bmatrix} 1 \\ \delta\theta_x \\ \delta\theta_y \\ \delta\theta_z \end{bmatrix} \end{cases} \tag{3.20}$$

3.3.2 Non-linear optimization: construction of the cost function

As anticipated, during a rendezvous with a non-cooperative S/C, the target variation of pose within two consecutive image acquisitions may be too great to enable linearisation of the trajectory in the image frame, and the tracking could be rapidly lost. A space debris may be in tumbling, with a total rotation rate that can exceed 6 deg/s [Bonnal et al., 2013]. In this section we provide an analytical formulation of our estimation method based on the optimization of a non-linear cost function. Given a fitting non linear function $\hat{y}(p)$ of a vector of N parameters p, and a set of M data points y_i, the parameters can be estimated minimizing the sum of the weighted squares of the errors between the measured data y_i and the fitting function $\hat{y}(p)$. The resulting scalar cost function is:

$$\chi^2(p) \quad = \sum_i^M (y_i - \hat{y}_i(p))^2 = (Y - \hat{Y}(p))^T W (Y - \hat{Y}(p)) \tag{3.21}$$

where the weight matrix W is equal to diag$(1/\sigma_i^2)$, with σ_i the standard deviation of the error associated to measurement y_i. The set of measurement data Y will be a vector composed by the projections of the points $[u_i, v_i]^T$ (i.e., the measured coordinates in image frame of the matched point corresponding to X_i^{tg}) along the projected normal $n_i = [n_{u_i}, n_{v_i}]^T$.

$$y_i = n_{u_i} u_i + n_{v_i} v_i, \quad Y = [y_1, y_2, ..., y_i, ..., y_M]^T \tag{3.22}$$

The weight matrix W is set to be equal to the identity matrix of size $M \times M$, since there is no way to know the error related to each matched image point $[u_i,\, v_i]$.

The pose parameter p to be estimated is the full pose at the instant k, and not the pose increment as it was done for RAPiD algorithm. In the pose parameter p, rotation will be described using the quaternion $q = q_{cam-tg} = [q_0,\, q_1,\, q_2,\, q_3]^T$, since quaternions have the lowest dimensionality possible for a globally non-singular representation of the rotational group $SO(3)$ [Markley, 2004]. The parameter vector p is therefore a 7×1 vector equal to $[tr_x,\, tr_y,\, tr_z,\, q_0,\, q_1,\, q_2,\, q_3]$. Analogously to Eq.(3.1) For each control point X_i^{tg} which has found a match in the image, the corresponding coordinates expressed in camera RF are:

$$X_i^{cam} \quad = tr + q \otimes X_i^{tg} \otimes q^* \tag{3.23}$$

Reminding the notation defined in Appendix A.2, we use $Q_{rot_j}(X_i^{tg}, q)$ (with $j = 1:3$) in order to refer to the first, the second and the third component of the vector resulting from the rotation described by the function $Q_{rot}(X_i^{tg}, q) = q \otimes X_i^{tg} \otimes q^*$, we project the point X_i^{cam} into the image frame using the pinhole camera model (i.e., Eqs.(3.6),(3.7)). The fitting function becomes:

$$\hat{y}_i \quad = n_i^T \begin{bmatrix} \hat{u}_i(X_i^{tg}, p) \\ \hat{v}_i(X_i^{tg}, p) \end{bmatrix} = n_{u_i} \left(f_x \frac{tr_x + Q_{rot_1}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(X_i^{tg}, q)} + c_x \right) + n_{v_i} \left(f_y \frac{tr_x + Q_{rot_2}(X_i^{tg}, q)}{tr_z + Q_{rot_3}(X_i^{tg}, q)} + c_y \right) \tag{3.24}$$

and the non linear function will be $\hat{Y}(p) = [\hat{y}_1(p), \hat{y}_2(p), ..., \hat{y}_i(p), ..., \hat{y}_M(p)]^T$. The nonlinear optimization is solved using the version of the Levenberg-Marquardt (LM) algorithm described by [Gavin, 2011] and detailed in Appendix D.4. The Jacobian of the nonlinear functions $\hat{Y}(p)$ has to be derived in order to compute at each iteration the update of parameter p. For each matched model point X_i^{tg}, corresponding to the measured point y_i and the fitting function $\hat{y}_i(X_i^{tg}, p)$, the Jacobian $\mathbb{J}_i$ is the 1×7 vector:

$$\mathbb{J}_i = \frac{\partial \hat{y}_i}{\partial p} = \begin{bmatrix} n_{u_i} \frac{\partial \hat{u}_i}{\partial tr_x} + n_{v_i} \frac{\partial \hat{v}_i}{\partial tr_x}, & \dots, & n_{u_i} \frac{\partial \hat{u}_i}{\partial q_3} + n_{v_i} \frac{\partial \hat{v}_i}{\partial q_3} \end{bmatrix} \tag{3.25}$$

The computation of the elements of the Jacobian with respect to the translational terms is straightforward, while it is more complex for the elements derived with respect to the quaternion. In Appendix A.2.3 the derivation of analytical Jacobian for the rotation of a vector $X_i^{tg} = [x_i, y_i, z_i]$ with respect to a quaternion q is carried out. The Jacobian is:

$$\frac{\partial (q \otimes X_i^{tg} \otimes q^*)}{\partial q} = \begin{bmatrix} \dfrac{\partial Q_{rot_1}(X_i^{tg}, q)}{\partial q} \\ \dfrac{\partial Q_{rot_2}(X_i^{tg}, q)}{\partial q} \\ \dfrac{\partial Q_{rot_3}(X_i^{tg}, q)}{\partial q} \end{bmatrix} = \begin{bmatrix} A & D & C & -B \\ B & -C & D & A \\ C & B & -A & D \end{bmatrix} \tag{3.26}$$

where:

$$\begin{aligned} A = \frac{\partial Q_{rot_1}}{\partial q_0} = \frac{\partial Q_{rot_2}}{\partial q_3} = -\frac{\partial Q_{rot_3}}{\partial q_2} &= 2(q_0 x_i - q_3 y_i + q_2 z_i) \\ B = \frac{\partial Q_{rot_2}}{\partial q_0} = \frac{\partial Q_{rot_3}}{\partial q_1} = -\frac{\partial Q_{rot_1}}{\partial q_3} &= 2(q_3 x_i + q_0 y_i - q_1 z_i) \\ C = \frac{\partial Q_{rot_1}}{\partial q_2} = \frac{\partial Q_{rot_3}}{\partial q_0} = -\frac{\partial Q_{rot_2}}{\partial q_1} &= 2(-q_2 x_i + q_1 y_i + q_0 z_i) \\ D = \frac{\partial Q_{rot_1}}{\partial q_1} = \frac{\partial Q_{rot_2}}{\partial q_2} = \frac{\partial Q_{rot_3}}{\partial q_3} &= 2(q_1 x_i + q_2 y_i + q_3 z_i) \end{aligned} \tag{3.27}$$

The terms of the Jacobian matrix in Eq.(3.25) for the $\hat{u}_i$ and $\hat{v}_i$ components are therefore:

$$\begin{cases} \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial tr_x} = f_x \dfrac{1}{tr_z + Q_{rot_3}(X_i^{tg},q)} \\ \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial tr_y} = 0 \\ \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial tr_z} = -f_x \dfrac{tr_x + Q_{rot_1}(X_i^{tg},q)}{tr_z + Q_{rot_3}(\Big(X_i^{tg},q)\Big)^2} \\ \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial q_0} = f_x \dfrac{A\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) - C\left(tr_x + Q_{rot_1}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \\ \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial q_1} = f_x \dfrac{D\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) - B\left(tr_x + Q_{rot_1}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \\ \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial q_2} = f_x \dfrac{C\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) + A\left(tr_x + Q_{rot_1}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \\ \dfrac{\partial \hat{u}_i(X_i^{tg},p)}{\partial q_3} = f_x \dfrac{-B\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) - D\left(tr_x + Q_{rot_1}(X_i^{tg},)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \end{cases} \tag{3.28}$$

$$\begin{cases} \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial tr_x} = 0 \\ \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial tr_y} = f_y \dfrac{1}{tr_z + Q_{rot_3}(X_i^{tg},q)} \\ \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial tr_z} = -f_y \dfrac{tr_y + Q_{rot_2}(X_i^{tg},q)}{tr_z + Q_{rot_3}(\Big(X_i^{tg},q)\Big)^2} \\ \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial q_0} = f_y \dfrac{B\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) - C\left(tr_y + Q_{rot_2}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \\ \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial q_1} = f_y \dfrac{-C\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) - B\left(tr_y + Q_{rot_2}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \\ \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial q_2} = f_y \dfrac{D\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) + A\left(tr_y + Q_{rot_2}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \\ \dfrac{\partial \hat{v}_i(X_i^{tg},p)}{\partial q_3} = f_y \dfrac{A\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right) - D\left(tr_y + Q_{rot_2}(X_i^{tg},q)\right)}{\left(tr_z + Q_{rot_3}(X_i^{tg},q)\right)^2} \end{cases} \tag{3.29}$$

The obtained Jacobian is used at each iteration to analytically compute the direction of the parameter increment, starting from the initial guess p_0 which is taken to be equal to the estimated pose at instant $k-1$. In this optimisation process, some approximations are introduced: at each update of the pose within the LM algorithm, a new set of control points and normal vectors should be computed by the masking algorithm, and new matches with the binary image should be found. Nevertheless, such a process would dramatically increase the computational load of the algorithm. For this reason, the set of control points, matched points and normal vectors will be kept constant for all the loops within each LM run. The algorithm shows very good performance even if in presence of this approximation, as it will be illustrated in Sec.3.4.

3.3.3 Use of silhouette contours

One of the major drawbacks of applying the Canny edge extraction on the greyscale images is that the algorithm detects not only geometrical edges, but also texture discontinuity, which are not managed in the masking algorithm. Moreover, the direction of the light could make invisible some geometrical edge due to the particular reflective texture of the MLI (Multi Layer Insulation), which usually covers S/Cs. Finally, shadows contours are identified by the Canny edge extractor as edges (e.g., as it is possible to see for the shadow contour highlighted in red in Fig.3.2(a), which appears in the output image of the Canny edge detector in Fig.3.2(b)). All these conditions can lead to possible false matching of the control points with a non-geometrical edge. As this work does not consider the option of using GPU acceleration to render also texture discontinuities (i.e., as done in [Petit et al., 2012]) or shadows, the most robust solution to reduce the number of false matching is to rely only on the external contour of the target (i.e., the silhouette contours). If the target is artificially illuminated by the chaser, or if there are proper sun illumination conditions (see [Fehse, 2003]), the silhouette contour of the target can be retrieved applying threshold and morphological operators on the greyscale image before the Canny edge extraction. This has been done for the image shown in Fig.3.2(d), whose Canny edge extraction output is shown in Fig.3.2(e). Although the study presented in this Chapter relies only in the use of a visible (VIS) camera, the coupling with an infra-red (IR) thermal camera would provide benefits to the silhouette extraction, increasing the robustness towards illumination conditions, shadows and reflective materials. Moreover, the use of multi-spectral imagery could help the segmentation process in presence of the Earth in the background. Providing a solution for background segmentation is out of the scope of this study. However, in Sec.5, we will provide some details of a segmentation algorithm developed in Thales Alenia Space and integrated to the final navigation solution proposed by this thesis.

As only the silhouette contours are now extracted, a new masking algorithm capable of detecting only the external perimeter of the target has been implemented. The algorithm is detailed in Appendix D.1.3, and it has lower computational load with respect to the former masking algorithm. For the pose corresponding to Fig.3.2(a), the computational time of the new algorithm (Fig.3.2(f)) is decreased by a factor of 5 with respect to the latency time of former masking algorithm (Fig.3.2(c)). Averagely, the latency is reduced

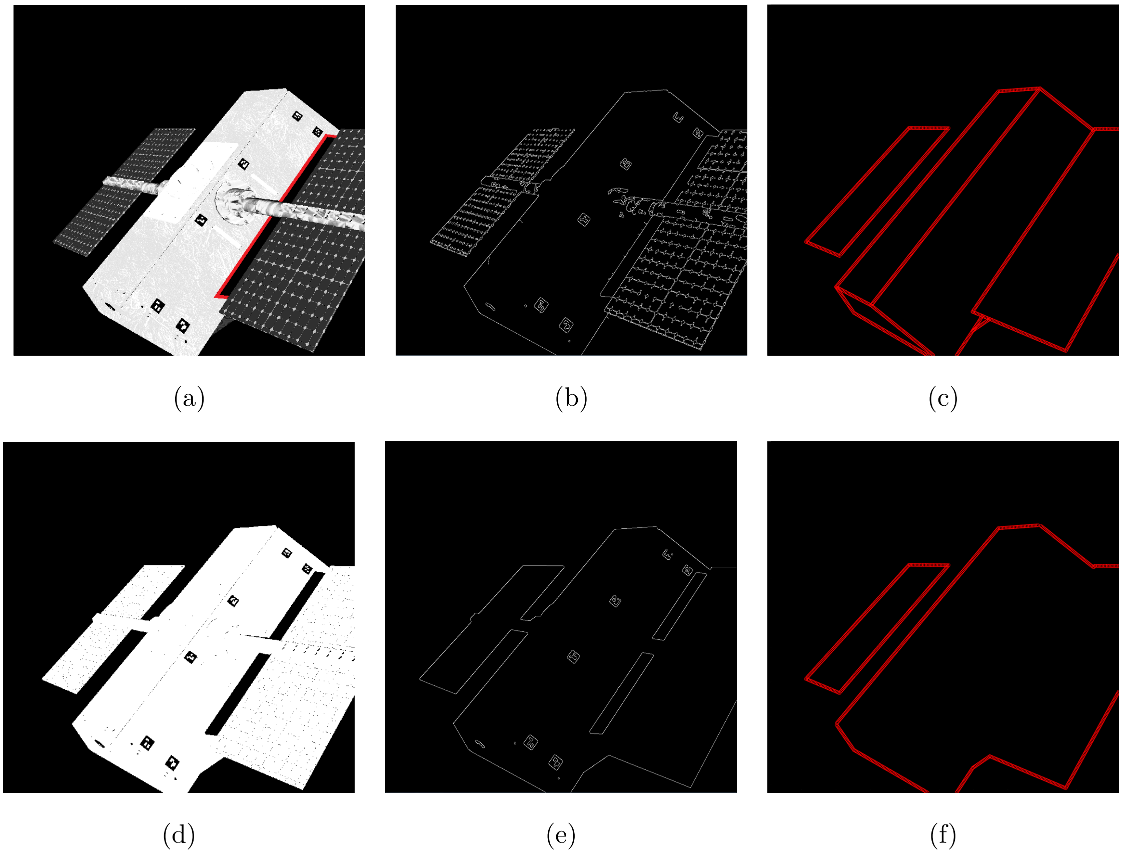


Figure 3.2: 3.2(a) Greyscale image. 3.2(b) Canny edge extraction on greyscale image. 3.2(c) Projection of all visible geometrical edges. 3.2(d) Thresholded image. 3.2(e) Canny edge extraction on thresholded image. 3.2(f) Projection of the external perimeter edges.

by a factor of $4 - 5$. The new masking algorithm applied to the thresholded image, together with a proper choice of d_{toll}, has proven to generate fewer mismatches than the former technique, thus allowing not to consider outliers rejection algorithms such as RANSAC proposed by [Zisserman, 1995] or the IRLS technique proposed by [Drummond and Cipolla, 2002, Kelsey et al., 2006], techniques that increase the computational load of the algorithm.

3.3.4 Integration of the measurements in the navigation filter

The tracking algorithm discussed in the previous sections provide a measurement of the relative pose target-chaser, but no measurement of the relative translational velocity and rotation rate. As anticipated in Sec.2, certain close proximity operations require the knowledge of the complete relative state of the target. The coupling of the IP-CV algorithm with a dynamic KF enables the estimation of both translational and rotational velocity of the target. Moreover, the measurements computed by the IP-CV tracking algorithm can be affected by a consistent delay, due to the high computational load associated to operation such as image segmentation, the 3D model projection, and the non-linear optimisation. There will be therefore a delay between the time of acquisition (i.e., image capture by the camera), and the time in which the measurement becomes available. In addition, due to the high computational load of the IP-CV algorithm, measurements could be available at a lower rate with respect to the one of the navigation filter. For this reason, the filter must implement appropriate technique to merge delayed and infrequent measurements.

Besides the benefits already listed, the coupling of the tracking algorithm with the dynamic KF extremely increases the robustness of the tracking. In fact recursive tracking algorithms, although they provide very precise measurements, are prone to divergence because they limit the search of the new pose measurement in the neighborhood of the previous estimate (i.e., model-image matching is done by local search in proximity of the projected model at the previously estimated pose). Mismatching are usually due to noisy images and poor silhouette extraction, but can also be "intrinsic" to the method. This could be the case when the S/C, rotating, passes from a configuration (instant $k-1$) in which a geometrical edge is visible, to a configuration (instant k) in which that edge becomes invisible (or vice versa). This phenomenon is known as self-occlusion. In such a case, the edge will be present in the *a priori* projected model, but will be invisible in the Canny image (or vice versa), increasing the probability to match the control points with the wrong edge in the image. This makes the optimization algorithm converge to a local minimum which does not represent the real pose of the target. Usually, model-based recursive algorithms cannot recover autonomously from these local minima and the tracking diverges. Even if our algorithm has shown good convergence property with respect to RAPiD, there are some configurations in which the tracking is lost (e.g., when the faces of the solar arrays become visible or invisible). For this reason, the coupling with the KF becomes crucial: the dynamic filter propagates a physical model which predicts the motion of the target and makes edges appear and disappear according to the prediction model.

In this section, the dynamic model implemented in the navigation filter, as well as the applied delay management technique, are briefly recalled. The complete formalisation of the method and its application to the space RDV navigation problem are discussed in Sec.2 and Appendix C. Then, in Sec.3.3.4.2, we will explain how to optimally couple the IP-CV measurements with the KF.

3.3.4.1 Recalls of the proposed navigation filter

The translational motion is modelled according to the well known Clohessy-Wiltshire-Hill's equations [Fehse, 2003], a system of linear differential equations that describes the relative motion of the chaser with respect to the target. The motion is expressed in the target Local Orbital Frame (LOF, with x axis along the radial Earth-target, z axis along target orbit angular momentum and y axis completing the right-handed trihedron):

$$\begin{cases} \ddot{x} - 3\omega^2 x - 2\omega\dot{y} & = \gamma_x \\ \ddot{y} + 2\omega\dot{x} & = \gamma_y \\ \ddot{z} + \omega^2 z & = \gamma_z \end{cases} \tag{3.30}$$

where ω is target's orbit angular rate and $\gamma_{x,y,z}$ are the control accelerations acting on chaser centre of mass (COM). The state transition matrix associated to the system in Eq.2.15 can be analytically computed, thus the relative translational dynamics can be written in the form of a linear time-discrete system. The rotational dynamics is modelled according to the non-linear prediction model:

$$\begin{cases} \dot{q}_{i-tg} & = \frac{1}{2} q_{i-tg} \otimes \begin{bmatrix} 0 \\ \omega_{i-tg}^{tg} \end{bmatrix} \\ \dot{\omega}_{i-tg}^{tg} & = -I_{tg}^{-1} \left(\omega_{i-tg}^{tg} \times I_{tg}\, \omega_{i-tg}^{tg} \right) \end{cases} \tag{3.31}$$

where q_{i-tg} is the attitude quaternion from inertial to target RF, ω_{i-tg}^{tg} is target rotation rate with respect to the inertial RF expressed in target RF and I_{tg} is the inertia matrix of the target at its COM.

In order to integrate delayed measurements, Larsen's method is implemented (see Sec.2.2.2). This method relies on the computation, throughout the delay period, of a correction term to add to the filter estimate when the delayed measurement becomes available. It requires only two matrix multiplications at each time step, as well as the storage of the predicted state and error covariance matrix relative to the time step in which the measurement was acquired. Moreover, Larsen's method allows to merge multi-rate measurements, enabling the use of multiple sensors and tracking algorithms. Larsen's method is sub-optimal for non-linear systems and in the presence of multi-rate measurements, but has a very low computational load compared to optimal methods (e.g., Filter Recalculation method, [Prasad et al., 2002, Li et al., 2004]). As proven by the analysis in Sec.2.4, for the RDV problem, Larsen's method is the best trade-off between optimality and computational load, with a performance comparable to the one of the optimal method.

3.3.4.2 Computation of the measurement noise covariance matrix R

One of the main problem related to the IP-CV algorithms is to have a good characterization of the measurement noise covariance matrix R, which is needed to integrate the measurements into the Kalman filter. Nevertheless, it may be particularly difficult to have a good characterization of R because the measurement noise depends on many factors, such as the intrinsic noise of the sensor, the relative distance camera-target, the relative rotation rate and velocity, the camera capture rate, the illumination conditions and even the target relative pose itself. When using the LM algorithm for the estimation of a parameter (in our case the parameter p_k, which correspond, from the point of view of the KF, to the measured pose at instant k), the covariance of the estimated parameter p_k, and therefore of the measured pose, can be computed from the Jacobian and the weight matrix, according to the formula [Gavin, 2011]:

$$R = [\mathbb{J}^T W \mathbb{J}]^{-1} \, . \tag{3.32}$$

However, as anticipated, the value of the weighting matrix W is not known. For this reason, it became necessary to detect another index of merit of the obtained measurement. The value of the χ^2 function evaluated at the estimated parameter p_k, divided by the degree of freedom of the problem (i.e., $\nu = M - N + 1$) is used as an indicator of the level of confidence of the measurement. This value is named *reduced cost function* and it also used as one of the convergence criteria in the LM algoritm.

$$r = \frac{\chi^2(p_k)}{\nu} \tag{3.33}$$

The value of r is a sort of averaged squared reprojection error, and it will be referred to as *residual* in the following sections. For a given target geometry, camera parameters and relative camera-target distance, the residual can be correlated to the noise covariance matrix of the associated measurements (i.e., a very low residual indicates that the reprojection error is minimal and therefore that the measurement is reliable). This correlation is obtained experimentally during offline calibration.

3.4 Performance analysis

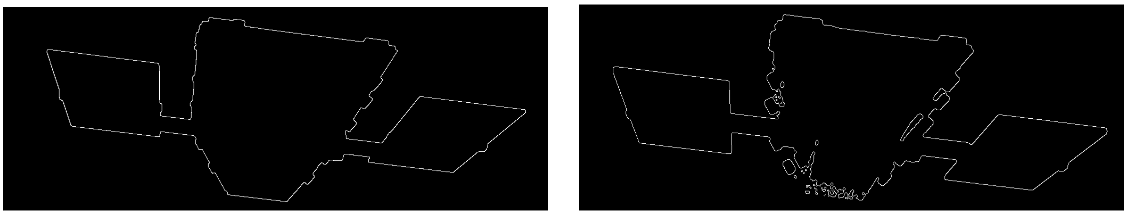
The proposed non-linear tracking algorithm is tested in order to characterize the performance at different distances. First, the tracking algorithm is tested with a constant calibration of the Kalman Filter, i.e., with a constant value of measurement noise covariance matrix R. Two different conditions are analyzed. In a first set of simulations, the estimation chain is tested using perfect silhouette images. This excludes illumination conditions, shadows and reflective textures from the possible sources of false matching. In the second set of simulations, realistic synthetic images are used. Then, the simulations are repeated using a dynamic calibration of the KF, i.e., exploiting the residual r introduced in Sec.3.3.4.2 to dynamically update the measurement noise covariance matrix R.

3.4.1 Simulation scenarios

In all the simulated scenarios, the images are acquired at a rate of 1 Hz. A latency time of 1 second for the IP-CV algorithm is assumed. Therefore, the measurements become available by the filter after a delay of 1 second, which corresponds to a delay of 10 time step, assuming a filter run frequency of 10 Hz.

In the set of simulations relying on perfect target silhouettes, target and chaser are at a fixed distance and do not translate with respect to the inertial frame. Chaser's attitude is fixed with the camera pointing towards the target, while the target is rotating according to the angular momentum equation under the effect of its initial conditions.

The second set of simulations is run on a full GNC rendezvous simulator modelling the LEO environment, with orbital perturbations affecting both chaser's and target's dynamics (see Appx.C.1). The simulations are performed in closed-loop, meaning that the estimated pose is used to control camera pointing and chaser position, thus adding a source of error. The chaser is equipped with a spot light illuminating the target. The images are simulated using Thales Alenia Space high fidelity image simulator SpiCam. The images are affected by shadows, reflections and blur, so that the silhouette retrieved by the Canny edge extraction is highly affected by noise. Fig. 3.3 shows, for the same relative pose, the difference between the perfect silhouette used in the first set of simulations and the noisy one used in the second set of simulations.



(a) First set of simulations

(b) Second set of simulations

Figure 3.3: Comparison between perfect (3.3(a)) and noisy (3.3(b)) silhouettes

Each set of simulations is repeated twice, once with static calibration of the KF, once with dynamic calibration of the KF. We will refer to case *S.A* and *D.A* (respectively

Static A and Dynamic A) for the cases relying on perfect silhouettes and respectively with static and dynamic KF calibration. We will refer to case $S.B$ and $D.B$ (respectively Static B and Dynamic B) for the cases relying on the high fidelity RDV simulator and respectively with static and dynamic KF calibration.

For each one of the 4 cases, the algorithm has been tested at a distance of 30 m, 20 m and 10 m, with a camera having a 30 deg FOV and a size of 1024×1024 pixels, i.e., in Eq.(3.6) we have $c_x = c_y = 512$ and $f_x = f_y = f = c_x / \tan(\text{FOV}/2) = 1911$. The target has a size of $4 \times 3 \times 1$ meters. These dimensions, translated in pixel, correspond, in the image frame, to an object size of $255 \times 191 \times 64$ pixels at 30 m (i.e., Fig.3.4(a)), $382 \times 287 \times 96$ pixels at 20 m (i.e., Fig.3.4(b)), and $764 \times 573 \times 191$ pixels at 10 m (i.e., Fig.3.4(c)). The 3D model projected by the masking algorithm is simplified and has fewer details than the CAD model used to generate the images, which on the other hand accurately reproduces the geometry of the S/C.

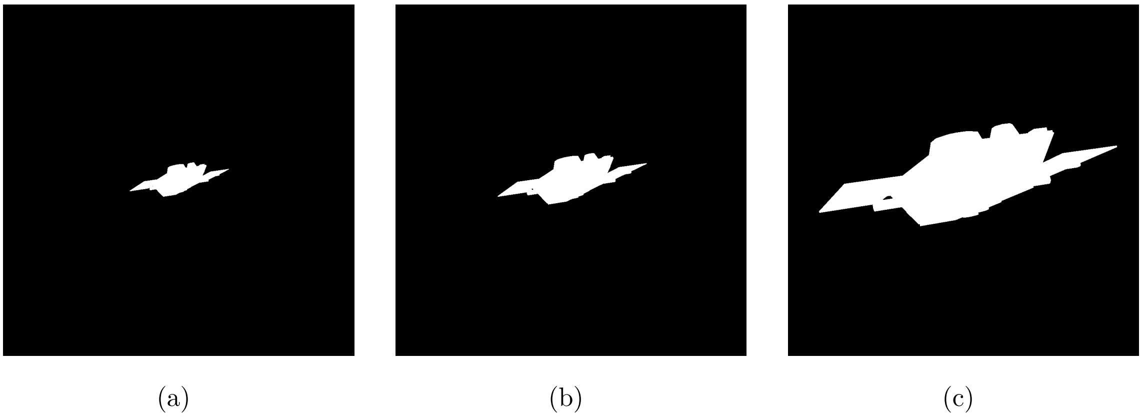

(a) (b) (c)

Figure 3.4: 3.4(a) Silhouette of the satellite at 30 m, 3.4(b) Silhouette of the satellite at 20 m, 3.4(c) Silhouette of the satellite at 10 m.

For each distance and each case, 100 simulations of 200 seconds have been run, varying on the initial conditions of target attitude and rotation rate. The initial relative attitude quaternions have been uniformly sampled to cover all the attitude “space”, while target rotation rate components along each body axis are random variables uniformly distributed in the interval $[-1deg/s, +1deg/s]$.

3.4.2 Simulation results

Tables 3.1 and 3.2 show the performance of each case at each distance, averaged on the 100 simulations. In Tab.3.1 the performance of the cases with static KF calibration are shown, while Tab.3.2 shows the performance of the cases with dynamic KF calibration. For the attitude estimation, the error is given in the axis-angle representation, which provides a positive scalar representation of the error (see Eq.(2.28)). The mean and the root-mean-square (RMS) are shown both for the error of the measured relative quaternion ($\delta\theta_{meas}$) and for the error of estimated relative quaternion ($\delta\theta_{est}$). For the relative target/chaser rotation rate estimation error ($[\delta\omega_{xest}, \delta\omega_{y_{est}}, \delta\omega_{zest}]$, expressed in chaser RF), the standard deviation σ is shown. The performance of the method in cases $S.A$ and $D.A$ can be considered as the intrinsic performance of the algorithm, because no external sources of error are present (i.e., external edges are perfectly extracted from binary silhouette images, and there are not pointing errors). Thus, cases $S.A$ and $D.A$ must be considered as a benchmark to compare respectively cases $S.B$ and $D.B$.

For what concerns the tests with static calibration, is should be noted that the mean of the angular error is lower for the measured quaternion, while the root-mean-square error is lower for the estimated quaternion. This is due to the static calibration of matrix R. The KF is calibrated in order to be robust towards divergent measurements, giving relatively more confidence to the prediction model than to the measurements. When the measurements are good, the estimation is less precise than the measurement itself, but when measurement start diverging, the estimation brings the measurement back to the correct pose. The performance of the quaternion estimation can be increased by using a time-varying calibration of the KF that computes in real-time the value of the measurement noise covariance R, as it is visible in Tab.3.2.

The performance of case $S.B$ can be directly compared to the benchmark performance of case $S.A$. A degradation of the performance occurs: at 20 m the method has diverged in 10 runs, and at 30 m in 26 runs. The value shown in Table 3.1 are averaged on the simulations in which the tracking was ensured. The high degradation of the performance at a distance of 30 m is due to the reduced size of the S/C in the image frame. As the distance increases, the number of control points used to compute the cost function decreases: this increases the relative weight of a mismatching with the noisy silhouette.

Table 3.1: Performance of the non-linear tracking algorithm with constant KF calibration

	Case S.A: Perfect silhouette						Case S.B: High fidelity RDV simulator					
	30 m		20 m		10 m		30 m		20 m		10 m	
	mean	RMS	mean	RMS	mean	RMS	mean	RMS	mean	RMS	mean	RMS
$\delta\theta_{meas}$ $[deg]$	1.91	3.34	1.23	2.22	0.80	1.47	6.73	8.73	3.82	5.20	1.73	2.51
$\delta\theta_{est}$ $[deg]$	2.13	3.20	1.34	2.01	0.83	1.25	7.21	9.23	4.12	5.41	1.79	2.42
	σ		σ		σ		σ		σ		σ	
$\delta\omega_{xest}$ $[deg/s]$	0.065		0.047		0.023		0.109		0.073		0.046	
$\delta\omega_{y_{est}}$ $[deg/s]$	0.169		0.115		0.057		0.191		0.122		0.067	
$\delta\omega_{zest}$ $[deg/s]$	0.141		0.087		0.044		0.167		0.113		0.063	

In cases $D.A$ and $D.B$, the baseline value of R used in the previous simulation is multiplied in run-time by a factor $\mathcal{R}$ that depends on the value of residual associated to the pose measurement (e.g., $\mathcal{R} < 1$ if r is small, and $\mathcal{R} >> 1$ is r is big). The values of $\mathcal{R}$ and the ranges of r are pre-defined via off-line scheduling. Looking at Table 3.2 it should be noted that not only the global performance is increased, but also that the estimated quaternion has become more precise that the measured one. Moreover, it should be noted how the mean and root mean squared errors of the measured quaternion are decreased with respect to the static cases This underlines how the measurements of a recursive tracker are dependent on the estimated pose itself: if the pose estimation improves, the measurements improve too. For the baseline simulation $D.A$, the mean angular error of the estimated quaternion is 1.08 deg at 30 m, 0.74 deg at 20 m, and 0.47 deg at 10 m. Also for case $D.B$ we can observe an improvement of the performance with respect to case $S.B$, with the algorithm that diverges in 6 runs at 20 m and 17 runs at 30 m. Nevertheless, there is still a significant degradation passing from case $D.A$ to case $D.B$. To overcome this degradation of the performance, different solutions can be envisaged. The simpler solution from an algorithmic point of view is to equip the chaser with multiple cameras having different FOV in order to rely, during each RDV phases, on an optimized sensor. However, this solution may be not applicable from a system point of view. The coupling of the VIS camera with a thermal IR camera will surely improve the silhouette extraction. Nevertheless, in order not to compromise the mission in case of failure of a camera sensor, it is necessary to develop a solution that ensures the tracking even with the lowest number of sensors available. For this reason, the development of ad-hoc pre-processing techniques to enhance the silhouette extraction should be envisaged (see Sec.5.1). Finally, the coupling of the tracking algorithm with a pose acquisition algorithm will enable re-initialization in cases where the divergence of the tracking can not be avoided.

Table 3.2: Performance of the non-linear tracking algorithm with dynamic KF calibration

	Case D.A: Perfect silhouette						Case D.B: High fidelity RDV simulator					
	30 m		20 m		10 m		30 m		20 m		10 m	
	mean	RMS	mean	RMS	mean	RMS	mean	RMS	mean	RMS	mean	RMS
$\delta\theta_{meas}$ $[deg]$	1.74	3.11	1.15	2.09	0.73	1.33	4.70	7.15	3.04	5.00	1.60	2.69
$\delta\theta_{est}$ $[deg]$	1.08	1.64	0.74	1.10	0.47	0.68	4.34	5.53	2.66	3.49	1.40	1.75
	σ		σ		σ		σ		σ		σ	
$\delta\omega_{x_{est}}$ $[deg/s]$	0.046		0.034		0.015		0.052		0.042		0.035	
$\delta\omega_{y_{est}}$ $[deg/s]$	0.131		0.087		0.044		0.080		0.062		0.044	
$\delta\omega_{z_{est}}$ $[deg/s]$	0.106		0.066		0.033		0.066		0.053		0.041	

3.5 Real-time implementation

The tracking algorithm discussed in the current chapter was supposed to be tested in real-time on Thales Alenia Space robotic bench in Cannes. The robotic bench is composed by two industrial robotic arms IRB 2400/16 made by ABB which simulate the behavior of the target from one side, and of the chaser on the other. On the target arm it is mounted a down-scaled mock-up of the target model used in this thesis.

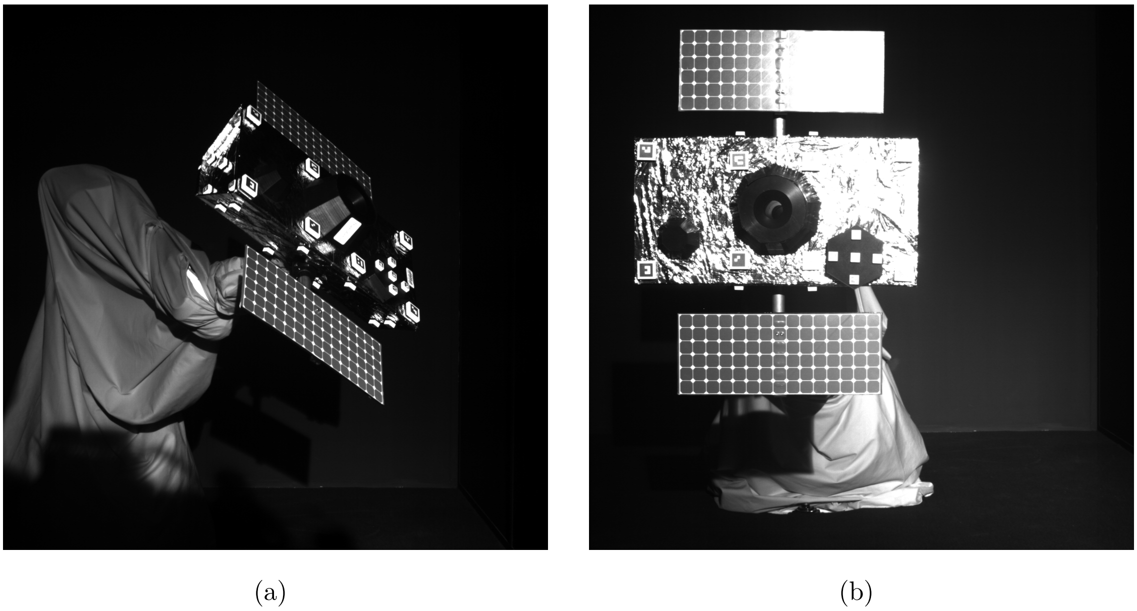

(a) (b)

Figure 3.5: Images of the target mock-up mounted on Thales Alenia Space robotic bench at different relative poses

Figure 3.5 shows the target mock-up at two different relative poses. For a series of reason, it was not possible to complete the validation campaign before the end of the thesis. The main issues encountered were related to the knowledge of the ground-truth actuated by the robotic arms, the extrinsic calibration of the camera and the illumination conditions. In particular, it has proven to be difficult to isolate the target mock-up from the robotic bench background, also because of the reflective texture of the mock-up. Moreover, a poor knowledge of the actuated ground-truth prevented us to correctly evaluate the performance of the tracking algorithm. However, all preliminary steps needed to test the tracking algorithm in real-time on the robotic bench have been carried out, and the tracking algorithm proved real-time estimation capability.

The most important outcome of the real-time test preparation has been the re-coding of the IP-CV algorithms on a static programming language (i.e., C++). The “translation” from Matlab code to C++ has been carried out exploiting Matlab auto-coding application. The Matlab codes (i.e., the masking algorithm, the matching algorithm, the optimization relying on the LM algorithm) have been modified to comply with some general rules for auto-coding (e.g., declaration of type and size of outputs and inputs), and a generic “example” script implementing the full tracking algorithm was written. From this inputs, the Matlab coder generated a library of C++ functions and an example *main*. The functions were then integrated on a C++ script implementing the image pre-processing with Open-CV functions and the pose estimation with our developed code. An important lesson was learned from the auto-coding procedure. When using Matlab coder, all the Matlab functions belonging to Matlab libraries are re-coded in C++ (e.g., *eye*(), *norm*(), *inv*(), etc). However, they are not re-coded in a "universal" way, but only in compliance with the way they are used in the parent function (e.g., if the function *norm*() is applied only to 3 × 1 vectors in the parent function, the auto-coded norm function in C++ will bug if it receives a 4 × 1 input). Thus, the "translation" of a function belonging to the standard Matlab libraries is not unique. This may generate an issue when trying to integrate multiple auto-coded libraries. For this reason, we recommend to rely on user-defined functions instead of functions belonging to the standard Matlab libraries when auto-coding is foreseen.

The porting of the tracking algorithm in C++ has allowed us to better assess the computational load of the method. The pose estimation algorithm was tested on a 2.70 GHz Intel Core i7 processor. The algorithm proved to be very fast, with an average latency -from the Canny edge extraction to the cost function minimization on 500 processed images- of 60.1 millisecond at 40 m (i.e., 16.64 images/s), $62,5$ millisecond at 20 m (i.e., 16 images/s), and 85.8 millisecond par image (i.e., 11.66 images/s) at 10 m. The latency has been decreased of a factor close to 20 when passing from Matlab to C++. It goes without saying that the performance achieved will degrade when the processing chain will run on a space embedded target, but the latency on the testing processor suggests that real-time on space-enabled processor is achievable.

3.6 Conclusion

The current thesis chapter has addressed the problem of monocular model-based pose tracking for space rendezvous, proposing an innovative method relying on silhouette-edges extraction and tracking. The method has been formally developed and implemented. The coupling with the dynamic filter introduced in Chapter 2 enables the robust estimation of the relative pose, as well as the estimation of the translational velocity and rotation rate, of the target. The method has shown high performance and robustness when using perfect silhouette images, and suffered a loss of robustness at high relative distances when tested in closed-loop on realistic images affected by shadows, blur and reflections. Indeed, the use of a dynamic calibration of the KF improves the performance. Different solutions to strengthen the estimation have been proposed, such as the coupling of the visible camera with a thermal infra-red camera in order to improve the silhouette extraction. The proposed estimation chain provides a cost effective solution with a very low computational load, thus compatible with typical space processing capabilities. However, as all the recursive pose tracker, the proposed method does not allow to initialize the pose from the so-called *lost in space* condition (i.e., no prior information of the target pose), neither it is capable of recovering from big pose estimation error. In the next chapter, the problem of pose acquisition will be assessed.

Chapter 4

Pose Estimation by Detection

Contents

As discussed in Chapter 3, traditional 3D tracking has a recursive nature, which provides a strong prior on the pose for each new frame and makes image feature identification and matching relatively easy [Lepetit et al., 2005]. However, recursive methods require initialization and can diverge in presence of local minima (see Sec.3.4.2). For this reason, a recursive tracking algorithm must be complemented with a pose estimation by detection algorithm to enable initial pose acquisition and fault detection. In a *detection* algorithm, the pose is retrieved by exploiting a-priori information on the geometry and appearance of the tracked object, but with no knowledge of the pose at previous instants. Since space-qualified microprocessors have relatively low computational resources, the pose estimation by detection algorithm must be computationally inexpensive. Moreover, it has to cope with the peculiarities of the RDV problem, such as harsh illumination conditions or the presence of textureless and reflective materials on the target.

4.1 Related work

When using monocular vision, the atavistic problem to model and recognize a 3D information from 2D images has to be faced. Monocular pose estimation by detection can be performed using geometric methods or by template matching.
In geometric methods, the observed 2D features in the input image are matched with a database of features computed offline. Then the pose is retrieved by solving the Perspective-n-Point (PnP) problem. Geometric approaches based on local feature are proposed for S/C pose acquisition in [Sharma et al., 2018b, Pesce et al., 2019, Rondao and Aouf, 2018]. In [Sharma et al., 2018b], the weak gradient elimination technique is employed to eliminate the background and the Hough transform is used to extract strong edge features, which are then merged into higher-level features. The correspondences between the image and the 3D model are obtained by paring each detected feature group with each analogous group of the 3D model. Unlike template matching techniques, only a small set of possible matches is hypothesized and then verified. Each combination of feature correspondences is employed to solve the PnP problem using the Efficient-PnP method [Lepetit et al., 2009]. This leads to multiple solutions, which are classified into "high" and "low" confidence poses according to the magnitude of the reprojection error. Only when the reprojection error is below a certain threshold the pose is considered to be correctly acquired. Even if this method doesn't require the storage of the templates, but only the storage of the S/C 3D model, it can still be considered computationally heavy, with an average execution time on a standard 2.5 GHz Intel-Core i-5-4570T processor of more than 8 seconds. In [Pesce et al., 2019] a set of image-model correspondences between the detected features and the 3D model (i.e., *consensus set*) is selected and used to compute a first guess pose solving the PnP problem. The obtained solution allows to re-project the 3D model features on the focal plane and to compare them to the ones detected in the original image, in order to obtain an enlarged consensus until a given threshold is reached. However, methods relying on local feature detection lack robustness in the adverse illumination conditions encountered by spaceborne systems.
In template matching approaches, a *training* set of views of the object is acquired offline to generate a database of templates that are compared at run-time with the input image.

Classical template matching approaches compare the pixels intensities of the templates and the input image according to similarity measures or alignment functions [Petit, 2013]. However, these methods are computationally expensive and lack robustness to illumination changes [Petit, 2013] . Other approaches rely on templates of local features, such as image gradient orientations (e.g., [Dalal and Triggs, 2005, Hinterstoisser et al., 2010]) or binary templates of the extracted edges. An edge-based template matching approach relying on a similarity measure derived from *Chamfer Matching* [Barrow et al., 1977] and on an unsupervised clustering technique based on affinity propagation [Frey and Dueck, 2007] is proposed for S/C pose estimation in [Petit, 2013].

The template matching approaches described are computationally complex because of the evaluation of a large number of possible pose hypotheses and real-time may not be achievable on space qualified processors. They might be appropriate for pose initialization, but cannot be used as a backup algorithm to help detecting divergence in the recursive tracking algorithm. For this reason, recent works have focused on using Convolutional Neural Networks (CNNs) for pose estimation directly from greyscale images in a end-to-end fashion [Sharma and D'Amico, 2020, Sharma et al., 2018a]. However, the reported accuracy is currently still lower than that of geometric methods [Sharma et al., 2018b]. Hybrid approaches, where CNNs are used to extract keypoints (i.e., local features) from the image while the PnP solver is used to compute the pose, have recently shown very good accuracy with synthetic images [Kisantal et al., 2020]. However, this still raises the issue of relying on local feature detection.

The approach proposed in this chapter is to rely on template matching with global features instead of local ones. Global features such as image moments or Fourier descriptors provide a low-dimensional representation of the target's silhouette on a binary image. As for a typical appearance-based template approach, this representation can be matched to its nearest neighbor in a database constructed offline from a sufficiently rich sample of possible poses. However, due to the small amount of data stored for each view, the search for the optimal matching remains computationally inexpensive, resulting in a fast estimation procedure that can be run in parallel of a frame-to-frame tracking algorithm. If global features are computed from binary silhouette images, they become independent from illumination conditions. However, in this case, the pose estimation algorithm should be complemented with a segmentation algorithm, especially when complex background such as the Earth is present in the image

4.2 Pose estimation with global descriptors

Pose estimation methods usually rely on local image features. Local image features, such as object edges and corners, are in general compared to a model using symbolic pattern matching techniques [Breuers, 1999]. Once a structural match between the image features and the object model is found, the object pose is estimated through an iterative fitting technique. On the other side, global image features are in general converted to a pose estimate in a single processing step by means of statistical methods, such as the nearest neighbor search or neural networks. This makes pose estimation based on global image features less complex compared to methods based on local image features. The method consists in computing global features for the current view of the object whose pose is to be determined, and comparing them with a database of features computed offline and representing the whole search space in order to find the best match.

Early attempts to use global features for shape recognition were motivated by aircraft identification applications and relied on Hu's moment invariants ([Dudani et al., 1977, Reeves et al., 1988, Glais and Ayoun, 1994, Breuers, 1999]) and Fourier descriptors ([Wallace and Wintz, 1980, Reeves et al., 1988, Chen and Ho, 1991, Glais and Ayoun, 1994]). Dudani proposed in [Dudani et al., 1977] to use these features not only for aircraft recognition but also for pose estimation, inspiring follow-up work in [Reeves et al., 1988, Chen and Ho, 1991, Glais and Ayoun, 1994, Breuers, 1999].Unfortunately, Hu's invariants, although very commonly used, are now known to form an incomplete set, see Section 4.4, which results in limited discrimination capabilities. During the past two decades, progress in pattern recognition led to the development of more powerful sets of rotation-invariant global features, which however were never tested for the pose estimation problem. In this paper, we consider complex moments (CMs), Zernike moments (ZMs) and Fourier descriptors (FDs). CMs directly improve on Hu's invariants [Flusser et al., 2009], see Section 4.4.1. ZMs form a set of orthogonal moments, see Section 4.4.2, with advantages in terms of information redundancy and image representation capabilities [Abu-Mostafa and Psaltis, 1984, Teh and Chin, 1988]. Some authors claim that Pseudo-ZM [Gishkori and Mulgrew, 2019] are more robust than ZM with respect to additive noise, but their independence (see Section 4.4) is considered questionable [Flusser et al., 2009]. ZMs were used in [Gandía and Casas, 2003] to determine the orientation of a S/C from silhouette images, as we do here. However, the authors did not fully exploit the power of the ZMs since they do not use the rotation invariance to reduce the complexity of the problem, as we explain in Section 4.3. In [Chang and Ghosh, 2000] the amplitude of ZMs up to the 29^{th} order is used as rotation invariant to estimate the pose of airplanes, but the use of only amplitudes leads to an incomplete set of descriptors, which results in limited discrimination capabilities, as we explain in Section 4.4.2. Moreover, in [Chang and Ghosh, 2000] the in-plane rotation (see Section 4.3) is retrieved by comparing the phase of the ZMs in the database with the phase of the ZMs computed on the observed image. This approach leads to a doubling of the dimension of the descriptor database. In Section 4.3, we describe a method to retrieve the in-plane rotation that requires storing only one descriptor for each database view. The amplitude of the Pseudo-ZMs is also used as rotation invariant in [Khotanzad and Liou, 1996], but this work proposes a method to compute only two of the six degrees-of-freedom (DOF) of the pose. A coarse initialization of two attitude angles using ZM is done in [Rondao et al., 2020]. Finally, Pseudo-ZMs are used in [Fu and Sun, 2017] for aircraft pose estimation from contour images. First, the authors determine with Pseudo-ZMs two of the three attitude angles of the observed object. Then, shape context descriptors are used to retrieve the remaining DOF. However, as we details in Section 4.3, the remaining DOF can be computed using geometric moments, resulting in a simpler and faster estimation method.

Moment-based invariants are generally computed using the whole silhouette of the observed object, since the performance of boundary moments quickly deteriorates in the presence of noise and discretization effects on the images [Reeves et al., 1988, Barczak et al., 2011], and thus CMs and ZMs are generally sensitive to the distribution of the "mass" in the image silhouette. On the other hand, FDs are computed only from the contours of the observed object, hence are more sensitive to changes in the object boundaries [Reeves et al., 1988]. This motivates our goal of comparing Fourier descriptors and moment-based invariants.

This Chapter is organized as follows. In Section 4.3, we present a detection algorithm for the pose estimation of a S/C of known geometry, based on global features computed from a single binary image. First, the method determines two of the three Euler angles that describe the attitude of the S/C using rotation invariant features. Then, the remaining degrees of freedom, i.e., the relative translation and the in-plane rotation, are retrieved using geometric moments (GMs). In Section 4.4 the theory underlying the computation of rotation invariants based on CMs, ZMs and FDs is recalled, and we propose a novel method to compute rotation invariant FDs. Then, Section 4.5 analyzes and compares the performance of these three sets of global descriptors, characterizing their behavior in different conditions typically encountered in operational RDV scenarios. To the best of our knowledge, no prior work provides such an evaluation of the performance of modern global descriptors for the problem of detecting and estimating the pose of a $3D$ object.

4.3 Proposed method

The interest in using global features such as Fourier descriptors [Wallace and Wintz, 1980, Reeves et al., 1988, Chen and Ho, 1991, Glais and Ayoun, 1994] or image moments [Dudani et al., 1977, Reeves et al., 1988, Glais and Ayoun, 1994, Breuers, 1999] is that these features can be made invariant to translation, to scaling, and most importantly to rotation. If a bi-dimensional shape is described by such invariant features, the value of the features will not depend on the position of the shape centroid (translation invariance), on the shape dimension (scaling invariance), and on the rotation of the shape in the image plane (rotation invariance). The shape of the projected silhouette does depend both on the relative attitude and the relative position of the observed object, and the contribution of the translation is coupled with the contribution of the attitude. However, under the hypothesis of weak-perspective model, those effects can be decoupled at the cost of an acceptable degradation of the estimation performance (as we show in Sec.4.5). The weak perspective model can be assumed when the depth of the object along the line of sight is small compared to the distance from the camera [Petit, 2013], or when the FOV is relatively small or the object is placed near to the center of the FOV (i.e., $tr^{cam}_{cam-tg_x} \sim 0$ and $tr^{cam}_{cam-tg_y} \sim 0$ in Eq.(3.1)). These are indeed conditions that are met during the pose acquisition phase. In the weak perspective model it is assumed that all points on a 3D object are at the same distance z from the camera without significant errors in the projection with respect to the full pinhole perspective model. Under this hypotheses, it can be assumed that the distance $tr^{cam}_{cam-tg_z}$ affects only the scale of the projected silhouette, while the components $tr^{cam}_{cam-tg_x}$ and $tr^{cam}_{cam-tg_y}$ affect only the position of the silhouette centroid. Thus, if global invariant features are used do describe the S/C silhouette at a given pose, the value of the features will depend only on the roll and pitch angles φ and ϑ of the matrix R_{cam-tg} (Eq.(3.1)). In fact, the yaw angle ψ only affects the rotation of the projected shape in the plane image, as visible in Fig.4.1.

The principle of the pose estimation algorithm that we propose is the following. During an off-line process, a set of synthetic views of the target, referred to as *training* images in the following, is generated for a sufficiently large number of discrete values of the pairs $(\varphi, \vartheta) \in]-\pi, \pi] \times]-\pi/2, \pi/2]$. When generating the database, the yaw angle $\psi = \psi_{train}$ is set to zero. Moreover, the camera is pointing perfectly towards the target COM, i.e., $tr^{cam}_{cam-tg_y} = tr^{cam}_{cam-tg_y} = 0$, and the distance $tr^{cam}_{cam-tg_z} = d$ is kept fixed for all the views in a database. This set-up is described by Fig.4.1. The choice of distance $d = d_{train}$ can in fact affect the performance of the pose estimation, as we discuss in Section 4.5.2. For a given pair (φ, ϑ), the position of the camera expressed in the target RF is $tr^{tg}_{cam-tg} = d_{train} \cdot [\sin\vartheta, \; -\cos\vartheta \sin\varphi, \; -\cos\vartheta \cos\varphi]^T$. Under the assumption that

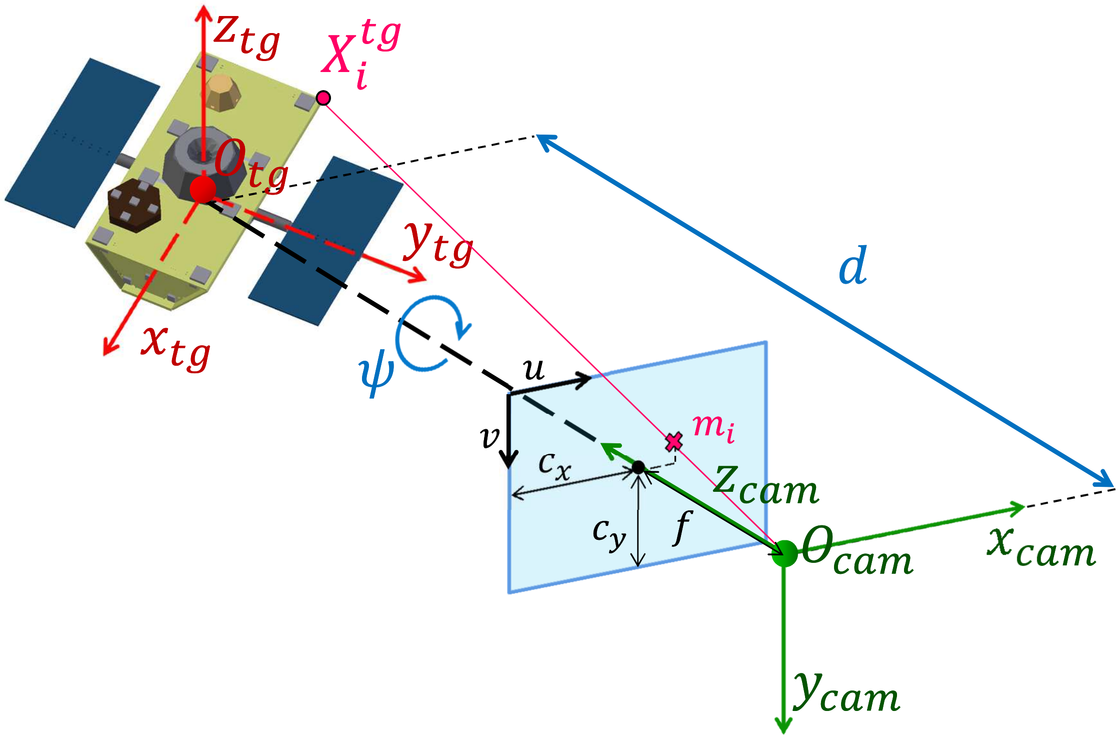


Figure 4.1: Schematic representation of the pose estimation problem using a monocular image

the camera's optical axis is pointing towards the target's COM, the locus of the points $tr_{cam-tg}^{tg}(d_{train}, \varphi, \vartheta)$ is represented by the sphere of radius d_{train} centered at the target COG. Thus, the relative attitudes used to generate the training images can be assigned by selecting N_w random points on the sphere. In order to avoid oversampling of the polar zone, which would be the result of a uniform sampling of φ and ϑ, the points are assigned using the algorithm suggested in [Kuffner, 2004]

$$\begin{cases} \varphi & = 2\pi \text{ rand}_\varphi - \pi \\ \vartheta & = \text{asin}(1 - 2\text{rand}_\vartheta) \end{cases} \tag{4.1}$$

where rand_φ and rand_ϑ are two independent random variables uniformly distributed in the interval $]0,\ 1]$. Note that although random attitude sampling was used here for benchmarking the different sets of global descriptor, a deterministic sampling may be more adequate for the nominal pose estimation solution. In that case, a spiral scheme could be used to generate uniformly distributed samples on the unit sphere [Koay, 2011].

Once the images corresponding to the N_w viewpoints are generated, the global invariant descriptors are computed for each view. The size of the resulting database is $N_w \times N_f$, with N_f the dimension of the feature vector. At run-time, when the camera acquires a new image of the target (referred to as *test* image in the sequel), the algorithm computes the descriptor vector associated to the resulting view and finds in the database the pair $(\varphi_{meas}, \vartheta_{meas})$ with the closest descriptor vector (minimizing the Euclidean distance), i.e., performs a nearest neighbor search. The two remaining degrees of freedom, i.e., the yaw angle ψ and the relative camera-target distance d, are estimated using geometric moments (GMs), according to the procedure described in the following. From a mathematical point of view, moments are projections of a function into a polynomial basis [Flusser et al., 2009]. GMs rely on the most intuitive power basis $p_{pq}(x, y) = x^p y^q$. The general definition for a geometric moment of an image is:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y)\, \mathrm{d}x\, \mathrm{d}y\,, \tag{4.2}$$

where $f(x, y)$ is a function equal to the intensity level of the image in correspondence of the point (x, y). When binary images such as object silhouettes are considered, $f(x, y)$ is equal to 1 inside silhouette and 0 elsewhere. For binary images, the lower order geometric moments are associated with geometrical properties of the observed silhouette. For example, m_{00} is the area, while $m_{10}/m_{00} = x_c$ and $m_{01}/m_{00} = y_c$ are the x and y coordinates of the silhouette's centroid. Geometric moments can be easily made translation and scale invariant. Translation invariance is obtained by computing the *central* geometric moments μ_{pq}, according to:

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_c)^p (y - y_c)^q f(x, y)\, \mathrm{d}x\, \mathrm{d}y\,. \tag{4.3}$$

Scale invariance is obtained by dividing the central geometric moments by an appropriated power of the area:

$$\nu_{pq} = \frac{\mu_{pq}}{m_{00}^{w}}, \quad \text{with} \qquad w = \frac{p+q}{2} + 1\,. \tag{4.4}$$

From a given image, the yaw angle ψ of R_{cam-tg} can be retrieved using the second order central moments, as follows.
It is known in fields such as structural engineering that it is possible to determine the principal axis of inertia of a cross-section given its second order central geometric moments μ_{20}, μ_{11}, μ_{02}. Given a certain shape, it is possible to compute its central GM μ'_{pq} after an in-plane rotation of an angle α using its central GM μ_{pq} before the rotation:

$$\mu'_{pq} = \sum_{l_1=0}^{p} \sum_{l_2=0}^{q} r_{11}^{l_1} r_{21}^{l_2} r_{12}^{p-l_1} r_{11}^{q-l_2} \mu_{l_1+l_2, p+q-(l_1+l_2)} \tag{4.5}$$

where r_{11}, r_{12}, r_{21}, r_{22} are the elements of the rotation matrix that describes the in-plane rotation of the shape:

$$\begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{4.6}$$

Given this relation, it is possible to compute the *inclination* $\tilde{\psi}_0$ of the major axis of inertia of the observed shape with the following procedure. Applying a rotation of $\tilde{\psi}_0$ to the reference system and computing the rotated centered moments, one will obtain the principal moments of inertia of the shape. The second order rotated moments can be obtained both with Eq.(4.5), or applying Eq.(4.2) for the rotated and centered coordinates

$$\mu'_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(x\cos(\tilde{\psi}_0) + y\sin(\tilde{\psi}_0)\right)^p \left(-x\sin(\tilde{\psi}_0) + y\cos(\tilde{\psi}_0)\right)^q f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{4.7}$$

The resulting second order rotated centered moments are:

$$\begin{cases} \mu'_{20} = & \mu_{20}\cos^2(\tilde{\psi}_0) + \mu_{02}\sin^2(\tilde{\psi}_0) + 2\mu_{11}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) \\ \mu'_{02} = & \mu_{20}\sin^2(\tilde{\psi}_0) + \mu_{02}\cos^2(\tilde{\psi}_0) - 2\mu_{11}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) \\ \mu'_{11} = & -\mu_{02}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) - \mu_{20}\sin(\tilde{\psi}_0)\cos(\tilde{\psi}_0) + \mu_{11}\left(\cos^2(\tilde{\psi}_0) - \sin^2(\tilde{\psi}_0)\right) \end{cases} \tag{4.8}$$

The angle $\tilde{\psi}_0$ can be easily computed by imposing the mixed moment μ_{11} to be null, has it happens when the second order moments are computed with respect to the principal axis of inertia. Alternatively, one can maximize μ'_{20}. Applying the double-angles trigonometric formulas, the third equation of system (4.8) becomes:

$$\begin{aligned} \mu'_{11} = & -\frac{1}{2}(\mu_{20} - \mu_{02})\sin(2\tilde{\psi}_0) + \mu_{11}\cos(2\tilde{\psi}_0) = 0 \\ \tilde{\psi}_0 = & \frac{1}{2}\mathrm{atan2}\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \end{aligned} \tag{4.9}$$

However, to recover the full in-plane angle of rotation $\psi_0 \in]-\pi,\pi]$ of the silhouette, we still need to determine a specific direction along the axis of inertia, i.e., distinguish between $\psi_0 = \tilde{\psi}_0$ or $\psi_0 = \tilde{\psi}_0 - \pi$.

For this, the third order GMs can be used, since they change sign under a rotation of π [Tahri, 2004]. Central moments in the image frame rotated by an angle $\tilde{\psi}_0$ are given by Eq.(4.5), with $\alpha = \tilde{\psi}_0$ in Eq.(4.6). Then, as a convention, we define the in-plane rotation ψ_0 to be the direction for which the moment μ'_{30} after rotation by ψ_0 is positive, i.e.,

$$\psi_0 = \begin{cases} \tilde{\psi}_0 & \text{if } \mu'_{30} > 0\,, \\ \tilde{\psi}_0 - \pi & \text{if } \mu'_{30} < 0\,. \end{cases} \tag{4.10}$$

The ambiguity can be resolved only if the silhouette is not rotationally symmetric. In the case of a silhouette having an N-fold rotation symmetry (N-FRS) (i.e., if it repeats itself after rotation around its centroid by $2\pi j/N$, for all $j = 1, ..., N$), N solutions are possible. In such case, only an observation of the target pose on a set of consecutive frames could help solve the ambiguity.

Once $\psi_{0_{test}}$ is determined for the current image, we can deduce the measured yaw angle ψ_{meas} of the camera frame by the following procedure. For the training view in the database that best matches the current view, which we recall was generated with $\psi = \psi_{train} = 0$, we can similarly compute the angle $\psi_{0_{train}}$ of the major axis of inertia (these angles can be precomputed and stored with each view). Then ψ_{meas} is determined by

$$\psi_{meas} = \psi_{0_{test}} + (\psi_{train} - \psi_{0_{train}}). \tag{4.11}$$

Finally, we obtain a measure d_{meas} of the relative camera-target distance along the optical axis from the zeroth order moment $m_{00_{test}}$ of the current silhouette image, namely

$$d_{meas} = \sqrt{m_{00_{train}}/m_{00_{test}}} \cdot d_{train}, \tag{4.12}$$

where $m_{00_{train}}$ is the zeroth order moment for the best matching training view, which again can be precomputed and stored in the database.

When constructing the database, the camera optical axis z_{cam} is always pointing towards the target's COM. However, during the pose acquisition phase of a RDV, the camera will likely be pointing towards the observed silhouette centroid instead of the target COM. It may also be the case that, for some operational reasons such as the need for aligning chaser and target docking interfaces, pointing exactly toward the COM cannot be done. In such a case, recalling (3.7) and exploiting the fact that $d = \sqrt{(tr^{cam}_{cam-tg_x})^2 + (tr^{cam}_{cam-tg_y})^2 + (tr^{cam}_{cam-tg_z})^2} \sim tr^{cam}_{cam-tg_z}$, the components $tr^{cam}_{cam-tg_x}$ and $tr^{cam}_{cam-tg_y}$ can be approximated by

$$\begin{cases} tr^{cam}_{cam-tg_x} &= \dfrac{d}{f}\,(x_{c_{test}} - c_x) \\ tr^{cam}_{cam-tg_y} &= \dfrac{d}{f}\,(y_{c_{test}} - c_y) \end{cases} \tag{4.13}$$

where $(x_{c_{test}}, y_{c_{test}})$ are the coordinate of the observed silhouette centroid. We remind that f denotes the focal length of the camera and (c_x, c_y) the principal point of the image. The identities (4.13) can be used to ensure camera pointing even before the target's full pose has been acquired.

By using rotation invariants, we can drastically reduce the dimension of the database that needs to be stored and increase the speed of the search. For example, if the library were built using a uniform discretization of 5 deg for the Euler angles, the database table would contain $N_w = 72 \times 36 \times 72 = 186624$ synthetic views if ψ were included in the search space, but only $N_w = 72 \times 36 = 2592$ synthetic views if rotations invariants where used. In Sections 4.4.1, 4.4.2, and 4.4.3 we provide details about the computation of invariant global features using CMs, ZMs, and FDs.

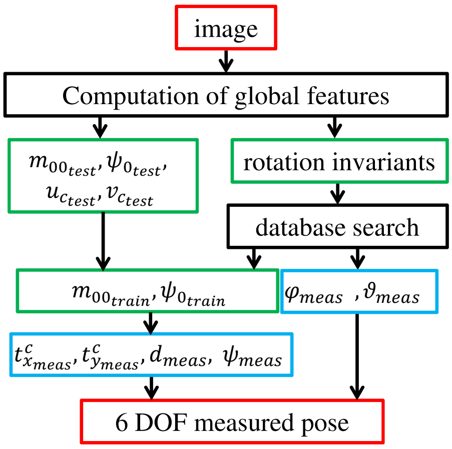


image
Computation of global features
$m_{00_{test}}, \psi_{0_{test}}, u_{c_{test}}, v_{c_{test}}$
rotation invariants
database search
$m_{00_{train}}, \psi_{0_{train}}$
φ_{meas} , ϑ_{meas}
$t^c_{x_{meas}}, t^c_{y_{meas}}, d_{meas}, \psi_{meas}$
6 DOF measured pose

Figure 4.2: Structure of the proposed pose estimation algorithm.

4.4 Computation of the invariant global features

While designing translation and scale invariants based on moments is generally straightforward, see (4.3) and (4.4), rotation invariance is more difficult to obtain. In a seminal paper [Hu, 1962], Hu introduced a set of 7 rotation invariants based on combination of second and third order geometric moments:

$$\begin{aligned}
Hu_1 =& \quad m_{20} + m_{02} \\
Hu_2 =& \quad (m_{20} - m_{02})^2 + 4m_{11}^2 \\
Hu_3 =& \quad (m_{30} - 3m_{12})^2 + (m_{03} - 3m_{21})^2 \\
Hu_4 =& \quad (m_{30} + m_{12})^2 + (m_{03} + m_{21})^2 \\
Hu_5 =& \quad (m_{30} - 3m_{12})(m_{30} + m_{12})\left[(m_{30} + m_{12})^2 - 3(m_{03} + m_{21})^2\right] \\
& \quad -(m_{30} - 3m_{21})(m_{03} + m_{21})\left[3(m_{30} + m_{12})^2 - (m_{03} + m_{21})^2\right] \\
Hu_6 =& \quad (m_{20} - m_{02})\left[(m_{30} + m_{12})^2 - (m_{03} + m_{21})^2\right] + 4m_{11}(m_{30} + m_{12})(m_{03} + m_{21}) \\
Hu_7 =& \quad -(m_{30} - 3m_{12})(m_{30} + m_{12})\left[(m_{30} + m_{12})^2 - 3(m_{03} + m_{21})^2\right] \\
& \quad -(m_{30} - 3m_{21})(m_{03} + m_{21})\left[3(m_{30} + m_{12})^2 - (m_{03} + m_{21})^2\right]
\end{aligned} \tag{4.14}$$

where, if the geometric moments m_{ij} are replaced by the central moments μ_{ij} (Eq.(4.3)) or by the normalized moments ν_{ij} (Eq.(4.4)), the relations give invariants not only to rotation but also to translation (i.e., central moments), or to translation and scaling (i.e., with normalized central moments). Hu's set is one of the first set of invariant that has been used for visual pattern recognition. Unfortunately, Hu's approach cannot be generalized, so that only invariants up to the third order can be derived, limiting the descriptive power of these features. On the other hand, by choosing other polynomial basis functions to compute the moments, as for CMs and ZMs, one can derive general rules for the computation of rotation invariants of up to any order. In particular, a set of invariants up to a given order r must be *independent* and *complete* [Flusser et al., 2009]. A set is independent if none of its elements can be expressed as a function of the other elements. It is complete if any rotation invariant up to the order r can be expressed by means of the set elements only. A complete set contains both *true invariants* and *pseudo-invariants*. True invariants are invariants that do not change sign under reflection, while pseudo-invariants (also known as *skew-invariants*) do [Flusser et al., 2009]. The capability of pseudo-invariants to detect mirror reflections is very important for the pose estimation problem, since a mirrored image corresponds to a different S/C attitude.

4.4.1 Rotation invariants with complex moments

CMs are computed by projecting the image function on the basis $p_{pq}(x,y) = (x+iy)^p(x-iy)^q$, where i is the imaginary unit. The complex moment c_{pq} of order $p+q$ is defined by the following formula:

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x+iy)^p(x-iy)^q f(x,y)\,\mathrm{d}x\,\mathrm{d}y\,. \tag{4.15}$$

Translation and scaling invariance can be obtained as for the standard geometric moments, replacing m_{00}, m_{10} and m_{01} in (4.3) and (4.4) by c_{00}, $\mathrm{Re}(c_{10})$ and $\mathrm{Im}(c_{10})$ respectively. Each CM can be expressed in terms of geometric moments of the same order, ad vice-versa, according to [Flusser et al., 2009]:

$$\begin{aligned} c_{pq} &= \sum_{k=0}^{p}\sum_{j=0}^{q}\binom{p}{k}\binom{q}{j}(-1)^{q-j}\cdot j^{p+q-k-j}\cdot m_{k+j,p+q-k-j} \\ m_{pq} &= \frac{1}{2^{P+q}i^q}\sum_{k=0}^{p}\sum_{j=0}^{q}\binom{p}{k}\binom{q}{j}(-1)^{q-j}\cdot c_{k+j,p+q-k-j} \end{aligned} \tag{4.16}$$

CMs carry the same amount of information as GMs, but are more convenient to derive an independent and complete set (also referred to as a *basis*) of rotational invariants up to any order, as shown by Flusser [Flusser, 2000]. The key points of the approach are recalled here. If the complex moments are expressed in polar coordinates ($x = r\cos\theta$, $y = r\sin\theta$), (4.15) becomes:

$$\begin{aligned} c_{pq} &= \int_0^{\infty}\int_0^{2\pi}(r\cos\theta + ir\sin\theta)^p(r\cos\theta - ir\sin\theta)^q f(r,\theta)\; r\;\mathrm{d}r\;\mathrm{d}\theta \\ &= \int_0^{\infty}\int_0^{2\pi} r^{p+q+1}\mathrm{e}^{i(p-q)\theta}f(r,\theta)\;\mathrm{d}r\;\mathrm{d}\theta\,. \end{aligned} \tag{4.17}$$

From (4.17) we can see that the complex conjugate of a CM satisfies $c^*_{pq} = c_{qp}$ and that a pure rotation of the image around the origin by an angle α changes CMs according to

$$c'_{pq} = \mathrm{e}^{-i(p-q)\alpha}c_{pq}\,, \tag{4.18}$$

which shows that the magnitude of a CM is a rotation invariant. However CM magnitudes do not generate a complete set. Instead, considering complex moments up to the order $r \geq 2$, we construct the following basis of rotation invariants [Flusser, 2000]

$$\mathcal{B}_r = \left\{ \Phi_{pq} := c_{pq}\, c_{q_0p_0}^{p-q} \mid p \geq q \text{ and } p+q \leq r \right\} , \tag{4.19}$$

where p_0 and q_0 that can be arbitrary chosen such that $p_0 + q_0 \leq r$, $p_0 - q_0 = 1$, $q_0 \neq 0$, and $c_{q_0p_0} \neq 0$. Rotation invariance follows from

$$c'_{pq} c'^{p-q}_{q_0p_0} = \mathrm{e}^{-i(p-q)\alpha} c_{pq} \cdot \left(c_{q_0p_0} \mathrm{e}^{-i(q_0-p_0)\alpha} \right)^{p-q} = c_{pq} c_{q_0p_0}^{p-q} . \tag{4.20}$$

The exponents p_0, q_0 are generally chosen as small as possible, e.g., $p_0 = 2$ and $q_0 = 1$, because high order moments are more sensitive to noise [Teh and Chin, 1988]. Each basis element Φ_{pq} such that $p \neq q$, except $\Phi_{p_0q_0}$, provides two real-valued invariants, corresponding to the real and imaginary parts of Φ_{pq}. It can be proven that the real part is a true invariant, while the imaginary part is a pseudo-invariant. Hu's invariants can be expressed from the elements of $\mathcal{B}_3$ (i.e., $\{\Phi(1,1),\ \Phi(2,1),\ \Phi(2,0),\ \Phi(3,0)\}$, with $p_0 = 2$ and $q_0 = 1$):

$$\begin{aligned}
Hu_1 &= \Phi(1,1) &&= c_{11} \\
Hu_2 &= \frac{|\Phi(2,0)|^2}{|\Phi(2,1)|^2} &&= c_{20}c_{02} \\
Hu_3 &= \frac{|\Phi(3,0)|^2}{|\Phi(2,1)|^3} &&= c_{30}c_{03} \\
Hu_4 &= \Phi(2,1) &&= c_{21}c_{12} \\
Hu_5 &= \mathrm{Re}\,(\Phi(3,0)) &&= \mathrm{Re}\,(c_{30}c_{12}^3) \\
Hu_6 &= \mathrm{Re}\,(\Phi(2,0)) &&= \mathrm{Re}\,(c_{20}c_{12}^2) \\
Hu_7 &= \mathrm{Im}\,(\Phi(3,0)) &&= \mathrm{Im}\,(c_{30}c_{12}^3)
\end{aligned} \tag{4.21}$$

The equivalency can be demonstrated using Eq(4.16). From Eq.(4.21) it is possible to see how Hu's set is not independent, as it holds that $Hu_3 = (Hu_5^2 + Hu_7^2)/Hu_4^3$. Moreover it is not possible to fully reconstruct the set of CMs basis $\mathcal{B}_3$ from Hu's invariants, meaning that the set is also incomplete. In particular, it is not possible to determine the sign of $\mathrm{Im}\,(\Phi(2,0))$ from Hu's set, meaning that the set is missing a skew-invariant. Thus, complex moment invariants up to the third order (six invariant) will surely provide better recognition performance than the seven Hu's invariants.

Note that if an object has an N-FRS, then all its complex moments with non-integer $(p-q)/N$ are equal to zero. In [Flusser and Suk, 2006] Flusser provide an extension of (4.19):

$$\mathcal{B}_N = \left\{ \Phi_N(p,q) \equiv c_{pq} c_{q_0p_0}^k | p \geq q \wedge p+q \leq r \wedge k \equiv (p-q)/N \text{is integer} \right\} \tag{4.22}$$

with p_0 and q_0 that can be arbitrary chosen such that $p_0 + q_0 \leq r$, $p_0 - q_0 = N$ and $c_{q_0 p_0} \neq 0$.

For our pose estimation algorithm, CM invariants up to order 10 (i.e., $N_f = 62$) are computed using the basis in Eq.(4.19) and stored in the database. The elements $\Phi_{00} = c_{00}$ and $\Phi_{10} = c_{10} c_{12}$ are not included in the set because c_{00} and c_{10} are already used to achieve scaling and translation invariance. The list of the CMs basis up to the 10^{th} order is shown in Table 4.1.

Because high-order moments have much higher magnitude than low order moments, the components of the feature vector must be normalized before applying the minimum Euclidean distance criterion. This issue, which is already present with Hu's invariants, has been addressed in the literature using different approaches, e.g., using z-score normalization [Dudani et al., 1977, Breuers, 1999] or variance balancing [Reeves et al., 1988, Khotanzad and Liou, 1996]. Here, we follow the approach suggested in [Mallick, 2018] of replacing each feature F by F_n, with

$$F_n = \operatorname{sign}(F) \cdot \log(|F|)\,, \tag{4.23}$$

which was found to provide the best recognition capabilities for CMs invariants.

Table 4.1: Complex Moments basis up to 10^{th} order.

$\Phi_{11} = c_{11}$	$\Phi_{32} = c_{32} c_{12}$	$\Phi_{80} = c_{80} c_{12}^{8}$	$\Phi_{44} = c_{44}$
$\Phi_{21} = c_{21} c_{12}$	$\Phi_{60} = c_{60} c_{12}^{6}$	$\Phi_{71} = c_{71} c_{12}^{6}$	$\Phi_{54} = c_{54} c_{12}$
$\Phi_{20} = c_{20} c_{12}^{2}$	$\Phi_{51} = c_{51} c_{12}^{4}$	$\Phi_{62} = c_{62} c_{12}^{4}$	$\Phi_{10,0} = c_{10,0} c_{12}^{10}$
$\Phi_{30} = c_{30} c_{12}^{3}$	$\Phi_{33} = c_{33}$	$\Phi_{53} = c_{53} c_{12}^{2}$	$\Phi_{91} = c_{91} c_{12}^{8}$
$\Phi_{40} = c_{40} c_{12}^{4}$	$\Phi_{70} = c_{70} c_{12}^{7}$	$\Phi_{90} = c_{90} c_{12}^{9}$	$\Phi_{82} = c_{82} c_{12}^{6}$
$\Phi_{31} = c_{31} c_{12}^{2}$	$\Phi_{61} = c_{61} c_{12}^{5}$	$\Phi_{81} = c_{81} c_{12}^{7}$	$\Phi_{73} = c_{73} c_{12}^{4}$
$\Phi_{22} = c_{22}$	$\Phi_{52} = c_{52} c_{12}^{3}$	$\Phi_{72} = c_{72} c_{12}^{5}$	$\Phi_{64} = c_{64} c_{12}^{2}$
$\Phi_{50} = c_{50} c_{12}^{5}$	$\Phi_{43} = c_{43} c_{12}$	$\Phi_{63} = c_{63} c_{12}^{3}$	$\Phi_{55} = c_{55}$
$\Phi_{41} = c_{41} c_{12}^{3}$			

4.4.2 Rotation invariants with Zernike moments

Zernike Moments are a family of orthogonal moments, i.e., the corresponding polynomial basis satisfies

$$\iint\limits_{\Omega} p_{pq}(x,y) \cdot p_{jk}(x,y)\,\mathrm{d}x\mathrm{d}y = 0 \tag{4.24}$$

for all $q \neq j$, $p \neq k$, where Ω is called the *region of orthogonality* and must contain the support of the image f, which must therefore typically be rescaled. For ZMs, Ω is taken to be the unit disk. Given an image expressed in polar coordinates, ZMs are defined by [Flusser et al., 2009]

$$A_{nl} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 V^*_{nl}(r,\theta) f(r,\theta)\, r \; \mathrm{d}r \mathrm{d}\theta \tag{4.25}$$

where n is a nonnegative integer called the order, $l \in \{-n, -n+2, \ldots, n\}$ is called the repetition (note that the difference $n - |l|$ is always even), and V_{nl} denotes the Zernike polynomials

$$V_{n,l}(r,\theta) = R_{n,l}(r) \mathrm{e}^{il\theta} \tag{4.26}$$

with radial part

$$R_{n,l}(r) = \sum_{s=0}^{(n-|l|)/2} (-1)^s \frac{(n-s)!}{s!((n+|l|)/2-s)!((n-|l|)/2-s)!} r^{n-2s} \,. \tag{4.27}$$

The radial part can be also written as

$$R_{n,l}(r) = \sum_{k=|l|,|l|+2\ldots}^{n} B_{nlk} r^k, \quad \text{with} \quad B_{nlk} = \frac{(-1)^{(n-k)/2}((n+k)/2)!}{((n-k)/2)!((k+l)/2)!((k-l)/2)!} \tag{4.28}$$

where the coefficients B_{nlk} allow writing the relation between ZMs and GMs:

$$A_{nl} = \frac{n+1}{\pi} \sum_{k=|l|,|l|+2\ldots}^{n} \sum_{j=0}^{(k-|l|)/2} \sum_{m=0}^{|l|} \binom{(k-|l|)/2}{j} \binom{|l|}{m} w^m B_{nlk} m_{k-2j-m,2j+m}, \quad \text{with} \quad w = \begin{cases} -1 & l > 0 \\ 1 & l \leq 0 \end{cases} \tag{4.29}$$

The radial functions satisfy $R_{n,-l}(r) = R_{nl}(r)$, so that $A_{n,-l} = A^*_{nl}$ and ZMs with repetition $l = 0$ are real valued moments.

Different methods have been proposed to normalize the image to the unit disk. In [Khotanzad and Hong, 1990], each shape is resized so that its zeroth order GM m_{00} is set to a predetermined value, while in [Gandía and Casas, 2003] a fixed-dimension bounding box is used instead of the zeroth order moment. In this paper, we simply transform the coordinates of the points belonging to the object's silhouette to normalized central polar coordinates, i.e., $\theta = \mathrm{atan}\left((x-x_c)/(y-y_c)\right)$ and $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}/r_{max}$, with r_{max} the maximum value among the radii of the considered silhouette. This approach, also in [Khotanzad and Liou, 1996], ensures that all the points in the object's silhouette are used to compute the moments, and also provides translation invariance. Scaling invariance is obtained by dividing A_{nl} by the zeroth order moment A_{00}, as suggested by [Flusser et al., 2009].

Teague [Teague, 1980] was the first to propose a set of rotation invariants based on ZMs, up to the eighth order, but gave no general rule to derive invariants of higher order. As for CMs, some authors [Khotanzad and Liou, 1996, Khotanzad and Hong, 1990, Chang and Ghosh, 2000] use only the magnitude $|A_{nl}|$, but this provides an incomplete set of invariants because the information carried by the pseudo-invariants is lost. Wallin [Wallin and Kubler, 1995] noted that ZMs, which are complex valued moments, behave as CMs under rotation, so that rotation invariance can be obtained by multiplying ZMs by an appropriate phase-cancellation term, see (4.18) and the discussion below. As suggested by [Flusser et al., 2009], this term has to be searched among the ZMs with repetition 1, starting from A_{31}. Similarly to CMs, these moments are equal to zero for objects having a rotational symmetry. In this case, the normalization moment should be searched within the ZMs having repetition 2. Given the normalizing moment $A_{n_r l_r}$, the normalized ZM Z_{nl} is

$$Z_{nl} = A_{nl}\mathrm{e}^{-il\phi}, \quad \text{with} \quad \phi = \frac{1}{l_r}\mathrm{atan}\left[\frac{\mathrm{Im}(A_{n_r l_r})}{\mathrm{Re}(A_{n_r l_r})}\right]. \tag{4.30}$$

Table 4.2: Zernike radial polynomial up to 10^{th} order

$R_{00} = 1$	$R_{60} = 20r^6 - 30r^4 + 12r^2 - 1$	$R_{88} = r^8$
$R_{11} = r$	$R_{62} = 15r^6 - 20r^4 + 6r^2$	$R_{91} = 126r^9 - 280r^7 + 210r^5 - 60r^3 + 5r$
$R_{20} = 2r^2 - 1$	$R_{64} = 6r^6 - 5r^4$	$R_{93} = 84r^9 - 169r^7 + 105r^5 - 20r^3$
$R_{22} = r^2$	$R_{66} = r^6$	$R_{95} = 36r^9 - 56r^7 + 21r^5$
$R_{31} = 3r^3 - 2r$	$R_{71} = 35r^7 - 60r^5 + 30r^3 - 4r$	$R_{97} = 9r^9 - 8r^7$
$R_{33} = r^3$	$R_{73} = 21r^7 - 30r^5 + 10r^3$	$R_{99} = r^9$
$R_{40} = 6r^4 - 6r^2 + 1$	$R_{75} = 7r^7 - 6r^5$	$R_{10,0} = 252r^{10} - 630r^8 + 560r^6 - 210r^4 + 30r^2 - 1$
$R_{42} = 4r^4 - 3r^2$	$R_{77} = r^7$	$R_{10,2} = 210r^{10} - 504r^8 + 420r^6 - 140r^4 + 15r^2$
$R_{44} = r^4$	$R_{80} = 70r^8 - 140r^6 + 90r^4 - 202 + 1$	$R_{10,4} = 120r^{10} - 252r^8 + 168r^6 - 35r^4$
$R_{51} = 10r^5 - 12r^3 + 3r$	$R_{82} = 56r^8 - 105r^6 + 60r^4 - 10r^2$	$R_{10,6} = 45r^{10} - 72r^8 + 28r^6$
$R_{53} = 5r^5 - 4r^3$	$R_{84} = 28r^8 - 42r^6 + 15r^4$	$R_{10,8} = 10r^{10} - 9r^8$
$R_{55} = r^5$	$R_{86} = 8r^8 - 7r^6$	$R_{10,10} = r^{10};$

Each Z_{nl} leads to two real-valued invariants, except for $Z_{n_r l_r}$ and for the moments with repetition $l = 0$. Z_{00} and Z_{10} are not included in the set. In fact, A_{00} is already used to obtain scaling invariance, while A_{10} is always zero when central coordinates are used. For the pose estimation algorithm, ZM invariants up to the 10^{th} order (i.e., $N_f = 62$) are computed and stored in the database. The radial polynomials used to compute ZMs are reported in Table 4.2. Unlike GMs and CMs, ZMs values have a smaller dynamic range [Flusser et al., 2009], which simplifies the process of feature matching in the database. Some authors claim that the presence of factorial terms in the radial polynomials increases the computation time needed to compute ZMs, especially for higher order moments, and methods to speed up the computation of the moments are proposed in [Prata and Rusch, 1989, Hwang and Kim, 2006, Wee et al., 2004]. Prata method [Prata and Rusch, 1989] exploits the recurrence relation:

$$R_{nl}(r) = \frac{2rn}{n+l} R_{n-1,l-1}(r) - \frac{n-l}{n+l} R_{n-2,l}(r) \tag{4.31}$$

However, as discussed in Section 4.5, the optimal performance of the ZM-based pose estimation algorithm is obtained with moments up to the seventh and ninth order. For such relatively low orders, the radial polynomial coefficients can be stored and thus do not need to be computed on-line.

4.4.3 Rotation invariants with Fourier descriptors

Fourier descriptors (FDs) provide a mean for representing the boundary of a two-dimensional shape. Indeed, since a closed curve can be represented by a periodic function of a continuous parameter, it admits a Fourier transform, whose coefficients can be used as global descriptors both for shape recognition and shape retrieval [Arbter et al., 1990]. In the first works discussing the use of FDs for pattern recognition, x and y coordinates were considered separately and two Fourier transforms were computed. However Ref.[Zhang et al., 2002] explains how Fourier descriptors must be derived from a *shape signature*, which can be any 1-D function representing 2-D areas or boundaries. Different shape signatures have been used in literature to derive FDs, but the most popular starts by defining the complex central coordinate position sequence

$$z(n) = (x(n) - x_c) + i(y(n) - y_c), \quad \text{for } n = 1, \dots, N_p, \tag{4.32}$$

with N_p the number of points belonging to the contour and the coordinates of the contour's centroid given by $x_c = \frac{1}{N_p} \sum_{n=1}^{N_p} x(n)$, $y_c = \frac{1}{N_p} \sum_{n=1}^{N_p} y(n)$. The FDs are defined by computing the discrete Fourier transform (DFT) of z using the fast Fourier transform (FFT)

$$Z(\omega) = \sum_{n=1}^{N_p} z(n) \exp\left(-i \frac{2\pi(n-1)}{N_p} \omega\right), \quad \text{for } \omega = 0, \dots, N_p - 1. \tag{4.33}$$

Scaling invariance is obtained by dividing the DFT sequence by $|Z(1)|$. Translation invariance is obtained by discarding the coefficient $Z(0)$, which is indeed equal to 0 when complex central coordinates are used to define the position sequence.

The behavior of FDs under rotation is similar to that of CMs and ZMs, see (4.18). However, the Fourier transform depends also on the starting point used to describe the contour. If this starting point is shifted by m positions, the resulting transform $\tilde{Z}(\omega)$ is

$$\tilde{Z}(\omega) = Z(\omega) \exp\left(\frac{2\pi i}{N_p} \omega m\right) . \tag{4.34}$$

A solution is to use only the magnitude of the transform $|Z(\omega)|$ as descriptor, as in [Zahn and Roskies, 1972, Chen et al., 2004, Conseil et al., 2007, Barczak et al., 2011], since it is invariant to rotation and independent of the choice of initial point. However the resulting feature set will be incomplete, since every harmonic $Z(\omega)$ provides in fact two invariants, i.e., its magnitude and phase or alternatively real and imaginary parts. As with CMs and ZMs, the imaginary parts of FDs are pseudo-invariants [Zahn and Roskies, 1972]. In [Persoon and Fu, 1977], the rotation and starting point are determined by a search for the best matching through all the possible shapes, but this procedure increases the computational time of the nearest neighbor search. In this paper, we propose a method to achieve simultaneously rotation invariance and independence with respect to the choice of initial point. First, using the GMs of the contour, the in-plane rotation angle ψ_0 is computed as described in Section 4.3. By convention, the initial point of the contour is taken to be the intersection of the contour with the half line originating from the figure centroid and with direction ψ_0. In the case of multiple intersections, we select the farthest one from the centroid. The sequence $X = [x_1 + iy_1, ..., x_j + iy_j, ..., x_{N_p} + iy_{N_p}]^T$ is then rearranged starting from the selected initial point and following the contour counterclockwise. Then, the coordinates of the points are rotated by an angle $-\psi_0$, i.e., we let

$$X_r = X\, \mathrm{e}^{-i\psi_0}\,. \tag{4.35}$$

and compute the DFT of the sequence X_r. This procedure is similar to the approach used in [Kuhl and Giardina, 1982] and [Chen and Ho, 1991], where the ellipse described by the first harmonic phasor is used instead of the inertia ellipsoid. Note that none of these methods can be applied when ellipse degenerates into a circle, i.e., when the object's silhouette has an N-FRS. Note that curves presenting a N-FRS have zero amplitude harmonics for all indices that are not integral multiples of N [Zahn and Roskies, 1972]. For the pose estimation algorithm, FDs up to the 99^{th} harmonic (i.e., $N_f = 198$) are computed and stored in the database.

4.5 Application and performance analysis

In this section, the performance of the method presented in Sec.4.3 is characterized for different invariants and conditions. We aim to provide guidelines for choosing the type and number of descriptors (choice of order) among ZMs, CMs and FDs that provides the best compromise between pose estimation accuracy and computation time. The sets of invariant based on CMs, ZMs and FDs described in Sec.4.3 have never been characterized and compared on the same recognition problem, and in particular on the problem of pose estimation of a 3-D object. Some works have provided comparison between two different set of features, which, however, were often incomplete. Ref. [Reeves et al., 1988]

compares the performance in aircraft recognition and pose estimation of the seven Hu's invariants (after variance balancing, cf. Sec.4.5.1) and Fourier descriptor, showing comparable recognition properties. More comparisons have been carried out in the scope of image classification. Hu's invariants and FDs have been compared in Ref.[Chen et al., 2004] for the problem of letter recognition, with FDs demonstrating better performance for low resolution images. Also in Ref.[Conseil et al., 2007] FD magnitudes up to the 64 harmonics outperformed Hu's invariants in the context of hand posture recognition. However, Hu's set is made of only seven invariants which are known to form an incomplete set (cf. Sec.4.4.1). The invariance with respect to scaling and rotation of CM invariants up to the 4^{th} order and FD magnitudes up to the 9^{th} harmonic was analyzed in Ref.[Barczak et al., 2011] using images of the American Sign Language (ASL): CMs invariants computed on silhouettes and FDs resulted having almost the same performance, while boundary CMs invariants resulted to be worst. Images of the ASL have been used also in Ref.[Otiniano-Rodrıguez et al., 2012] to compare the recognition performance of ZMs invariants and Hu's invariants. ZMs invariants up to the 9^{th} order resulted having the best accuracy; however, Hu's invariants were computed on boundary images, which are known to provide a lower recognition power with respect to silhouette images. In Ref.[Teh and Chin, 1988] various moments, including CMs, GMs and some orthogonal moments among which ZMs, are examined and compared with respect to sensitivity to image noise, information redundancy, and capability for image representation. The study showed that higher order moments are more vulnerable to noise, and that the orthogonal moments are better than the other types of moments in terms of information redundancy, as was also demonstrated in [Abu-Mostafa and Psaltis, 1984]. In terms of overall performance, Zernike and pseudo-Zernike moments (which are derived from the ZMs by releasing from the condition imposing $n - |l|$ to be even) outperformed the others.
However, even if more than one source states that ZMs based invariants globally performs better than other sets, no works have directly compared the three set introduced in Sec.4.3. This section proposes a deep analysis of the performance of the three described sets in order to characterize their behavior under different conditions that can be encountered in an operational scenario.

4.5.1 Simulation setup

The geometry of the target S/C used in the simulations is inspired from the structure of the Iridium-NEXT satellites and shown in Fig. 4.3. The synthetic views are generated with Thales Alenia Space rendering engine SpiCam. The target has a size of 4 × 3 × 1 meters and the views are generated for a camera with a FOV of 30° and image size of 1024×1024

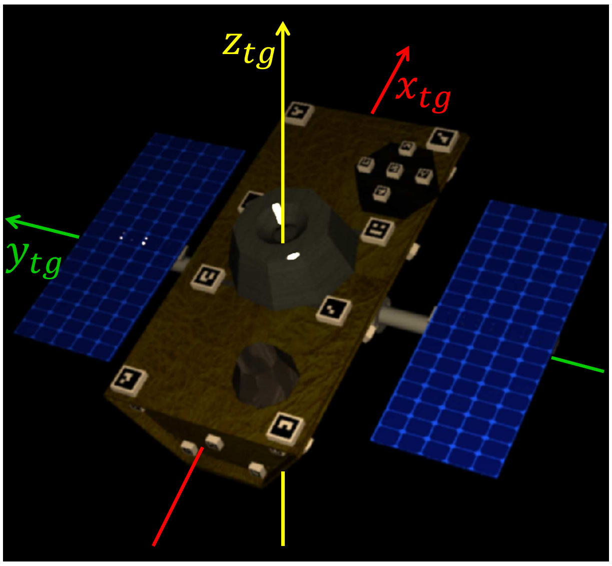
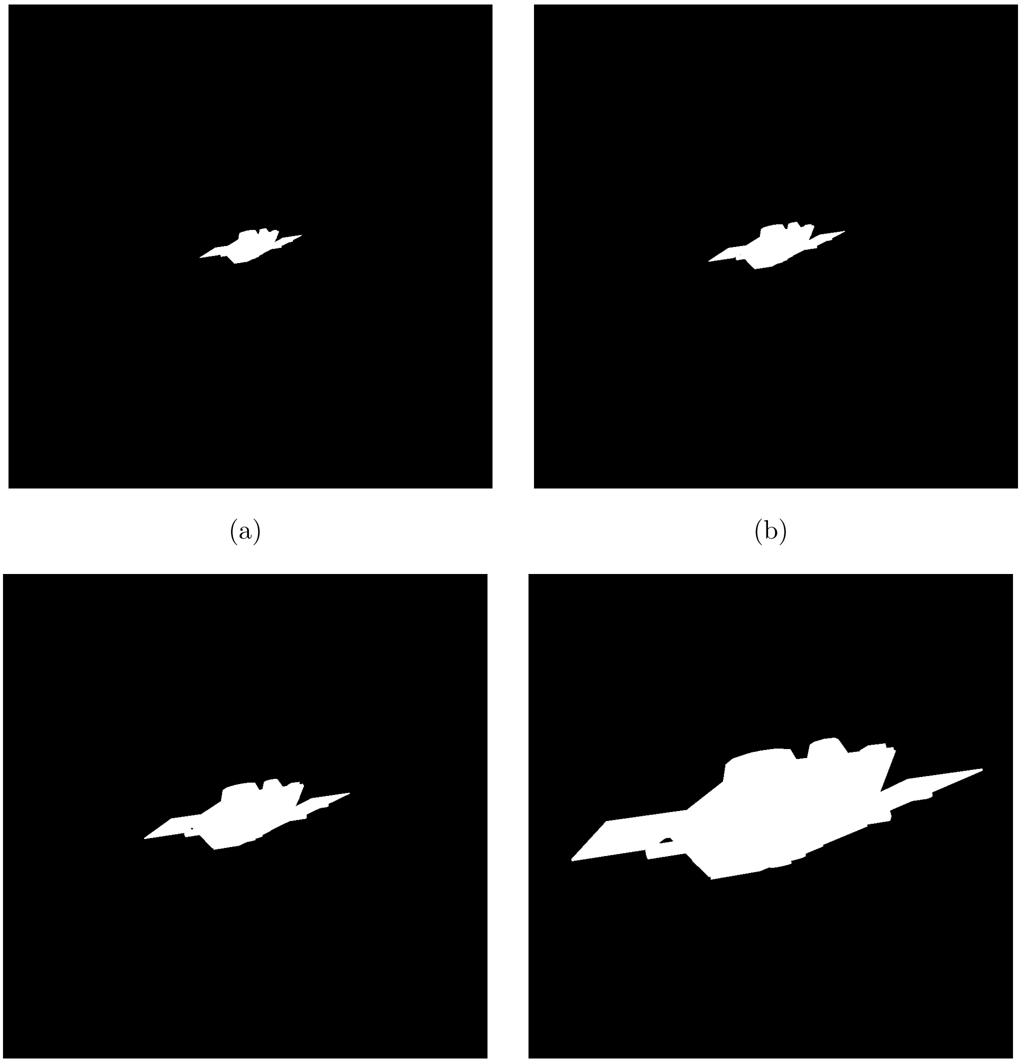


Figure 4.3: Geometry and body reference frame of the target spacecraft

pixels, i.e., on Fig. 4.1 we have $c_x = c_y = 512$ and $f = c_x/\tan(\text{FOV}/2) = 1911$. The target dimensions, translated in pixel, correspond to a projected size of 191 × 143 × 48 pixels at 40 m, 255 × 191 × 64 pixels at 30 m, 382 × 287 × 96 pixels at 20 m, and 764 × 573 × 191 pixels at 10 m (see Fig.4.4). The main structure of the S/C, composed of the central body and the lateral solar arrays, has two symmetry planes, (O_{tg}, x_{tg}, z_{tg}) and (O_{tg}, y_{tg}, z_{tg}), see Fig. 4.1. Some elements on the central body such as antennas and a docking fixture break the symmetry but are relatively small and visible only for a restricted range of attitudes. The ambiguity of determining the pose of a symmetric body was already noted in previous work on aircraft pose estimation and classification [Dudani et al., 1977, Chen and Ho, 1991, Breuers, 1999]. With the method of [Breuers, 1999], for any triplet of Euler angles $[\varphi,\ \vartheta,\ \psi]$, 8 solutions are possible. Two solutions are due purely to the presence of a symmetry plane, i.e., $[\varphi,\ \vartheta,\ \psi]$ and $[-\varphi,\ -\vartheta,\ \psi-\pi]$. Then, two more solutions, i.e., $[\pi-\varphi,\ -\vartheta,\ \psi]$ and $[\pi+\varphi,\ \vartheta,\ \psi-\pi]$, are due to the impossibility of distinguishing between mirror images using Hu's invariants. Moreover, for all of the 4 solutions described, the method in [Breuers, 1999] cannot disambiguate the in-plane rotation between $[\varphi,\ \vartheta,\ \psi]$ and $[\varphi,\ \vartheta,\ \psi-\pi]$. In contrast, by using a complete set of invariants containing also pseudo-invariants, as discussed in Section 4.4, and by estimating the in-plane rotation as proposed in Section 4.3, we need only to consider the two potential solutions $[\varphi,\ \vartheta,\ \psi]$ and $[-\varphi,\ -\vartheta,\ \psi-\pi]$ for every output of the pose estimation algorithm.

In order to characterize the intrinsic performance of the descriptor without including



(a)

(b)

(c)

(d)

Figure 4.4: 4.4(a) Silhouette of the satellite at 40 m, 4.4(b) Silhouette of the satellite at 30 m, 4.4(c) Silhouette of the satellite at 20 m, 4.4(d) Silhouette of the satellite at 10 m.

the problem of distinguish between two symmetric attitudes, the tests described in this Section are done using training images and test images corresponding to attitudes in the semi-sphere where $\varphi \geq 0$. The training database has a size of $N_w = 5000$ and the methods are tested on 2000 images. Both the training and the test attitudes are generated by modifying the sampling scheme (4.1) to consider only nonnegative values of φ, i.e., with $\varphi = \pi \, \mathrm{rand}_\varphi$. Only for the test attitudes, also ψ is randomly generated such that $\psi = 2\pi \, \mathrm{rand}_\psi - \pi$, where rand_ψ is a random variable uniformly distributed in the interval]0, 1]:

$$\begin{cases} \varphi & = \pi \, \mathrm{rand}_\varphi \\ \vartheta & = \mathrm{asin}(1 - 2\mathrm{rand}_\vartheta) \end{cases} \tag{4.36}$$

Test images are generated with perfect camera pointing (i.e., $tr^{cam}_{cam-tg} = [0,\, 0,\, d]$), except for the images used in Sec.4.5.2.5, where the algorithm is tested in the presence of camera pointing errors. Even if the object is symmetric, the number of rotation symmetries of the projected silhouette is always smaller or equal to 1. It is equal to 1 if the projected shape has one axis of reflection [Shen et al., 2000]. This implies that the invariants proposed in Section 4.4.1 and Section 4.4.2 can be used, except for attitudes corresponding to a camera position close to the sphere poles, i.e., $\vartheta \sim 0$ and $\varphi \sim 0$ or $\varphi \sim \pi$. For these attitudes, the projected S/C shape has a 2-FRS and a different set of invariants should be computed to enable recognition. Nevertheless, according to [Flusser et al., 2009], the order of the normalizing moment used to obtain rotation invariance should be kept as low as possible to improve the performance of the recognition, as higher order moments are more sensitive to noise. In order to avoid degrading the global performance of the algorithm for the sake of improving the recognition of just two isolated orientations, the rotation-normalizing moments are kept equal to c_{12} for CMs and A_{31} for ZMs. Moreover, in correspondence of the polar attitudes, the rule for the determination of the in-plane angle of rotation ψ_0 described in Section 4.3 will provide two solutions. However, this ambiguity can be resolved by imposing a continuity constraint between two consecutive pose estimates, as will be developed in Chapter 5.

4.5.2 Simulation results

We describe in this section the results of four different simulation experiments. In Paragraph 4.5.2.1, our aim is to characterize the degradation of the pose estimation performance when the S/C in the test images is at a distance d_{test} different from the distance d_{train} used to build the database. Pararagraph 4.5.2.2 compares the performance of the different global descriptors when the resolution of both test and training images changes. In Paragraph 4.5.2.3, the robustness against the resizing of the test images is investigated, while Paragraph 4.5.2.4 studies the influence of the database size N_w on the estimation error. Finally, in Paragraph 4.5.2.5, the robustness to camera pointing errors is analyzed. For any test image, we compute the estimation error in the axis-angle representation using unit quaternions. If $q_{cam-tg_{true}}$ is the true relative attitude quaternion associated with the camera-target pose in a test image, and $q_{cam-tg_{meas}}$ the relative attitude quaternion estimated by the algorithm, the error $\delta_{\varphi\vartheta\psi}$ is computed as:

$$\begin{aligned} \delta q &= q^*_{cam-tg_{true}} \otimes q_{cam-tg_{meas}} \\ \delta_{\varphi\vartheta\psi} &= 2\,\text{atan}\left(\left|\sqrt{\delta q_1^2 + \delta q_2^2 + \delta q_3^2}/\delta q_0\right|\right) \end{aligned} \tag{4.37}$$

where quaternions are written $q = q_0 + q_1 i + q_2 j + q_3 k$, with q_0 denoting the scalar part, and q^* denotes the quaternion conjugate $q^* = q_0 - (q_1 i + q_2 j + q_3 k)$. The quantity $\delta_{\varphi\vartheta\psi}$ represents the smallest rotation that aligns the measured quaternion with the true one, and its value is always in the interval $[0, 180°]$. We assume that a test image has found an acceptable match if $\delta_{\varphi\vartheta\psi} < 20°$. Indeed, when using a frame-to-frame tracking algorithm, a pose estimation error in this range can typically be corrected, while outside of this range the matching result can be rejected as outlier, as done in Chap.5. Finally, we record as performance indices: i) the *accuracy* (also denoted $\% < 20°$), defined here as the percentage of detections with an estimation error lower than 20°; ii) the *mean* of the error $\delta_{\varphi\vartheta\psi}$ over all the test samples such that $\delta_{\varphi\vartheta\psi} < 20°$; and iii the *mean* of the measured distance d_{meas} computed using all the test samples.

A baseline test is performed using a database built with $d_{train} = 20$ m and test images taken at the same distance. Table 4.3 shows the performance of the CM, ZM, and FD invariants as a function of the moment or harmonic order. The best accuracy for CMs is 90.45%, obtained with invariants of the 5^{th} order, with a mean angular error of 2.67°. The best accuracy of ZM-based invariants is 96.95%, obtained with moments up to the 9^{th} order, with a mean angular error of 1.74°. However, little performance improvement is observed beyond the 7^{th} order. The optimal performance of FDs is obtained using coefficients up to the 10^{th} harmonic, with an accuracy of 85.05% and a mean angular error of 3.03°, and no improvements is observed for higher order harmonics. In general, even when the accuracy of CM, ZM, and FD invariants is comparable (e.g., 4^{th} order CMs, 3^{rd} order ZMs and 10^{th} harmonic FDs), the mean angular error using ZMs is lower. The descriptors show a comparable performance in the estimation of the distance d_{meas}.

4.5.2.1 Effect of a variation of the test distance with a constant training distance

The descriptors based on ZMs, CMs, and FDs are theoretically invariant on a continuous image, but this invariance degrades for a digital image due to pixel discretization [Huang and Leng, 2010, Barczak et al., 2011]. Thus, if the descriptors for a test image with the target at distance d_{test} are compared to a database of descriptors computed with the target at a different distance d_{train}, the quality of the matching may be reduced in practice. Moreover, below a certain value of d, it is no more possible to assume that the distance affects only the scale: its contribution to the shape of the projected silhouette may entail an additional degradation of the performance.

To investigate this issue and understand if one can use a database with a single distance to estimate the S/C pose at different distances or if it is necessary to store views covering

Table 4.3: Performance for $d_{train} = 20$ m, $d_{test} = 20$ m.

Complex moment invariants

order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$\% < 20$ deg	84.15	86.65	**90.45**	89.55	88.40	88.20	86.95	88.50
$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	2.95	2.99	**2.67**	2.75	2.83	2.92	2.92	3.15
d_{meas} *mean* [m]	20.14	20.13	20.15	20.13	20.09	20.08	20.05	**20.02**

Zernike moment invariants

order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$\% < 20$ deg	85.35	91.85	95.25	95.45	96.40	96.25	**96.95**	96.80
$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	2.12	1.86	1.80	1.74	1.75	**1.73**	1.74	1.76
d_{meas} *mean* [m]	20.07	20.01	20.01	20.01	20.02	**20.01**	20.01	20.01

Fourier descriptors

harmonic:	3^{rd}	4^{th}	5^{th}	7^{th}	9^{th}	10^{th}	20^{th}	99^{th}
$\% < 20$ deg	53.65	74.65	81.50	84.30	84.95	**85.05**	85.05	85.00
$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	4.65	3.34	3.23	3.01	**3.01**	3.03	3.05	3.05
d_{meas} *mean* [m]	20.13	20.21	20.12	20.04	20.03	**20.01**	19.98	19.98

a wide range of relative distances, we use the same training database built with $d_{train} = 20$ m and the same test attitudes, but perform tests with different values of d_{test}. Results are shown in Table 4.4 for $d_{test} = 30$ m, $d_{test} = 40$ m and $d_{test} = 10$ m. At 30 m, the maximal accuracy is obtained for CMs up to the 6^{th} order (85.80%), for ZMs up to the 9^{th} order (94.85%), and for FDs up to the 20^{th} harmonic (i.e., 72.85%). Zernike invariants are the most stable, with an accuracy loss of only 2.1% compared to the baseline test. The mean angular error is increased for all the methods. Trends are confirmed when the test distance increases. At 40 m, FDs are the most affected (best accuracy equal to 61.25% for the 10^{th} harmonic), followed by CMs (best accuracy equal to 78.40% with moments up to the 6^{th} order). Zernike invariants confirm their higher stability, with the best performance obtained for the 9^{th} order. The accuracy of 91.35% for ZMs is still higher than the best performance of CM and FD in the baseline test, and the mean angular error of 3.02° is comparable. On the other hand, if the test distance is *decreased* with respect to the training distance, as illustrated in Table 4.4 for $d_{test} = 10$ m, the moment-based descriptors (CMs and ZMs) show a substantial performance degradation. The best accuracy for CM invariants (67.45%) is obtained with moments up to the 6^{th}

Table 4.4: Effect of a variation of the test distance with a constant training distance ($d_{train} = 20$ m)

Complex moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$d_{test} = 30$ m	% < 20 deg	73.95	77.55	83.95	**85.80**	83.00	83.15	81.40	82.45
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	5.19	4.94	5.03	**4.90**	5.26	5.21	5.88	5.64
	d_{meas} *mean* [m]	30.45	30.39	30.53	30.36	30.21	30.25	**30.01**	30.02
$d_{test} = 40$ m	% < 20 deg	66.55	71.35	75.65	**78.40**	71.40	73.65	71.00	74.05
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	6.29	6.25	6.46	**6.23**	6.80	6.72	7.35	7.19
	d_{meas} *mean* [m]	40.56	40.59	40.71	40.67	**40.15**	40.24	39.52	39.78
$d_{test} = 10$ m	% < 20 deg	57.20	60.65	64.85	**67.45**	63.30	64.65	64.60	64.60
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	7.93	**7.84**	8.69	8.40	8.92	8.72	8.86	8.82
	d_{meas} *mean* [m]	9.79	9.70	9.75	**9.79**	9.71	9.70	9.72	9.71

Zernike moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$d_{test} = 30$ m	% < 20 deg	78.85	86.90	93.00	92.50	93.95	93.20	**94.85**	94.40
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	3.06	2.57	2.55	2.42	2.42	**2.37**	2.41	2.45
	d_{meas} *mean* [m]	30.19	30.11	30.11	**30.11**	30.12	30.12	30.12	30.12
$d_{test} = 40$ m	% < 20 deg	71.40	80.20	89.35	89.65	90.95	89.40	**91.35**	91.25
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	3.90	3.19	3.29	3.02	3.03	**2.96**	3.02	3.11
	d_{meas} *mean* [m]	40.28	40.22	**40.21**	40.22	40.22	40.22	40.23	40.22
$d_{test} = 10$ m	% < 20 deg	62.20	72.90	80.80	84.80	87.00	86.65	**87.70**	87.35
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	5.04	4.64	4.62	**4.11**	4.12	4.15	4.20	4.28
	d_{meas} *mean* [m]	9.87	**9.89**	9.89	9.88	9.87	9.88	9.87	9.87

Fourier descriptors

	harmonic:	3^{rd}	4^{th}	5^{th}	7^{th}	9^{th}	10^{th}	20^{th}	99^{th}
$d_{test} = 30$ m	% < 20 deg	43.80	61.80	69.50	72.10	72.7	72.65	**72.85**	72.75
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	5.69	4.29	4.2	3.70	3.76	**3.75**	3.82	3.82
	d_{meas} *mean* [m]	30.53	30.48	30.32	30.10	30.05	**29.99**	29.92	29.90
$d_{test} = 40$ m	% < 20 deg	34.45	48.80	57.80	60.40	61.10	**61.25**	61.00	60.95
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	6.58	5.10	5.00	**4.42**	4.48	4.50	4.58	4.63
	d_{meas} *mean* [m]	41.08	40.76	40.53	40.16	40.07	**40.01**	39.91	39.90
$d_{test} = 10$ m	% < 20 deg	45.65	80.80	84.60	90.70	**92.25**	91.25	90.25	90.00
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	6.95	5.31	5.14	**4.13**	4.15	4.14	4.22	4.23
	d_{meas} *mean* [m]	9.75	9.77	9.66	9.85	**9.86**	9.84	9.83	9.83

order, with a mean angular error of 8.40°. The degradation of ZM invariant is smaller (accuracy = 87.70%, mean angular error = 4.20° for the 9^{th} order), but higher than the degradation incurred at 40 m. This relatively large performance loss can be attributed to the fact that at 10 m the distance starts having a non-negligible contribution to the shape of the projected silhouette. On the other hand, FD shows an opposite trend, with a accuracy of 92.25% for the 9^{th} harmonic, a value which is even higher than the one obtained in the baseline test. The trends are summarized in Fig. 4.5, which shows as a function of d_{test} the evolution of best performance obtained for each set of descriptors (attained for 6^{th} order CMs, 9^{th} order ZMs, and 10^{th} harmonic for FDs). The measured distance d_{meas} tends to be underestimated when $d_{test} < d_{train}$, and overestimated when $d_{train} < d_{test}$, as a consequence of the effect of d on the shape of the projected silhouette.

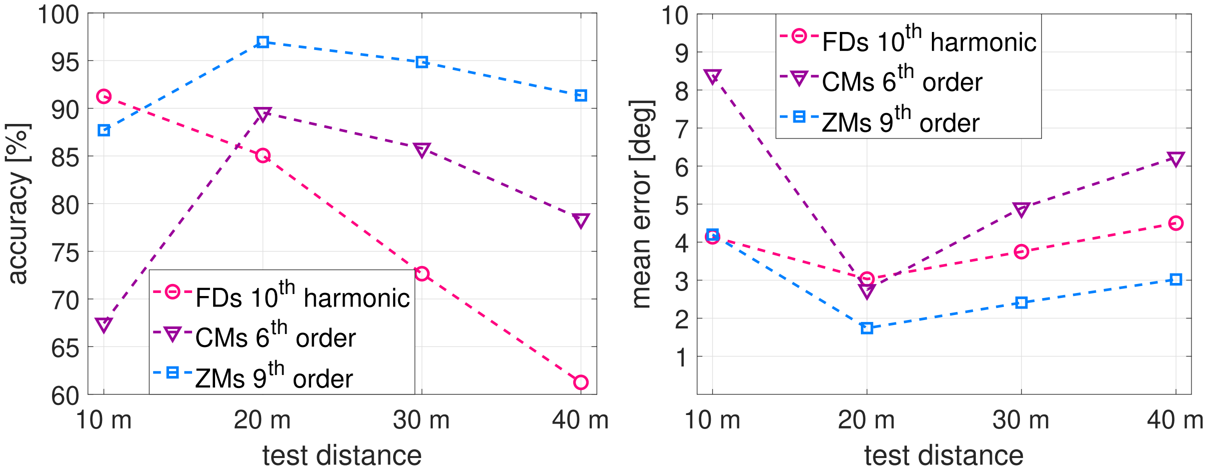


Figure 4.5: Effect of a variation of the test distance with a constant training distance $d_{train} = 20$ m.

4.5.2.2 Effect of a variation in the resolution of both the test and the training images

In order to better isolate in the previous tests the true effect of a mismatch between d_{test} and d_{train} from a potential degradation in performance simply due to a lower silhouette resolution as the distance increases, we performed additional simulation experiments assuming $d_{test} = d_{train}$, with this distance equal to 10 m and 40 m, complementing the baseline test at 20 m. The numerical results are shown in Table 4.5 and Fig. 4.6 displays the performance indices of the 6^{th} order CMs, the 9^{th} order ZMs, and the 10^{th} harmonic FDs as a function of the test distance. The accuracy of CMs and ZMs is only slightly affected by the distance. For ZM invariants up to the 9^{th} order, the accuracy and mean angular error are only slightly affected by the distance, going from 97.15% and 1.75° at 10 m to 95.25% and 1.98° at 40 m respectively. On the other hand, the mean angular error of both CMs and FDs degrades more clearly for a lower image resolution. The mean angular error of CM invariants up to the 6^{th} order is equal to 2.57° at 10 m and 3.41° at 40 m, and the mean angular error of FD invariant up to the 10^{th} harmonic is equal to 2.86° at 10 m and 3.87° at 40 m. An interesting result is the trend in the accuracy of FDs, which monotonically increases as the image resolution increases, with a maximum of 96.65% (10^{th} harmonic) at a distance of 10 m. This performance is comparable to the accuracy of ZM invariants, even though the error obtained with ZMs remains always lower.

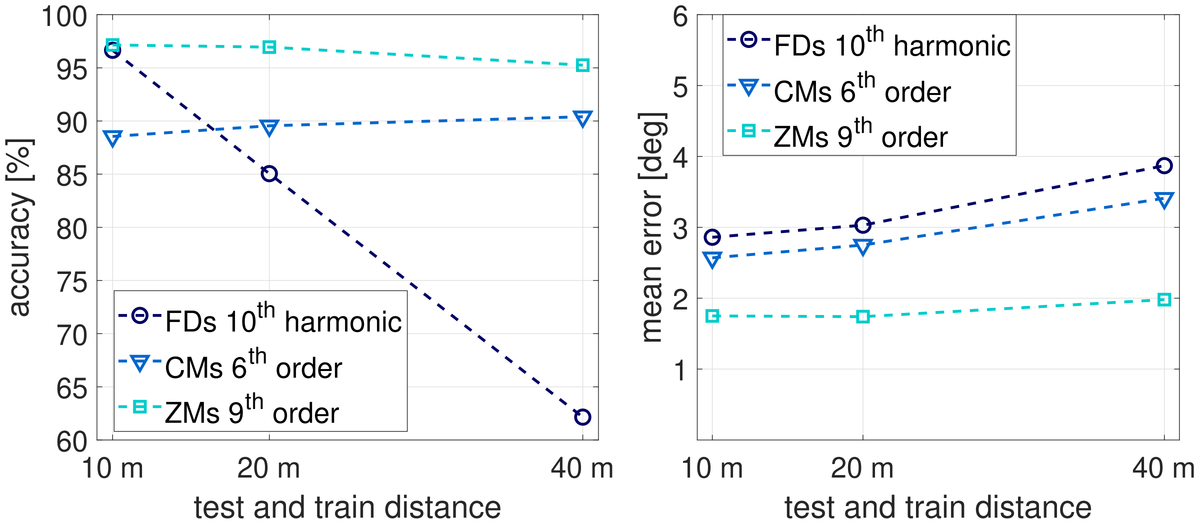


Figure 4.6: Effect of a variation in the resolution of both the test and the training images ($d_{test} = d_{train}$).

4.5.2.3 Effect of a resizing of the test images

The largest contributor to the algorithm latency is the time needed to compute moment invariants (see Sec.4.5.4). The execution time increases linearly with the number of pixels to be processed, and this may suggest to resize the acquired image before computing the descriptors. In this paragraph we test the performance of the invariants at $d_{train} = d_{test} =$ 20 m, where the test images have been resized from 1024 × 1024 pixels to 256 × 256 and 512 × 512 pixels using OpenCV resize() function [OpenCV development team, 2019a]. When computing descriptors from the resized images, (4.12) needs to be corrected to allow computing the correct value of d_{meas}:

$$d_{meas} = \sqrt{m_{00_{train}}/m_{00_{test}}} \cdot d_{train} \cdot \sqrt{n_{pixel_R}/n_{pixel_0}}, \tag{4.38}$$

Table 4.5: Effect of a variation of the resolution of both the test and the training images ($d_{train} = d_{test}$)

Complex moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$d_{test} = 40$ m	$\% < 20$ deg	82.00	84.65	**90.75**	90.40	89.30	89.45	88.15	89.15
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	4.07	3.77	3.54	**3.41**	3.54	3.72	3.83	3.92
	d_{meas} *mean* [m]	40.29	40.19	40.19	40.20	40.09	**39.99**	40.02	40.10
$d_{test} = 10$ m	$\% < 20$ deg	82.90	85.05	**89.45**	88.55	87.55	86.60	85.65	85.70
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	2.7	2.69	**2.56**	2.57	2.78	2.91	3.01	3.17
	d_{meas} *mean* [m]	10.08	10.04	10.03	10.04	10.02	10.01	9.99	**10.00**

Zernike moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$d_{test} = 40$ m	$\% < 20$ deg	84.35	86.70	93.40	92.95	94.90	94.50	**95.25**	94.75
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	2.41	2.03	2.04	1.97	1.98	1.98	1.98	**1.96**
	d_{meas} *mean* [m]	40.07	40.03	40.04	40.02	40.03	**40.02**	40.03	40.02
$d_{test} = 10$ m	$\% < 20$ deg	85.50	93.05	95.90	96.20	96.80	97.05	97.15	**97.20**
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	1.96	1.87	1.80	1.76	**1.73**	1.74	1.75	1.77
	d_{meas} *mean* [m]	10.03	10.01	10.01	10.00	**10.00**	10.01	10.00	10.01

Fourier descriptors

	harmonic:	3^{rd}	4^{th}	5^{th}	7^{th}	9^{th}	10^{th}	20^{th}	99^{th}
$d_{test} = 40$ m	$\% < 20$ deg	38.45	51.35	59.25	61.85	**62.90**	62.15	62.35	62.25
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	5.65	4.35	4.07	**3.80**	3.93	3.87	3.90	3.88
	d_{meas} *mean* [m]	41.01	40.61	**40.29**	39.69	39.69	39.64	39.60	39.59
$d_{test} = 10$ m	$\% < 20$ deg	62.55	89.20	92.10	96.30	96.50	**96.65**	96.05	95.85
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	4.53	3.30	3.14	**2.79**	2.85	2.86	2.85	2.85
	d_{meas} *mean* [m]	10.05	9.93	9.93	10.00	10.00	**10.00**	10.00	10.00

where n_{pixel_R} is the total number of pixel in the resized image and n_{pixel_0} is the total number of pixel in the original image. The results are displayed in Table 4.6.

Moment invariants show an accuracy comparable to the baseline test, and a slightly increased angular error. ZM invariants show a higher degradation in the estimation of the distance with respect to CMs. However, this issue can be overcome using the original image to compute m_{00}, and the resized image to compute rotation invariants. The performance of the FD is very poor. However, as discussed in Sec.4.5.4, the very low time needed to compute FDs suggests that image resizing is unnecessary when working with Fourier invariants.

Table 4.6: Effect of a resizing of the test image. ($d_{train} = d_{test} = 20$ m)

Complex moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
256×256	% < 20 deg	78.30	82.75	87.40	**87.55**	86.25	86.25	85.40	86
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	3.85	3.69	**3.49**	3.52	3.58	3.63	3.74	3.9
	d_{meas} *mean* [m]	20.02	19.99	**19.99**	19.98	19.96	19.92	19.91	19.86
512×512	% < 20 deg	81.9	84.70	**89.90**	88.90	87.00	87.10	86.30	87.30
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	3.53	3.43	3.16	**3.15**	3.30	3.41	3.52	3.64
	d_{meas} *mean* [m]	19.98	19.97	19.99	**20.00**	19.96	19.93	19.84	19.86

Zernike moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
256×256	% < 20 deg	78.55	89.65	94.20	94.30	95.25	95.55	**95.95**	95.90
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	3.52	2.84	2.70	2.49	2.46	2.42	**2.38**	2.41
	d_{meas} *mean* [m]	**20.02**	19.92	19.91	19.91	19.90	19.90	19.90	19.89
512×512	% < 20 deg	83.05	91.40	94.55	94.60	95.60	95.55	**96.55**	96.35
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	2.66	2.20	2.11	1.98	1.95	**1.95**	1.96	1.98
	d_{meas} *mean* [m]	**19.94**	19.89	19.88	19.87	19.87	19.87	19.87	19.87

Fourier descriptors

	harmonic:	3^{rd}	4^{th}	5^{th}	7^{th}	9^{th}	10^{th}	20^{th}	99^{th}
256×256	% < 20 deg	25.50	40.35	45.45	48.40	48.75	48.65	48.75	**48.80**
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	7.44	5.98	5.74	**5.30**	5.39	5.35	5.37	5.40
	d_{meas} *mean* [m]	20.39	20.12	**20.10**	19.55	19.52	19.47	19.40	19.41
512×512	% < 20 deg	37.4	55.15	60.70	63.35	**63.80**	63.65	63.60	63.70
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	5.88	4.46	4.26	**4.01**	4.03	4.05	4.15	4.16
	d_{meas} *mean* [m]	20.24	20.12	**20.04**	19.71	19.68	19.64	19.61	19.60

4.5.2.4 Effect of the database size N_w

Increasing N_w typically leads to a smaller distance between a given target orientation and its nearest neighbor in the database, and hence to a smaller estimation error on average. However, a larger database requires more memory as well as more computation time for matching, although as we discuss below the latter is typically much smaller than the time needed to compute the invariants. The effect of reducing the database size N_w is evaluated for $d_{train} = d_{test} = 20$ m. Table 4.7 shows the results for $N_w = 1000$ and $N_w = 3000$, in addition to the baseline scenario with $N_w = 5000$.

The estimated average distance is not displayed in the table as no appreciable differences with respect to the baseline test were found. Fig. 4.7 compares the performance indices as a function of the ZM-based invariants' order for the different values of N_w.

Table 4.7: Effect of database size N_w. with $d_{train} = d_{test} = 20$ m.

Complex moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$N_w = 1000$ m	% < 20 deg	71.80	74.10	**77.95**	77.30	76.70	75.10	74.70	75.95
	mean [deg]	**4.82**	4.94	4.92	5.05	5.27	5.26	5.17	5.22
$N_w = 3000$ m	% < 20 deg	80.30	82.95	**86.85**	86.80	85.65	85.75	84.10	84.75
	mean [deg]	3.35	3.44	**3.23**	3.36	3.51	3.54	3.50	3.68

Zernike moment invariants

	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$N_w = 1000$ m	% < 20 deg	75.35	79.95	86.15	87.55	89.60	89.50	**90.85**	90.25
	mean [deg]	3.70	3.49	3.53	3.40	3.42	**3.34**	3.35	3.35
$N_w = 3000$ m	% < 20 deg	82.65	89.25	93.75	94.40	95.25	95.15	**95.85**	95.45
	mean [deg]	2.44	2.23	2.21	2.15	2.13	2.14	**2.13**	2.14

Fourier descriptors

	harmonic:	3^{rd}	4^{th}	5^{th}	7^{th}	9^{th}	10^{th}	20^{th}	99^{th}
$N_w = 1000$ m	% < 20 deg	49.50	68.70	75.30	77.90	**79.20**	78.95	79.10	79.05
	mean [deg]	5.36	4.56	4.46	**4.29**	4.37	4.41	4.48	4.49
$N_w = 3000$ m	% < 20 deg	53.15	73.20	80.55	82.75	83.15	83.35	**83.40**	83.20
	mean [deg]	4.86	3.68	3.56	**3.33**	3.34	3.38	3.42	3.42

As expected, both the accuracy and mean angular error improve with larger values of N_w. However, the marginal improvement also decreases as N_w becomes large, so that above a certain threshold, the performance gains may not be worth the higher storage and computational costs.

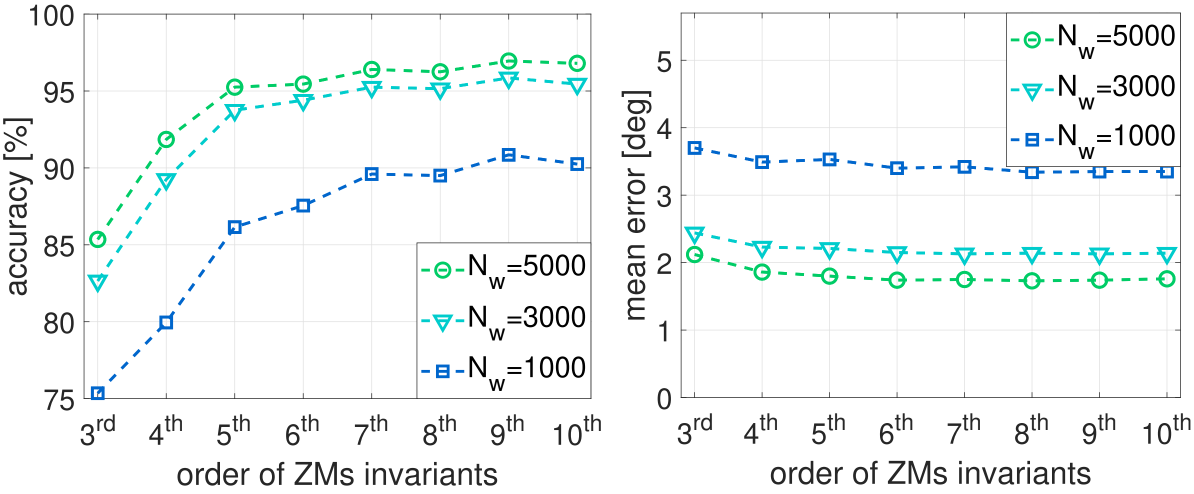
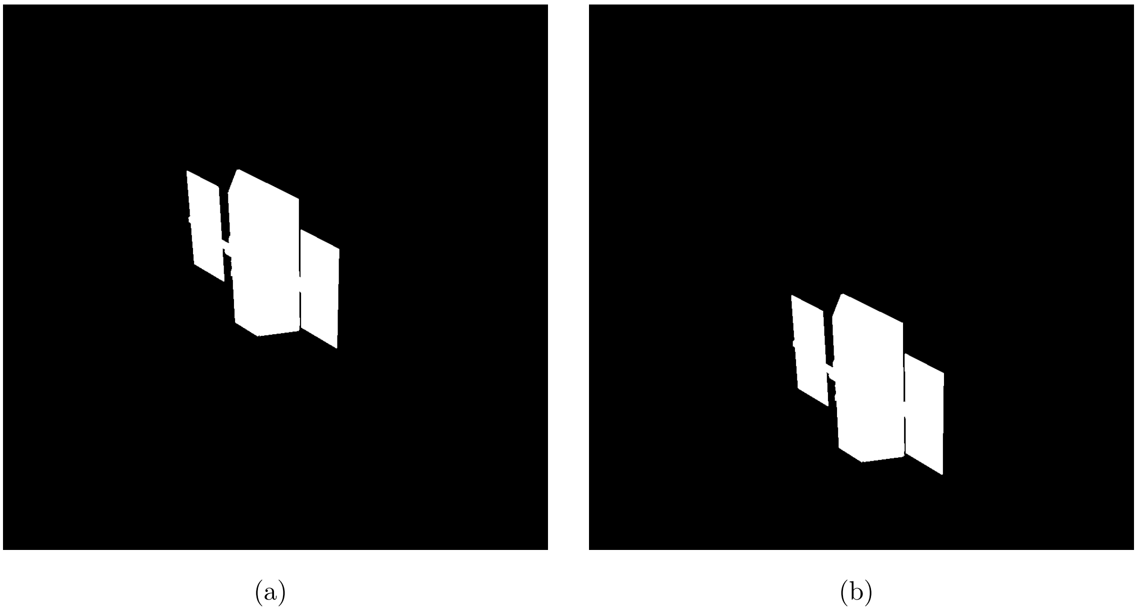


Figure 4.7: Performance of the ZM invariants for different database sizes N_w.

4.5.2.5 Effect of a pointing error


(a)

(b)

Figure 4.8: Silhouette of the target satellite at 20 m, at a relative attitude of $\vartheta = -35.02°$, $\psi = 87.99°$. In Fig.4.8(a) the pointing error is $\Delta\gamma = 1.43°$, in Fig.4.8(b) it is $\Delta\gamma = 5.85°$.

Table 4.8: Effect of a pointing error. $d_{train} = d_{test} = 20$ m

		Complex moment invariants							
	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$\Delta\gamma \in [1.41°, 2°]$	% < 20 deg	83.00	86.35	**90.40**	89.80	88.85	88.50	87.65	88.50
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	3.57	3.55	3.36	**3.43**	3.53	3.62	3.65	3.80
	Δtr_x *mean* [cm]	4.75	4.67	**4.63**	4.65	4.71	4.65	4.74	4.74
	Δtr_y *mean* [cm]	4.89	4.83	4.80	4.78	**4.69**	4.75	4.88	4.81
	Δtr_z *mean* [m]	0.605	0.537	**0.499**	0.521	0.539	0.563	0.638	0.624
$\Delta\gamma \in [4.25°, 6°]$	% < 20 deg	80.30	84.70	**88.05**	87.65	85.80	85.95	85.70	86.25
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	6.08	6.01	5.84	**5.83**	5.89	5.93	6.17	6.26
	Δtr_x *mean* [cm]	6.70	6.20	6.32	**6.05**	6.36	6.20	6.64	6.62
	Δtr_y *mean* [cm]	6.82	6.18	6.11	**5.96**	6.42	6.54	6.59	6.53
	Δtr_z *mean* [m]	0.701	0.577	0.587	**0.553**	0.654	0.641	0.691	0.690

		Zernike moment invariants							
	order:	3^{rd}	4^{th}	5^{th}	6^{th}	7^{th}	8^{th}	9^{th}	10^{th}
$\Delta\gamma \in [1.41°, 2°]$	% < 20 deg	85.40	91.85	95.70	95.70	96.65	96.85	**97.35**	97.15
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	2.77	2.57	2.54	2.47	**2.45**	2.47	2.47	2.47
	Δtr_x *mean* [cm]	4.49	4.39	4.39	4.38	**4.37**	4.37	4.38	4.37
	Δtr_y *mean* [cm]	4.53	4.46	4.46	4.46	**4.45**	4.46	4.46	4.45
	Δtr_z *mean* [m]	0.292	0.162	0.156	0.156	**0.156**	0.157	0.157	0.157
$\Delta\gamma \in [4.25°, 6°]$	% < 20 deg	84.25	90.70	94.90	94.60	95.90	96.05	**97.10**	96.65
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	5.47	5.29	5.27	5.24	**5.20**	5.21	5.21	5.22
	Δtr_x *mean* [cm]	5.07	4.54	4.56	4.53	4.52	**4.50**	4.50	4.50
	Δtr_y *mean* [cm]	5.16	4.71	4.68	4.66	4.65	**4.63**	4.63	4.63
	Δtr_z *mean* [m]	0.318	0.176	0.176	0.169	0.168	**0.166**	0.167	0.168

		Fourier descriptors							
	harmonic:	3^{rd}	4^{th}	5^{th}	7^{th}	9^{th}	10^{th}	20^{th}	99^{th}
$\Delta\gamma \in [1.41°, 2°]$	% < 20 deg	53.45	74.15	80.65	84.15	**84.80**	84.60	84.55	84.50
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	4.99	3.88	3.62	3.48	**3.48**	3.49	3.51	3.51
	Δtr_x *mean* [cm]	6.84	5.67	5.53	4.72	**4.70**	4.71	4.72	4.72
	Δtr_y *mean* [cm]	6.56	5.63	5.45	**4.86**	4.88	4.89	4.89	4.89
	Δtr_z *mean* [m]	2.211	1.235	1.088	**0.586**	0.593	0.599	0.610	0.613
$\Delta\gamma \in [4.25°, 6°]$	% < 20 deg	53.05	75.00	81.00	84.10	84.45	**84.60**	84.45	84.45
	$\delta_{\varphi\vartheta\psi}$ *mean* [deg]	6.81	6.05	5.78	**5.74**	5.75	5.77	5.79	5.77
	Δtr_x *mean* [cm]	13.26	8.66	8.04	5.82	**5.81**	5.84	5.93	5.91
	Δtr_y *mean* [cm]	14.08	9.10	8.69	**6.27**	6.28	6.31	6.29	6.28
	Δtr_z *mean* [m]	2.189	1.166	1.038	**0.562**	0.566	0.571	0.584	0.585

While the database is constructed by centering the image at the target's COM, during the pose acquisition phase in a real RDV the camera will likely point at the silhouette's centroid, causing a small pointing error. In some scenario, it may be even impossible to keep the target in the center of the FOV, causing a larger pointing error. The presence of this error affects not only the position of the image's centroid (which does not influence the invariants), but also the projected shape of the S/C, because of the camera's perspective projection. It is therefore important to understand whether the presence of such pointing errors affects the algorithm. To do this, we added angular offsets to the camera RF around the x_c and y_c axis and tested the algorithm at a distance of 20 m, using the database built with $d_{train} = 20$ m. Two sets of 2000 images were generated. In the first one, each image is generated adding a total pointing error $\Delta\gamma$ such that $\Delta\gamma \in [1.42°, 2°]$ (see Fig.4.8(a)). In the second set, the error is $\Delta\gamma \in [4.25°, 6°]$ (see Fig.4.8(b)). For these tests we collect also the performance on the estimation of the relative position vector $tr^{cam}_{cam-tg_{meas}}$. In particular, the *mean* of the absolute errors $\Delta tr_{x,y,z} = |tr^{cam}_{cam-tg_{meas x,y,z}} - tr^{cam}_{cam-tg_{true x,y,z}}|$, is shown. The results are displayed in Table 4.8 and show that all the methods are relatively robust to pointing errors. The accuracy of the invariants remains almost the same as for the baseline test, while the mean angular errors increase slightly. It is interesting to see how the attitude error grows almost linearly with the pointing error $\Delta\gamma$. ZM based invariants show lower angular errors and position errors than the other sets of invariants.

4.5.3 Discussion of the results

From the tests described in this section, the following important conclusions can be drawn.

- If the target is close, the shape of its projected silhouette can change significantly with the distance, resulting in a significant decrease in performance for moment-based invariants. This phenomenon depends on the camera sensor properties and on the target geometry. In our experiments, it starts occurring for a distance of about 10 m, i.e., about two-and-a-half times the maximum dimension of the target. FDs, on the other hand, suffer less from this issue.

- The accuracy of moment-based invariants is only mildly affected by the image resolution, see Fig. 4.6, and by the image resizing, see Par. 4.5.2.3. Thus, images could be sub-sampled before computing moments in order to decrease the computation time of the algorithm. This results in a small performance loss, which might be acceptable if the algorithm is only used to detect the divergence of a classical iterative tracking algorithm.

- ZM invariants up to the 4^{th} order (i.e., 13 features) always perform better than CM and FD invariants of any order or harmonic. The accuracy for FDs is comparable to that of ZMs only for high resolution images.

- The accuracy of FDs is highly affected by the test distance, and the accuracy always increases as the test distance decreases, regardless of the training distance. This can be explained as follows. The majority of the spectral content of the projected S/C silhouette, which is a relatively simple shape, is contained in the first 10 harmonics. For a low resolution image, the rasterization effect shifts some of the spectral content of the S/C shape to be to higher frequencies, resulting in loss of information. On the other hand, for high resolution images, the spectral content is correctly distributed in the first harmonics and the matching accuracy is enhanced. This feature is also very interesting because for short distances the computation time for the silhouette's moments increases due to the large number of pixels to be processed. The computation of FDs on the other hand needs less time, since only contour points have to be processed. Thus, it could be useful to switch from moment-based descriptors to FDs as the test distance becomes sufficiently small.
- The distance is always correctly recovered using (4.12), even when $\delta_{\varphi\vartheta\psi}$ is higher than 20°. Hence, even in the case of an incorrect matching, the best match is an image having a "mass" distribution similar to the one of the S/C silhouette. This provides a method to estimate the camera-target distance d using monocular vision, even when it is not possible to estimate the attitude correctly.

It should be noted that all the tests have been carried out using perfect binary images, and that a degradation should be expected when using real images of the target spacecraft. When using real images, the extracted binary silhouettes are very likely to be affected by segmentation errors. However, the design or choice of robust background subtraction and segmentation algorithms are out of the scope of this study.

4.5.4 Computation time and Memory requirements

Fig.4.9 shows the evolution of the computation time needed to compute the descriptors, as a function of the invariants' order, averaged over the images of the test set at 20 m. The times are normalized with respect to the time needed to compute ZM invariants up to the 10^{th} order. Using MATLAB on a 2.70 GHz Intel Core i7 processor, this time is equal to 649.95 milliseconds. The absolute times are displayed in Table 4.9. Note however that these absolute computation times are mostly indicative, since the implementation could be optimized and would be done in a statically compiled programming language or directly in hardware on a real system, which can lead to significant improvements. For ZMs and CMs,

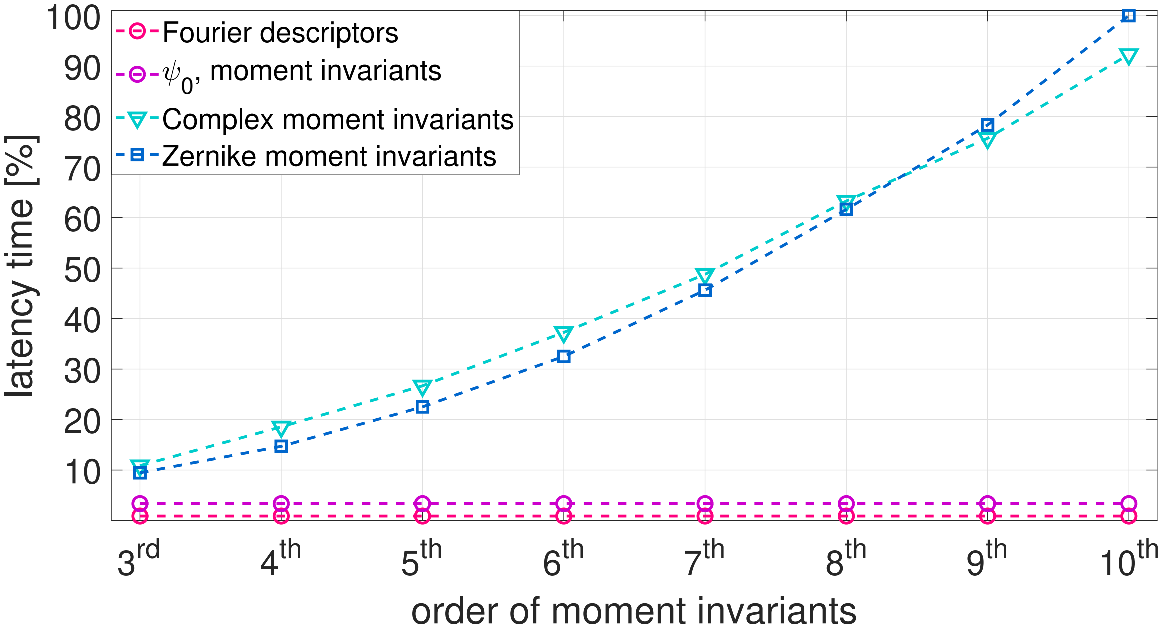


Figure 4.9: Computation time normalized with respect to the 10^{th} order ZM invariant computation time

Table 4.9: Absolute computation time of invariants averaged on 2000 images at $d = 20$ m

	3^{rd} order	4^{th} order	5^{th} order	6^{th} order	7^{th} order	8^{th} order	9^{th} order	10^{th} order
ZMs [ms]	61.5	95.52	146.32	211.26	296.5	400.51	509.16	649.95
CMs [ms]	70.67	120.64	173.39	242.18	316.93	411.29	492.03	599.92

FDs [ms]	5.80

ψ_0 [ms]	21.71

the total computation time of the algorithm is the sum of the time needed to compute the n^{th} order invariants and the time needed to compute the in-plane rotation angle ψ_0. For FDs, the computation time includes the time needed to extract the edges from the silhouette's image (using the OpenCV function findContours() [OpenCV development team, 2019b]), as well as the time to compute the in-plane rotation. In fact, for FDs, ψ_0 must be calculated before computing the Fourier transform. The FFT computes all the harmonics up to the size of the input sequence simultaneously, hence the computation time does not depend on the invariants' order. We see that the computation time for CMs and ZMs is comparable, while FDs are much faster to compute than moment-based descriptors. Note that the computation of the 7^{th} order ZM invariants requires only half of the time needed to compute the 10^{th} order ones, and that in every test, these invariants have shown an accuracy very close to the optimal one (which was always obtained for the 9^{th} or 10^{th} order) as well as similar mean angular errors. Hence, 7^{th} order ZM invariant offer a good compromise between computational cost and estimation performance. CM invariants offer no particular advantage, as they are always outperformed by ZM invariants and take essentially the same amount of time to compute. Finally, the size of the database affects only moderately the overall computation time of the algorithm, since the time necessary for the nearest neighbor search is largely dominated by the descriptor computation time. For example, with a database of size $N_w = 5000$, matching requires less than 0.1% of the time needed to compute the 10^{th} order ZM invariants. Thus, the choice of the value of N_w should be driven only by the memory available to store the database and by the precision required for the algorithm.,Note again that since the size of the database grows exponentially with the number of parameters to discretize, this analysis relies crucially on the fact that the distance and yaw angles are not included in the matching process, thanks to the invariance properties of the descriptors. The memory needed for the descriptor database storage can be computed by multiplying the value of 8 bytes (i.e., the dimension of a *double* type variable) by the total number of *doubles* in the database, which is given by $N_w \times (N_f + 4)$ (i.e., the number of the rotation invariant N_f, plus $m_{00_{train}}$, $\psi_{0_{train}}$, φ_{train}, and ϑ_{train}). A database of ZMs invariants up to the 9^{th} order (i.e., $N_f = 53$) with $N_w = 5000$ has a size of 2.28 Mbytes. A navigation solution using two databases of ZMs invariants up to the 7^{th} order (i.e., $N_f = 34$) computed respectively for $d_{train} = 40$ m and $d_{train} = 20$ m, and a database of FDs up to the 10^{th} harmonic (i.e., $N_f = 18$) computed at $d_{train} = 10$ m, requires a size of 3.93 MBytes if $N_w = 5000$, and 2.353 MBytes if $N_w = 3000$. These requirements are compatible with the resources available on typical space qualified avionics.

4.6 Conclusion

In this Chapter we have proposed a template matching method to estimate the pose of a non-cooperative target during space rendezvous from a single binary image capturing the target's silhouette. The method is suitable for initial pose acquisition and for detecting faults and deviations in other on-board trackers, and represents a novel approach for the pose estimation of a spacecraft in a rendezvous. Three types of global descriptors, based on complex moments, Zernike moments, and Fourier descriptors, are introduced and compared in order to match the silhouette in a database of pose-dependent feature vectors generated offline. By exploiting the scale and rotation invariance of these descriptors, the approach requires discretizing only two pose angles to construct the database, leading to fast computation times appropriate for real-time implementations. Our performance analysis shows that Zernike moment invariants provide the highest accuracy and robustness in off-nominal conditions. Fourier descriptors show comparable performance with a much lower computational cost, but only for high resolution images or short target distances. This suggests that these two types of descriptors are complementary and could be used in combination. The next chapter will focus on the post-processing of the algorithm outputs to detect outliers and discriminate between symmetric attitudes.

Chapter 5

Integrated Solution

Contents

In this Chapter, the tools developed in Chapters 2, 3, and 4 are merged together to provide an integrated navigation solution to cover the final approach rendezvous phase. As anticipated, the development of background segmentation techniques was out of the scope of this thesis. However, the navigation chain has been tested also on synthetic images having Earth in the background after the application of a background subtraction algorithm. The algorithm was developed by Thales Alenia Space in the scope of rendezvous-related projects. For this reason, the background subtraction algorithm will be briefly described in Sec.5.1. In Sec.5.2 we will explain how to exploit the detection algorithm developed in Chap.4 to cover the *range-only* RDV phase, while in Sec.5.3 we details the 6 degrees-of-freedom integrated solution. Finally, in Sec.5.4 the navigation solution is applied to a simulated RDV scenario, and in Sec.5.5 the conclusions are drawn.

5.1 Background subtraction algorithms

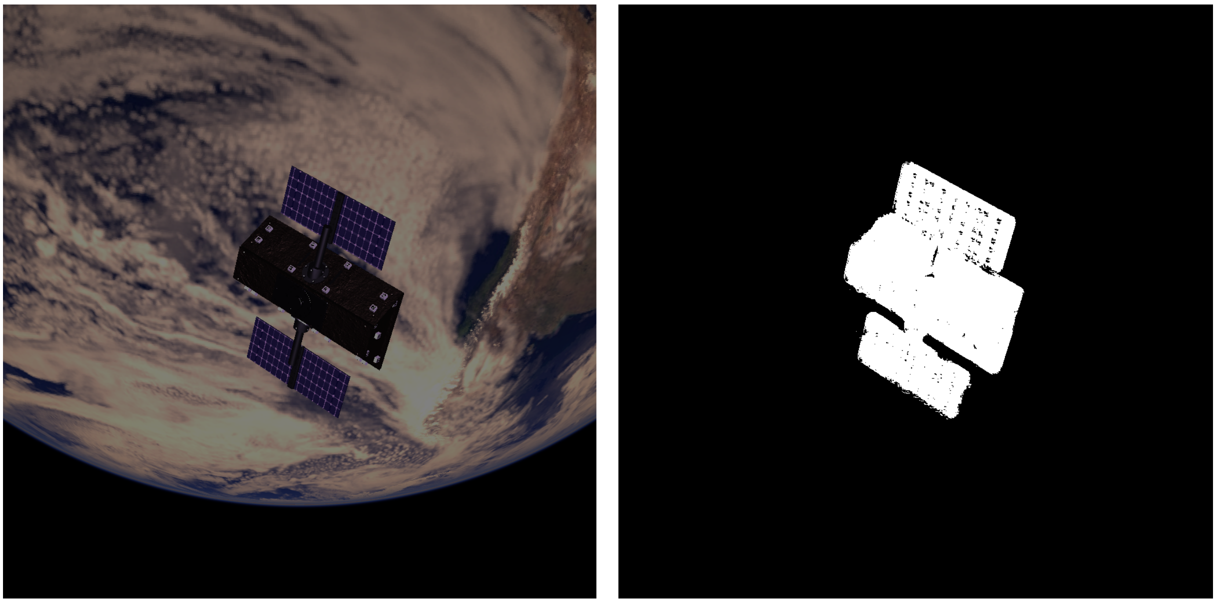


Figure 5.1: Example of input and output images of the segmentation algorithm

In order to robustly apply the pose estimation algorithms, it is necessary to isolate the target spacecraft from any possible complex background present on the captured image. In this section we will present the use of neural network applied to space image segmentation. Recently major advances have been achieved in the domain of computer vision thanks to neural network, especially convolutional neural networks. This allows reaching in many applications a precision that was complicated to achieve until now, and image segmentation strongly benefits from these new techniques. The use of neural network for image segmentation has already been explored on classic cases, with famous network architectures like Mask-R CNN [He et al., 2018], SegNet [Badrinarayanan et al., 2017], or U-NET [Ronneberger et al., 2015]. In the case considered for this study, the need is to precisely extract the mask of a satellite in space environment, with complex illumination conditions and backgrounds such as the Earth. The solution must be embeddable, meaning that the network must have a low computational load. Fully convolutional neural networks with transposed convolutions layers applied to a classical encoder-decoder architecture have been selected for this study. The convolution layers offer a higher level of representation at each step, while the encoder architecture decreases the feature space to force the categorization of pixel into relevant classes. Finally, the transpose convolution layers reconstruct the segmented image. This approach seems to be will fitted to our case, with acceptable resource consumption with a network composed by 6 convolutional layers and 13500 parameters to be trained. A meta-optimization operation was performed to find the best fitted meta-parameter of the network (e.g., depth of the encoding/decoding, number of convolution kernels). The output of the neural network is a gray-scale image where the intensity of the pixel corresponds to the probability the pixel has of belonging to the spacecraft silhouette. The segmentation has been tested with two independent dataset generated with two different synthetic image generators, one of them being highly representative. The network was previously trained on 9000 labeled images randomly generated by the image generator, and then tested on an independent dataset generated by the same generator on a given test case. The network was able to extract the satellite in almost every case, even when it was barely visible due to the distance or the presence of shadows. The mask borders are precise with a low noise of only few pixels. Some outputs showed the issue of misdetection of some spacecraft parts, either on the inside of the mask -especially for very short range detections-, or on thin parts like the solar array connections. The generated mask can present some false detection with secondary blobs, but these blobs are always smaller and with a lower pixel intensity than the satellite blob. A simple post processing algorithm is applied to extract the main blob: a 20% downscaled image exploration is performed to detect the blob, which is then extracted using a watershed algorithm. The final output is a binary image. For some images, this post-processing resulted in a separation of the solar arrays from the satellite core. This issue has been solved applying morphological filtering on the full scale image before the extraction of the main blob. This post-processing technique can also help filling missing S/C parts that are inside the detected silhouettes. With the filtering, the algorithm has a complexity of $\mathcal{O}(n)$, with n the size of the image. While no precise metric was measured at this stage due to the difficulty to find a relevant one, it was assessed that only the 0.2% of the images was presenting defaults making them non exploitable for pose estimation. The impact of the proposed segmentation method on the pose estimation algorithms will be shown in Sec.5.4, where the full navigation solution is tested.

5.2 Range-only navigation mode

As anticipated in 1.2, the 6 degrees of freedom (6-DOF) pose estimation is required only for the final approach, starting from a relative distance of $40 - 20$ m from the target. Before reaching this distance, different navigation modes are activated sequentially, and their activation depends mainly on the apparent size of the tracked object in the image. When the apparent size of an object in the field of view of the camera is smaller than 1 pixel, the object behaves as a point source. For an object this far away the luminance (i.e., luminous intensity per unit area of light traveling in a given direction) of the object is more important that the apparent size. The luminance depends on many factors, such as the viewing geometry, the direction of the Sun, the direction of the observer, and the reflectivity properties of the target. When the target starts being identifiable in the image, the so-called Line-of-Sight (LOS) navigation (also referred to as *angles-only* navigation) can start [D'Amico et al., 2013]. LOS navigation is bearing-only: it enables to correctly identify the camera-target direction, allowing to point the sensor towards the target, but does not provide range measurements.

As the size of the target in the image frame increases, model-based method can be used to measure the camera-target relative distance. When the range starts being measurable, the relative position camera-target can be fully retrieved. We will refer to this navigation mode as *range-only* mode. For a given target geometry and camera sensor, there is a range of distances in which the method based on global descriptors described in Chapter 4 does not allow the correct detection of the relative attitude of the observed object, but does provide reliable range measurements. The relative camera-target translation vector tr_{cam-tg}^{cam}, that we decompose in its coordinates x, y, z (i.e., $tr_{cam-tg}^{cam} = [x_{cam-tg}^{cam}, y_{cam-tg}^{cam}, z_{cam-tg}^{cam}] = [x, y, z]$) can be retrieved as described in Sec.4.3. When current view is matched with a view in the training database, the relative distance camera-target is computed according to Eq.(4.12), which is recalled here:

$$d_{meas} = \sqrt{m_{00_{train}}/m_{00_{meas}}} \cdot d_{train} \tag{5.1}$$

Then, using the coordinates of the current spacecraft silhouette ($x_{c_{meas}}$, $y_{c_{meas}}$), and the projection equation (3.7), one can write:

$$\begin{cases} \dfrac{x}{z} & = \dfrac{1}{f_x}\left(x_{c_{meas}} - c_x\right) \\ \dfrac{y}{z} & = \dfrac{1}{f_y}\left(y_{c_{meas}} - x_y\right) \\ d_{meas} & = \sqrt{x^2 + y^2 + z^2} \end{cases} \tag{5.2}$$

From the system (5.2), the relative position can be retrieved. Then, under the hypothesis of weak perspective model (see Sec.4.3), (5.2) can be approximate to $d_{meas} \sim z$, and the computation of the relative position is simplified.

Figure 5.2 shows the measured relative camera-target position in camera frame obtained using the detection method developed in Chapter 4 relying on Zernike moment invariants up to the 9^{th} order. The figure shows the results for a straight-line approach from 200 m to 40 m, using the training database built at a relative distance of 40 m computed for the tests described in Sec 4.5.2.2. The sensor properties are the same used in 4.5.2 (i.e., FOV= 30 deg, size of the image = 1024 × 1024 pixels, and $c_x = c_y = 512$), and figure 5.3 shows the appearance of the target spacecraft at a relative distance of 200 m.

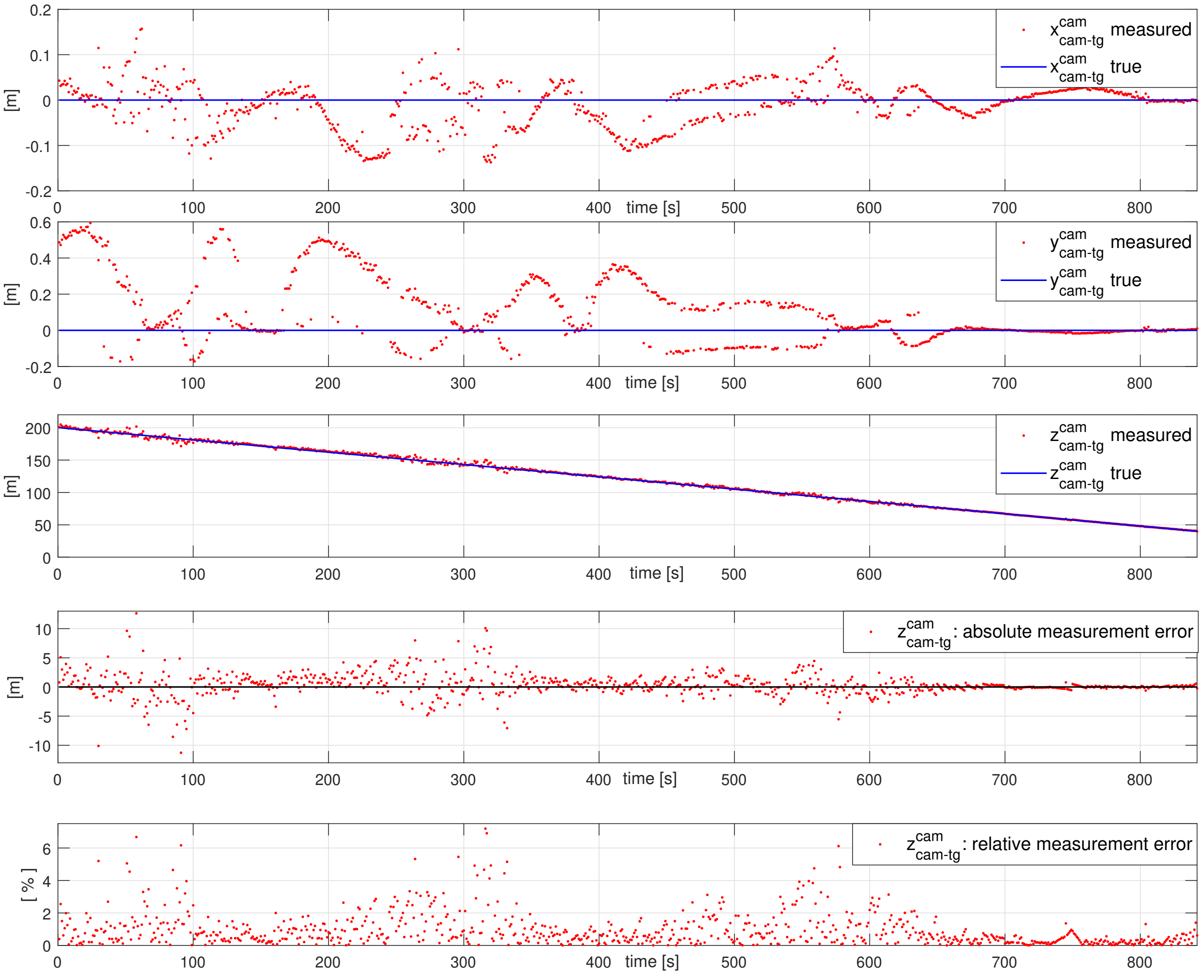


Figure 5.2: Measured camera-target position frame using ZM invariants

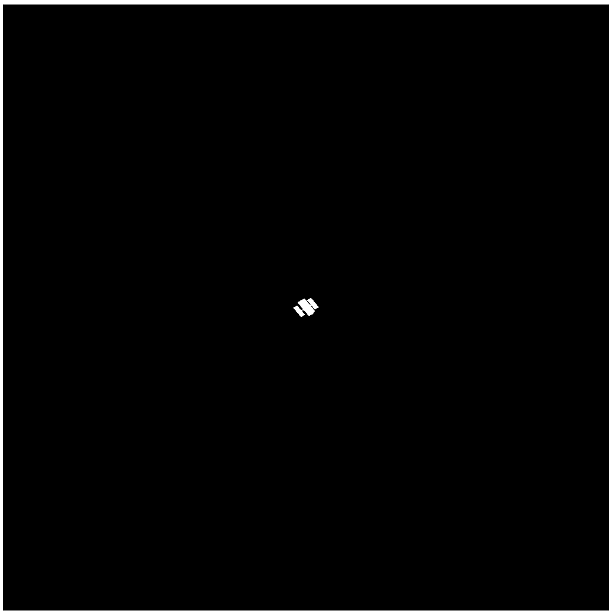


Figure 5.3: Silhouette of the target at 200 m with a 1024 × 1024 pixels camera, FOV=30°

From the figure it is visible how the measurement error decreases as the distance decreases, which is both due to the fact that the target size in the image increases, and the relative distance gets closer to the training distance. The fourth graphical in Fig.5.2 shows the relative percentage error, which sets below the 2% when the range is lower than 70 m. The coupling of these measurements with the translational Kalman filter based on the CWH equations described in 2.3.1 will improve the estimation and provide a smoother estimate, as it will be shown in the simulation of Sec.5.4. Note that this simulation is not representative of a real world scenario, where noise and segmentation errors should be expected and will cause the degradation of the performance. However, the results shown in Fig.5.2 allow understanding which is the performance of the method as a function of the apparent size of the S/C in the image frame. The results show that, for the considered target geometry, ZM invariants allow the estimation of the range starting from a distance of $z_0 = 200$ m with a field of view $\mathrm{FOV}_0 = 30$ deg. For a given object at a given attitude, the size of the projected silhouette in the image frame depends on the value of the range and the camera parameters. Thus, if range measurements are needed from a distance of z_1 m, this provide a constraint on the field of view FOV_1 of the camera to be used. Calling Δu the size of the projected target silhouette at a given attitude along the axis x_{cam}, and being ΔX the real-size dimension of the target along the same axis, one can write:

$$\begin{aligned} \Delta u &= f_x \frac{\Delta X}{z} = \frac{c_x}{\tan(FOV/2)} \frac{\Delta X}{z} \\ \frac{\Delta u}{\Delta x} &= \frac{c_x}{z \tan(FOV/2)} = \text{const} \end{aligned} \tag{5.3}$$

and therefore derive an expression for the value of FOV_1 that provides at a distance z_1 the same performance as the one of the reference camera with FOV_0 at a distance z_0:

$$\begin{aligned} \frac{c_{x0}}{z_0 \tan(FOV_0/2)} &= \frac{c_{x1}}{z_1 \tan(FOV_1/2)} \\ FOV_1 &= 2\text{atan}\left(\tan(FOV_0/2)\frac{c_{x1}}{c_{x0}}\frac{z_0}{z_1}\right) \end{aligned} \tag{5.4}$$

Considering two cameras having the same sensor size, in order to have range estimation from $z_1 = 1000$ m a NAC (Narrow-Angle-Camera) with a FOV_1 smaller than 6.13 deg is required. However it should be reminded that the operational range of a camera has also a lower bound, and that model-based algorithms start being unusable when the target occupies more than the 90% of the FOV. Therefore a NAC must always be coupled with a WAC (Wide-Angle-Camera), whose operational range superposes with the NAC operational range.

It should be reminded that the proposed method is not analytically exact, but provides a good approximation under the assumption of weak-perspective model, which is a reasonable model when the depth of the object along the line of sight is small compared to the distance from the camera, or when the camera field of view is small.

5.3 6 Degree-Of-Freedom pose estimation

Starting from a relative distance of $40 - 20$ m, the 6 DOF pose estimation should be initialized. First, a pose acquisition phase is needed: during this phase the servicer is on a holding point and acquires the 6-DOF pose of the target. Once that the pose is initialized, the tracking can start. During the tracking phase, the performance of the estimation is enhanced by the knowledge of the rigid body rotational dynamics using Kalman filtering techniques.

With a 1024 × 1024 pixels 30 deg FOV WAC, the pose can be reliably retrieved from a distance of 40 m using the Global Descriptor (GB) based detection algorithm developed in Chap.4. Using Zernike moment invariants, at a distance of 40 m with a train database built at the same distance, the method allows the detection of the observed pose with an accuracy of 95% and a mean error of 2 deg (see Tab.4.5). In order to have a reliable initialization of the pose, a methodology must be developed in order to robustly discard the 5% of the outlying detections. Moreover, the method presented in Chapter 4 do not allow to distinguish between two symmetric attitudes. In Section 5.3.2 we will present a methodology that allows initializing the pose discarding the outliers and eliminating the ambiguity due to the symmetry of the observed object, starting from measurement provided by the detection method proposed in Chap.4.

Once the pose is acquired, the tracking starts. The 6-DOF pose tracking will rely on two different and independent algorithms: a pose estimation by detection algorithm (i.e., *detector*, Sec.5.3.3.1) and a frame-by-frame recursive tracker (i.e., *tracker*, Sec.5.3.3.2). The *detector* relies on the measurements computed using the GD method, and exploits a kinematic KF to reject outliers and discriminate between multiple solutions caused by the target symmetries. The measurements computed by the detector are completely independent from the ones computed by the recursive tracker. On the other hand, the recursive tracker is based on the algorithm developed in Chapter 3 and it is coupled with the dynamic filter described in Chapter 2. Only the state estimated by the recursive tracker and the dynamic filter will be considered by the Guidance and the Control functions. The role of the detector is to supervise the recursive tracker in order to provide a corrected measurement in case of tracker divergences.

Before detailing each algorithm in a dedicated section, we provide in Fig. 5.4 an overview of the logic of the 6-DOF pose estimation. As the 6-DOF pose estimation phase starts, the *pose acquisition* algorithm is repeated until a set of conditions is met. If after a given number of iterations (i.e., *it max* in the diagram) the conditions are not satisfied, an error is sent. Otherwise, if the conditions are met, the pose is considered to be acquired and both the *tracker* and the *detector* are initialized.

If, during the tracking, the *detector* notices that its measurements are no more reliable, it can be reinitialized by the *pose acquisition* function. This reinitialization will not affect the *tracker*, if the tracker estimate is still considered reliable (i.e., if the condition *tracker initialized* in Fig.5.4 is true).

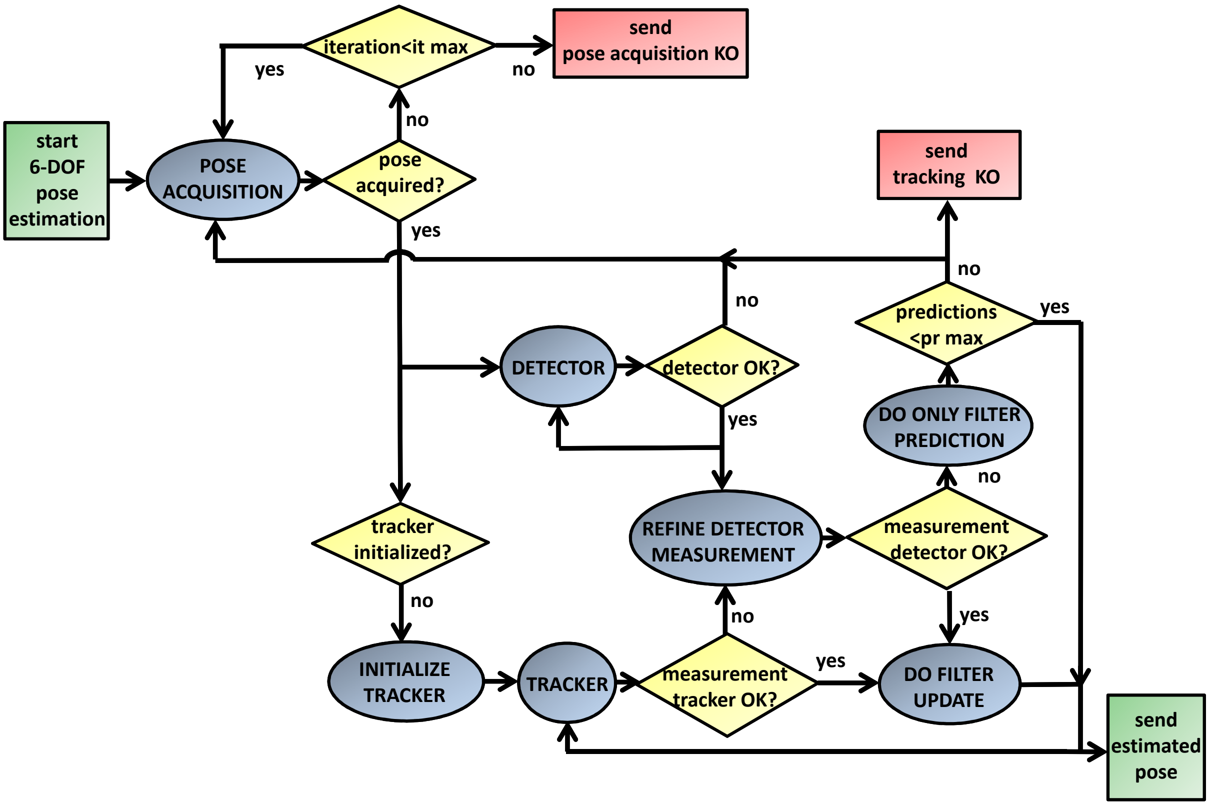


Figure 5.4: Logic of the 6-DOF pose estimation algorithm

The *tracker* behaves as follows. First, it computes a measurement and checks if it is reliable. If the measurement is reliable, it is sent to the dynamic filter to obtain the current pose estimate. Otherwise, the *tracker* asks the *detector* for a measurement. If the *detector* has an available measurement, the measurement is refined and, if reliable, it used in the filter update step. Otherwise, if the refined *detector* measurement is not considered as reliable, or if no *detector* measurements are available (i.e., the *detector* is in re-initialization), the dynamic filter performs only the prediction step (i.e., the filter operates in open-loop) and the predicted state is used as estimated state. If the filter propagates in open-loop for more time step than a given threshold (i.e., *pr max* in the diagram), then the dynamic filter sends an error and asks for the re-initialization of both the tracker and the detector.

5.3.1 Impact of target symmetry on the pose estimation

For sake of simplicity, the diagram in Fig.5.4 does not show the initialization of the symmetry, which will be discussed in Sec.5.3.2. However, the ambiguity brought by the target symmetric geometry needs to be addressed with ad-hoc solutions that depend on the particular geometry of the target, as well as on the type of mission.

For the particular case considered in this study, we do not consider the pose to be fully acquired as long as the symmetry ambiguity isn't solved. This is due to the fact that, for a given range of roll angles, there are some parts of the model that break the symmetry. These parts are clearly identifiable at the distance where the pose is acquired.

Other target satellites may have even more than two possible solutions for a given projected silhouette (i.e., in presence of axial symmetry, as for the satellite depicted in Fig.5.5) and no prominent 3D parts breaking the symmetry. In such a case, the tracking must be initialized before the detection of the symmetry, and only at very close range it would be possible to select the exact solution using visual features such as markers or the shape of the docking fixture.

However, it is important to notice that the non-detection of the symmetry does not prevent the dynamic filter to converge, and even allows to correctly estimate the rotation rate of the target expressed in chaser (and inertial) frame. We provide in this section more details.

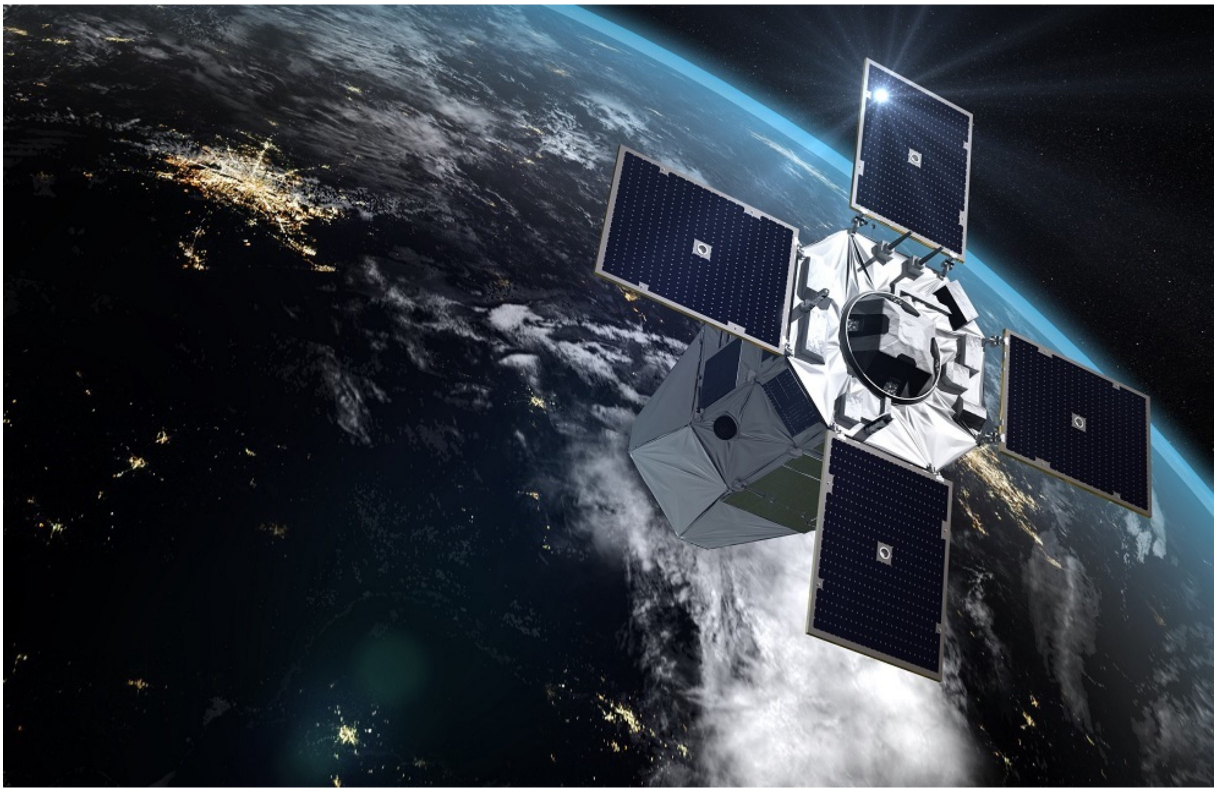


Figure 5.5: Rendering of the CSO-1 (Composante Spatiale Optique) satellite, which has a 4 fold rotation symmetry with respect to the nadir axis. Images credits: Ministère des Armées, *www.defense.gouv.fr*

For a 3D rigid object, there is no relative motion between the body frame and the "symmetric" body frame. For example, for the satellite in Fig.5.5, the attitude of the body frame can be determined only with an uncertainty corresponding to a rotation of $\pi/2$, π, and $3\pi/2$, around the nadir axis. For the model used in this study (see Fig.4.3), the body frame can be determined with an uncertainty of π around the body axis z_{tg} (i.e., x_{tg} and y_{tg} axis can be mistaken for $-x_{tg}$ and $-y_{tg}$). Let's call $q_{tg-tgSym}$ and $R_{tg-tgSym}$ the quaternion and the rotation matrix indicating the (constant) offset between the real body frame and the "symmetric" one:

$$\begin{aligned} q_{tg-tgSym} &= [0 \quad 0 \quad 0 \quad 1]^T \\ R_{tg-tgSym} &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.5}$$

For a given target attitude (expressed with respect to the inertial frame) $q_{i-tg} = [q_0\ , q_1\ , q_2\ , q_3]^T$, the "symmetrical" attitude $q_{i-tgSym}$ is:

$$q_{i-tgSym} \quad = q_{i-tg} \otimes q_{tg-tgSym} = [-q_3 \quad q_2 \quad -q_1 \quad q_0]^T \tag{5.6}$$

Using the formula in Eq.(A.5) one can obtain the relation between the two attitudes expressed in Euler angles:

$$\begin{aligned} eu_{i-tg} &= [\varphi \quad \vartheta \quad \psi]^T \\ eu_{i-tgSym} &= [-\varphi \quad -\vartheta \quad \psi]^T \end{aligned} \tag{5.7}$$

Since the quaternion $q_{tg-tgSym}$ is constant, the angular rates $\omega^{tgSym}_{tg-tgSym}$ and $\omega^{tg}_{tg-tgSym}$ are null (i.e., $\dot{q}_{tg-tgSym} = \frac{1}{2} q_{tg-tgSym} \otimes \omega^{tgSym}_{tg-tgSym} = \frac{1}{2}\omega^{tg}_{tg-tgSym} \otimes q_{tg-tgSym} = 0$). This allows to write an important relation on the angular rates. Being $\omega^{tg}_{i-tg} = [\omega_x,\ \omega_y,\ \omega_z]^T$ the true angular rate expressed in the true body axis, one can write:

$$\begin{aligned} \omega^{tgSym}_{i-tg} &= R^T_{tg-tgSym}\omega^{tg}_{i-tg} \\ \omega^{tgSym}_{i-tgSym} &= \omega^{tgSym}_{i-tg} + \omega^{tgSym}_{tg-tgSym} = R^T_{tg-tgSym}\omega^{tg}_{i-tg} + 0 = [-\omega_x \quad -\omega_y \ \omega_z]^T \end{aligned} \tag{5.8}$$

Intuitively this relation could be obtained by differentiation of the Euler angles (i.e., $\omega_{x/y_{sym}} = \frac{d\varphi/\vartheta_{sym}}{dt} = \frac{-\varphi/\vartheta_{t+1} - (-\varphi/\vartheta_t)}{dt} = -\omega_{x/y}$, and $\omega_{z_{sym}} = \frac{d\psi_{sym}}{dt} = \frac{\psi_{t+1} - \pi - (\psi_t - \pi)}{dt} = \omega_z$). This relation is very important because it says that, if the estimated attitude is not the true but the symmetric one, the Kalman filter rotation rate estimation will converge to the values of $[-\omega_x,\ -\omega_y,\ \omega_z]$. If at a given instant t the navigation algorithm notices that it is estimating the wrong state $[q_{i-tgSym},\ \omega^{tgSym}_{i-tgSym}]$, the state can be corrected using the equations in (5.8) and (5.6). The state covariance matrix P can be corrected using the jacobian of the (linear) transformation $[q_{i-tgSym},\ \omega^{tgSym}_{i-tgSym}] = \mathbb{J}[q_{i-tg},\ \omega^{tg}_{i-tg}]$:

$$\mathbb{J} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad P_{sym} = \mathbb{J}P\mathbb{J}^T \tag{5.9}$$

The mathematical relations that leads to the derivation of (5.8), (5.6), and (5.9) can be generalized for any geometric model and value of $q_{tg-tgSym}$. This would help to switch from one filter estimate to its symmetrical one as soon as the ambiguity is solved. However, it is very important to notice how the angular rate expressed in coordinates different from the target body coordinates is correctly estimated. Let's consider the relative camera-target rotation rate expressed in camera coordinates:

$$\omega^{cam}_{cam-tg} = -\omega^{cam}_{i-cam} + q_{cam-tg} \otimes \omega^{tg}_{i-tg} \otimes q^*_{cam-tg} \tag{5.10}$$

if the algorithm is estimating the symmetric attitude, the equation becomes:

$$\begin{aligned}\omega_{cam-tgSym}^{cam} &= -\omega_{i-cam}^{cam} + q_{cam-tgSym} \otimes \omega_{i-tgSym}^{tgSym} \otimes q_{cam-tgSym}^{*} \\ &= -\omega_{i-cam}^{cam} + q_{cam-tgSym} \otimes (\omega_{i-tg}^{tgSym} + \omega_{tg-tgSym}^{tgSym}) \otimes q_{cam-tgSym}^{*} \\ &= -\omega_{i-cam}^{cam} + q_{cam-tg} \otimes \left[q_{tg-tgSym} \otimes \left(\omega_{i-tg}^{tgSym} + 0\right) \otimes q_{tg-tgSym}^{*}\right] \otimes q_{cam-tg}^{*} \\ &= -\omega_{i-cam}^{cam} + q_{cam-tg} \otimes \omega_{i-tg}^{tg} \otimes q_{cam-tg}^{*} \\ &= \omega_{cam-tg}^{cam}\end{aligned} \tag{5.11}$$

This demonstration uses as “global” reference frame the camera frame *cam*, but can be repeated for the chaser *ch* frame and the inertial *i* frame. This result is very important because it states that the relative rotation rate of the target in camera (or chaser) axis is correctly estimated even in presence of a pose estimation error due to the symmetry. Therefore, for those applications where only the rotation rate is needed in order to synchronize the chaser motion with the target motion before capture or berthing (e.g., during active debris removal with non prepared S/C), it will not be necessary to solve the ambiguity related to the multiple symmetrical solutions.

5.3.2 Pose acquisition function

The pose acquisition function exploits measurements from the GD-based method. Due to the sensitivity of the acquisition phase, Zernike moment invariants are used, as they provide the best performance. The method described in 4.3 needs to be associated to some post-processing of the measurements to enable robust pose initialization, and in this section we details the post-processing procedure.

First of all, the training database must be extended also to the negative values of the roll angles. The database built in Sec.4.5.2 contained 5000 views corresponding to training attitudes where the roll angle was always positive (i.e., $\varphi \geq 0$). In fact, for the particular target geometry considered in this study, due to the presence of two symmetry plane, for each solution $[\varphi,\ \vartheta,\ \psi]$, the symmetrical solution $[-\varphi,\ -\vartheta,\ \psi - \pi]$ is also possible. The presence of small objects on the main body of the target breaks this symmetry for a range of roll angles such that $|\varphi| \in [80,\ 100]$ deg. Thus, the database is extended adding 500 views satisfying the following conditions (in degrees):

$$\begin{cases} \varphi &= -(80 + \ 20\,\mathrm{rand}_{\varphi}) \\ \vartheta &= \mathrm{asin}(1 - 2\,\mathrm{rand}_{\vartheta}) \\ \psi &= 0 \end{cases} \tag{5.12}$$

For these attitudes, the descriptor vector is computed according to Sec. 4.4.2. Using this “extended” training database, the raw measurements of the Euler attitude angles along a certain time-window will appear as in Fig.5.6. The blue solid lines represent the true Euler angles φ, ϑ, and ψ, while the blue dashed lines represent the symmetric solution branches. In the first graph it is possible to see how value of negative φ are registered only for $|\varphi| \in [80,\ 100]$, due to the particular construction of the database. The objective of the pose acquisition function is to identify a time-window in which there are not outlying measurements. For outlying measurement (or *outlier*) we mean a measurement which is not attributable to any of the two symmetric solutions.

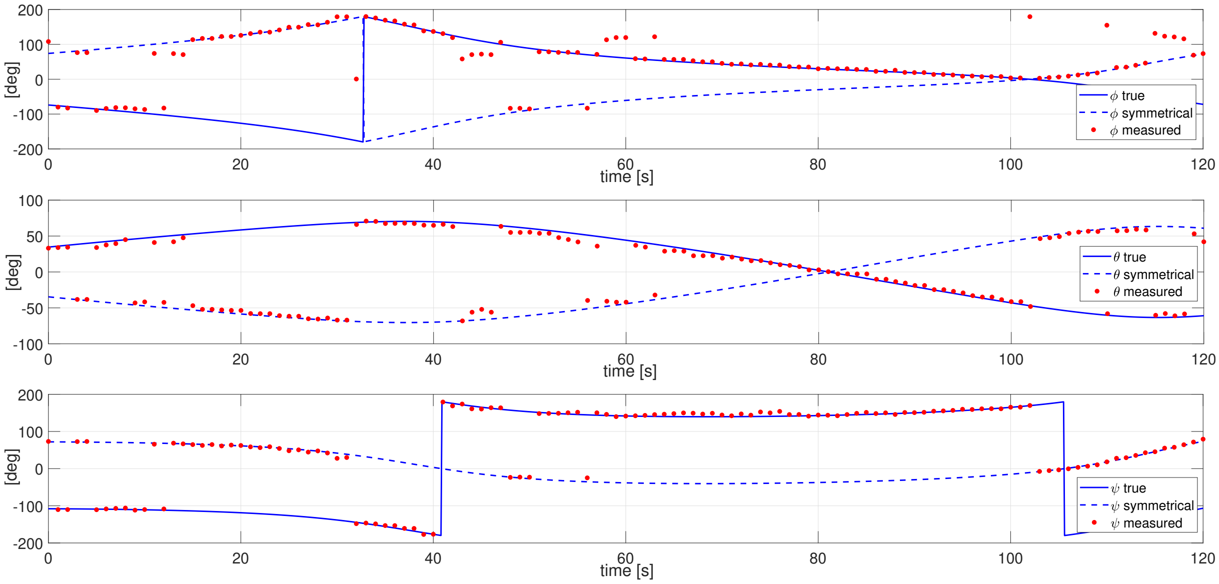


Figure 5.6: ZM-based detection algorithm raw measurements at a distance of 40 m

To initialize the pose, three conditions have to be satisfied.

1) Initially, we search for a time-window of a given length (l_{tw}) along which the difference between two consecutive attitude measurements is below a given threshold δ_{t1}. The fulfillment of such a condition ensures, in the observed time-window, that the measurements are "continuous" and therefore compatible with the rigid body dynamics, and that no outliers are present (the probability that in the considered time-window there is a series of outlying measurements close one to the other exists but remains low). At the current time-step $t = i$, the new acquired measurement $[\varphi_i,\ \vartheta_i,\ \psi_i]$ is compared with the measurement $[\varphi_{i-1},\ \vartheta_{i-1},\ \psi_{i-1}]$ acquired at the previous instant $t = i - 1$.
   Two index indicating the angular distance between the two measurements are computed (similarly to what was done in 4.5.2 to evaluate the recognition performance of the GD methods), one taking into account only the roll and pitch angles ($\delta_{\varphi\vartheta_i}$),

and the second taking into account the full attitude ($\delta_{\varphi\vartheta\psi_i}$).

$$
\begin{cases}
\delta q & = \text{quat}([\varphi_i,\ \vartheta_i,\ 0]) \otimes \text{quat}^*([\varphi_{i-1},\ \vartheta_{i-1},\ 0]) \\
\delta_{\varphi\vartheta_i} & = 2\left|\text{atan}\left(\sqrt{\delta q_1^2 + \delta q_2^2 + \delta q_3^2}/\delta q_0\right)\right|
\end{cases}
$$

$$
\begin{cases}
\delta q & = \text{quat}([\varphi_i,\ \vartheta_i,\ \psi_i]) \otimes \text{quat}^*([\varphi_{i-1},\ \vartheta_{i-1},\ \psi_{i-1}]) \\
\delta_{\varphi\vartheta\psi_i} & = 2\left|\text{atan}\left(\sqrt{\delta q_1^2 + \delta q_2^2 + \delta q_3^2}/\delta q_0\right)\right|
\end{cases}
\tag{5.13}
$$

The operator *quat*() indicates the operation that passes from the Euler angles to the corresponding quaternion (i.e., Eq.(A.6)). These indices are saved, together with the current measurement, on a shifting memory which collects the last l_{tw} (from $t = i + 1 - l_{tw}$ to $t = i$) values of the indices $\delta_{\varphi\vartheta_t}$, $\delta_{\varphi\vartheta\psi_t}$, and measurements.

It should be reminded that the roll and pitch angles are computed at the same time thought a nearest neighbor search, while the yaw angle is computed in a second step using geometric moments (see Sec.4.3). First, the condition on the continuity of the roll and pitch angles has to be satisfied. If all the stored values of $\delta_{\varphi\vartheta_t}$, $t \in [i+1-l_{tw}, i]$ are less than δ_{t_1}, the first condition $cond_1$ is satisfied. A reasonable value of the time-window size l_{tw} is 10, while for δ_{t1} a value of 20 deg can be used.

2) Then, the continuity of the yaw angle is checked. The second condition $cond_2$ is automatically satisfied if all the stored values of $\delta_{\varphi\vartheta\psi_t}$, $t \in [i+1-l_{tw}, i]$ are less than δ_{t_1}. However, as outlined in Sec.4.5.1, for the considered target geometry, there is an indetermination in the estimation of the yaw angle for those couple of $[\varphi,\ \vartheta]$ that correspond to a projected silhouette having a 2-fold rotation symmetry. In such a case, the GD based method cannot distinguish between two values of ψ shifted of π. This can lead to the presence of values of $\delta_{\varphi\vartheta\psi_t}$ close to 180 deg. Thus, an error in ψ close to 180 deg can indeed be corrected. A threshold δ_{t_2} is defined in order to set the acceptable difference between two consecutive yaw angles (a value of 20 deg can be used). From $t = i + 1 - l_{tw}$ to $t = i - 1$, the yaw discontinuity δ_{ψ_t} =min$(|\psi_{t+1} - \psi_t|, |\psi_{t+1} - \psi_t| - 2\pi)$ is computed. If at least one value of δ_{ψ_t} is in the interval $[\delta t_2, \pi - \delta t_2]$, the time-window is discarded since the discontinuity in ψ cannot be attributed to the 2-FRS ambiguity. On the other hand, if a value of δ_{psi_t} is in the interval $[\pi - \delta t_2, \pi]$, a counter c_ψ (initialized at 0) is incremented by one and the value of the yaw angle ψ_{t+1} is shifted of π. When the algorithm has reached $t = i - 1$, c_ψ will indicate the number of shifts that have been performed along the considered time-window.

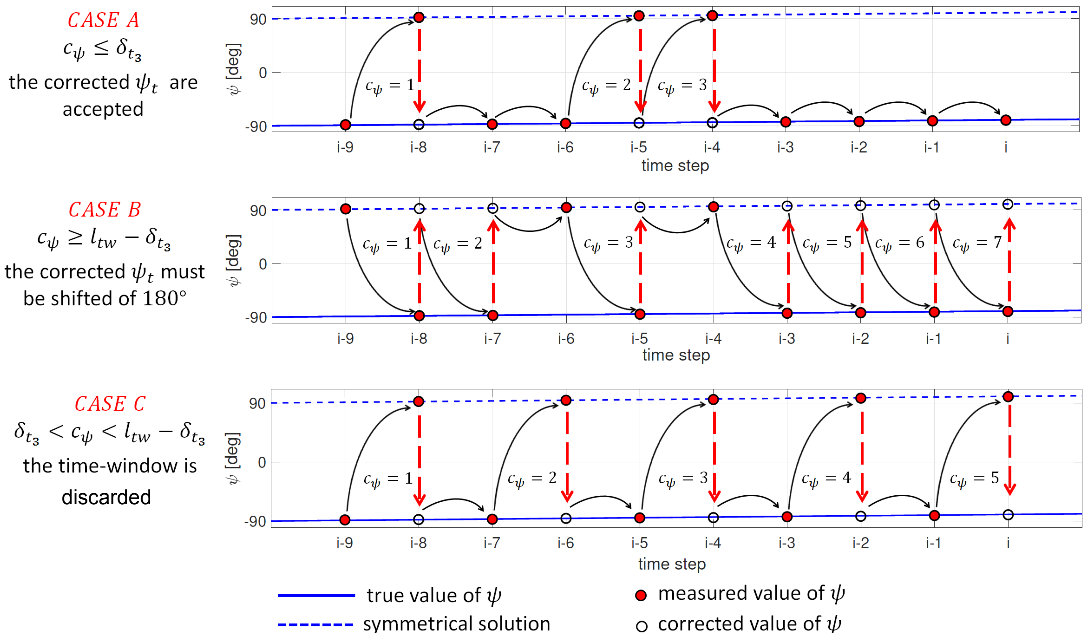


Figure 5.7: Correction algorithm detecting discontinuity of π in the yaw measurements

The value of c_ψ can go from 0 (no corrections have been performed) to $l_{tw}-1$ (all the value of ψ_t except for $t = i+1-l_{tw}$ have been corrected). A value of $c_{\psi_{max}}$ must be defined to set the maximal number of accepted corrections over a time-window. In this work, a value of $c_{\psi_{max}} = 30\% l_{tw}$ has been chosen. If $c_\psi \le c_{\psi_{max}}$ (Fig. 5.7, Case A), the corrections of the ψ_t are accepted, and the values of $\delta_{\varphi\vartheta\psi_t}$, $t \in [i{+}2{-}l_{tw}, i]$ are recomputed in order to check if they verify $cond_2$ (i.e., $\delta_{\varphi\vartheta\psi_t} < \delta_{t_1}$). If $c_\psi \ge l_{tw}{-}c_{\psi_{max}}$ (Fig. 5.7, Case B), then all the corrected values of ψ must be shifted of π, and then checked for the verification of $cond_2$. Finally, if $c_{\psi_{max}} < c_\psi < l_{tw} - c_{\psi_{max}}$(Fig. 5.7, Case C), it means that there are not enough values of ψ on the same solution branch, and therefore it is impossible to solve the ambiguity caused by the 2-FRS, and the time-window is discarded.

3) If both $cond_1$ and $cond_2$ are verified, then the current (and eventually corrected) measurement $q_{cam-tg_i} = \text{quat}([\varphi_i,\ \vartheta_i,\ \psi_i])$, together with the relative position tr^{cam}_{cam-tg} computed according to Sec.5.2, is used as prior to refine the measured pose using the contour-based non-linear tracking algorithm described in 3.3, obtaining $\tilde{q}_{cam-tg_i}$, $\tilde{tr}^{cam}_{cam-tg}$. Then, the reprojection error of the refined pose is computed according to Eq.(3.33), and compared with a threshold δ_{t_3}. The value of δ_{t_3} cannot be generalized because it depends on many factors, such as the relative range, the camera calibration matrix, the target geometry and even the camera-target relative attitude. Thus, a table of δ_{t_3} values must be pre-computed offline. As already spec-

ified in Sec.3.3.4.2, such a table should already be computed in order to provide a correlation between the measurements computed by the tracking algorithm and the value of the corresponding measurement noise covariance matrix. In a preliminary study, for a given geometry and camera properties, a simple parametrization of δ_{t_3} as a function of the camera-target range provides satisfying results. If the reprojection error is less than δ_{t_3}, the third condition $cond_3$ is verified and the pose $\tilde{q}_{cam-tg_i}$, $\tilde{x}^{cam}_{cam-tg}$ is accepted as initial pose. The time instant i is then referred to as t_{acq_f} and the time instant $i + 1 - l_{tw}$ is referred to as t_{acq_i}.

5.3.2.1 Symmetry detection

If $cond_3$ is verified, the pose $\tilde{q}_{cam-tg_i}$, $\tilde{x}^{cam}_{cam-tg}$ is considered a valid measurement to start the tracking. However, the ambiguity on the attitude $\tilde{q}_{cam-tg_i}$ due to the target symmetry remains. As anticipated, this ambiguity can be solved only if $|\varphi| \in [80, 100]$ deg. If the measured roll angle is in this range during the pose acquisition time-window, when $cond_2$ is verified, two refined poses and residuals are computed using the tracking algorithm. The first is computed using the relative attitude $q_{cam-tg_i} = \text{quat}[\varphi_i,\ \vartheta_i,\ \psi_i]$, and the second using the symmetric relative attitude $q^{sym}_{cam-tg_i} = \text{quat}([-\varphi_i,\ -\vartheta_i,\ \psi_i - \pi])$. The pose providing the lower residual is selected (if the difference between the two residual is above a certain threshold) and the residual is compared to δ_{t_3} to check for $cond_3$. If $q^{sym}_{cam-tg_i}$ results being the pose with the lower residual, all the collected measurements $[\varphi_t,\ \vartheta_t,\ \psi_t]$, $t \in [i + 1 - l_{tw}, i]$ are “switched” to the symmetrical solution branches ($[\varphi_t,\ \vartheta_t,\ \psi_t] \to [-\varphi_t,\ -\vartheta_t,\ \psi_t - \pi]$). The symmetry is then considered to be initialized and $i = t_{acq_f} = t_{sym}$. The tracking is initialized even if during the acquisition time-window the roll angles do not enter in the interval allowing the discrimination of the symmetric attitudes, Then, as soon as the roll angle measured by the tracking algorithm enters the range $|\phi| \in [80,\ 100]$ deg, the procedure described above is repeated applying the criterion based on the residual previously described. More details are provided in Sec.5.3.3.2.

5.3.2.2 Filter initialization

When the initial attitude q_{cam-tg} is acquired, the rotational dynamic filter has to be initialized. There is no need to do this for the translational dynamics since the translational KF has already been activated during the Range-only mode. All the measurement acquired (and eventually corrected) during the pose acquisition time-window are taken into account: the measurements are propagated inside a kinematic rotational filter from $t = t_{acq_i}$ to $t = t_{acq_f}$. The filter state will be composed by the target absolute attitude quaternion q_{i-tg} and its rotation rate with respect to the inertial frame expressed in body axis ω_{i-tg}^{tg}. The prediction model is described in Appx.C.3.1, where also the state jacobian matrix of the considered model is introduced. The sensitivity matrix C is the same as the sensitivity matrix of the dynamic filter whose computation is carried out in Sec.2.3.2. In order to compute the sensitivity matrix C, it will be necessary to store at each instant the latest l_{tw} values of the estimated chaser attitude quaternion q_{i-ch} and eventually of the quaternion q_{ch-cam} if it is time-varying (e.g., the camera is placed on a articulated robotic arm). The sensitivity matrix is equal to $C_t = \Sigma(q^*_{ch-cam_t} \otimes q^*_{i-ch_t})$, see Eq.(2.20)). The prediction of the state and the state covariance matrix is performed according to the CD-EKF formulation (see Appx.B.2). The initialization of the filter is given by:

$$\begin{cases} q_{i-tg_{tacq_i}|t_{acq_i}} & = q_{i-ch_{tacq_i}} \otimes q_{ch-cam_{tacq_i}} \otimes q_{cam-tg_{tacq_i}} \\ \omega^{tg}_{i-tg_{tacq_i}|t_{acq_i}} & = [0,\ 0,\ 0]^T \\ x_{t_{acq_i}|t_{acq_i}} & = \begin{bmatrix} q_{i-tg_{tacq_i}|t_{acq_i}} \\ \omega^{tg}_{i-tg_{tacq_i}|t_{acq_i}} \end{bmatrix} \\ P_{t_{acq_i}|t_{acq_i}} & = \text{eye}(7) \end{cases} \tag{5.14}$$

Successively the kinematic filter goes from $t = t_{acq_i} + 1$ to $t = t_{acq_f}$. This allows to have a robust initialization of the target rotational state at the current instant t_{acq_f}, and allows the state covariance matrix P to start converging. The obtained values of the state covariance matrix $P_{t_{acq_f}}$ and the state $x_{t_{acq_f}}$ will be used as initialization for the dynamic rotational KF described in Sec.2.3.2.

The filter initialization function has a latency that can be neglected with respect to the pose acquisition function described in the previous paragraph, which is dominated by the time needed to compute moment invariants (see Sec.4.5.4).

5.3.3 Pose tracking functions

We have already explained that the 6-DOF pose tracking will rely on two different and independent algorithms, i.e., the *detector* and the *tracker*. The measurements computed by the detector are completely independent from the ones computed by the recursive tracker. The role of the detector is to provide a corrected measurement in case of tracker divergences.

5.3.3.1 Detector

The logic of the detection algorithm is very simple. At the current time step t, the GD method developed in Chapter 4 is used to obtain a measurement of the relative camera-target Euler angles $eu^{meas}_{cam-tg} = [\varphi,\ \vartheta,\ \psi]$. From these angles, the symmetrical solution $eu^{sym}_{cam-tg} = [-\varphi,\ -\vartheta,\ \psi - \pi]$ is built. The goal then is to determine which is the more likely solution according to the target state at the previous instants. Thus, a kinematic filter is associated to the detection method. The filter structure and initialization are discussed in Sec.5.3.2.2. From the estimated state at the instant $t-1$, the predicted target absolute state at the instant t is computed, and, using the current camera absolute attitude quaternion q_{i-cam}, the predicted q_{cam-tg} and the associated Euler angles $eu_{cam-tg} = [\varphi^{pred},\ \vartheta^{pred},\ \psi^{pred}]$ are obtained. The predicted quaternion serves as *anchor point* to select the most likely measurement. The angular distances of the predicted attitude from the current measurement and from the symmetric measurement are computed, both with and without taking into account the yaw angle.

$$\begin{cases} \delta q &= \text{quat}([\varphi,\ \vartheta,\ 0]) \otimes \text{quat}^*([\varphi^{pred},\ \vartheta^{pred},\ 0]) \\ \delta_{\varphi\vartheta_{meas}} &= 2\left|\text{atan}\left(\sqrt{\delta q_1^2+\delta q_2^2+\delta q_3^2}/\delta q_0\right)\right| \end{cases}$$
$$\begin{cases} \delta q &= \text{quat}([\varphi,\ \vartheta,\ \psi]) \otimes \text{quat}^*([\varphi^{pred},\ \vartheta^{pred},\ \psi^{pred}]) \\ \delta_{\varphi\vartheta\psi_{meas}} &= 2\left|\text{atan}\left(\sqrt{\delta q_1^2+\delta q_2^2+\delta q_3^2}/\delta q_0\right)\right| \end{cases}$$
$$\begin{cases} \delta q &= \text{quat}([-\varphi,\ -\vartheta,\ 0]) \otimes \text{quat}^*([\varphi^{pred},\ \vartheta^{pred},\ 0]) \\ \delta_{\varphi\vartheta_{sym}} &= 2\left|\text{atan}\left(\sqrt{\delta q_1^2+\delta q_2^2+\delta q_3^2}/\delta q_0\right)\right| \end{cases} \tag{5.15}$$
$$\begin{cases} \delta q &= \text{quat}([-\varphi,\ -\vartheta,\ \psi-\pi]) \otimes \text{quat}^*([\varphi^{pred},\ \vartheta^{pred},\ \psi^{pred}]) \\ \delta_{\varphi\vartheta\psi_{sym}} &= 2\left|\text{atan}\left(\sqrt{\delta q_1^2+\delta q_2^2+\delta q_3^2}/\delta q_0\right)\right| \end{cases}$$

If $\delta_{\varphi\vartheta_{sym}} < \delta_{\varphi\vartheta_{meas}}$, the symmetric solution is the one closer to the prediction, and the measurement must be corrected:

$$\begin{aligned} &\text{if}\ \ \delta_{\varphi\vartheta_{sym}} < \delta_{\varphi\vartheta_{meas}} \\ &\begin{cases} eu^{meas}_{cam-tg_t} &= eu^{sym}_{cam-tg_t} \\ \delta_{\varphi\vartheta_{meas}} &= \delta_{\varphi\vartheta_{sym}} \\ \delta_{\varphi\vartheta\psi_{meas}} &= \delta_{\varphi\vartheta\psi_{sym}} \end{cases} \end{aligned} \tag{5.16}$$

Once that the ambiguity on the two symmetric solution is solved, the algorithm searches for eventual correctable discontinuities in the yaw angle. Similarly to what was done in the pose acquisition algorithm, the angular distance δ_ψ =min$(|\psi-\psi^{pred}|, |\psi-\psi^{pred}|-2\pi)$ is computed. If δ_{psi} is in the interval $[\pi-\delta t_2, \pi]$, then ψ must be switched of π and the value $\delta_{\varphi\vartheta\psi_{meas}}$ must be updated:

$$\begin{aligned} &\text{if}\ \ \pi-\delta_{t_2} \leq \delta_\psi \leq \pi \\ &\begin{cases} \psi &= \psi-\pi \\ \delta q &= \text{quat}([\varphi,\ \vartheta,\ \psi]) \otimes \text{quat}^*([\varphi^{pred},\ \vartheta^{pred},\ \psi^{pred}]) \\ \delta_{\varphi\vartheta\psi_{meas}} &= 2\left|\text{atan}\left(\sqrt{\delta q_1^2+\delta q_2^2+\delta q_3^2}/\delta q_0\right)\right| \end{cases} \end{aligned} \tag{5.17}$$

At the end of these two step, the corrected measurement is considered only if both $\delta_{\varphi\vartheta_{meas}}$ and $\delta_{\varphi\vartheta\psi_{meas}}$ are below a threshold which can be set to be equal to δ_{t_1} (e.g., 20 deg). If the conditions are satisfied, then the measurement can be used in the update step of the kinematic filter. Otherwise, the measurement is considered to be an outlier and it is discarded. In such a case, the KF estimated state will be equal to the KF predicted state. A counter c_{kine} is increased by one any time a measurement is discarded, and it is reset to zero any time a measurement is accepted and used in the KF update step. The counter indicates the number of time step during which the state has been propagated without any measurement update. If the counter exceed a certain threshold $c_{kine_{max}}$ (with $c_{kine_{max}} \sim 20$ step), the state is no more considered as reliable and the detector must be reinitialized though the procedure discussed in Sec. 5.3.2. In such a case, a flag indicating the availability of the detector measurements must be set to *false* as long as a new pose is acquired. It should be noted that this reinitialization will not affect the recursive tracker, which only in case of its own divergence will “consult” the detector.

It has already been anticipated that, if the symmetry has not been initialized during the pose acquisition phase, it is the task of the recursive tracker to initialize it. When the recursive tracker has solved the ambiguity on the symmetry, it sends a flag to the detector indicating whether it has to switch on the symmetric solution branch or not. If at the generic instant t the flag (sent from the recursive tracker at the time step $t-1$) indicates to the detector that it must switch to the symmetric solution, the kinematic KF estimated state and covariance matrix at $t-1$ must be corrected according to Eqs.(5.6), (5.8), and (5.9). In such a way, the predicted state at t will be placed on the correct solution branch and will allow to select the correct measurement.

5.3.3.2 Tracker

The tracker is based on the contour-based non-linear tracking algorithm discussed in Sec.3.3. This algorithm uses, at the time step t, the estimated relative pose at the instant $t-1$ (obtained from the dynamic KF estimate) to project the 3D a priori model, select the control points and match them to the edges extracted from the current captured image. Then the new pose is computed minimizing a non-linear cost function. This measured pose is then incorporated in the dynamic KF, as discussed in Sec.3.3.4. Both the IP-CV algorithm and the dynamic filter are activated at $t = t_{acq_f}$ and initialized through the procedure discussed in Sec.5.3.2.2.

While the IP-CV “core” of the tracker (masking algorithm, matching, cost function minimization) has not been modified for the implementation of the full solution, its interface with the dynamic filter has been changed in order to allow a dynamic calibration of the filter and to benefit from the addition of the detection algorithm.

The measurement computed by the frame-by-frame tracker is associated to a residual. As explained in Sec.3.3.4.2, for a given target geometry, camera parameters and relative camera-target distance, the residual can be correlated to the noise covariance matrix R of the associated measurements (i.e., a very low residual indicates that the reprojection error is minimal and therefore the measurement is reliable). For the simulations carried out in the following Section (i.e., Sec.5.4), we have decided to use a discrete tuning of R. For a given distance, three thresholds are set, i.e., r_{low}, r_{high}, and r_{max}. The thresholds are computed experimentally during a series of tests where the residual is correlated to the magnitude of the measurement error. If the residual is below r_{low}, the value of R is reduced of one order of magnitude with respect to the nominal value, i.e., $R = 0.1\ R_{nom}$. If the residual is in the interval $[r_{low}, r_{high}]$, then $R = R_{nom}$. If the residual is in the interval $]r_{high},\ r_{max}]$, the covariance matrix is increased of two order of magnitude, i.e., $R = 100\ R_{nom}$. Finally, if the residual exceeds the threshold r_{max}, the measurement computed by the recursive tracker is discarded. In such a case, the recursive tracker “asks” the detector for its measured pose (downstream of the kinematic KF). Three scenarios are possible:

- The validity flag of the detector estimate is *true*: the estimated relative pose sent by the detector is refined using the contour-based tracking algorithm and the residual is computed. If the residual is below r_{max} the measurement is accepted and used in the dynamic KF to perform the state update. The measurement noise covariance matrix is set to be $R = 100\ R_{nom}$.

- The validity flag of the detector estimate is *true*, but the residual obtained refining the detector estimated pose is above the threshold r_{max}. The measurement is discarded, and the state in the dynamic Kalman filter is propagated in “open-loop” (i.e., only prediction step).

- The validity flag of the detector estimate is *false*. This is the worst case scenario, since the detector estimate is not available to correct the divergence in the recursive tracker. In such a scenario, the state of the dynamic filter is propagated without update step as long as the pose is re-acquired.

If the dynamic filter does not receive any update for a given number of time steps (i.e., the number of open-loop cycles of the KF exceed the threshold *pr max* in Fig.5.4), it is the filter that ask for a full re-initialization of the pose. If the pose is not reacquired after a given number of time-steps (i.e., *it max* in Fig.5.4), an error is sent at higher level. If the tracking is lost, the chaser should stop its approach motion to begin a new acquisition phase or even perform a collision avoidance maneuver if necessary.

Symmetry detection

If the symmetry ambiguity has not been resolved during the pose acquisition phase, this task is transferred to the tracker. If the estimated relative camera-target Euler angle ϕ at a certain time step enters the interval $|\psi| \in [80, 100]$, at the following time step the contour-based pose estimation algorithm is run using both the current and the symmetric pose as initial guess. The pose providing the lower residual is selected (if the difference between the two residual is above a certain threshold). If the selected pose is on the solution branch opposite to the current estimate, also the dynamic KF state must be corrected, according to Eqs.(5.6), (5.8), and (5.9). In this case, a flag is sent to the detector in order to allow it performing the same correction. In this case, the latency of the tracker is almost doubled, and the delay is managed by the dynamic KF.

5.3.4 6-DOF pose estimation logic

The diagram in Fig.5.8 shows the detailed logic of the pose estimation function. For sake of simplicity, the error messages shown in Fig. 5.4 are not included in the work flow.

The diagram is divided in two “environments”, the *image processing* block and the *navigation function* block. This division aims at separating the functions that process the images to obtain a valid measurement, from the navigation function which receives as input a measurement and outputs an estimated state. The pose acquisition function, the detector and the tracker are gathered in the *image processing* block, while the dynamic filter is contained in the *navigation function* block, where they may be other blocks such as the chaser absolute rotational dynamics and the chaser absolute translational dynamics estimators. This separation may be also performed at hardware level, by implementing all the image processing on dedicated hardware such as a FPGA.

The communication between the blocks is performed by numerical variables and Boolean flags. Flags travel on the dashed lines, while variables travel on the solid lines.

The flags that are connected to a block indicate that the action inside the block must be performed only if the flag is true. There are four flags connected to a block:

- *do_pose_acq*: if *true*, the pose acquisition function must be executed. It is initialized as *true.*
- *do_detector*: if *true*, the detector must be executed. It is initialized as *false.*

- *do_tracker*: if *true*, the tracker and the dynamic filter must be executed. It is initialized as *false*.
- *do_update*: if *true*, the the dynamic filter uses the measurement form the image processing function to do the state update, otherwise the filter evolves in open-loop. It is initialized as *false*.

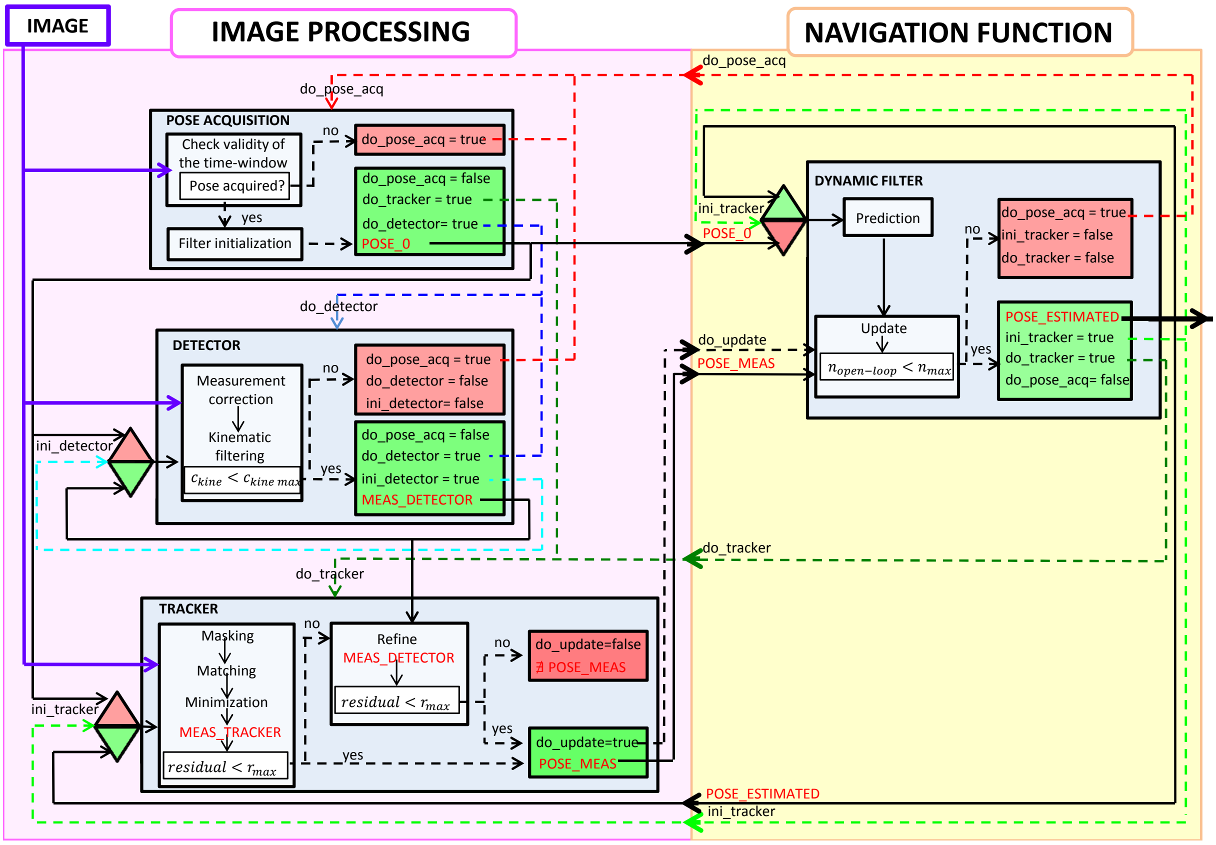


Figure 5.8: Detailed logic of the 6-DOF pose estimation

The flags connected to the bi-color diamonds operate as selectors: if the flag is *true*, the output of the diamond is equal to the input connected to the green triangle; if the flag is *false*, the output of the diamond is the input connected to the red triangle. There are two flags connected to a diamond:

- *ini_detector*: it is *false* if the detector needs an initialization (in such a case the new acquired pose from the pose acquisition function is used as prior in the algorithm). It is initialized as *false*.
- *ini_tracker*: it is *false* if the tracker needs an initialization (in such a case the new acquired pose from the pose acquisition function is used as prior in the algorithm). It is initialized as *false*.

The diagram aims also at highlighting the input-output flow from the image processing function and the navigation function. The navigation function receives, as input, the initialized pose *POSE_0* from the pose acquisition function if the tracking is not yet initialized, and the measured pose (i.e. *POSE_MEAS*) with the *do_update* flag from the tracker. The outputs of the navigation function are the inputs of the image processing function at the next time-step. They are the *POSE_ESTIMATED*, the *ini_tracker* flag, and the *do_tracking* flag, which all enter the tracker block, and the *do_pose_acq* flag which enters the pose acquisition function and activate it if the tracking is lost, i.e., when the number of open-loop cycles ($n_{open-loop}$) of the dynamic KF exceeds the threshold n_{max}. The estimated pose is also send towards the guidance and the control functions, which are not included in the diagram. Also signals such as the *do_pose_acq* and the *do_tracking* should be sent at higher level (i.e., to the mission and vehicle management (MVM) or the failure detection, identification and recovery (FDIR) functions), in order to activate different modes (e.g., collision avoidance maneuvers, stop of the forced motion and so on).

5.4 Application to an operational scenario

The navigation architecture has been tested on a forced motion scenario from a relative distance of 100 m to 5 m, using the 1024×1024 pixels, 30 deg FOV camera that has been used also in the simulations of Chapters 3 and 4. In the first 2 minutes of the simulation, the chaser approaches the target along the V-bar (i.e., y_{LOF} axis) at a relative speed of 0.5 m/s. During this phase, only range measurements are provided using the algorithm discussed in Sec.5.2. When the chaser reaches a distance of 40 m from the target, it stops the motion and start the 6-DOF pose acquisition phase, according to the method discussed in Sec.5.3.2. After 2 minutes, the chaser starts another forced motion towards the target at a relative speed of 0.2 m/s. The approach velocities have been chosen in order to cover all the phases in a relatively short simulation, but remain high compared to the approach rates usually used during close proximity operations (e.g., for the MEV-1 mission, the approach from 80 m to docking lasted some hours, see Sec.1.2.3). However, the choice of high approach velocities allows us to test the tracking algorithm in presence of high displacements of the target from one image frame to the following one. The target rotation rate is initialized to be 1.5 deg/s around each body axis, which is an upper bound of the rate that can be expected during a rendezvous [Bonnal et al., 2013]. The camera acquisition frequency is 1 Hz, while the navigation filter run frequency is 10 Hz. In order to run the detection algorithm, three different database have been built. The first one is built at a distance of $d_{train} = 40$ m, and it is used in the range-only phase, in the pose acquisition phase and during the tracking, down to a relative distance of 30 m. The second database is built at $d_{train} = 20$ m and it is used for a relative distance in the interval [30, 15] m. For the distances below 15 m a database built at $d_{train} = 10$ m is used. All the databases have the size of 5500×56 (i.e., $N_w = 5500$ samples, $N_f = 51$ Zernike rotation invariants up to the 9^{th} order, and 4 more global features being $\psi_{0_{train}}$, $m_{00_{train}}$, φ_{train}, and ϑ_{train}). 9^{th} order ZM invariants have been chosen as they have proven to provide the optimal performance in Sec.4.5.2. As the distance decreases, the interval of distances in which a database built at a given distance d_{train} can provide reliable detections become smaller. This is due to the fact that the weak perspective approximation is no more applicable for shorter ranges, and the effect of the relative distance on the shape of the projected silhouette becomes non-negligible. For a 30 deg FOV camera and the considered target geometry, the detection algorithm does not provide reliable measurements below ~ 8 m.

5.4.1 Discussion of the results

The simulation is run with three different series of images:

- CASE A: The first set of images are perfect binary silhouettes of the target satellite. The simulation run with this set is the baseline simulation which shows the intrinsic performance of the pose estimation algorithm.

- CASE B: The second set of images is composed by grey-scale images obtained illuminating the target satellite with a spot-light placed on the chaser. This could be the case of a rendezvous performed approaching the target from the -R-bar side, or a rendezvous on a geo-stationary orbit performed along the V-bar. The images are affected by shadows, reflections and blur, so that the silhouette retrieved is affected by multiple sources of noise. The silhouettes are retrieved using a simple thresholding technique.

- CASE C: The third set of images is composed by grey-scale images obtained illuminating the target satellite with a spot-light placed on the chaser, and Earth in the background. This is the case that is more likely to happen during rendezvous on a Low Earth orbit. The method discussed in Sec.5.1 is used to extract the silhouette from the background.

Tables 5.1 and 5.2 show the performance in the estimation of the target attitude and the relative position of all cases, as a function of the range. The metric used for the rotational dynamics performance (Tab.5.1) is the angular error $\delta_{\varphi\vartheta\psi}$. The metric used for the translational dynamic (Tab.5.2) is the range error. Both the mean knowledge error μ (i.e., the MKE, which is the mean of the Absolute Knowledge Error AKE according to [Ott et al., 2011]) and the root mean square of the AKE are listed. It is evident how the performance is degraded in cases B and C. However at close range (i.e., below 20 m), the performance is satisfying in all the scenarios. For what concerns the translation estimation performance, it is interesting to notice how the overall performance of CASE B (i.e., no Earth in background, simple thresholding) is worse than the overall performance of CASE C (i.e., Earth in background, background subtraction and silhouette post-processing). This suggests that the post-processing technique proposed in Sec.5.1 should be used also in case of deep-space background to enhance the silhouette extraction.

Table 5.1: Comparison of the performance of the 6-DOF pose estimation algorithm as a function of the range

Rotational dynamics, estimated $\delta_{\varphi\vartheta\psi}$ [deg]

range [m]:		$= 40$	$\in]40, 30]$	$\in]30, 20]$	$\in]20, 10]$	< 10	*overall*
CASE A	μ	0.98	1.11	0.39	0.43	1.09	0.80
	rms	1.72	1.40	0.45	0.57	1.17	1.28
CASE B	μ	1.30	4.10	1.27	1.02	1.19	1.75
	rms	1.44	5.07	1.47	1.23	1.38	2.52
CASE C	μ	3.76	5.92	2.36	2.52	2.16	3.51
	rms	4.68	7.22	2.42	2.57	2.23	4.48

Table 5.2: Comparison of the performance of the 6-DOF pose estimation algorithm as a function of the range

Translational dynamics: estimated range error [m]

range [m]:		$= 40$	$\in]40, 30]$	$\in]30, 20]$	$\in]20, 10]$	< 10	*overall*
CASE A	μ	0.195	0.267	0.107	0.043	0.045	0.150
	rms	0.206	0.279	0.115	0.047	0.046	0.180
CASE B	μ	0.591	1.845	0.507	0.192	0.024	0.679
	rms	0.649	1.903	0.564	0.224	0.029	0.940
CASE C	μ	0.240	0.422	0.229	0.275	0.366	0.291
	rms	0.307	0.430	0.261	0.288	0.369	0.329

Table 5.3 shows the overall performance (from $t = t_{acq_f}$) of more metrics. For the rotational dynamics, the mean of $\delta_{\varphi\vartheta\psi_{detector}}$, $\delta_{\varphi\vartheta\psi_{tracker}}$, and $\delta_{\varphi\vartheta\psi_{est}}$ (the angular error re-

spectively of the detector measurements, of the tracker measurements, and of the dynamic filter estimates) is shown. Also the mean of the AKE of the estimated rotation rate $\delta_{\omega_{x_{est}}}$, $\delta_{\omega_{y_{est}}}$, and $\delta_{\omega_{z_{est}}}$ is listed. For the translational dynamics, the mean of the position estimation error (i.e., $\delta_{x_{ext}}$, $\delta_{y_{ext}}$, and $\delta_{z_{ext}}$) and the mean of velocity estimation error (i.e., $\delta_{\dot{x}_{ext}}$, $\delta_{\dot{y}_{ext}}$, and $\delta_{\dot{z}_{ext}}$) are listed. The translational state is expressed in LOF frame, so that the y component indicates the direction of the motion, and it is therefore characterized by higher errors. For CASE C, the overall angular error of the detector is very high due to the fact that the detector diverges between 40 m and 30 m and is therefore reinitialized.

For what concerns the performance of CASE C, significant improvement could be obtained by training the background subtraction neural network with a bigger training dataset and increasing the number of layers in the network, but this could lead to an augmentation of the computational load of the algorithm. The performance of the estimation algorithm on the test CASE C shows that the navigation solution is quite robust with respect to the presence of noise and errors on the extracted silhouette. This means that, in the presence of deformation of the target satellite due to the flexibility of the appendages or to damaged parts, the algorithm ensures a certain level of robustness. However, in the presence of big non-modeled differences, the 3D a priori model upon which the estimation algorithms rely could be updated from ground after an inspection phase. In fact, the segmentation algorithm is the only part of the navigation solution that requires a relatively long training phase on-ground, but it could be made model-invariant by increasing the training dataset size with images of different spacecrafts and increasing the number of the

Table 5.3: Overall comparison

			CASE A	CASE B	CASE C
$\delta_{\varphi\vartheta\psi_{detector}}$	[deg]	μ	3.22	6.21	12.14
$\delta_{\varphi\vartheta\psi_{tracker}}$	[deg]	μ	0.95	1.93	4.17
$\delta_{\varphi\vartheta\psi_{est}}$	[deg]	μ	0.80	1.75	3.51
δ_{ω_x}	[deg/s]	μ	0.06	0.11	0.15
δ_{ω_y}	[deg/s]	μ	0.04	0.07	0.11
δ_{ω_z}	[deg/s]	μ	0.10	0.18	0.19
δ_x	[m]	μ	0.008	0.012	0.024
δ_y	[m]	μ	0.150	0.679	0.291
δ_z	[m]	μ	0.009	0.014	0.021
$\delta_{\dot{x}}$	[m/s]	μ	0.000	0.001	0.001
$\delta_{\dot{y}}$	[m/s]	μ	0.003	0.017	0.009
$\delta_{\dot{z}}$	[m/s]	μ	0.001	0.001	0.001

network layers. For what concerns the detection algorithm, a new feature database can be built in less than a hour once that the new model has been estimated, and then it can be updated from ground.

The following Sections provide figures and tables that detail the performance of the navigation solution for CASE A (Sec.5.4.1.1), CASE B (Sec.5.4.1.2), and CASE C (Sec.5.4.1.3), that we discuss in the following.

Figures 5.9, 5.12, and 5.15, show, for each one of the cases, the estimated relative position and velocity in LOF frame. As the approach is performed along the V-bar, y^{LOF} corresponds to the direction of motion, and y^{LOF}_{LOF-ch}, $\dot{y}^{LOF}_{LOF-ch}$, are the variables affected by the biggest estimation error. The pose is acquired at $t_{acq_f} = 145$ s in CASE A and CASE B, and at $t_{acq_f} = 155$ s in CASE C. Before t_{acq_f}, the navigation function is in range-only mode and the dynamic translational filter (i.e., violet dots) uses the raw measurements coming from the ZM-based detection algorithm (i.e., magenta dots). After t_{acq_f}, the 6-DOF pose tracking starts and the dynamic translational filter (i.e., blue dots) uses the measurements coming from the tracker (i.e., cyan dots). The relative range error in percentage is also shown.

Figures 5.11, 5.14, and 5.17, show, for each one of the cases, the estimated absolute attitude and rotation rate of the target with respect to the inertial frame. The plots show the estimated quaternion q_{i-tg} and rotation rate ω^{tg}_{i-tg} from $t = t_{acq_f}$ (i.e., magenta dots), obtained using the dynamic filter and the tracker measurements. When the dynamic filter uses the refined detector measurements to compute the estimate, the state is plotted in violet. When no measurements are available and the filter evolves in open-loop, the state is plotted in cyan. The quaternion measurements obtained during the time acquisition window (i.e., $t_{acq_i} \leq t \leq t_{acq_f}$) are displayed in blue. The green dot indicates the time instant in which the symmetry ambiguity is solved and the estimate is switched to the right solution branch ($t = 176$ s for all the simulations).

Finally, Figures 5.10, 5.13, and 5.16 show the angular error of the estimated attitude. The plot shows the angular error from $t = t_{acq_f}$ (i.e., magenta dots), obtained using the dynamic filter and the tracker measurements. When the dynamic filter uses the refined detector measurements to compute the estimate, the error is plotted in violet. When no measurements are available and the filter evolves in open-loop, the angular error is plotted in cyan. The angular errors obtained during the time acquisition window (i.e., $t_{acq_i} \leq t \leq t_{acq_f}$) are displayed in blue. The green dots indicate the angular error before the ambiguity on the symmetry is solved at $t = 176$ seconds. The angular errors computed before $t = 176$ s (i.e., blue and green dots) have been obtained shifting the estimate of π. Note that the angular error is not always decreasing as the relative distance decreases (e.g., there is a peak around $t = 260$ s). This is due to the fact that the estimation performance does not depend only on the relative distance, but also on the current attitude of the target. In particular, around $t \sim 260$ s, the target solar arrays become invisible (i.e., in Fig.4.3, we have that the camera optical axis is parallel to y_{tg}). We know from Chapter 3 that the particular condition in which some surface of the tracked object passes from being visible to being invisible is particularly challenging for contour-based tracking algorithms, and can lead to the divergence of the algorithm. This also explains why, around $t \sim 270$ s, detector measurements are used.

The performance of the tracker measurements, the detector measurements, and the estimated state as a function of the range are collected for all the simulations. Tables 5.4, 5.6, and 5.8 show the performance of the rotational dynamics estimation. The collected indices are the mean and the RMS of $\delta_{\varphi\vartheta\psi_{detector}}$, $\delta_{\varphi\vartheta\psi_{tracker}}$, and $\delta_{\varphi\vartheta\psi_{est}}$ (the angular error respectively of the detector measurements, of the tracker measurements, and of the dynamic filter estimates), and the mean and RMS of the AKE of the estimated rotation rate $\delta_{\omega_{x_{est}}}$, $\delta_{\omega_{y_{est}}}$, and $\delta_{\omega_{z_{est}}}$. Tables 5.5, 5.7, and 5.9 show the performance of the translational dynamics estimation. The collected indices are the mean and the RMS of the absolute range error (i.e., along y_{LOF}) of the detector measurement, the tracker measurement, and the KF estimate. For the KF estimate, also the mean and the RMS of the absolute range rate error (i.e., named *velocity error estimate* in the tables) are shown.

One should note that the performance of the estimated state achieved in CASE A and CASE B below 10 m is compliant with all the docking and berthing requirements expressed in Sec.1.2.2 (i.e., a navigation error at least two time smaller than the accuracy demanded on the real relative state). However in CASE C the performance is degraded with respect to the other cases, and the estimation errors below 10 m do not assure the margin required to safely control the chaser in docking and berthing operations.

5.4.1.1 Case A: Perfect silhouette images

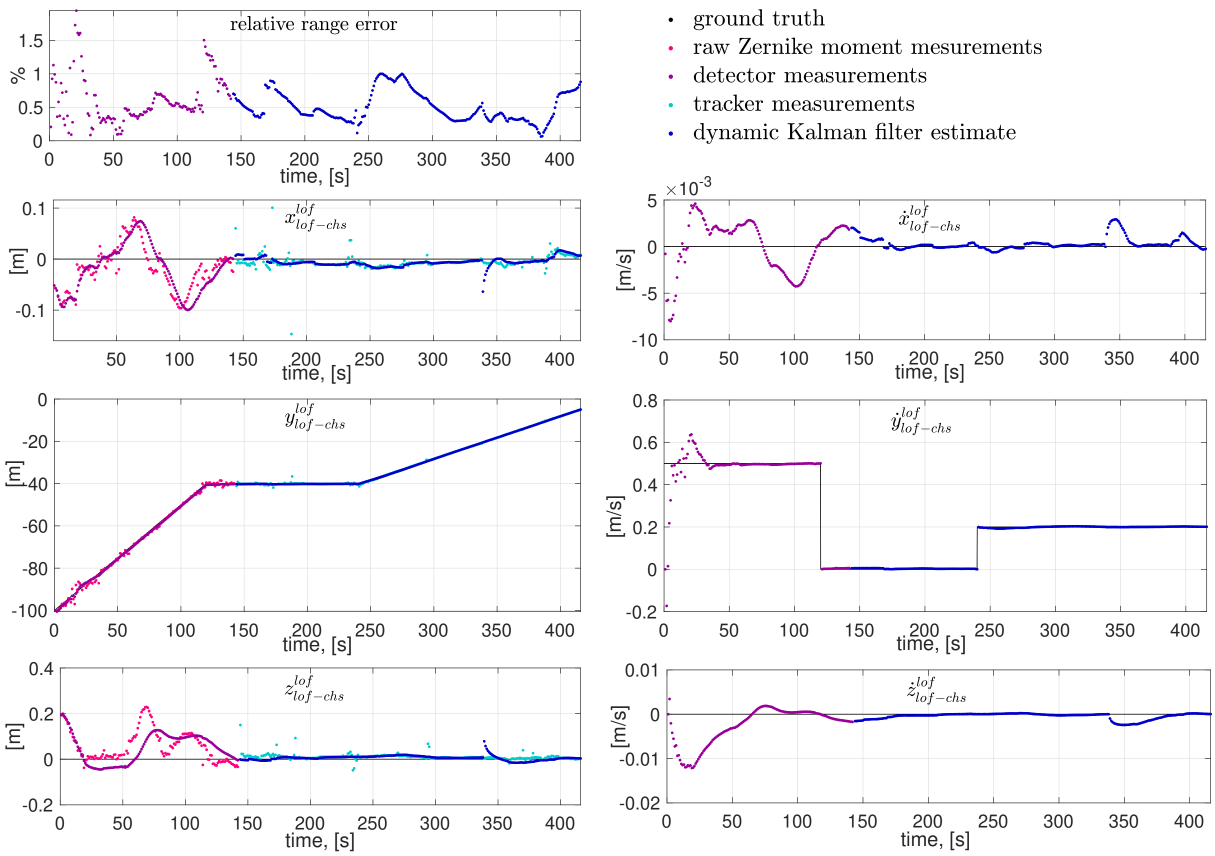


Figure 5.9: Relative target-chaser translational state in LOF, perfect silhouette images

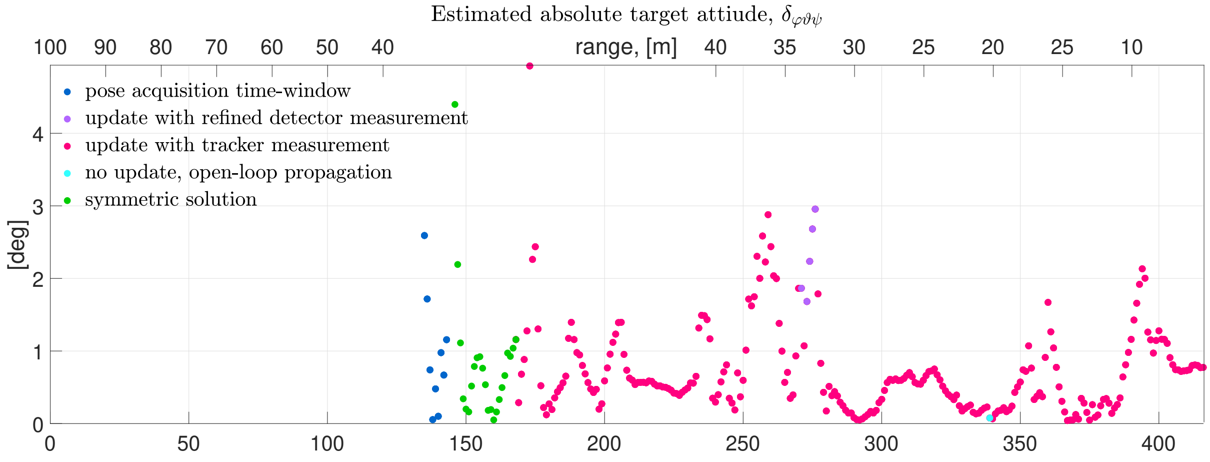


Figure 5.10: Angular error of the estimated target attitude quaternion, perfect silhouette images

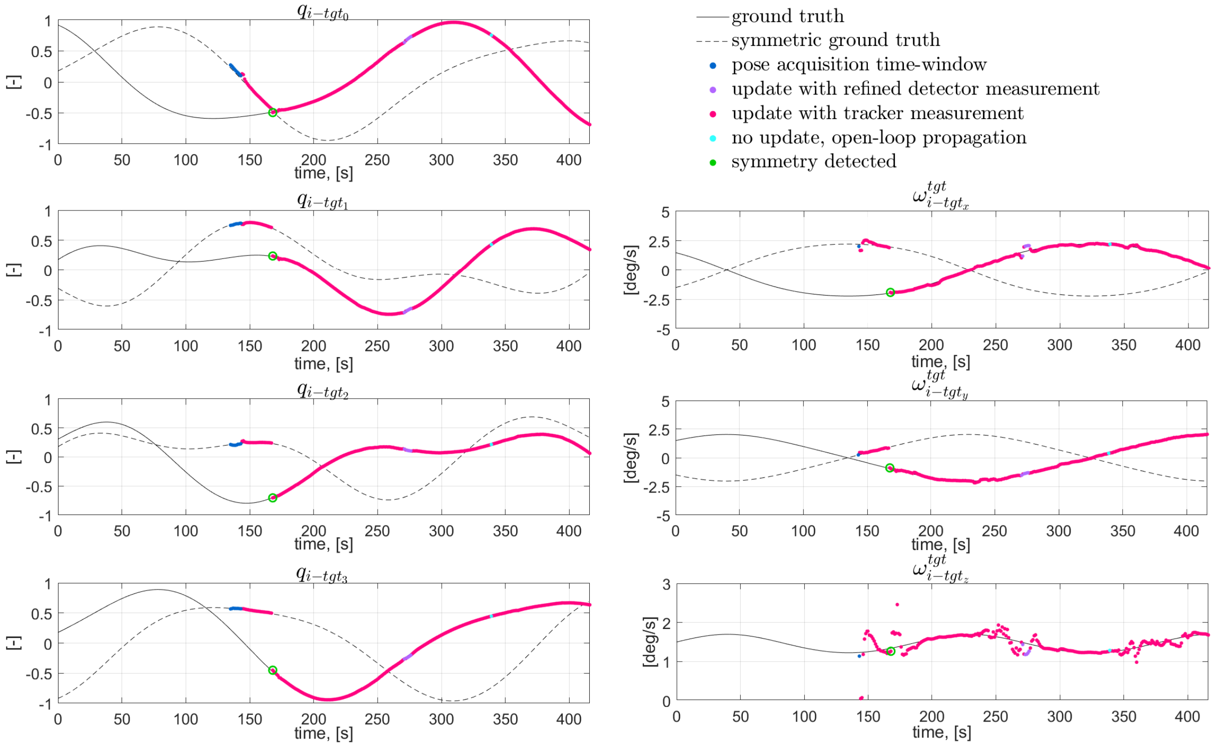


Figure 5.11: Absolute target rotational state, perfect silhouette images

Table 5.4: Performance of the rotational dynamics estimation, perfect silhouette images

			Rotational dynamics					
		range [m]:	= 40	∈]40, 30]	∈]30, 20]	∈]20, 10]	< 10	*overall*
$\delta_{\varphi\vartheta\psi}$ detector	[deg]	μ	2.75	2.27	1.57	1.69	12.97	3.22
		rms	18.11	2.52	1.77	2.21	13.95	11.73
$\delta_{\varphi\vartheta\psi}$ tracker	[deg]	μ	1.07	1.25	0.63	0.61	1.17	0.95
		rms	2.42	1.62	1.19	0.94	1.37	1.78
$\delta_{\varphi\vartheta\psi}$ estimated	[deg]	μ	0.98	1.11	0.39	0.43	1.09	0.80
		rms	1.72	1.40	0.45	0.57	1.17	1.28
δ_{ω_x} estimate	[deg/s]	μ	0.07	0.11	0.04	0.05	0.01	0.06
		rms	0.12	0.18	0.06	0.07	0.02	0.11
δ_{ω_y} estimate	[deg/s]	μ	0.05	0.04	0.03	0.02	0.01	0.04
		rms	0.07	0.05	0.04	0.02	0.02	0.05
δ_{ω_z} estimate	[deg/s]	μ	0.14	0.14	0.01	0.07	0.07	0.10
		rms	0.26	0.18	0.02	0.10	0.09	0.18

Table 5.5: Performance of the translational dynamics estimation, perfect silhouette images

			Translational dynamics					
		range [m]:	= 40	∈]40, 30]	∈]30, 20]	∈]20, 10]	< 10	*overall*
range error detector	[m]	μ	0.060	0.080	0.075	0.030	0.167	0.071
		rms	0.079	0.093	0.076	0.038	0.189	0.092
range error tracker	[m]	μ	0.292	0.289	0.127	0.061	0.050	0.196
		rms	0.499	0.361	0.156	0.084	0.073	0.344
range error estimate	[m]	μ	0.195	0.267	0.107	0.043	0.045	0.150
		rms	0.206	0.279	0.115	0.047	0.046	0.180
velocity error estimate	[m/s]	μ	0.003	0.008	0.002	0.001	0.002	0.003
		rms	0.004	0.029	0.002	0.001	0.002	0.012

5.4.1.2 Case B: Grey-scale images

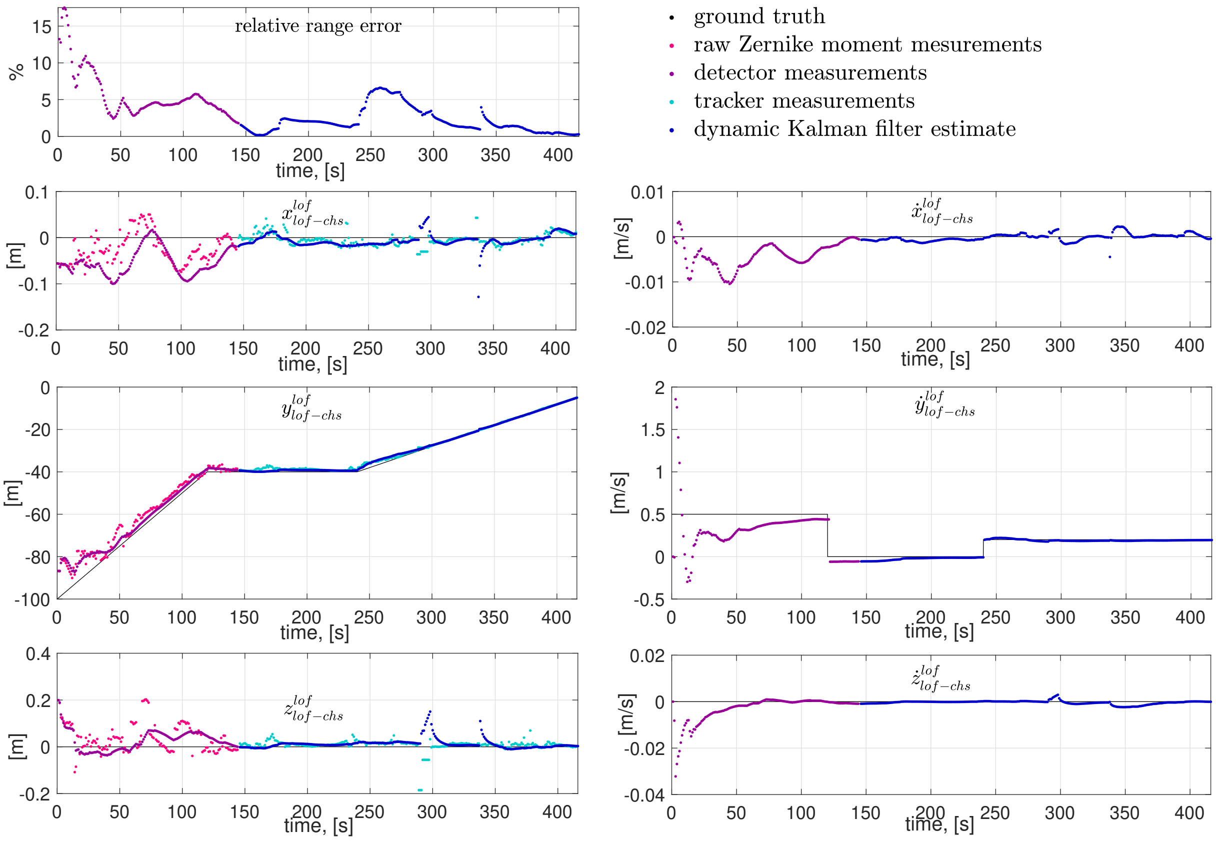


Figure 5.12: Relative target-chaser translational state in LOF, grey-scale images

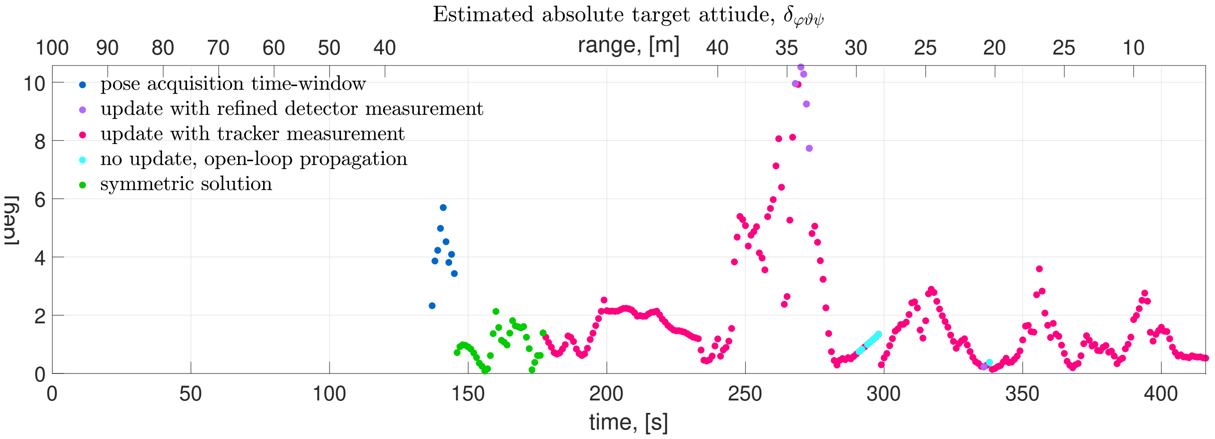


Figure 5.13: Angular error of the estimated target attitude quaternion, grey-scale images

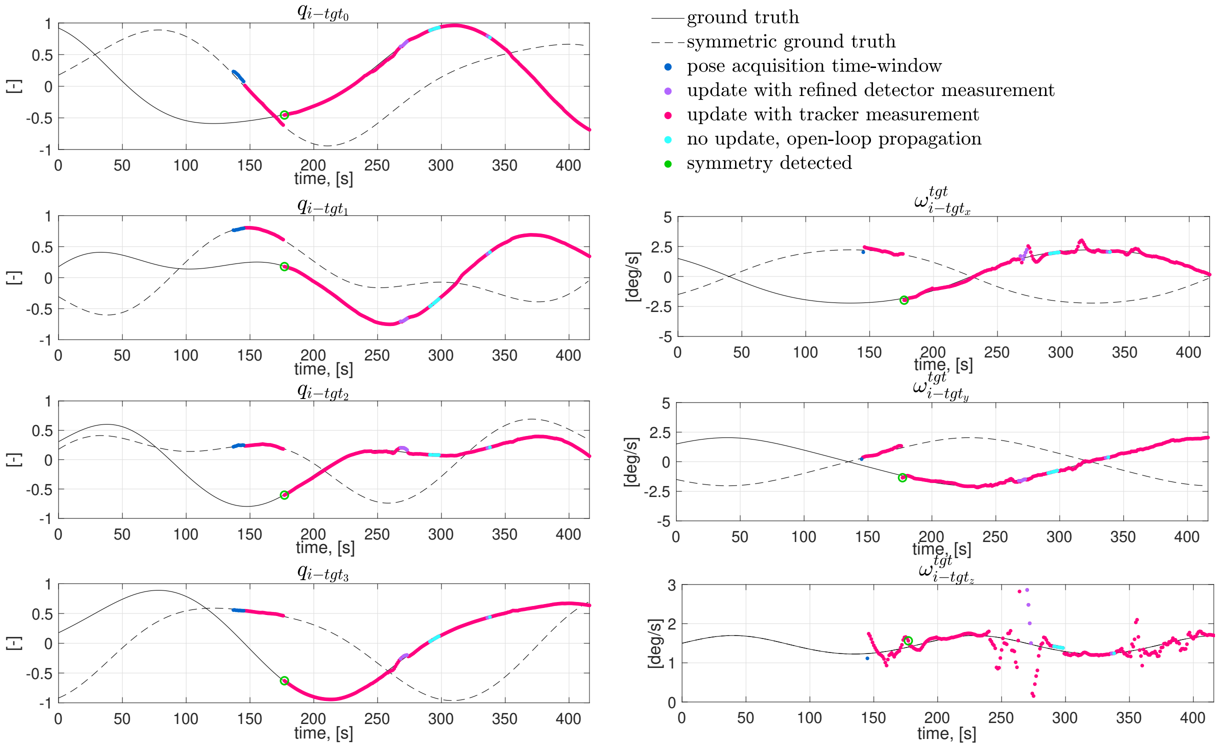


Figure 5.14: Absolute target rotational state, grey-scale images

Table 5.6: Performance of the rotational dynamics estimation, grey-scale images

			Rotational dynamics					
		range [m]:	$= 40$	$\in]40, 30]$	$\in]30, 20]$	$\in]20, 10]$	< 10	*overall*
$\delta_{\varphi\vartheta\psi}$ detector	[deg]	μ	6.49	5.99	5.40	3.37	12.60	6.21
		rms	18.29	6.84	6.37	4.46	13.60	12.44
$\delta_{\varphi\vartheta\psi}$ tracker	[deg]	μ	1.31	3.92	2.24	1.17	1.20	1.93
		rms	1.51	4.68	3.52	1.56	1.47	2.79
$\delta_{\varphi\vartheta\psi}$ estimated	[deg]	μ	1.30	4.10	1.27	1.02	1.19	1.75
		rms	1.44	5.07	1.47	1.23	1.38	2.52
δ_{ω_x} estimate	[deg/s]	μ	0.10	0.18	0.16	0.09	0.02	0.11
		rms	0.12	0.28	0.24	0.13	0.03	0.18
δ_{ω_y} estimate	[deg/s]	μ	0.09	0.07	0.07	0.05	0.02	0.07
		rms	0.11	0.09	0.09	0.06	0.03	0.09
δ_{ω_z} estimate	[deg/s]	μ	0.09	0.57	0.04	0.14	0.09	0.18
		rms	0.13	0.87	0.05	0.21	0.12	0.39

Table 5.7: Performance of the translational dynamics estimation, grey-scale images

			Translational dynamics					
		range [m]:	$= 40$	$\in]40, 30]$	$\in]30, 20]$	$\in]20, 10]$	< 10	*overall*
range error detector	[m]	μ	1.094	1.293	0.931	0.482	0.202	0.902
		rms	1.246	1.312	0.935	0.511	0.217	1.037
range error tracker	[m]	μ	1.168	1.604	0.470	0.236	0.089	0.844
		rms	1.292	1.778	0.535	0.263	0.119	1.111
range error estimate	[m/s]	μ	0.591	1.845	0.507	0.192	0.024	0.679
		rms	0.649	1.903	0.564	0.224	0.029	0.940
velocity error estimate	[m/s]	μ	0.026	0.017	0.012	0.011	0.006	0.017
		rms	0.031	0.032	0.012	0.011	0.006	0.024

5.4.1.3 Case C: Segmented images

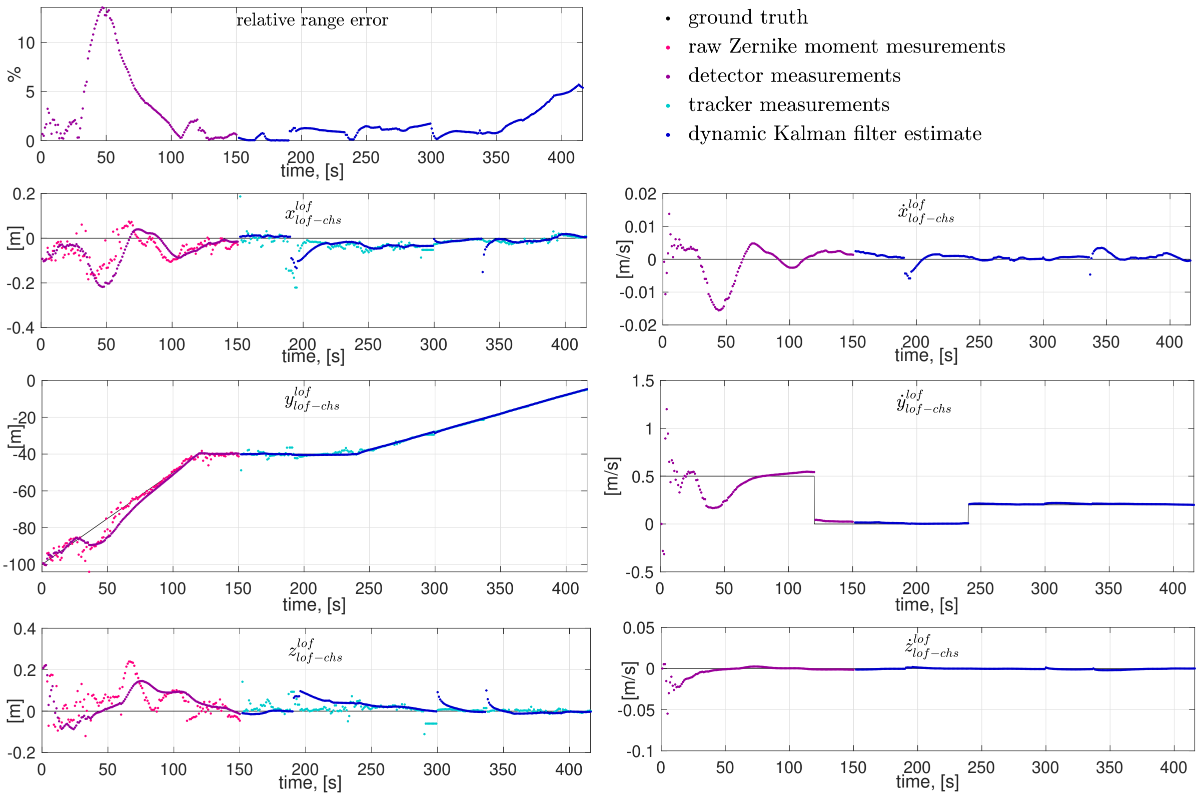


Figure 5.15: Relative target-chaser translational state in LOF, segmented silhouette images

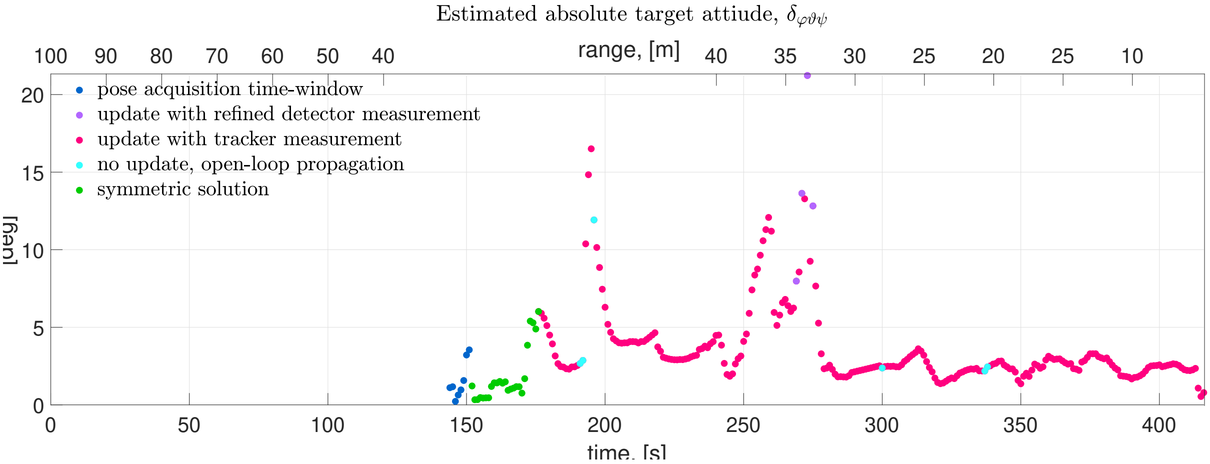


Figure 5.16: Angular error of the estimated target attitude quaternion, segmented silhouette images

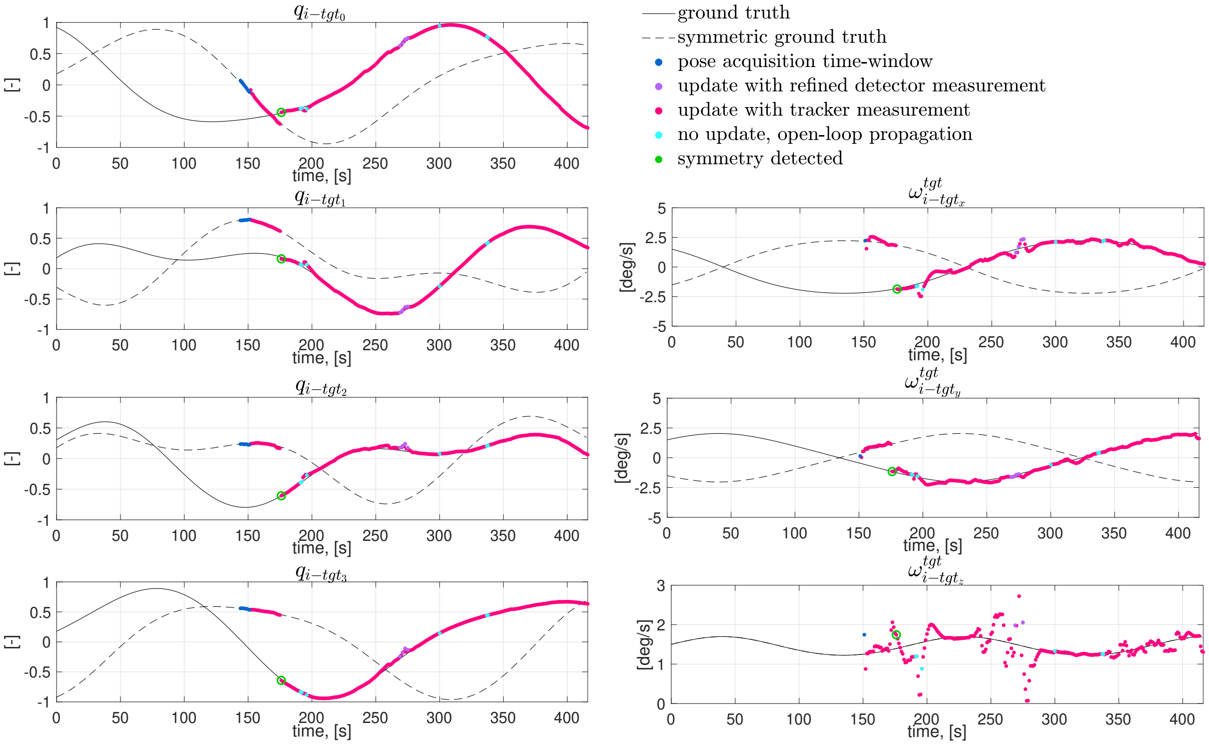


Figure 5.17: Absolute target rotational state, segmented silhouette images

Table 5.8: Performance of the rotational dynamics estimation, segmented silhouette images

			Rotational dynamics					
	range [m]:		$= 40$	$\in]40,\ 30]$	$\in]30,\ 20]$	$\in]20,\ 10]$	< 10	*overall*
$\delta_{\varphi\vartheta\psi}$ detector	[deg]	μ	6.98	37.32	3.61	4.60	12.30	12.14
		rms	18.94	43.04	3.87	5.01	14.78	22.34
$\delta_{\varphi\vartheta\psi}$ tracker	[deg]	μ	4.27	8.22	2.53	2.56	2.29	4.17
		rms	5.67	13.16	2.68	2.60	2.37	6.83
$\delta_{\varphi\vartheta\psi}$ estimated	[deg]	μ	3.76	5.92	2.36	2.52	2.16	3.51
		rms	4.68	7.22	2.42	2.57	2.23	4.48
δ_{ω_x} estimate	[deg/s]	μ	0.23	0.20	0.07	0.11	0.05	0.15
		rms	0.32	0.28	0.09	0.14	0.06	0.23
δ_{ω_y} estimate	[deg/s]	μ	0.17	0.11	0.10	0.04	0.09	0.11
		rms	0.21	0.15	0.14	0.06	0.14	0.16
δ_{ω_z} estimate	[deg/s]	μ	0.20	0.49	0.02	0.09	0.13	0.19
		rms	0.32	0.68	0.020	0.11	0.17	0.36

Table 5.9: Performance of the translational dynamics estimation, segmented silhouette images

			Translational dynamics					
range [m]:			= 40	∈]40, 30]	∈]30, 20]	∈]20, 10]	< 10	*overall*
range error detector	[m]	μ	0.420	1.546	0.292	0.467	0.735	0.648
		rms	0.465	1.607	0.403	0.492	0.754	0.832
range error tracker	[m]	μ	1.022	0.486	0.185	0.228	0.268	0.539
		rms	1.601	0.626	0.287	0.261	0.286	0.986
range error estimate	[m]	μ	0.240	0.422	0.229	0.275	0.366	0.291
		rms	0.307	0.430	0.261	0.288	0.369	0.329
velocity error estimate	[m/s]	μ	0.008	0.012	0.013	0.009	0.003	0.009
		rms	0.009	0.031	0.014	0.009	0.004	0.016

5.4.2 Latency of the algorithms

The segmentation stage has been tested on a 2.20 GHz Intel Core i7 development laptop equipped with a Nvidia GTX 1070 GPU. While real-time performances can be achieved (i.e., 36 millisecond for each image, thus an average of 27.78 images/s on the development laptop), this is not representative of a real world scenario were the processing chain must run on a space embedded target. The process of porting the neural network on a space compatible FPGA, with the objective of achieving at least a working frequency of 1 Hz and ideally 10 Hz, is on-going. There are also possibilities to decrease, if needed, the neural network resource consumption by alleviating it (e.g., reducing the number of layers and/or the number of kernel in the layers) at the cost of its precision, but this could be compensated by efficient post processing able to correct this. Moreover, reducing the size of the image before entering the CNN could largely speed up the processing with acceptable precision trade-off.

The pose estimation algorithms were tested on a 2.70 GHz Intel Core i7 processor. All the algorithms are autocoded in C++ from MATLAB code. The average time needed to compute ZM invariants up to the 9^{th} order is equal to 208 millisecond, 387 milliseconds and 1.16 seconds respectively at 40 m, 20 m, and 10 m. The latency of the algorithm at 10 m, as well as its computational load (i.e., due to the number of silhouette points to process in order to compute the rotation invariants), is not compatible with real-time implementation. However, this computation time can be drastically reduced resizing the image before computing the moment invariants. The results shown in Sec.4.5.2.3 prove that the method is robust to a reduction of the size of the test images. Note however that this absolute computation time is mostly indicative, since the computation of moment invariants can be optimized [Prata and Rusch, 1989, Hwang and Kim, 2006, Wee et al., 2004], which could lead to significant improvements. The time needed to compute the moment invariants is the dominant part of the detection algorithm: with a database of the size considered in this study, the matching requires less than 0.1% of the time needed to compute the 9^{th} order ZM invariants at 20 m.

On the same processor, the tracking algorithm (from the Canny edge extraction to the cost function minimization), has an average latency of 60.1 millisecond at 40 m (i.e., 16.64 images/s), $62,5$ millisecond at 20 m (i.e., 16 images/s), and 85.8 millisecond par image (i.e., 11.66 images/s) at 10 m.

In the pose acquisition phase, at each time step, the latency will be the summation of the time needed for the segmentation and the time needed for the computation of the ZM invariants. The latency needed to compute the angular distances between two consecutive measurements and eventually correct the ψ angle is almost negligible. At the time instant where $cond_1$ and $cond_2$ are satisfied, the latency of the pose acquisition function must take into account also the execution time of the contour-based tracking algorithm. If the symmetry must be detected, the latency of the tracker must be counted twice. The filter initialization function has a latency that can be neglected with respect to the total latency of the pose acquisition function. During the 6-DOF tracking, if the detector and the tracker are executed in parallel (which is possible since they are independent one from the other), the latency is equal to the time needed by the segmentation plus the highest latency among the tracker and the detector. However, if the tracker measurement isn't accepted, the detector measurement must be corrected using the tracker and the latency of the tracker is therefore doubled.

The performance shown in the previous paragraph has been obtained in presence of a very high relative dynamics (i.e., very high approach velocity of the chaser, and high rotation rate of the target), which is very unlikely to happen in a real-world scenario. This allows us to state that if, after implementation of the algorithm on a space-qualified processor, the working frequency of 1 Hz is not reached, the acquisition rate could be slightly decreased without any major degradation of the estimation performance.

5.5 Conclusion

The current chapter has addressed the problem of monocular model-based pose estimation for close-proximity operations during space rendezvous using a visible monocular camera. A complete navigation solution, covering all the navigation phases from the 6 degrees-of-freedom pose acquisition to the robust full pose tracking, has been presented and tested in different scenarios. The pose estimation relies on a contour-based recursive tracker which provides very precise measurements, and on a global-feature based detection algorithm which allows pose initialization and measurement correction in case of divergence of the tracker. A background subtraction algorithm based on convolutional neural network has been integrated to the navigation algorithms. The method enables the extraction of the target silhouette from the acquired image, allowing to reach satisfying estimation performance even in presence of complex background such as the Earth. The segmentation algorithm would certainly benefit from the use of multi-spectral images, especially in the thermal infrared spectrum, and from the increase of the training database size. With the implementation of this improvements, the estimation performance in presence of complex background is likely to get closer to the baseline performance with perfect silhouette images.

The proposed estimation chain provides a cost effective solution with a relatively low computational load. The latency of the navigation functions on standard PCs allows to assume that, after the implementation of the algorithms on a dedicated hardware such as a FPGA, the navigation chain will satisfy the requirement of 1 Hz acquisition rate. Future works will focus on the porting of the navigation algorithms on a space-qualified hardware.

Chapter 6

Conclusion

This thesis went through the development of a navigation solution for autonomous space rendezvous with non-cooperative target based on the use of a monocular visible camera.

After an introduction on the operational context of space rendezvous in Chapter 1, the problem of incorporating delayed and multi-rate measurements in a navigation filter for the estimation of the dynamics of a non-cooperative target has been assessed in Chapter 2. A dynamic filter for the estimation of full target rotational and translational state exploiting relative pose measurement has been formalized. Two delay management techniques have been compared: Larsen's method, which provides a fast but sub-optimal solution, and Filter Recalculation method, which always provides the optimal estimate but has a higher computational load. The Monte Carlo validation campaign has shown that Larsen's method performance is comparable to Filter Recalculation method performance. The latter shows remarkably better performance only in the transient phase of simulations exploiting interim measurements but at the expense of a higher computational and storage need. When a delayed measurement arrives, Filter Recalculation method computational load is multiplied by a factor equal to the number of delay samples, which might be incompatible with the critical applications run by the on-board computer for this particular time step. This suggests that, in applications where the on-board resources are limited, Larsen's method is preferable since it provides a faster estimation without any significant degradation of the steady-state performance.

Then, in Chapter 3, the problem of monocular model-based pose tracking for space rendezvous has been assessed, and an innovative method relying on silhouette-edges extraction and tracking has been proposed. The method has been formally developed and implemented. The coupling with the dynamic filter introduced in Chapter 2 enables the robust estimation of the relative pose, as well as the estimation of the translational velocity and rotation rate, of the target. The method has shown high performance and robustness when using perfect silhouette images, and suffered a loss of robustness at high relative distances when tested in closed-loop on realistic images affected by shadows, blur and reflections. Indeed, the use of a dynamic calibration of the KF improves the performance. Different solutions to strengthen the estimation have been proposed, such as the coupling of the visible camera with a thermal infra-red camera in order to improve the silhouette extraction. The proposed estimation chain provides a cost effective solution with a very low computational load, thus compatible with typical space processing capabilities. However, as all the recursive pose tracker, the proposed method does not allow to initialize the pose from the so-called *lost in space* condition (i.e., no prior information of the target pose), neither it is capable of recovering from big pose estimation error.

For this reason, we have proposed in Chapter 4 a template matching method to estimate the pose of a non-cooperative target during space rendezvous from a single binary image capturing the target's silhouette. The method is suitable for initial pose acquisition and for detecting faults and deviations in other on-board trackers, and represents a novel approach for the pose estimation of a spacecraft in a rendezvous. Three types of global descriptors, based on complex moments, Zernike moments, and Fourier descriptors, are introduced and compared in order to match the silhouette in a database of pose-dependent feature vectors generated offline. By exploiting the scale and rotation invariance of these descriptors, the approach requires discretizing only two pose angles to construct the database, leading to fast computation times appropriate for real-time implementations. Our performance analysis shows that Zernike moment invariants provide the highest accuracy and robustness in off-nominal conditions. Fourier descriptors show comparable performance with a much lower computational cost, but only on for high resolution images or short target distances. This suggests that these two types of descriptors are complementary and could be used in combination.

Finally, in Chapter 5, the algorithms developed in the previous chapters have been merged in order to develop a complete navigation solution, covering all the navigation phases from the 6 degrees-of-freedom pose acquisition to the robust full pose tracking. The pose estimation relies on the contour-based recursive tracker of Chap.3, which provides very precise measurements, and on the global-feature based detection algorithm of Chap.4, which allows pose initialization and measurement correction in case of divergence of the tracker. A method to post-process the outputs of the detection algorithm to eliminate outliers and provide a robust pose acquisition has been developed. A background subtraction algorithm based on convolutional neural network has been integrated to the navigation algorithms. The method enables the extraction of the target silhouette from the acquired image, allowing to reach satisfying estimation performance even in presence of complex background such as the Earth. The segmentation algorithm would certainly benefit from the use of multi-spectral images, especially in the thermal infrared spectrum, and from the increase of the training database size. With the implementation of this improvements, the estimation performance in presence of complex background is likely to get closer to the baseline performance with perfect silhouette images.

The proposed estimation chain provides a cost effective solution with a relatively low computational load. The latency of the navigation functions on standard PCs allows to assume that, after the implementation of the algorithms on a dedicated hardware such as a FPGA, the navigation chain will satisfy the requirement of 1 Hz acquisition rate.

Future works in Thales Alenia Space will go towards three main directions:

- Improvement of the background subtraction technique
- Porting of the navigation solution on FPGA
- Test campaign on the Thales Alenia Space robotic test-bench in Cannes

APPENDICES

Appendix A

Kinematics of rotations

Contents

A.1 Convention for rotations

In order to use an unambiguous notation in the whole thesis, in this section we detail the convention and operators that will be used to describe rotations and vectors in different reference frames.

Rotations are described by quaternions, according to Hamilton convention (see [Sola, 2015]). A quaternion q is a four-component object with a three-vector part q_v and a scalar part q_0:

$$q = \begin{bmatrix} q_0 \\ q_v \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{A.1}$$

It is used to define a rotation of a angle α around the vector q_v normalized, such that:

$$q = \begin{bmatrix} q_0 \\ q_v \end{bmatrix} = \begin{bmatrix} \cos\left(\dfrac{\alpha}{2}\right) \\ \dfrac{q_v}{|q_v|}\sin\left(\dfrac{\alpha}{2}\right) \end{bmatrix} \tag{A.2}$$

The correspondence between the quaternion q and the rotation matrix R that corresponds to the same rotation is described by the following equation:

$$R(q) = (q_0^2 - q_v^T q_v) I_{3\times3} + 2q_v q_v^T + 2q_0 [q_v]_\times \tag{A.3}$$

where $I_{3\times3}$ is the 3×3 identity matrix. The notation $[v]_\times$ for a given vector $v = [v_1,\ v_2,\ v_3]^T$ indicates the skew-symmetric cross-product matrix equal to:

$$[v]_\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}, \quad \text{and therefore} \quad [q_v]_\text{x} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \tag{A.4}$$

while the relation between the quaternion components and the corresponding Euler angles is given by:

$$\begin{cases} \varphi & \text{atan2}\left(\dfrac{2q_2q_3 + 2q_0q_1}{1 - 2(q_1^2 + q_2^2)}\right) \\ \vartheta & \text{asin}\Big(2q_0q_2 - 2q_1q_3\Big) \\ \psi & \text{atan2}\left(\dfrac{2q_1q_2 + 2q_0q_3}{1 - 2(q_2^2 + q_3^2)}\right) \end{cases} \tag{A.5}$$

and

$$\begin{cases} q_0 = & \cos\left(\dfrac{\varphi}{2}\right)\cos\left(\dfrac{\vartheta}{2}\right)\cos\left(\dfrac{\psi}{2}\right) + \sin\left(\dfrac{\varphi}{2}\right)\sin\left(\dfrac{\vartheta}{2}\right)\sin\left(\dfrac{\psi}{2}\right) \\ q_1 = & \sin\left(\dfrac{\varphi}{2}\right)\cos\left(\dfrac{\vartheta}{2}\right)\cos\left(\dfrac{\psi}{2}\right) - \cos\left(\dfrac{\varphi}{2}\right)\sin\left(\dfrac{\vartheta}{2}\right)\sin\left(\dfrac{\psi}{2}\right) \\ q_2 = & \cos\left(\dfrac{\varphi}{2}\right)\sin\left(\dfrac{\vartheta}{2}\right)\cos\left(\dfrac{\psi}{2}\right) + \sin\left(\dfrac{\varphi}{2}\right)\cos\left(\dfrac{\vartheta}{2}\right)\sin\left(\dfrac{\psi}{2}\right) \\ q_3 = & \cos\left(\dfrac{\varphi}{2}\right)\cos\left(\dfrac{\vartheta}{2}\right)\sin\left(\dfrac{\psi}{2}\right) - \sin\left(\dfrac{\varphi}{2}\right)\sin\left(\dfrac{\vartheta}{2}\right)\cos\left(\dfrac{\psi}{2}\right) \end{cases} \tag{A.6}$$

The Hamiltonian convention for quaternions is also referred to as "*local to global*". Given a vector x^a expressed in reference frame a, and the same vector expressed in the reference frame b (i.e., x^b), the following relation allows to pass from one frame to the other:

$$x^a = q_{a-b} \otimes x^b \otimes q_{a-b}^* \tag{A.7}$$

(where q^* is the quaternion conjugate $q^* = [q_0,\ -q_1,\ -q_2,\ -q_3]^T$), and $\otimes$ indicates the quaternion product such that:

$$q_a \otimes q_b = \begin{bmatrix} q_{a_0}q_{b_0} - q_{a_1}q_{b_1} - q_{a_2}q_{b_2} - q_{a_3}q_{b_3} \\ q_{a_0}q_{b_1} + q_{a_1}q_{b_0} + q_{a_2}q_{b_3} - q_{a_3}q_{b_2} \\ q_{a_0}q_{b_2} - q_{a_1}q_{b_3} + q_{a_2}q_{b_0} + q_{a_3}q_{b_1} \\ q_{a_0}q_{b_3} + q_{a_1}q_{b_2} - q_{a_2}q_{b_1} + q_{a_3}q_{b_0} \end{bmatrix} \tag{A.8}$$

To rotate a 3 components vector x using quaternions formalism, a dimension is added to transform the vector into a quaternion with null scalar part $q_0 = 0$ and with $q_v = x$. For sake of simplicity, we will not change the notation of vectors when used in quaternion products. The same rotation can be described using the rotation matrix R_{a-b}

$$x^a = R_{a-b} x^b \tag{A.9}$$

Finally, the derivative of an attitude quaternion and a rotation matrix must be introduced. Given the rotation rate ω_{a-b} corresponding to the attitude quaternion q_{a-b}, the rotation rate can be expressed either in the global (a) reference frame (ω^a_{a-b}) or in the local (b) reference frame (ω^b_{a-b}). The quaternion derivative will be:

$$\begin{aligned} \dot{q}_{a-b} &= \frac{1}{2} q_{a-b} \otimes \omega^b_{a-b} \\ &= \frac{1}{2}\, \omega^a_{a-b} \otimes q_{a-b} \end{aligned} \tag{A.10}$$

while, reminding the definition of the cross-product matrix $[\,]_\times$, the derivative of the rotation matrix will be:

$$\begin{aligned} \dot{R}_{a-b} &= R_{a-b} [\omega^b_{a-b}]_\times \\ &= [\omega^a_{a-b}]_\times R_{a-b} \end{aligned} \tag{A.11}$$

A.2 Jacobian of quaternion operations

In the following section the analytic computation of the Jacobians of a series of operation such as the computation of the quaternion conjugate, the quaternion product and the rotation of a vector from one frame to another will be presented.

A.2.1 Quaternion product

Let's denote the quaternion product $q_a \otimes q_b$ with the function $Q_{prod}(q_a, q_b)$:

$$q_a \otimes q_b = \begin{bmatrix} q_{a0} q_{b0} - q_{a1} q_{b1} - q_{a2} q_{b2} - q_{a3} q_{b3} \\ q_{a0} q_{b1} + q_{a1} q_{b0} + q_{a2} q_{b3} - q_{a3} q_{b2} \\ q_{a0} q_{b2} - q_{a1} q_{b3} + q_{a2} q_{b0} + q_{a3} q_{b1} \\ q_{a0} q_{b3} + q_{a1} q_{b2} - q_{a2} q_{b1} + q_{a3} q_{b0} \end{bmatrix} \tag{A.12}$$

The Jacobians are:

$$\frac{\partial (q_a \otimes q_b)}{\partial q_a} = \mathbb{J}_{Q_{prod}(q_a, q_b)/q_a} = \begin{bmatrix} q_{b0} & -q_{b1} & -q_{b2} & -q_{b3} \\ q_{b1} & q_{b0} & q_{b3} & -q_{b2} \\ q_{b2} & -q_{b3} & q_{b0} & q_{b1} \\ q_{b3} & q_{b2} & -q_{b1} & q_{b0} \end{bmatrix} \tag{A.13}$$

$$\frac{\partial(q_a \otimes q_b)}{\partial q_b} = \mathbb{J}_{Q_{prod}(q_a,q_b)/q_b} = \begin{bmatrix} q_{a_0} & -q_{a_1} & -q_{a_2} & -q_{a_3} \\ q_{a_1} & q_{a_0} & -q_{a_3} & q_{a_2} \\ q_{a_2} & q_{a_3} & q_{a_0} & -q_{a_1} \\ q_{a_3} & -q_{a_2} & q_{a_1} & q_{a_0} \end{bmatrix} \tag{A.14}$$

A.2.2 Quaternion conjugate

Let's denote the quaternion conjugate q^* of a quaternion q with the function $Q_{conj}(q)$:

$$q^* = \begin{bmatrix} q_0 \\ -q_1 \\ -q_2 \\ -q_3 \end{bmatrix} \tag{A.15}$$

The Jacobian is:

$$\frac{\partial q^*}{\partial q} = \mathbb{J}_{Q_{conj}(q)/q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{A.16}$$

A.2.3 Rotation of a vector

Let's denote the rotation of a vector ω with respect to a quaternion q with the function $Q_{rot}(\omega, q)$ such that:

$$Q_{rot}(w, q) = q \otimes \begin{bmatrix} 0 \\ \omega \end{bmatrix} \otimes q^* \tag{A.17}$$

Let's now call $w = [0, \omega]^T = [0, \omega_1, \omega_2, \omega_3]^T$ and $(q \otimes w) = qw$:

$$Q_{rot}(w, q) = q \otimes w \otimes q^* = qw \otimes q^* \tag{A.18}$$

The Jacobian, with respect to q and with respect to ω, can be written as:

$$\frac{\partial(q \otimes w \otimes q^*)}{\partial q} = \mathbb{J}_{Q_{rot}(\omega,q)/q} = \frac{\partial(qw \otimes q^*)}{\partial qw}\frac{\partial(q \otimes w)}{\partial q} + \frac{\partial(qw \otimes q^*)}{\partial q^*}\frac{\partial(q^*)}{\partial q} \tag{A.19}$$

$$\frac{\partial(q \otimes w \otimes q^*)}{\partial \omega} = \mathbb{J}_{Q_{rot}(\omega,q)/\omega} = \frac{\partial(qw \otimes q^*)}{\partial qw}\frac{\partial(q \otimes w)}{\partial \omega} \tag{A.20}$$

Let's now focus on the computation of $\mathbb{J}_{Q_{rot}(\omega,q)/q}$. Using the formula in (A.13) and the definition of quaternion conjugate in (A.15), the first quantities are computed:

$$\frac{\partial(qw \otimes q^*)}{\partial qw} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{A.21}$$

$$\frac{\partial(qw)}{\partial q} = \frac{\partial(q \otimes w)}{\partial q} = \begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{A.22}$$

The first part of the Jacobian will be therefore:

$$\frac{\partial(qw \otimes q^*)}{\partial qw}\frac{\partial(qw)}{\partial q} =$$
$$= \begin{bmatrix} q_1\omega_1 + q_2\omega_2 + q_3\omega_3 & -q_0\omega_1 + q_3\omega_2 - q_2\omega_3 & -q_3\omega_1 - q_0\omega_2 + q_1\omega_3 & q_2\omega_1 - q_1\omega_2 - q_0\omega_3 \\ q_0\omega_1 - q_3\omega_2 + q_2\omega_3 & q_1\omega_1 + q_2\omega_2 + q_3\omega_3 & -q_2\omega_1 + q_1\omega_2 + q_0\omega_3 & -q_3\omega_1 - q_0\omega_2 + q_1\omega_3 \\ q_3\omega_1 + q_0\omega_2 - q_1\omega_3 & q_2\omega_1 - q_1\omega_2 - q_0\omega_3 & q_1\omega_1 + q_2\omega_2 + q_3\omega_3 & q_0\omega_1 - q_3\omega_2 + q_2\omega_3 \\ -q_2\omega_1 + q_1\omega_2 + q_0\omega_3 & q_3\omega_1 + q_0\omega_2 - q_1\omega_3 & -q_0\omega_1 + q_3\omega_2 - q_2\omega_3 & q_1\omega_1 + q_2\omega_2 + q_3\omega_3 \end{bmatrix} \tag{A.23}$$

Let's now compute the second part of the Jacobian. Using (A.14):

$$\frac{\partial(qw \otimes q^*)}{\partial q^*} = \begin{bmatrix} qw_0 & -qw_1 & -qw_2 & -qw_3 \\ qw_1 & qw_0 & -qw_3 & qw_2 \\ qw_2 & qw_3 & qw_0 & -qw_1 \\ qw_3 & -qw_2 & qw_1 & qw_0 \end{bmatrix} \tag{A.24}$$

where, using (A.12):

$$qw = q \otimes w = \begin{bmatrix} -q_1\omega_1 - q_2\omega_2 - q_3\omega_3 \\ q_0\omega_1 - q_3\omega_2 + q_2\omega_3 \\ q_3\omega_1 + q_0\omega_2 - q_1\omega_3 \\ -q_2\omega_1 + q_1\omega_2 + q_0\omega_3 \end{bmatrix} \tag{A.25}$$

Reminding (A.15), the second part of the Jacobian will be:

$$\frac{\partial(qw \otimes q^*)}{\partial q^*}\frac{\partial(q^*)}{\partial q} =$$
$$= \begin{bmatrix} -q_1\omega_1 - q_2\omega_2 - q_3\omega_3 & +q_0\omega_1 - q_3\omega_2 + q_2\omega_3 & +q_3\omega_1 + q_0\omega_2 - q_1\omega_3 & -q_2\omega_1 + q_1\omega_2 + q_0\omega_3 \\ q_0\omega_1 - q_3\omega_2 + q_2\omega_3 & q_1\omega_1 + q_2\omega_2 + q_3\omega_3 & -q_2\omega_1 + q_1\omega_2 + q_0\omega_3 & -q_3\omega_1 - q_0\omega_2 + q_1\omega_3 \\ q_3\omega_1 + q_0\omega_2 - q_1\omega_3 & q_2\omega_1 - q_1\omega_2 - q_0\omega_3 & q_1\omega_1 + q_2\omega_2 + q_3\omega_3 & q_0\omega_1 - q_3\omega_2 + q_2\omega_3 \\ -q_2\omega_1 + q_1\omega_2 + q_0\omega_3 & q_3\omega_1 + q_0\omega_2 - q_1\omega_3 & -q_0\omega_1 + q_3\omega_2 - q_2\omega_3 & q_1\omega_1 + q_2\omega_2 + q_3\omega_3 \end{bmatrix} \tag{A.26}$$

It can be noted that the first line of $\dfrac{\partial(qw \otimes q^*)}{\partial qw}\dfrac{\partial(qw)}{\partial q}$ is opposite to the first line of $\dfrac{\partial(qw \otimes q^*)}{\partial q^*}\dfrac{\partial(q^*)}{\partial q}$, while the other lines are equals. Defining the following quantities:

$$\begin{aligned} A &= 2(q_0\omega_1 - q_3\omega_2 + q_2\omega_3) \\ B &= 2(q_3\omega_1 + q_0\omega_2 - q_1\omega_3) \\ C &= 2(-q_2\omega_1 + q_1\omega_2 + q_0\omega_3) \\ D &= 2(q_1\omega_1 + q_2\omega_2 + q_3\omega_3) \end{aligned} \tag{A.27}$$

the expression of the Jacobian becomes:

$$\frac{\partial(q \otimes w \otimes q^*)}{\partial q} = \frac{\partial(qw \otimes q^*)}{\partial qw}\frac{\partial(q \otimes w)}{\partial q} + \frac{\partial(qw \otimes q^*)}{\partial q^*}\frac{\partial(q^*)}{\partial q} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ A & D & C & -B \\ B & -C & D & A \\ C & B & -A & D \end{bmatrix} \tag{A.28}$$

Then the dimension must be reset to a matrix 3 × 4 by eliminating the first line, since the quantity for which we are calculating the Jacobian is a 3 element vector (the rotated ω). Introducing the notation $Q_{rot_j}(\omega, q)$, $j = 1 : 3$ in order to refer to the first, the second and the third component of the vector resulting from the rotation described by the function $Q_{rot}(\omega, q)$, the Jacobian can be written as:

$$\frac{\partial Q_{rot}(\omega, q)}{\partial q} = \mathbb{J}_{Q_{rot}(\omega,q)/q} = \begin{bmatrix} \dfrac{\partial Q_{rot_1}(\omega, q)}{\partial q} \\ \dfrac{\partial Q_{rot_2}(\omega, q)}{\partial q} \\ \dfrac{\partial Q_{rot_3}(\omega, q)}{\partial q} \end{bmatrix} = \begin{bmatrix} \mathbb{J}_{Q_{rot_1}(\omega,q)-q} \\ \mathbb{J}_{Q_{rot_2}(\omega,q)-q} \\ \mathbb{J}_{Q_{rot_3}(\omega,q)-q} \end{bmatrix} \tag{A.29}$$

where $\mathbb{J}_{Q_{rot_j}(\omega,q)/q}$ is the 1 × 4 vector:

$$\begin{cases} \mathbb{J}_{Q_{rot_1}(\omega,q)/q} &= \begin{bmatrix} A & D & C & -B \end{bmatrix} = \begin{bmatrix} \dfrac{\partial Q_{rot_1}(\omega, q)}{\partial q_0} & \dfrac{\partial Q_{rot_1}(\omega, q)}{\partial q_1} & \dfrac{\partial Q_{rot_1}(\omega, q)}{\partial q_2} & \dfrac{\partial Q_{rot_1}(\omega, q)}{\partial q_3} \end{bmatrix} \\ \mathbb{J}_{Q_{rot_2}(\omega,q)/q} &= \begin{bmatrix} B & -C & D & A \end{bmatrix} = \begin{bmatrix} \dfrac{\partial Q_{rot_2}(\omega, q)}{\partial q_0} & \dfrac{\partial Q_{rot_2}(\omega, q)}{\partial q_1} & \dfrac{\partial Q_{rot_2}(\omega, q)}{\partial q_2} & \dfrac{\partial Q_{rot_2}(\omega, q)}{\partial q_3} \end{bmatrix} \\ \mathbb{J}_{Q_{rot_3}(\omega,q)/q} &= \begin{bmatrix} C & B & -A & D \end{bmatrix} = \begin{bmatrix} \dfrac{\partial Q_{rot_3}(\omega, q)}{\partial q_0} & \dfrac{\partial Q_{rot_3}(\omega, q)}{\partial q_1} & \dfrac{\partial Q_{rot_3}(\omega, q)}{\partial q_2} & \dfrac{\partial Q_{rot_3}(\omega, q)}{\partial q_3} \end{bmatrix} \end{cases} \tag{A.30}$$

Let's now focus on the Jacobian with respect to w. Using Eq.(A.13) and (A.15):

$$\frac{\partial(qw \otimes q^*)}{\partial qw} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \\ -q_1 & q_0 & -q_3 & q_2 \\ -q_2 & q_3 & q_0 & -q_1 \\ -q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{A.31}$$

and, using Eq.(A.14)

$$\frac{\partial(q \otimes w)}{\partial w} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{A.32}$$

therefore:

$$\begin{aligned} \frac{\partial(q \otimes w \otimes q^*)}{\partial w} &= \frac{\partial(qw \otimes q^*)}{\partial qw}\frac{\partial(q \otimes w)}{\partial w} \\ &= \begin{bmatrix} q_0^2+q_1^2+q_2^2+q_3^2 & 0 & 0 & 0 \\ 0 & q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 0 & 2q_1q_2+2q_0q_3 & q_0^2-q_1^2+q_2^2-q_3^2 & 2q_3q_2-2q_0q_1 \\ 0 & 2q_1q_3-2q_0q_2 & 2q_3q_2-2q_0q_1 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \end{aligned} \tag{A.33}$$

This matrix actually must be reduced eliminating the first line and the firs column, which correspond to the null scalar component added to obtain the four-element w from the three-element ω.

$$\begin{aligned} \frac{\partial(q \otimes w \otimes q^*)}{\partial \omega} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \frac{\partial(qw \otimes q^*)}{\partial qw} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ 2q_1q_2+2q_0q_3 & q_0^2-q_1^2+q_2^2-q_3^2 & 2q_3q_2-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_3q_2-2q_0q_1 & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \end{aligned} \tag{A.34}$$

Appendix B

Kalman filtering theory

Contents

This appendix provides an overview of the theory underlying optimal filtering using Kalman filter (KF). Several formulations and extension of Kalman filter exist, depending on whether the filter is applied to a linear or a non-linear system, a time-discrete or a time-continuous system. In Sec.B.1, the derivation of the KF for a linear system, both in the time-continuous and in the time-discrete formulation, is presented. Moreover, the theory underlying the passage from the time-continuous to the time-discrete formulation is discussed. In Sec.B.2, the extension to the estimation of a non-linear time-continuous system with time-discrete measurement is presented.

B.1 Kalman filtering for linear systems

B.1.1 The Continuous Kalman Filter

Let's take a process $x(t)$ and a measurement $y(t)$ whose dynamics is governed by the following linear equations:

$$\begin{cases} \dot{x}(t) &= A(t)x(t) + B(t)u(t) + G(t)w(t) \\ y(t) &= C(t)x(t) + v(t) \end{cases} \tag{B.1}$$

where $u(t)$ is a deterministic quantity (i.e., the input), $w(t)$ and $v(t)$ are respectively the process and the measurement noise. These two quantities are two uncorrelated zero mean Gaussian processes whose covariances Q and R are defined according to:

$$\begin{cases} \mathbb{E}\left[w(t)w^T(\tau)\right] & = Q(t)\delta(t-\tau) \\ \mathbb{E}\left[v(t)v^T(\tau)\right] & = R(t)\delta(t-\tau) \\ \mathbb{E}\left[v(t)w^T(\tau)\right] & = 0 \end{cases} \tag{B.2}$$

with $\mathbb{E}[\nu]$ denoting the expected value of the random variable ν. The objective of the Kalman Filter is to provide the best estimate $\hat{x}$ of the process exploiting the difference between the real measurement y (affected by noise) and the predicted measurement $(\hat{y})$. The equation governing the process estimate is therefore written in the form:

$$\begin{cases} \dot{\hat{x}}(t) & = A(t)\hat{x}(t) + B(t)u(t) + K(t)[y(t) - \hat{y}(t)] \\ \hat{y}(t) & = C(t)\hat{x}(t) \end{cases} \tag{B.3}$$

Introducing the state error $e(t)$:

$$e(t) = \hat{x}(t) - x(t) \quad , \tag{B.4}$$

Eqs.(B.1) and (B.3) allows to derive the dynamics of the error $e(t)$:

$$\begin{aligned} \dot{e}(t) & = A(t)\hat{x}(t) + B(t)u(t) + K(t)[y(t) - \hat{y}(t)] - A(t)x(t) - B(t)u(t) - G(t)w(t) \\ & = A(t)\hat{x}(t) - A(t)x(t) + K(t)[C(t)x(t) + v(t) - C(t)\hat{x}(t)] - G(t)w(t) \\ & = [A(t) - K(t)C(t)]e(t) + K(t)v(t) - G(t)w(t) \end{aligned} \tag{B.5}$$

which can be written as:

$$\dot{e}(t) \quad = E(t)e(t) + z(t) \qquad \text{with} \quad \begin{cases} E(t) & = A(t) - K(t)C(t) \\ z(t) & = -G(t)w(t) + K(t)v(t) \end{cases} \tag{B.6}$$

The Gaussian processes $v(t)$ and $w(t)$ are not correlated, thus the covariance of $z(t)$ can be written in the form:

$$\mathbb{E}\left[z(t)z^T(\tau)\right] \quad = \left[G(t)Q(t)G^T(t) + K(t)R(t)K^T(t)\right]\delta(t-\tau) \tag{B.7}$$

The state error covariance $P(t) = \mathbb{E}\left[e(t)\mathrm{e}^T(t)\right]$ can be computed reminding the general solution for a linear system of the first order such as the one in Eq.(B.6). Denoting $\Phi_e(t, t_0)$ the transition matrix of the system in Eq.(B.6) (i.e., $\Phi_E(t, t_0) = \mathrm{e}^{E(t)(t-t_0)}$, and $\Phi_E^T(t, t_0) = \mathrm{e}^{E^T(t)(t-t_0)}$), we have:

$$P(t) = \quad \mathbb{E}\left[\left(\Phi_E(t,t_0)e(t_0) + \int_{t_0}^{t}\Phi_E(t,\tau)z(\tau)d\tau\right)\left(\Phi_E(t,t_0)e(t_0) + \int_{t_0}^{t}\Phi_E(t,\tau)z(\tau)d\tau\right)^T\right] \tag{B.8}$$

Assuming that $e(t)$ and $z(t)$ are uncorrelated, Eq.(B.8) becomes:

$$P(t) = \quad \mathbb{E}\left[\Phi_E(t,t_0)e(t_0)\mathrm{e}^T(t_0)\Phi_E^T(t,t_0)\right] + \mathbb{E}\left[\left(\int_{t_0}^{t}\Phi_E(t,\tau)z(\tau)z^T(\tau)\Phi_E^T(t,\tau)d\tau\right)\right] \tag{B.9}$$

Being $\mathbb{E}\left[e(t_0)\mathrm{e}^T(t_0)\right] = P(t_0)$, and reminding Eq.(B.7), the expression of the state error covariance becomes:

$$P(t) = \quad \Phi_E(t,t_0)P(t_0)\Phi_E^T(t,t_0) + \int_{t_0}^{t}\Phi_E(t,\tau)\left[G(\tau)Q(\tau)G^T(\tau) + K(\tau)R(\tau)K^T(\tau)\right]\Phi_E^T(t,\tau)d\tau \tag{B.10}$$

Replacing $(t-\tau)$ with $(t-t_0)+(t_0-\tau)$, it is possible to write:

$$P(t) = \quad \Phi_E(t,t_0)\left(P(t_0) + \int_{t_0}^{t}\Phi_E(t_0,\tau)\left[G(\tau)Q(\tau)G^T(\tau) + K(\tau)R(\tau)K^T(\tau)\right]\Phi_E^T(t_0,\tau)d\tau\right)\Phi_E^T(t,t_0) \tag{B.11}$$

Eq.(B.11) is derived in order to obtain the expression of $\dot{P}(t)$, which is:

$$\begin{aligned}
\dot{P}(t) = \quad & E(t)\Phi_E(t,t_0)\left(P(t_0) + \int_{t_0}^{t}\Phi_E(t_0,\tau)\left[G(\tau)Q(\tau)G^T(\tau) + K(\tau)R(\tau)K^T(\tau)\right]\Phi_E^T(t_0,\tau)d\tau\right)\Phi_E^T(t,t_0) + \\
& +\Phi_E(t,t_0)\left(P(t_0) + \int_{t_0}^{t}\Phi_E(t_0,\tau)\left[G(\tau)Q(\tau)G^T(\tau) + K(\tau)R(\tau)K^T(\tau)\right]\Phi_E^T(t_0,\tau)d\tau\right)\Phi_E^T(t,t_0)E^T(t) + \\
& +\Phi_E(t,t_0)\frac{d}{dt}\left(\int_{t_0}^{t}\Phi_E(t_0,\tau)\left[G(\tau)Q(\tau)G^T(\tau) + K(\tau)R(\tau)K^T(\tau)\right]\Phi_E^T(t_0,\tau)d\tau\right)\Phi_E^T(t,t_0) \\
\dot{P}(t) = \quad & E(t)P(t) + P(t)E^T(t) + \Phi_E(t,t_0)\left(\Phi_E(t_0,t)\left[G(\tau)Q(\tau)G^T(\tau) + K(\tau)R(\tau)K^T(\tau)\right]\Phi_E^T(t_0,t)\right)\Phi_E^T(t,t_0)
\end{aligned} \tag{B.12}$$

From Eq.(B.12) the following differential Lyapunov equation is obtained:

$$\begin{aligned}
\dot{P}(t) &= E(t)P(t) + P(t)E^T(t) + G(t)Q(t)G^T(t) + K(t)R(t)K^T(t) \\
\dot{P}(t) &= \left[A(t) - K(t)C(t)\right]P(t) + P(t)\left[A(t) - K(t)C(t)\right]^T + G(t)Q(t)G^T(t) + K(t)R(t)K^T(t)
\end{aligned} \tag{B.13}$$

The optimal gain $K(t)$ is computed in order to minimize the trace of $\dot{P}(t)$. Knowing that the covariance matrix is symmetric ($P(t) = P(t)^T$), and applying the following trace properties

$$\begin{cases}
tr(ABC) & = tr(CAB) = tr(BCA) \\
\dfrac{\partial tr(AB)}{\partial A} & = B^T \\
\dfrac{\partial tr(ACA^T)}{\partial A} & = 2AC
\end{cases} \tag{B.14}$$

it is possible to write:

$$\frac{\partial tr(\dot{P}(t))}{\partial K(t)} \quad = -2P(t)C^T(t) + 2K(t)R(t) = 0 \tag{B.15}$$

which leads to the expression of the optimal Kalman gain $K(t)$:

$$K(t) \quad = P(t)C^T(t)R^{-1}(t) \tag{B.16}$$

Replacing the expression of $K(t)$ in Eq.B.13 with Eq.(B.16) leads to:

$$\begin{aligned}\dot{P}(t) &= A(t)P(t) - P(t)C^T(t)R^{-1}(t)C(t)P(t) + P(t)A^T(t) + ...\\ &\quad ... - P(t)C^T(t)R^{-1}(t)C(t)P(t) + G(t)Q(t)G^T(t) + P(t)C^T(t)R^{-1}(t)R(t)R^{-1}(t)C(t)P(t)\\ \dot{P}(t) &= A(t)P(t) + P(t)A^T(t) - P(t)C^T(t)R^{-1}(t)C(t)P(t) + G(t)Q(t)G^T(t)\end{aligned} \tag{B.17}$$

which is a *continuous Riccati equation* that allows computing the estimated covariance matrix. The KF operates as follows: once initialized with a guess of $\hat{x}(t_0)$ and $P(t_0)$, the gain $K(t)$ is computed using Eq.B.16. Then $\hat{x}(t)$ and $P(t)$ are computed through integration of Eq.B.3 and B.13.

B.1.2 The Discrete Kalman Filter

The Navigation functions operates in the discrete-time domain with a given rate. Thus, a discrete version of the Kalman Filter should be derived.

Let's consider the time-discrete process x_k and the time-discrete measurement y_k that are governed by the following linear time-discrete equation:

$$\begin{cases} x_k &= A_k x_{k-1} + B_k u_k + G_k w_k \\ y_k &= C_k x_k + v_k \end{cases} \tag{B.18}$$

where w_k and v_k represent respectively the process and the measurement noise, uncorrelated one from the other, white and with normal probability distributions, with covariance Q_k and R_k such that:

$$\begin{cases} \mathbb{E}\left[w_k w_j\right] &= \begin{cases} 0 & k \neq j \\ Q_k & k = j \end{cases} \\ \mathbb{E}\left[v_k v_j\right] &= \begin{cases} 0 & k \neq j \\ R_k & k = j \end{cases} \end{cases} \tag{B.19}$$

The *a priori* estimates of process and of measurement are defined as the system was not affected by noise:

$$\begin{cases} \hat{x}_{k|k-1} &= A_k \hat{x}_{k-1|k-1} + B_k u_k \\ \hat{y}_{k|k-1} &= C_k \hat{x}_{k|k-1} \end{cases} \tag{B.20}$$

As in the continuous case, the objective of the Kalman Filter is to exploit the difference between the real measurement y_k and the predicted measurement $\hat{y}_{k|k-1}$ to obtain the information needed to update the *a priori* estimate $\hat{x}_{k|k-1}$, and compute the optimal a posteriori estimate $\hat{x}_{k|k}$. The goal is to find an equation that computes the *a posteriori* estimate as a linear combination of the *a priori* estimate and the so called residual, which is the difference between the real measurement y_k and the predicted one ($\hat{y}_{k|k-1} = C_k \hat{x}_{k|k-1}$):

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \\ &= (I - K_k C_k)\hat{x}_{k|k-1} + K_k(C_k x_k + v_k) \end{aligned} \tag{B.21}$$

Let's introduce the *a priori* estimation error $e_{k|k-1}$ and the *a posteriori* estimation error $e_{k|k}$:

$$\begin{cases} e_{k|k} &= x_k - \hat{x}_{k|k} \\ e_{k|k-1} &= x_k - \hat{x}_{k|k-1} = A_k e_{k-1|k-1} + G_k w_k \end{cases} \tag{B.22}$$

Eq.(B.21) can be manipulated in order to express the dependence of the *a posteriori* error to the *a priori* one. Subtracting x_k from the right and left side of Eq.(B.21) one obtains:

$$\begin{aligned} \hat{x}_{k|k} - x_k &= (I - K_k C_k)\hat{x}_{k|k-1} + K_k C_k x_k - x_k + K_k v_k \\ e_{k|k} &= (I - K_k C_k) e_{k|k-1} - K_k v_k \end{aligned} \tag{B.23}$$

The optimal value of K_k is the one that minimize the trace of the *a posteriori* state error covariance $P_{k|k}$. Reminding that the process and the measurement noises are independent one from another, and that the noise has covariance equal to R_k, it is possible to write:

$$\begin{aligned} P_{k|k} &= \mathbb{E}\left[e_{k|k}\mathrm{e}^T_{k|k}\right] \\ &= \mathbb{E}\left[(I - K_k C_k)e_{k|k-1}\mathrm{e}^T_{k|k-1}(I - K_k C_k)^T\right] + \mathbb{E}\left[K_k v_k v_k^T K_k^T\right] \\ &= (I - K_k C_k)P_{k|k-1}(I - K_k C_k)^T + K_k R_k K_k^T \end{aligned} \tag{B.24}$$

where $P_{k|k-1}$ is the *a priori* state error covariance:

$$\begin{aligned} P_{k|k-1} &= \mathbb{E}\left[e_{k|k-1}\mathrm{e}^T_{k|k-1}\right] \\ &= \mathbb{E}\left[(A_k e_{k-1|k-1} + G_k w_k)(A_k e_{k-1|k-1} + G_k w_k)^T\right] \\ &= A_k P_{k-1|k-1} A_k^T + G_k Q_k G_k^T \end{aligned} \tag{B.25}$$

The Optimal Kalman Gain K_k is found imposing the partial derivative of the *a posteriori* state error covariance with respect to K_k to be equal to 0. Applying the same trace properties introduced for the time-continuous case in Eq.(B.14) we obtain:

$$\frac{\partial tr(P_{k|k})}{\partial K_k} = -P_{k|k-1}C_k^T - P_{k|k-1}C_k^T + 2K_k C_k P_{k|k} C_k^T + 2K_k R_k = 0 \tag{B.26}$$

which is solved by a value of K_k equal to:

$$K_k = P_{k|k-1}C_k^T(C_k P_{k|k} C_k^T + R_k)^{-1} \tag{B.27}$$

Once that the optimal gain is computed, the state is updated using Eq.(B.21), and the *a posteriori* error covariance is updated using Eq.(B.24):

$$\begin{aligned}
P_{k|k} &= (I - K_k C_k) P_{k|k-1} (I - K_k C_k)^T + K_k R_k K_k^T \\
&= P_{k|k-1} - K_k C_k P_{k|k-1} - P_{k|k-1} C_k^T K_k^T + K_k C_k P_{k|k-1} C_k^T K_k^T + K_k R_k K_k^T \\
&= P_{k|k-1} - K_k C_k P_{k|k-1} - P_{k|k-1} C_k^T K_k^T + K_k (C_k P_{k|k-1} C_k^T + R_k) K_k^T \\
&= P_{k|k-1} - K_k C_k P_{k|k-1} - P_{k|k-1} C_k^T K_k^T + K_k (K_k^{-1} P_{k|k-1} C_k^T) K_k^T \\
&= P_{k|k-1} - K_k C_k P_{k|k-1} - P_{k|k-1} C_k^T K_k^T + P_{k|k-1} C_k^T K_k^T \\
&= (I - K_k C_k) P_{k|k-1}
\end{aligned} \tag{B.28}$$

The discrete KF will therefore operate as follows. An *a priori* estimate of the state, the measurement, and the state error covariance are predicted using the state equations and the *a posteriori* estimate of the previous step. This phase is named *prediction*:

$$\begin{cases}
\hat{x}_{k|k-1} &= A_k \hat{x}_{k-1|k-1} + B_k u_k \\
\hat{y}_{k|k-1} &= C_k \hat{x}_{k|k-1} \\
P_{k|k-1} &= A_k P_{k-1} A_k^T + G_k Q_k G_k^T
\end{cases} \tag{B.29}$$

Then the gain is computed, taking into account the noise covariance and the *a priori* predicted error covariance:

$$K_k = P_{k|k-1} C_k^T (C_k P_{k|k} C_k^T + R_k)^{-1} \tag{B.30}$$

Finally, both the state and the state error covariance matrix are corrected in the *update* step obtaining the *a posteriori* estimates:

$$\begin{cases}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}) \\
P_{k|k} &= (I - K_k C_k) P_{k|k-1}
\end{cases} \tag{B.31}$$

In practice, the Kalman gain K_k is a sort of weight that discriminate which contribution between the *a priori* estimate and the measurement residual are going to be given more importance in the computation of the update. Taking the two limit cases, if the measurement noise covariance R_k tends to zero, K_k tends to C_k^{-1} and the update is simply calculated by inverting the relation state/measurement:

$$\begin{aligned}
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + C_k^{-1} (y_k - C_k \hat{x}_{k|k-1}) \\
&= C_k^{-1} y_k
\end{aligned} \tag{B.32}$$

On the other hand, if the measurement noise covariance matrix tends to infinity (i.e., the measurement are not reliable), or if the *a priori* error covariance tends to zero (i.e., the process prediction is completely reliable), the Kalman gain is equal to zero and therefore the *a posteriori* estimate is equal to the *a priori* one:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} \tag{B.33}$$

B.1.3 From the Continuous to Discrete Kalman Filter

When implementing optimal estimation of a time-continuous process with a digital filter, there is the need of discretizing the time-continuous equations that govern the dynamic of the process. In this section the correspondence between the continuous state matrix $A(t)$ and input matrix $B(t)$, and the discrete state matrix A_k and input matrix B_k is presented. Remembering the expression of the general solution for a system in the form of Eq.(B.1) for $t_0 = 0$:

$$x(t) \quad = \mathrm{e}^{At}x_0 + \int_0^t \Phi_f(t,\tau)Bu(\tau)d\tau = \mathrm{e}^{At}x_0 + \int_0^t \mathrm{e}^{At}\mathrm{e}^{-A\tau}Bu(\tau)d\tau \tag{B.34}$$

it is possible to write the solution at the consecutive instants $t_{k+1} = (k+1)T$ and $t_k = kT$:

$$\begin{cases} x(kT) & = \mathrm{e}^{AkT}x_0 + \int_0^{kT} \mathrm{e}^{AkT}\mathrm{e}^{-A\tau}Bu(\tau)d\tau \\ x((k+1)T) & = \mathrm{e}^{A(k+1)T}x_0 + \int_0^{(k+1)T} \mathrm{e}^{A(k+1)T}\mathrm{e}^{-A\tau}Bu(\tau)d\tau \end{cases} \tag{B.35}$$

Multiplying all the terms of the first equation in (B.35) by e^{AT} and solving the equation for $\mathrm{e}^{A(k+1)T}x_0$ we get:

$$\mathrm{e}^{A(k+1)T}x_0 \quad = \mathrm{e}^{AT}x(kT) - \int_0^{kT} \mathrm{e}^{A(k+1)T}\mathrm{e}^{-A\tau}Bu(\tau)d\tau \tag{B.36}$$

The value of $\mathrm{e}^{A(k+1)T}x_0$ found in Eq.(B.36) can be replaced in the second equation of(B.35):

$$x((k+1)T) \quad = \mathrm{e}^{AT}x(kT) + \left[\int_0^{(k+1)T} \mathrm{e}^{A(k+1)T}\mathrm{e}^{-A\tau}Bu(\tau)d\tau - \int_0^{kT} \mathrm{e}^{A(k+1)T}\mathrm{e}^{-A\tau}Bu(\tau)d\tau\right] \tag{B.37}$$

which is equivalent to:

$$x((k+1)T) \quad = \mathrm{e}^{AT}x(kT) + \int_{kT}^{(k+1)T} \mathrm{e}^{A(k+1)T}\mathrm{e}^{-A\tau}Bu(\tau)d\tau \tag{B.38}$$

Then, supposing that in the interval from kT to $(k+1)T$ the input $u(t) = u(kT)$ is constant and equal to $u(t_k)$, it is possible to take it out of the integral, obtaining:

$$x((k+1)T) \quad = \mathrm{e}^{AT}x(kT) + \int_{kT}^{(k+1)T} \mathrm{e}^{A[(k+1)T-\tau]}Bd\tau \;\; u(kT) \qquad \tau \in [kT, (k+1)T[\tag{B.39}$$

It is possible to see that as τ ranges from kT to $(k+1)T$ (i.e., the lower and the upper limit of the integration) the exponent ranges from T to 0. If a new variable $\lambda = (k+1)T - \tau$ is introduced, then $d\lambda = -d\tau$, and λ ranges from $\lambda = T$ to $\lambda = 0$ as τ ranges from $\tau = kT$ to $\tau = (k+1)T$, and Eq.(B.39) becomes:

$$\begin{aligned} x((k+1)T) \quad & = \mathrm{e}^{AT}x(kT) - \int_T^0 \mathrm{e}^{A\lambda}Bd\lambda \;\; u(kT) \qquad \lambda \in [0, kT[\\ & = \mathrm{e}^{AT}x(kT) + \int_0^T \mathrm{e}^{A\lambda}Bd\lambda \;\; u(kT) \qquad \lambda \in [0, kT[\end{aligned} \tag{B.40}$$

which can be written as:

$$\begin{aligned} x(t_{k+1}) &= \mathrm{e}^{AT}x(t_k) + \int_0^T \mathrm{e}^{A\lambda}Bd\lambda \;\; u(t_k) \\ x(t_{k+1}) &= A_k x(t_k) + B_k u(t_k) \end{aligned} \tag{B.41}$$

For a linear system, the correspondence between the continuous-time state and input matrices $A(t)$ and $B(t)$, and the discrete-time state and input matrices A_k, B_k is therefore:

$$A_k = e^{AT} = \Phi(T) \qquad B_k = \int_0^T \mathrm{e}^{AT}Bdt \tag{B.42}$$

In the scope of this thesis, this equivalence will be useful to model the relative translational dynamics of the chaser with respect target, as it will be detailed in Sec.C.2.2.

B.2 The Continuous-Discrete Extended Kalman Filter

The Kalman theory can be extended to non-linear dynamic systems by means of linearization. However, the obtained estimate is no more optimal, i.e., the computed Kalman gain is not the gain that minimize the *a posteriori* state error covariance matrix. In this section we details the theory underlying the Continuous-Discrete Extended Kalman Filter (CD-EKF), which is a version of the KF allowing to estimate a time-continuous non linear process with time-discrete measurements [Frogerais et al., 2012, Kulikov and Kulikova, 2014]. For the scope of this thesis, the CD-EKF will be used for the estimation of the rotational dynamics of the target. Let's take a dynamic system given in the following form:

$$\begin{cases} \dfrac{dx(t)}{dt} &= f\left(t, x(t)\right) + Bu(t) + Gw(t) \\ y(t_k) &= c(x(t_k)) + v(t_k) \Longleftrightarrow y_k = c(x_k) + v_k \end{cases} \tag{B.43}$$

where $x(t)$ is a time-continuous non-linear process, $f(t, x(t))$ is the nonlinear equation describing the dynamic of the system. $w(t)$ is the Gaussian process noise with covariance Q ($w(t) \sim \mathcal{N}(0, Q)$)and G is the noise input matrix. At the time instant t_k, the prediction of both the state and the covariance matrix P is computed integrating numerically from $t = t_{k-1}$ to $t = t_k$ the following coupled ordinary differential equations:

$$\begin{cases} \dfrac{dx}{dt} = & f(t, x) + Bu(t) \\ \dfrac{dP}{dt} = & \dfrac{\partial f(t, x)}{\partial x}P + \left(\dfrac{\partial f(t, x)}{\partial x}\right)^T P + GQ(t)G^T = R(P, t) \end{cases} \tag{B.44}$$

The second equation in the system is a differential Lyapunov equation that can be derived as follows. Firstly, the state equation can be linearized and written as:

$$\frac{dx(t)}{dt} \sim \frac{\partial f(t,x)}{\partial x} x(t) + Bu(t) + Gw(t) \tag{B.45}$$

The integration of this equation from an initial time t_0 to the running time t yields to:

$$x(t) = \Phi_f(t,t_0)x(t_0) + \int_{t_0}^{t} \mathrm{e}^{F(t-\tau)} Bu(\tau)d\tau + \int_{t_0}^{t} \mathrm{e}^{F(t-\tau)} Gw(\tau)d\tau \tag{B.46}$$

where Φ_f is the state transition matrix of the linearized process (i.e., $\Phi_f(t_i,t_j) = \mathrm{e}^{F(t_i-t_j)}$, with F the Jacobian of f, $F = \dfrac{\partial f(t,x)}{\partial x}$) The state error covariance matrix can be computed from its definition:

$$P(t) = \mathbb{E}\left[(x(t) - m(t))(x(t) - m(t))^T\right] \tag{B.47}$$

where $m(t)$ is the mean (i.e., the expected value) of $x(t)$:

$$\begin{aligned} m(t) = \mathbb{E}\left[x(t)\right] \quad &= \mathbb{E}\left[\Phi_f(t,t_0)x(t_0) + \int_{t_0}^{t} \mathrm{e}^{F(t-\tau)} Bu(\tau)d\tau + \int_{t_0}^{t} \mathrm{e}^{F(t-\tau)} Gw(\tau)d\tau\right] \\ &= \Phi_f(t,t_0)m(t_0) + \int_{t_0}^{t} \mathrm{e}^{F(t-\tau)} Bu(\tau)d\tau \end{aligned} \tag{B.48}$$

The computation of the state error covariance matrix ca be carried out similarly to what has been done for the time-continuous linear case:

$$\begin{aligned}
P(t) = \quad & \mathbb{E}\Bigg[\Big(\Phi_f(t,t_0)x(t_0) + \int_{t_0}^{t}\Phi_f(t,\tau)Bu(\tau)d\tau + \int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau - \Phi_f(t,t_0)m(t_0) - \int_{t_0}^{t}\Phi_f(t,\tau)Bu(\tau)d\tau\Big) ... \\
& ... \Big(\Phi_f(t,t_0)x(t_0) + \int_{t_0}^{t}\Phi_f(t,\tau)Bu(\tau)d\tau + \int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau - \Phi_f(t,t_0)m(t_0) - \int_{t_0}^{t}\Phi_f(t,\tau)Bu(\tau)d\tau\Big)^T\Bigg] \\
P(t) = \quad & \mathbb{E}\Bigg[\Big(\Phi_f(t,t_0)\big(x(t_0) - m(t_0)\big) + \int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau\Big)\ \Big(\Phi_f(t,t_0)\big(x(t_0) - m(t_0)\big) + \int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau\Big)^T\Bigg] \\
P(t) = \quad & \mathbb{E}\Bigg[\Big(\Phi_f(t,t_0)\big(x(t_0) - m(t_0)\big)\Big)\Big(\Phi_f(t,t_0)\big(x(t_0) - m(t_0)\big)\Big)^T\Bigg] + \mathbb{E}\Bigg[\Big(\Phi_f(t,t_0)\big(x(t_0) - m(t_0)\big)\Big)\ \Big(\int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau\Big)^T\Bigg] + ... \\
& ... + \mathbb{E}\Bigg[\Big(\int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau\Big)\ \Big(\Phi_f(t,t_0)\big(x(t_0) - m(t_0)\big)\Big)^T\Bigg] + \mathbb{E}\Bigg[\Big(\int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau\Big)\ \Big(\int_{t_0}^{t}\Phi_f(t,\tau)Gw(\tau)d\tau\Big)^T\Bigg] \\
P(t) = \quad & \Phi_f(t,t_0)\mathbb{E}\left[\big(x(t_0) - m(t_0)\big)\big(x(t_0) - m(t_0)\big)^T\right]\Phi_f^T(t,t_0) + 0 + 0 + \mathbb{E}\left[\int_{t_0}^{t}\Phi_f(t,t_0)\Phi_f(t_0,\tau)Gw(\tau)w(\tau)^T G^T \Phi_f^T(t_0,\tau)\Phi_f^T(t,t_0)d\tau\right] \\
P(t) = \quad & \Phi_f(t,t_0)\left(P(t_0) + \int_{t_0}^{t}\Phi_f(t_0,\tau)G\,\mathbb{E}\left[w(\tau)w(\tau)^T\right]G^T\Phi_f^T(t_0,\tau)d\tau\right)\Phi_f^T(t,t_0)
\end{aligned} \tag{B.49}$$

The final expression of the state error covariance will be:

$$P(t) = \Phi_f(t,t_0)\left(P(t_0) + \int_{t_0}^{t}\Phi_f(t_0,\tau)G\,QG^T\Phi_f^T(t_0,\tau)d\tau\right)\Phi_f^T(t,t_0) \tag{B.50}$$

Eq.(B.50) is derived in order to obtain $\dot{P}(t)$:

$$\begin{aligned}
\dot{P}(t) = \;& F\Phi_f(t,t_0)\left(P(t_0) + \int_{t_0}^{t} \Phi_f(t_0,\tau)G\,QG^T\Phi_f^T(t_0,\tau)d\tau\right)\Phi_f^T(t,t_0) + ...\\
& ... + \Phi_f(t,t_0)\left(P(t_0) + \int_{t_0}^{t} \Phi_f(t_0,\tau)G\,QG^T\Phi_f^T(t_0,\tau)d\tau\right)\Phi_f^T(t,t_0)F^T + ...\\
& ... + \Phi_f(t,t_0)\frac{d}{dt}\left(\int_{t_0}^{t} \Phi_f(t_0,\tau)G\,QG^T\Phi_f^T(t_0,\tau)d\tau\right)\Phi_f^T(t,t_0)\\
\dot{P}(t) = \;& FP(t) + P(t)F^T + \Phi_f(t,t_0)\left(\Phi_f(t_0,t)G\,QG^T\Phi_f^T(t_0,t)\right)\Phi_f^T(t,t_0)
\end{aligned} \tag{B.51}$$

This lead to the differential Lyapunov equation presented in (Eq.B.44):

$$\dot{P}(t) = FP(t) + P(t)F^T + GQG^T \tag{B.52}$$

This system in Eq.B.44 is solved numerically dividing the step $\Delta_t = t_k - t_{k-1}$ in sub-steps $t_{k,n}$ of size equal to δ, where $t_{k,1} < ... < t_{k,n} < ... < t_{k,\alpha}$, $t_{k,1} = t_{k-1}$, $t_{k,\alpha} = t_k$ and $\delta = t_{k,n} - t_{k,n-1}$. At each sub-step, the prediction is done applying an Explicit Runge-Kutta method of order i, as shown by the following equation:

$$\begin{cases}
x_{k,n} = & x_{k,n-1} + \left(\mathfrak{RK}_{\mathfrak{i}}^{\delta} f\right)(x_{k,n-1}, t_{k,n-1})\\
P_{k,n} = & P_{k,n-1} + \left(\mathfrak{RK}_{\mathfrak{i}}^{\delta} R\right)(P_{k,n-1}, t_{k,n-1})
\end{cases} \tag{B.53}$$

For the rotational dynamics KF developped in Chapter2, a fourth order scheme is used:

$$\left(\mathfrak{RK}_4^{\delta} f\right)(t,x) = \frac{\delta}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right) \tag{B.54}$$

where:

$$\begin{cases}
K_1 & = f(t,x)\\
K_2 & = f(t + \frac{\delta}{2}, x + \frac{\delta}{2}K_1)\\
K_3 & = f(t + \frac{\delta}{2}, x + \frac{\delta}{2}K_2)\\
K_4 & = f(t, x + \delta K_3)
\end{cases} \tag{B.55}$$

The so computed values of $x(t_k)$ and $P(t_k)$ are the a priori prediction of the state and the covariance matrix $\hat{x}_{k|k-1}$ and $P_{k|k-1}$. From this moment on the CD-EKF follows the steps of a classical Discrete Kalman Filter in order to compute the gain K_k and the update using the discrete measurement y_k. As in Section B.1.2, the goal is to find an equation that computes the *a posteriori* estimate as a linear combination of the *a priori* estimate and the difference between the real measurement $y(x_{k|k-1})$ and the predicted one $(\tilde{y}(\hat{x}_{k|k-1}))$:

$$\hat{x}_{k|k} \;= \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1}) \tag{B.56}$$

If the measurement is a non-linear function of the state, an approximation has to be introduced before computing the Kalman gain:

$$C_k = \frac{\partial c_i}{\partial x_j}(\hat{x}_{k|k-1}, 0) \tag{B.57}$$

The predicted and the real measurement can be approximated to:

$$\begin{aligned} \tilde{y}_k &\simeq C_k x_k + v_k \\ \tilde{\hat{y}}_k &\simeq C_k \hat{x}_{k|k-1} \end{aligned} \tag{B.58}$$

The equation of the *a posteriori* error can be now manipulated as in Eq.(B.23) and the *a posteriori* error is found:

$$e_k = (I - K_k C_k)e_{k|k-1} - K_k v_k \tag{B.59}$$

The covariance is computed taking into account that the process and the measurement noises are independent one from another, and that the measurement noise has discrete covariance equal to R_k:

$$\begin{aligned} P_{k|k} &= \mathbb{E}\left[e_{k|k}\mathrm{e}_k^T\right] \\ &= \mathbb{E}\left[(I - K_k C_k)e_{k|k-1}\mathrm{e}_k^{-T}(I - K_k C_k)^T\right] + \mathbb{E}\left[K_k v v^T K_k^T\right] \\ &= (I - K_k C_k)P_{k|k-1}(I - K_k C_k)^T + K_k R_k K_k^T \end{aligned} \tag{B.60}$$

where $P_{k|k-1}$ is the *a priori* error covariance matrix that was computed through numerical integration. The Kalman Gain K_k is found minimizing partial derivative of the *a posteriori* state error covariance matrix with respect to K_k, using the now well known trace's properties in Eq.(B.14):

$$\frac{\partial tr(P_{k|k})}{\partial K_k} = -P_{k|k-1}C_k^T - P_{k|k-1}C_k^T + 2K_k C_k P_{k|k} C_k^T + 2K_k R_k = 0 \tag{B.61}$$

This leads to a value of K_k equal to:

$$K_k = P_{k|k-1}C_k^T(C_k P_{k|k-1}C_k^T + R_k)^{-1} \tag{B.62}$$

As anticipated, unlike the linear cases, the so computed Kalman Gain is not the optimal gain even if it is found through minimization of the *a posteriori* covariance. In fact two approximations due to linearization issues have been introduced, namely the Jacobian A used to compute numerically the *a priori* covariance $P_{k|k-1}$ and the Jacobian C that linearize the non-linear measurement sensitivity matrix.

The *a posteriori* state estimate and state error covariance matrix can now be computed:

$$P_{k|k} = (I - K_k C_k)P_{k|k-1} \tag{B.63}$$

Alternatively, the covariance update can be derived using the so-called *Joseph Form*, which is claimed to be more numerically-stable [Crassidis and Junkins, 2011]:

$$P_{k|k} = (I - K_k C_k) P_{k|k-1} (I - K_k C_k)^T + K_k R K_k^T \tag{B.64}$$

However, Joseph's stabilized version requires more computations than the classical form. For the update of the state, the non-linear sensitivity function $c(x_k)$ can be directly used without need of exploiting the Jacobian C_k:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \Big(y_k - c(\hat{x}_{k|k-1}) \Big) \tag{B.65}$$

Appendix C

Dynamic models

Contents

This appendix provides an overview of the models governing the translational and rotational dynamics of both target and chaser during a space rendezvous. First, a short introduction on Keplerian orbits and the absolute translational dynamic model implemented in the RDV simulator is given. Then, the relative translational dynamics model underlying a rendezvous is presented, as well as the rotational dynamic model.

C.1 Absolute translational dynamics

According to the Keplerian two-body problem, the knowledge of the S/C state vector (position and velocity) at any instant allows the determination of a unique set of six orbital elements (a semi-major axis, e eccentricity, i inclination, Ω Right Anomaly of Ascending Node (RAAN), ω argument of periapsis and θ the true anomaly) and vice-versa. The RAAN and i define the orbital plane, a and e define the shape of the orbit

(i.e., "dimension" and type of conic), ω fixes the position of the periapsis on the orbital plane, while θ is the only time-varying parameter and indicates the angular position of the spacecraft with respect to the periapsis.
In order to define the orbital parameters, an inertial reference frame must be selected. When studying Earth orbits, the J2000 Earth centered equatorial frame is usually chosen (i.e., the ECI, which stands for Earth-Centered-Inertial)[ESA, 2012]:

- x_i axis direction is towards the mean equinox at J2000 (i.e., the intersection between the J2000 equatorial plane and the ecliptic plane).
- z_i axis is defined by the direction of the Earth mean rotation axis at J2000
- y_i axis completes the right-handed trihedron

The RF origin is at the Earth's center of mass, which can be assumed to be the orbital focus of all the artificial Earth satellite orbits, as Earth mass is in the order of 10^{24} kg. As anticipated, the angles Ω and i define the orbital plane of an object orbiting around Earth, and with ω they identify the so called *Orbital Reference Frame:*

- a rotation of Ω around inertial z_i axis defines x' and y' axis (x' is also called ascending node N, and defines the intersection between the equatorial plane and the orbital plane)
- a rotation of i around x' axis defines a z' axis (which corresponds to the angular momentum h of the orbit) and y'' .
- a third rotation of an angle ω around $z' = h$ axis rotates x' axis to x'' which intersects the perigee of the orbit.

The matrix that allows the passage from a vector x^i written in the inertial frame and a vector x^{orb} written in the orbital frame is obtained by multiplying the three rotation matrices R_Ω, R_i, R_ω :

$$x^i = R_{i-orb}\, x^{orb} = R_\Omega R_i R_\omega\, x^{orb} \tag{C.1}$$

where

$$R_\Omega = \begin{bmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{bmatrix} \quad R_\omega = \begin{bmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.2}$$

From the Orbital RF, it is possible to easily define the *Local Orbital Frame* (LOF) of a spacecraft, which become essential when dealing with relative translational dynamics. The

Local Orbital RF is a moving RF centered at the spacecraft COM. During rendezvous, the relative motion is studied as the relative motion of the chaser with respect to the target, expressed in the Local Orbital Frame of the target. . More than one convention exist to define this RF In this thesis, the *RVH* convention will be used [Curtis, 2013]. The Local Orbital Frame is defined such that:

- x_{LOF} axis is the radial that goes from Earth to the spacecraft, i.e., the R-bar.
- z_{LOF} axis is directed as h, the angular momentum of the spacecraft orbit, i.e., the H-bar.
- y_{LOF} axis completes the trihedron (for circular orbits, it is directed as the tangential velocity vector), i.e., the V bar.

The transformation that goes from the Orbital Frame to the Local Orbital Frame of a spacecraft is a rotation of an angle θ (where θ is the true anomaly of the spacecraft) around angular momentum vector h (i.e., z_{orb}).

$$x^{orb} = R_{orb-LOF}\, x^{LOF} = R_\theta\, x^{LOF} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} x^{LOF} \tag{C.3}$$

$$x^i = R_{i-LOF}\, x^{LOF} = R_\Omega R_i R_\omega R_\theta\, x^{LOF} \tag{C.4}$$

Since the rotations of ω and θ are performed around the same axis, they can be collected in the same rotation matrix R_u where $u = \omega + \theta$:

$$R_u = R_\omega R_\theta = \begin{bmatrix} \cos(\omega+\theta) & -\sin(\omega+\theta) & 0 \\ \sin(\omega+\theta) & \cos(\omega+\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.5}$$

and

$$x^i = R_{i-LOF}\ x^{LOF} = R_\Omega R_i R_u\, x^{LOF} \tag{C.6}$$

Usually another convention known as LVLH (Local Vertical, Local Horizontal) convention is used[Fehse, 2003]:

- z_{LVLH} axis is the radial that goes from the spacecraft to the Earth COM, i.e., -R-bar.
- y_{LVLH} axis is directed in the direction opposite to the angular momentum of spacecraft orbit, i.e., -H-bar.
- x_{LVLH} axis completes the trihedron (for circular orbits, it is directed as the tangential velocity vector), i.e., the V-bar.

In order to pass from the RVH convention to the LVLH convention, a rotation of $\pi/2$ around h followed by a rotation of $-\pi/2$ around the obtained x axis are needed:

$$x^{LVLH} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} x^{RVH} \tag{C.7}$$

For a Keplerian orbit, only θ is a time-varying orbital element, and it is responsible of the rotation of the LOF. Nevertheless, in the presence of perturbations, such as the atmospheric drag in LEO, the Earth's non uniform mass distribution, the solar pressure, and third body perturbation, the Keplerian orbital elements are not constant. Therefore, at any instant, the position and velocity components allow to compute the parameters of the osculating Keplerian orbit described by the spacecraft motion. Using the so called force-approach, the variation of the orbital parameters can be described by a system of differential equations called *Gauss Planetary Equations* where the perturbations are written in the form of accelerations. These equations directly relate the perturbing force components to the rate of change of the orbital elements. The equations that governs the variation of the elements can be derived expressing the accelerations either in the Local Orbital RF, with accelerations that have component f_r directed as the radial unit vector $\hat{r}$, f_θ perpendicular to $\hat{r}$ and in the orbital plane, and f_h directed as the unit vector $\hat{h}$ parallel to the angular momentum, or in the intrinsic reference frame (which is a frame where x is directed as the normal to the orbit trajectory $\hat{n}$, y is directed as the tangent to the orbit trajectory $\hat{t}$ and z is directed as $\hat{h}$. For completeness, we report here set of the Gauss equations used in the rendezvous simulator using for this study, with perturbing accelerations expressed in the local orbital reference frame:

$$\begin{cases} \dfrac{\mathrm{d}a}{\mathrm{d}t} &= \dfrac{2\sqrt{a^3}}{\sqrt{\mu(1-e^2)}}\left[e\sin\theta\cdot f_r + (1+e\cos\theta)\cdot f_\theta\right] \\ \dfrac{\mathrm{d}e}{\mathrm{d}t} &= \sqrt{\dfrac{a(1-e^2)}{\mu}}\left[\sin\theta\cdot f_r + (\cos\theta+\cos E)\cdot f_\theta\right] \\ \dfrac{\mathrm{d}i}{\mathrm{d}t} &= \dfrac{\sqrt{a(1-e^2)}}{\sqrt{\mu}(1+e\cos\theta)}\cos u\cdot f_h \\ \dfrac{\mathrm{d}\Omega}{\mathrm{d}t} &= \dfrac{\sqrt{a(1-e^2)}}{\sqrt{\mu}(1+e\cos\theta)}\dfrac{\sin u}{\sin i}\cdot f_h \\ \dfrac{\mathrm{d}\omega}{\mathrm{d}t} &= \sqrt{\dfrac{a(1-e^2)}{\mu}}\left[-\dfrac{\cos\theta}{e}\cdot f_r + \dfrac{2+e\cos\theta}{e(1+e\cos\theta)}\sin\theta\cdot f_\theta - \dfrac{\sin u\cot i}{1+e\cos\theta}\cdot f_h\right] \\ \dfrac{\mathrm{d}M}{\mathrm{d}t} &= \sqrt{\dfrac{\mu}{a^3}} - \sqrt{\dfrac{a}{\mu}}\dfrac{1-e^2}{e(1+e\cos\theta)}\left[(2e-\cos\theta-e\cos^2\theta)\cdot f_r + (2+e\cos\theta)\sin\theta\cdot f_\theta\right] \end{cases} \tag{C.8}$$

where μ is the gravitational constant of the planet and the variable M is the mean anomaly of the orbit, a parameter which is linearly proportional to the time and allows to compute the true anomaly θ [Curtis, 2013]. The orbit propagator used for this study integrates, at each time instant, the osculating orbital parameters, from which the absolute position and velocity of the S/C are derived. In the following paragraphs the perturbations that have been included in the RDV simulator used for this study are briefly detailed (i.e., the atmospheric drag, the solar pressure, and the non-uniform mass distribution).

Atmospheric drag

When the perigee of an orbit is below 1000 km, the atmospheric drag effects is not negligible and it increases as the altitude decreases. Drag, unlike other perturbation forces, is a non-conservative force and continuously decreases the energy of the orbit reducing its semi-major axis.

The effect of the atmosphere on a spacecraft should be studied taking into account the contribution of each spacecraft's surface and the orientation of each surface with respect to the relative velocity of the spacecraft in the atmospheric flow. It produces both a force and a torque. However, in a first approximation, the atmospheric drag acceleration can be modeled according to:

$$\mathbf{f}_d = -\frac{1}{2}\rho V^2 C_D \frac{A}{m}\hat{t} \tag{C.9}$$

where ρ is the atmospheric density at the considered altitude, A is the cross sectional area exposed to the flux, m is the mass of the spacecraft, C_D is the dimensionless drag coefficient of the spacecraft, V is the modulus of the S/C velocity. The components of $\hat{t}$ need to be expressed in the Orbital RF in order to identify the contribution f_r , f_θ and f_h. Actually, the atmosphere velocity due to the rotation of Earth and the presence of atmospheric winds should be taken into account to compute the S/C velocity in the aerodynamic flow. However, at the altitudes considered for the RDV scenarios, these contributions can be neglected and the absolute translational velocity of the S/C can be used. The highest uncertainty in Eq.(C.9) is brought by the atmosphere density, which is a function of both the altitude and the time (due to the effect of solar flux and the diurnal variation). For a first approximation we can consider a very simple model, the so-called *exponential model*:

$$\rho = \rho_0 e^{-\alpha(h-h_0)} \tag{C.10}$$

where h is the actual altitude of the spacecraft in kilometers, h_0 is the reference height of 200 km where the value of density $\rho_0 = 10^{-10}$ kg/m^3 is measured, and α is a scale factor equal to 0.016 km^{-1}. This model provide results corresponding to a low solar activity, and therefore it defines the lower boundary of density values for a given altitude. Another model provides the higher boundary of values, and corresponding to a high solar activity:

$$\rho = A_1 e^{(A_2 - A_3\sqrt{h})}, \quad \text{with} \begin{cases} A_1 & = 2.14\,10^{-10} \\ A_2 & = 8.27 \\ A_3 & = 1.68\,10^{-2} \end{cases} \tag{C.11}$$

During rendezvous operation, a difference in the values of $m/(C_D A)$ -also known as the ballistic coefficient- of the target and the chaser produces the so called *differential drag*, which causes deviation from the nominal relative trajectory.

Solar pressure

The solar-radiation pressure is induced by the light energy (photons) radiated from the sun. The perturbing acceleration of a satellite due to solar-radiation pressure acceleration can be computed by means of the following equation [ESA, 2015a]:

$$\mathbf{f}_{sp} = C_r \frac{A}{m} P_0 \left(\frac{r_0}{r}\right)^2 \hat{r}_{Sun-S/C} \tag{C.12}$$

where C_r is the solar radiation pressure reflectivity coefficient, A is the cross sectional area exposed to the flux, m is the mass of the spacecraft, r is the distance Sun-S/C, vector, r_0 is the reference distance equal to 1 Astronomic Unit (UA), P_0 is the solar radiation pressure at the distance r_0, and $\hat{r}_{Sun-S/C}$ is the Sun-S/C unit vector. The components of this vector need to be expressed in the Orbital RF in order to identify the contributions f_r , f_θ and f_h. Below 800 km of altitude, the perturbing acceleration from drag is greater than that from solar radiation pressure, while, above 800 km, acceleration from solar radiation pressure is greater [Larson and Wertz, 1992].

Non-uniform mass distribution

The non-uniform density distribution and the ellipsoid-shape of Earth are responsible of one of the most relevant perturbation effects. As the Earth has a bulge at the equator and a flattening at the poles, the equipotential surfaces of the Earth's gravity filed are not spheres, but they look like irregular ellipsoid. The gravitational potential to which a S/C in Earth orbit is subjected at a given position $r \cdot \hat{r}$ can be written as:

$$U = \frac{\mu}{r}\left[1 - \sum_{n=2}^{\infty} J_n \left(\frac{R}{r}\right)^n P_n(\sin\delta)\right] \tag{C.13}$$

where r is the Earth-S/C distance, μ is Earth gravitational constant, R is the equatorial radius of Earth, δ is the geodetic latitude of the S/C, J_n are the so-called *gravity harmonics* coefficients, and P_n indicates the Legendre polynomial of order n. The values of the harmonics depend on the shape and mass distribution of the central body. The second order zonal harmonic is $J_2 = 1.082629 \cdot 10^{-3}$, while the following harmonics are of order 10^{-6} or lower. Thus, a reasonably good accuracy can be maintained by simply including the J_2 effect.

The Legendre polynomial by definition are

$$P_n(x) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{dx^n}[(x^2-1)^n] \tag{C.14}$$

For the second order harmonics, Eq.(C.14) becomes:

$$P_2(\sin\delta) = \frac{1}{2}(3\sin^2\delta - 1) \tag{C.15}$$

so the approximated potential function becomes:

$$U = \frac{k}{r}\left[1 - J_2\left(\frac{R}{r}\right)^2\left(\frac{3}{2}\sin^2\delta - \frac{1}{2}\right)\right] \tag{C.16}$$

The contribution due to J_2 can be isolated from the potential of a perfect spherical body:

$$U_{J_2} = U - \frac{k}{r} = -\frac{k}{r}J_2\left(\frac{R}{r}\right)^2\left(\frac{3}{2}\sin^2\delta - \frac{1}{2}\right) \tag{C.17}$$

The declination δ can be expressed using the coordinates of the considered point expressed in the ECI RF, and the distance r:

$$z = r\sin\delta \quad \Rightarrow \quad \sin\delta = \frac{z}{r} \tag{C.18}$$

Eq.(C.17) becomes:

$$U_{J_2} = -\frac{k}{r}J_2\left(\frac{R}{r}\right)^2\left(\frac{3}{2}\frac{z^2}{r^2} - \frac{1}{2}\right) \tag{C.19}$$

The perturbing forces along each component can be derived by taking partial derivatives of U_{J_2}:

$$\mathbf{f}_{J_2} = \vec{\nabla}U_{J_2} = \frac{\partial U_{J_2}}{\partial r}\hat{r} + \frac{\partial U_{J_2}}{\partial z}\hat{z} \tag{C.20}$$

Once derived, Eq.(C.20) has to be expressed in the Local Orbital Frame. This can be done easily as the components of the unit vector $\hat{z}$ of the ECI, expressed in the Local Orbital RF, corresponds to the last row of the rotation matrix R_{i-LOF} of Eq.(C.6):

$$\hat{z}^{LOF} = \begin{bmatrix} \sin i \sin u \\ \sin i \cos u \\ \cos i \end{bmatrix}, \tag{C.21}$$

while the value $z/r = \sin i \sin u$. These relations allow to write the contribution of the non-uniform mass distribution in analytical form as a function of the orbital parameters, which allows to include this perturbation source directly in Eq.(C.8):

$$\begin{aligned} f_r &= -\frac{3}{2}\frac{kJ_2R^2}{r^4}\left(1-\sin^2 i\sin^2 u)\right) \\ f_\theta &= -\frac{3}{2}\frac{kJ_2R^2}{r^4}\left(2\sin^2 i\ \sin u\cos u\right) \\ f_h &= -\frac{3}{2}\frac{kJ_2R^2}{r^4}\left(2\sin i\ \cos i\ \sin u\right) \end{aligned} \tag{C.22}$$

C.2 Relative translational dynamics

As detailed in Sec.1.2.1, the navigation must switch from absolute to relative as the chaser gets closer to the target. The description of the relative translational dynamics of the couple chaser-target using the Gauss equations would be to complex to be used as a prediction model in a navigation filter. As it will be shown in this Section, after a linearization of the gravitational acceleration that is acting on the chaser, it is possible to write the relative dynamics of the chaser in the target LOF in the form of a linear system of second order differential equations. The motion of a body under the influence of a central force follows the Newton's law of gravitation [Fehse, 2003] :

$$\mathbf{F}_g(\mathbf{r}) = -G\frac{Mm}{r^2}\frac{\mathbf{r}}{r} = -\mu\frac{m}{r^3}\mathbf{r} \tag{C.23}$$

where

$$\begin{aligned} \mathbf{F}_g(\mathbf{r}) &= \text{gravitational force} \\ G &= \text{universal gravitational constant} \\ M &= \text{mass of the central body (e.g. Earth)} \\ m &= \text{mass of the spacecraft} \\ \mathbf{r} &= \text{the radius vector, } r = |\mathbf{r}| \\ \mu &= GM \end{aligned} \tag{C.24}$$

The central acceleration to which a S/C is subjected is:

$$\mathbf{f}_g(\mathbf{r}) = -\mu\frac{\mathbf{r}}{r^3} \tag{C.25}$$

Supposing that no perturbations are acting neither on the target nor on the chaser, from Eq.(C.25) the equations of motion of the target and the chaser are given by:

$$\begin{cases} \ddot{\mathbf{r}}_t = -\mu\dfrac{\mathbf{r}_t}{r_t^3} \\ \ddot{\mathbf{r}}_c = -\mu\dfrac{\mathbf{r}_c}{r_c^3} + \dfrac{\mathbf{F}}{m_c} \end{cases} \tag{C.26}$$

where $\mathbf{F}$ is the control force acting on the chaser and m_c is the mass of the chaser. The relative position $\mathbf{s}$ of the chaser with respect to the target, as well as the relative acceleration $\ddot{\mathbf{s}}$, expressed in the ECI frame, are:

$$\begin{cases} \mathbf{s} = \mathbf{r}_c - \mathbf{r}_t \\ \ddot{\mathbf{s}} = \ddot{\mathbf{r}}_c - \ddot{\mathbf{r}}_t \end{cases} \tag{C.27}$$

From Eqs.(C.26) and (C.27), one obtains

$$\ddot{\mathbf{s}} = -\mu\frac{\mathbf{r}_c}{r_c^3} + \mu\frac{\mathbf{r}_t}{r_t^3} + \frac{\mathbf{F}}{m_c} = \mathbf{f}_g(\mathbf{r}_c) - \mathbf{f}_g(\mathbf{r}_t) + \frac{\mathbf{F}}{m_c} \tag{C.28}$$

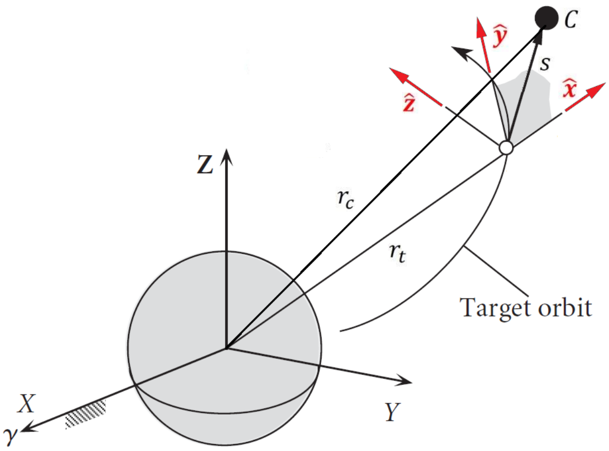


Figure C.1: Relative target-chaser position. The axis X, Y, Z identify the ECI RF, while the axis $\hat{x}$, $\hat{y}$, $\hat{z}$ identify target LOF

The expression of the gravitational force acting on the chaser $\mathbf{f}_g(\mathbf{r}_c)$ can be linearized around the vector $\mathbf{r}_t$ by means of Taylor expansion:

$$\mathbf{f}_g(\mathbf{r}_c) = \mathbf{f}_g(\mathbf{r}_t) + \left.\frac{d\mathbf{f}_g(\mathbf{r})}{d\mathbf{r}}\right|_{\mathbf{r}=\mathbf{r}_t} (\mathbf{r}_c - \mathbf{r}_t) \tag{C.29}$$

Considering that

$$\mathbf{r} = [r_x, r_y, r_z]^T \quad \text{and} \quad r = \sqrt{r_x^2 + r_y^2 + r_z^2}, \tag{C.30}$$

$\mathbf{f}_g(\mathbf{r})$ can be written as :

$$\mathbf{f}_g(\mathbf{r}) = -\mu\frac{\mathbf{r}_x}{r^3}\hat{i} + -\mu\frac{\mathbf{r}_y}{r^3}\hat{j} + -\mu\frac{\mathbf{r}_z}{r^3}\hat{k} \tag{C.31}$$

where $\hat{i}$, $\hat{j}$, $\hat{k}$ are the unit vectors directed along the ECI axis X, Y, Z (fig.C.1). Thus, the derivative $\frac{d\mathbf{f}_g(\mathbf{r})}{d\mathbf{r}}$ can be written in the form of the Jacobian:

$$\frac{d\mathbf{f}_g(\mathbf{r})}{d\mathbf{r}} = -\mu \begin{bmatrix} \frac{\partial(r_x/r^3)}{\partial r}\frac{\partial r}{\partial r_x} & \frac{\partial(r_x/r^3)}{\partial r}\frac{\partial r}{\partial r_y} & \frac{\partial(r_x/r^3)}{\partial r}\frac{\partial r}{\partial r_z} \\ \frac{\partial(r_y/r^3)}{\partial r}\frac{\partial r}{\partial r_x} & \frac{\partial(r_y/r^3)}{\partial r}\frac{\partial r}{\partial r_y} & \frac{\partial(r_y/r^3)}{\partial r}\frac{\partial r}{\partial r_z} \\ \frac{\partial(r_z/r^3)}{\partial r}\frac{\partial r}{\partial r_x} & \frac{\partial(r_z/r^3)}{\partial r}\frac{\partial r}{\partial r_y} & \frac{\partial(r_z/r^3)}{\partial r}\frac{\partial r}{\partial r_z} \end{bmatrix} \tag{C.32}$$

which, evaluated in $\mathbf{r} = \mathbf{r}_t$, gives

$$\left.\frac{d\mathbf{f}_g(\mathbf{r})}{d\mathbf{r}}\right|_{\mathbf{r}=\mathbf{r}_t} = -\frac{\mu}{r_t^3}\mathbf{M}, \quad \text{with} \quad \mathbf{M} = \begin{bmatrix} 1-3\frac{r_x^2}{r_t^2} & 3\frac{r_x r_y}{r_t^2} & 3\frac{r_x r_z}{r_t^2} \\ 3\frac{r_y r_x}{r_t^2} & 1-3\frac{r_y^2}{r_t^2} & 3\frac{r_y r_z}{r_t^2} \\ 3\frac{r_z r_x}{r_t^2} & 3\frac{r_z r_y}{r_t^2} & 1-3\frac{r_z^2}{r_t^2} \end{bmatrix} \tag{C.33}$$

Eq.(C.28) then becomes

$$\ddot{\mathbf{s}} = -\frac{\mu}{r_t^3}\mathbf{M}\mathbf{s} + \frac{\mathbf{F}}{m_c} \tag{C.34}$$

The aim now is to represent the chaser motion in the rotating target local orbital frame. Being the rotation rate of the LOF equal to $\boldsymbol{\omega}$, and identifying with the index * the derivatives take in the rotating LOF, it is possible to write:

$$\ddot{\mathbf{s}} = \ddot{\mathbf{x}}_c - \ddot{\mathbf{x}}_t = \dot{\boldsymbol{\omega}} \times \mathbf{s} + \boldsymbol{\omega} \times \boldsymbol{\omega} \times \mathbf{s} + \boldsymbol{\omega} \times \frac{\mathrm{d}^*\mathbf{s}}{\mathrm{d}t} + \frac{\mathrm{d}^{*2}\mathbf{s}}{dt^2} \tag{C.35}$$

Eq.(C.35) can be expressed in LOF coordinates, where $\mathbf{s}^{LOF} = [x,\, y,\, z]$, and $\boldsymbol{\omega}^{LOF} = [0,\, 0,\, \omega]$. Being $\dot{\boldsymbol{\omega}}$ equal to zero for a circular orbit, it is possible to write

$$\frac{\mathrm{d}^{*2}\mathbf{s}}{dt^2} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}, \quad \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{s}) = \begin{bmatrix} -\omega^2 x \\ -\omega^2 y \\ 0 \end{bmatrix}, \quad 2\boldsymbol{\omega} \times \frac{d^*\mathbf{s}}{\mathrm{d}t} = \begin{bmatrix} -2\omega\dot{y} \\ 2\omega\dot{x} \\ 0 \end{bmatrix}, \quad \frac{d\boldsymbol{\omega}}{\mathrm{d}t} \times \mathbf{s} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.36}$$

Moreover, knowing that $\mathbf{r}_t^{LOF} = [r,\, 0,\, 0]$, and reminding that for a circular orbit $\omega^2 = \mu/r^3$, Eq.(C.34) becomes:

$$\ddot{\mathbf{s}} = -\frac{\mu}{r_t^3}\mathbf{M}\mathbf{s} + \frac{\mathbf{F}}{m_c} = -\omega^2 \begin{bmatrix} -2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \frac{\mathbf{F}}{m_c} = -\begin{bmatrix} -2\omega^2 x \\ \omega^2 y \\ \omega^2 z \end{bmatrix} + \frac{\mathbf{F}}{m_c} \tag{C.37}$$

Substituting Eqs.(C.36),(C.37) into the Eq.(C.35), the well known linear system of differential equation for the relative motion, known as Hill's equations, is obtained:

$$\begin{aligned} \ddot{x} - 3\omega^2 x - 2\omega\dot{y} &= \gamma_x \\ \ddot{y} + 2\omega\dot{x} &= \gamma_y \\ \ddot{z} + \omega^2 z &= \gamma_z \end{aligned} \tag{C.38}$$

where the terms $\gamma_{x,y,z} = \frac{F_{x,y,z}}{m_c}$ are the control accelerations of the chaser. The motion of the z coordinate, which is also referred to as *out-of-plane* motion, is not coupled with the *in-plane* motion (x and y components).

The system in (C.38) can be written in the time-continuous state-representation form, by setting as state vector $\mathbf{x}_{ip} = [x, y, \dot{x}, \dot{y}]^T$ for the in-plane motion and $\mathbf{x}_{op} = [z, \dot{z}]^T$ for the out-of-plane motion:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3\omega^2 & 0 & 0 & 2\omega \\ 0 & 0 & -2\omega & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m_c} & 0 \\ 0 & \frac{1}{m_c} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix} \rightarrow \dot{\mathbf{x}}_{ip} = A_{ip}\mathbf{x}_{ip} + B_{ip}\mathbf{u}_{ip} \tag{C.39}$$

$$\begin{bmatrix} \dot{z} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix} + \frac{1}{m_c} \begin{bmatrix} 0 \\ 1 \end{bmatrix} F_z \rightarrow \dot{\mathbf{x}}_{op} = A_{op}\mathbf{x}_{op} + B_{op}\mathbf{u}_{op} \tag{C.40}$$

The homogeneous solution of the system in Eq.(C.38), which leads to the analytical computation of the transition matrix Φ_0 of the time-continuous linear state-space representation corresponding to Eqs.(C.39),(C.40), is known under the name of Clohessy-Wiltshire equations and will be detailed in Sec.C.2.1. It must be reminded that this linear system has been obtained by linearization of the gravitational force acting on the chaser. Because of this linearization, the accuracy of the CWH equations decreases as the target-chaser distance increases. In a LEO rendezvous mission, position errors will become significant at a distance of a few tens of kilometers from the origin. For example, in the radial direction (i.e., x), an error equal to $\Delta x = r(1 - cos(\frac{y}{r}))$ is always present due to the orbit curvature. For a target orbit at an altitude of $h = 800$km (i.e., $r = 6766km$), the error in the radial direction is $\Delta x = 7.0$ m at a distance of $y = 10$ km, and $\Delta x = 62.7m$ at a distance of $y = 30km$. If a curved definition of the y-coordinate were used, the useful range of the CWH equations could be significantly increased [Fehse, 2003].

C.2.1 Analytical solution for the homogeneous problem

The analytical solution of the set of second order differential equations in (C.38) is obtained computing firstly the homogeneous solution (i.e., $\gamma_{x,y,z} = 0$), and then adding a particular solution.

The computation of the homogeneous solution can be done in many ways, both in time domain or in Laplace domain. In the following section the solution in time domain proposed by [Curtis, 2013] is presented. Assuming that in $t = t_0$ the initial conditions are:

$$\begin{bmatrix} x(t_0) \\ y(t_0) \\ z(t_0) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \begin{bmatrix} \dot{x}(t_0) \\ \dot{y}(t_0) \\ \dot{z}(t_0) \end{bmatrix} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix} \tag{C.41}$$

the second equation in (C.38) cn be integrated:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\dot{y} + 2\omega x) = 0 \rightarrow \dot{y} + 2\omega x = \text{const} = \dot{y}_0 + 2\omega x_0 \tag{C.42}$$

It is possible therefore to write:

$$\dot{y} = \dot{y}_0 + 2\omega(x_0 - x) \tag{C.43}$$

and to substitute this result in the first equation of (C.38):

$$\ddot{x} - 3\omega^2 x - 2\omega\left[\dot{y}_0 + 2\omega(x_0 - x)\right] \rightarrow \ddot{x} + \omega^2 x = 2\omega\dot{y}_0 + 4\omega^2 x_0 \tag{C.44}$$

This is a classical second order differential equation which has solution in the form of:

$$x = A\sin(\omega(t - t_0)) + B\cos(\omega(t - t_0)) + \frac{1}{\omega^2}(2\omega\dot{y}_0 + 4\omega^2 x_0) \tag{C.45}$$

Evaluating this equation at $t = t_0$, the value of B is found:

$$x(t_0) = x_0 = B + \frac{2}{\omega}\dot{y}_0 + 4x_0 \rightarrow B = -3x_0 - \frac{2}{\omega}\dot{y}_0 \tag{C.46}$$

The value of A can be computed by differentiation of the expression of x, and evaluating the obtained result in $t = t_0$:

$$\dot{x} = \omega A\cos(\omega(t - t_0)) - \omega B\sin(\omega(t - t_0)) \rightarrow \dot{x}(t_0) = \dot{x}_0 = \omega A \rightarrow A = \frac{\dot{x}_0}{\omega} \tag{C.47}$$

The expression of x becomes therefore:

$$x = \frac{\dot{x}_0}{\omega}\sin(\omega(t - t_0)) + (-3x_0 - \frac{2}{\omega}\dot{y}_0)\cos(\omega(t - t_0)) + \frac{2}{\omega}\dot{y}_0 + 4x_0 \tag{C.48}$$

And, rearranged:

$$x = (4 - 3\cos(\omega(t - t_0)))x_0 + \frac{\sin(\omega(t - t_0))}{\omega}\dot{x}_0 + \frac{2}{\omega}(1 - \cos(\omega(t - t_0)))\dot{y}_0 \tag{C.49}$$

Deriving Eq.(C.49), the expression of $\dot{x}$ is obtained:

$$\dot{x} = 3\omega\sin(\omega(t - t_0)))x_0 + \cos(\omega(t - t_0))\dot{x}_0 + 2\sin(\omega(t - t_0))\dot{y}_0 \tag{C.50}$$

In order to compute y, the expression of x in Eq.(C.49) is inserted in Eq.(C.43):

$$\begin{aligned}\dot{y} &= \dot{y_0} + 2\omega\left[x_0 - (4 - 3\cos(\omega(t-t_0)))x_0 - \frac{\sin(\omega(t-t_0))}{\omega}\dot{x}_0 - \frac{2}{\omega}(1-\cos(\omega(t-t_0))\dot{y}_0\right] \\ &= 6\omega(\cos(\omega(t-t_0)) - 1)x_0 - 2sin(\omega(t-t_0))\dot{x}_0 + (4\cos(\omega(t-t_0)) - 3)\dot{y}_0\end{aligned} \tag{C.51}$$

Eq.(C.51) is then integrated in time:

$$y = 6\omega\left(\frac{1}{\omega}\sin(\omega(t-t_0)) - (t-t_0)\right)x_0 + \frac{2}{\omega}\cos(\omega(t-t_0))\dot{x}_0 + \left(\frac{4}{\omega}\sin(\omega(t-t_0)) - 3(t-t_0)\right)\dot{y}_0 + C \tag{C.52}$$

The integration constant C is found evaluating the equation in $t = t_0$

$$y_0 = \frac{2}{\omega}\dot{x}_0 + C \rightarrow C = y_0 - \frac{2}{\omega}\dot{x}_0 \tag{C.53}$$

The expression of y is therefore:

$$y = 6\left(\sin(\omega(t-t_0)) - \omega(t-t_0)\right)x_0 + y_0 + \frac{2}{\omega}(\cos(\omega(t-t_0)) - 1)\dot{x}_0 + \left(\frac{4}{\omega}\sin(\omega(t-t_0)) - 3(t-t_0)\right)\dot{y}_0 \tag{C.54}$$

For what concerns the third equation in (C.38), which describe the out-of-plane motion, the solution is decoupled and can be expressed as:

$$z = D\cos(\omega(t-t_0)) + E\sin(\omega(t-t_0)) \tag{C.55}$$

with

$$\begin{cases} D = & z_0 \\ E = & \dfrac{\dot{z}_0}{\omega} \end{cases} \tag{C.56}$$

Naming $t - t_0 = \tau$, the solution of the homogeneous problem is:

$$\begin{cases} x = & (4 - 3cos(\omega\tau))x_0 + \dfrac{\sin(\omega\tau)}{\omega}\dot{x}_0 + \dfrac{2}{\omega}(1 - cos(\omega\tau))\dot{y}_0 \\ y = & 6\left(\sin(\omega(\tau)) - \omega\tau\right)x_0 + y_0 + \dfrac{2}{\omega}(\cos(\omega\tau) - 1)\dot{x}_0 + \dfrac{1}{\omega}(4\sin(\omega\tau) - 3\omega\tau)\dot{y}_0 \\ \dot{x} = & 3\omega\sin(\omega\tau)x_0 + \cos(\omega\tau)\dot{x}_0 + 2\sin(\omega\tau)\dot{y}_0 \\ \dot{y} = & 6\omega(\cos(\omega\tau) - 1)x_0 - 2\sin(\omega\tau)\dot{x}_0 + (4\cos(\omega\tau) - 3)\dot{y}_0 \\ z = & \cos(\omega\tau)z_0 + \dfrac{1}{\omega}sin(\omega\tau)\dot{z}_0 \\ \dot{z} = & -\omega\sin(\omega\tau)z_0 + cos(\omega\tau))\dot{z}_0 \end{cases} \tag{C.57}$$

The homogeneous solution can be rearranged in matrix form, dividing the in-plane-motion from the out-of-plane motion:

$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 4-3cos\omega\tau & 0 & \dfrac{\sin(\omega\tau)}{\omega} & \dfrac{2}{\omega}(1-cos(\omega\tau)) \\ 6\left(\sin(\omega\tau)-\omega\tau\right) & 1 & \dfrac{2}{\omega}(\cos(\omega\tau)-1) & \dfrac{1}{\omega}(4\sin(\omega\tau)-3\omega\tau) \\ 3\omega\sin(\omega\tau) & 0 & \cos(\omega\tau) & 2\sin(\omega\tau) \\ 6\omega(\cos(\omega\tau)-1) & 0 & -2\sin(\omega\tau) & (4\cos(\omega\tau)-3) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ \dot{x}_0 \\ \dot{y}_0 \end{bmatrix} \tag{C.58}$$

$$\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos(\omega\tau) & \dfrac{1}{\omega}\sin(\omega\tau) \\ -\omega\sin(\omega\tau) & \cos(\omega\tau) \end{bmatrix} \begin{bmatrix} z_0 \\ \dot{z}_0 \end{bmatrix} \tag{C.59}$$

The solutions for both the in-plane motion and the out-of-plane one are written therefore in the form of:

$$\begin{cases} \mathbf{x}_{ip}(t) = & \Phi_{ip}(\tau)\mathbf{x}_{ip}(t_0) \\ \mathbf{x}_{op}(t) = & \Phi_{op}(\tau)\mathbf{x}_{op}(t_0) \end{cases} \tag{C.60}$$

We also know that the general expression for the solution of the homogeneous problem in the state-space domain can be written as:

$$\begin{cases} \mathbf{x}_{ip}(t) = & e^{A_{ip}(t-t_0)}\mathbf{x}_{ip}(t_0) = e^{A_{ip}(\tau)}\mathbf{x}_{ip}(t_0) \\ \mathbf{x}_{op}(t) = & e^{A_{op}(t-t_0)}\mathbf{x}_{op}(t_0) = e^{A_{op}(\tau)}\mathbf{x}_{op}(t_0) \end{cases} \tag{C.61}$$

which means that the matrices in Eqs.(C.58) and (C.59) are the the so-called *State Transition Matrix* (STM) of the system:

$$\begin{cases} \Phi_{ip}(t,t_0) = \Phi_{ip}(\tau) & = e^{A_{ip}(\tau)} = e^{A_{ip}(t-t_0)} \\ \Phi_{op}(t,t_0) = \Phi_{op}(\tau) & = e^{A_{op}(\tau)} = e^{A_{op}(t-t_0)} \end{cases} \tag{C.62}$$

The computation of these matrices is very important when writing the time-discrete representation of the problem, as it will be explained in Section C.2.2.

C.2.2 Discretization of the CWH equations

As demonstrated in Appendix B.1.3, given a continuous-time linear system described by the matrices A and B:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{C.63}$$

and given the corresponding time-discrete linear system described by the matrices A_k and B_k at the generic instant t_k:

$$x_k = A_k x_{k-1} + B_k u_k \tag{C.64}$$

it is possible to write, denoting T the generic time-interval between two consecutive time-steps (i.e., $t_{k+1} - t_k = T$):

$$A_k = e^{AT} = \Phi(T) \tag{C.65}$$

$$B_k = \int_0^T e^{AT} B dt \tag{C.66}$$

Thus, for the CWH system, the matrix A_k can be obtained by evaluating the STM of Eqs.(C.58),(C.59) evaluated in $\tau = T$.

On the other side, B_k must be computed by integrating Eq.(C.66). For the in-plane motion, it will be:

$$
\begin{aligned}
B_{k_{ip}} &= \int_0^T \begin{bmatrix} 4 - 3\cos\omega\tau & 0 & \dfrac{\sin(\omega\tau)}{\omega} & \dfrac{2}{\omega}(1-\cos(\omega\tau)) \\ 6\,(\sin(\omega\tau) - \omega\tau) & 1 & \dfrac{2}{\omega}(\cos(\omega\tau)-1) & \dfrac{1}{\omega}(4\sin(\omega\tau) - 3\omega\tau) \\ 3\omega\sin(\omega\tau) & 0 & \cos(\omega\tau) & 2\sin(\omega\tau) \\ 6\omega(\cos(\omega\tau)-1) & 0 & -2\sin(\omega\tau) & (4\cos(\omega\tau)-3) \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \dfrac{1}{m_c} & 0 \\ 0 & \dfrac{1}{m_c} \end{bmatrix} d\tau \\
&= \frac{1}{m_c}\int_0^T \begin{bmatrix} \dfrac{\sin(\omega\tau)}{\omega} & \dfrac{2}{\omega}(1-\cos(\omega\tau)) \\ \dfrac{2}{\omega}(\cos(\omega\tau)-1) & \dfrac{1}{\omega}(4\sin(\omega\tau)-3\omega\tau) \\ \cos(\omega\tau) & 2\sin(\omega\tau) \\ -2\sin(\omega\tau) & (4\cos(\omega\tau)-3) \end{bmatrix} d\tau = \frac{1}{m_c}\begin{bmatrix} -\dfrac{\cos(\omega\tau)}{\omega^2} & \dfrac{2}{\omega^2}(\omega\tau - \sin(\omega\tau)) \\ \dfrac{2}{\omega^2}(\sin(\omega\tau)-\omega\tau) & -\dfrac{4}{\omega^2}\cos(\omega\tau) - \dfrac{3}{2}\tau^2 \\ \dfrac{\sin(\omega\tau)}{\omega} & -\dfrac{2}{\omega}\cos(\omega\tau) \\ \dfrac{2}{\omega}\cos(\omega\tau) & \dfrac{4}{\omega}\sin(\omega\tau) - 3\tau \end{bmatrix}\Bigg|_T
\end{aligned}
\tag{C.67}
$$

leading to:

$$
B_{k_{ip}} = \frac{1}{m_c}\begin{bmatrix} \dfrac{1}{\omega^2}(1-\cos(\omega T) & \dfrac{2}{\omega^2}(\omega T - \sin(\omega T)) \\ \dfrac{2}{\omega^2}(\sin(\omega T) - \omega T) & \dfrac{4}{\omega^2}(1-\cos(\omega T)) - \dfrac{3}{2}T^2 \\ \dfrac{\sin(\omega T)}{\omega} & \dfrac{2}{\omega}(1-\cos(\omega T)) \\ \dfrac{2}{\omega}(\cos(\omega T)-1) & \dfrac{4}{\omega}\sin(\omega T) - 3T \end{bmatrix}
\tag{C.68}
$$

For the out-of-plane motion, the input matrix will be:

$$B_{k_{op}} = \int_0^T \begin{bmatrix} \cos(\omega\tau) & \frac{1}{\omega}\sin(\omega\tau) \\ -\omega\sin(\omega\tau) & \cos(\omega\tau) \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{m_c} \end{bmatrix} d\tau = \frac{1}{m_c}\int_0^T \begin{bmatrix} \frac{1}{\omega}\sin(\omega\tau) \\ \cos(\omega\tau) \end{bmatrix} d\tau$$

$$B_{k_{op}} = \frac{1}{m_c} \left. \begin{bmatrix} -\frac{1}{\omega^2}\cos(\omega\tau) \\ \frac{\sin(\omega\tau)}{\omega} \end{bmatrix} \right|_0^T = \frac{1}{m_c} \begin{bmatrix} \frac{1}{\omega^2}(1-\cos(\omega T)) \\ \frac{\sin(\omega T)}{\omega} \end{bmatrix} \tag{C.69}$$

C.2.3 Computation of open-loop maneuvers

The CWH equations allow to easily compute the ΔV required to perform open loop maneuvers to approach the target. In this section we detail two methods, the *two impulses rendezvous*, and the *straight line approaches*. It should be reminded that these methods are just a preliminary tool to generate rendezvous trajectories. First, the so-computed maneuvers are affected by the intrinsic linearization error underlying the CWH equations. Secondly, chaser's thruster have a finite thrust, so that the impulsive maneuvers which are theoretically considered as instantaneous ΔV are actually maneuvers that have a finite duration. Thirdly, these approaches are not compliant with the safety requirements for rendezvous operation. Finally, as anticipated in Sec.1.2.1, during the final approach the maneuvers must be performed in closed loop. However, for the scope of testing navigation algorithms, these methods provide an useful tool to generate close proximity trajectories.

C.2.3.1 Two impulses rendezvous

By definition, a two impulses rendezvous is a rendezvous maneuver characterized by two impulsive burns, the first one putting the chaser on a trajectory that will "impact" the target after a given time-of-flight TOF, and the second one realized before the impact in order to bring the relative velocity of the spacecrafts to zero. The knowledge of the transition matrix allows computing the initial ΔV_0 and the final ΔV_f that the chaser thrusters must provide in order to perform the maneuvers. The expression in Eq.(C.60) can be rearranged naming

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad \mathbf{r_0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \quad \mathbf{v_0} = \begin{bmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{bmatrix}. \tag{C.70}$$

The CWH system can be written as

$$\begin{cases} \mathbf{r}(\tau) = & \Phi_{rr}(\tau)\mathbf{r}_0 + \Phi_{rv}(\tau)\mathbf{v}_0 \\ \mathbf{v}(\tau) = & \Phi_{vr}(\tau)\mathbf{r}_0 + \Phi_{vv}(\tau)\mathbf{v}_0 \end{cases} \tag{C.71}$$

where the matrices $\Phi_{rr}, \Phi_{rv}, \Phi_{vr}, \Phi_{vv}$ are composed by the rearranged elements of the matrices Φ_{ip}, Φ_{op}:

$$\Phi_{rr} = \begin{bmatrix} 4 - 3cos\omega\tau & 0 & 0 \\ 6\left(\sin(\omega\tau) - \omega\tau\right) & 1 & 0 \\ 0 & 0 & \cos\left(\omega\tau\right) \end{bmatrix}, \Phi_{rv} = \begin{bmatrix} \dfrac{\sin(\omega\tau)}{\omega} & \dfrac{2}{\omega}(1 - cos(\omega\tau)) & 0 \\ \dfrac{2}{\omega}(\cos(\omega\tau) - 1) & \dfrac{1}{\omega}(4\sin(\omega\tau) - 3\omega\tau) & 0 \\ 0 & 0 & \dfrac{\sin(\omega\tau)}{\omega} \end{bmatrix}$$

$$\Phi_{vr} = \begin{bmatrix} 3\omega\sin(\omega\tau) & 0 & 0 \\ 6\omega(\cos(\omega\tau) - 1) & 0 & 0 \\ 0 & 0 & -\omega\sin(\omega\tau) \end{bmatrix}, \Phi_{vv} = \begin{bmatrix} \cos(\omega\tau) & 2\sin(\omega\tau) & 0 \\ -2\sin(\omega\tau) & (4\cos(\omega\tau) - 3) & 0 \\ 0 & 0 & \cos(\omega\tau) \end{bmatrix} \tag{C.72}$$

Let's now imagine that the chaser is, at the the instant $t = t_0 = 0$, at a relative distance from the target $\mathbf{r} = \mathbf{r_0}$, with a relative velocity $\mathbf{v} = \mathbf{v_0}^-$. Using the STM, it is possible to determine the relative velocity $\mathbf{v} = \mathbf{v_0}^+$ (i.e., the chaser relative velocity after the first burn) that is needed at instant $t = t_0 = 0$ to reach the target after the TOF prescribed (i.e., $\mathbf{r}(TOF) = 0$ m).

$$\begin{aligned} \mathbf{r}(TOF) &= \Phi_{rr}(TOF)\mathbf{r}_0 + \Phi_{rv}(TOF)\mathbf{v}_0^+ = \mathbf{0} \\ \mathbf{v}_0^+ &= -\Phi_{rv}^{-1}(TOF)\Phi_{rr}(TOF)\mathbf{r}_0 \end{aligned} \tag{C.73}$$

The initial $\Delta V_0 = \mathbf{v_0}^+ - \mathbf{v_0}^-$ is therefore computed. Once that $\mathbf{v_0}^+$ is determined, also $\mathbf{v_f} = \mathbf{v}(ToF)$ can be computed using the second equation in (C.71):

$$\begin{aligned} \mathbf{v}(TOF) = \mathbf{v}_f^- &= \Phi_{vr}(TOF)\mathbf{r}_0 + \Phi_{vv}(TOF)\mathbf{v}_0^+ \\ &= \left[\Phi_{vr}(TOF) - \Phi_{vv}(TOF)\Phi_{rv}^{-1}(TOF)\Phi_{rr}(TOF)\right]\mathbf{r}_0 \end{aligned} \tag{C.74}$$

The final ΔV_f is the one that brings to zero the relative velocity (i.e., $\mathbf{v}_f^+ = 0$ m/s), and therefore $\Delta V_f = -\mathbf{v}_f^-$.

C.2.3.2 Straight line approaches

In this section the R-bar and V-bar constant velocity straight line approaches are briefly described. As anticipated in Sec.1.2.1, these maneuvers are performed (in closed loop) during the final approach before contact (e.g., last 100 meters).

Constant velocity straight line V-bar approach

Starting from an hold point along the V-bar (which is a stable point where the chaser can stay for an undetermined time with nominal zero relative velocity), a straight line approach is started providing to the chaser an initial tangential ΔV_{y_i} directed towards the target. To keep the chaser moving at constant speed V_y along the V-bar, a radial continuous acceleration γ_x must be provided. At the end of the maneuver, a second ΔV_{y_f} of equal magnitude and opposite direction of ΔV_{y_i} is performed to stop the motion (i.e., $\Delta V_{y_f} = -\Delta V_{y_i} = -V_y$). Being $\dot{y}(t) = V_y$, the profile of $y(t)$ can be integrated, resulting in $y(t) = y_0 + V_y\, t$. By definition of the maneuver, the other components (i.e., $x(t)$ and $z(t)$) are always null. Thus it is possible to write:

$$\begin{cases} x(t) = & 0 \\ y(t) = & y_0 + V_y\, t \\ z(t) = & 0 \end{cases}, \quad \begin{cases} \dot{x}(t) = & 0 \\ \dot{y}(t) = & V_y \\ \dot{z}(t) = & 0 \end{cases} \tag{C.75}$$

Then, using the Hill's equations in Eq.(C.38), the value of the radial acceleration γ_x can be computed:

$$\begin{aligned} \ddot{x} - 3\omega^2 x - 2\omega\dot{y} &= \gamma_x & \qquad \gamma_x &= 2\omega V_y \\ \ddot{y} + 2\omega\dot{x} &= \gamma_y \quad \Rightarrow & \gamma_y &= 0 \\ \ddot{z} + \omega^2 z &= \gamma_z & \gamma_z &= 0 \end{aligned} \tag{C.76}$$

Constant velocity straight line R-bar approach

As for the approach along the V-bar, the initial hypothesis is that the chaser is on a hold point at zero relative velocity with respect to the target. However, such an hold point is not stable and requires the application of a continuous thrust profile. By definition the maneuver is performed with a constant velocity profile $\dot{x}(t) = V_x$. The maneuver starts with a radial boost providing an initial $\Delta V_{x_i} = V_x$ and ends with a final radial boost ΔV_{x_f} in the opposite direction of ΔV_{x_i} (i.e., $\Delta V_{x_f} = -\Delta V_{x_i}$). To perform the straight line approach, accelerations both in x and y direction must be applied during the whole maneuver. Analogously to the V-bar case, it is possible to compute the values of these accelerations using Hill's equations:

$$\begin{cases} x(t) = & x_0 + V_x\, t \\ y(t) = & 0 \\ z(t) = & 0 \end{cases}, \quad \begin{cases} \dot{x}(t) = & V_x \\ \dot{y}(t) = & 0 \\ \dot{z}(t) = & 0 \end{cases} \Rightarrow \quad \begin{aligned} \gamma_x &= -3\omega^2(x_0 + V_x\, t) \\ \gamma_y &= 2\omega V_x \\ \gamma_z &= 0 \end{aligned} \tag{C.77}$$

The straight line R-bar approach is therefore obtained applying a continuous thrust profile along the V-bar and a thrust profile inversely proportional to the position x along the R-bar.

C.3 Absolute rotational dynamics

As anticipated in Sec.A.1, there are several representations for the attitude of a rigid body, which is expressed as the relative orientation between a coordinate frame attached to the body and a fixed or inertial coordinate frame. For the scope of this thesis, the quaternion representation according to the Hamiltonian convention has been chosen to represent the rotational dynamics of a S/C. Although the 3x3 orthogonal attitude matrix is the fundamental representation of the spacecraft's attitude, the orthogonality requirement imposes six constraints on its nine elements, reflecting the fact that the special orthogonal group $SO(3)$ of rotation matrices has dimension three. Therefore, employing the nine elements of the attitude matrix as components of a state vector in a Kalman Filter leads to some complexity in enforcing the constraints. However, all the three-parameter representations of $SO(3)$ are singular or discontinuous for certain attitudes. This suggests the use of higher-dimensional non-singular parameterizations of the attitude, such as the quaternion representation, which is a four-dimensional representation upon which the unit norm constraint is added. According to Eq.(A.10), quaternions respond to the kinematic relation:

$$\dot{q}_{i-SC} = \frac{1}{2} q_{i-SC} \otimes \begin{bmatrix} 0 \\ \omega_{i-SC}^{SC} \end{bmatrix} \tag{C.78}$$

where i is the inertial frame and SC is the spacecraft frame.

C.3.1 Kinematic model for the absolute rotational dynamics propagation

A simple kinematic model could be used to propagate the attitude of spacecrafts having a very low dynamics (i.e., $\omega_{i-SC}^{SC} \sim 0$). The equation governing the evolution of the rotation rate would be:

$$\dot{\omega}_{i-SC}^{SC} = 0 \tag{C.79}$$

The navigation filter modeled according to this model is referred to as *kinematic filter*. In order to compute the optimal estimate for such a system according to the Kalman theory, it is necessary to know the Jacobian $F(x)$ of the non linear function $f(x)$, according to the formulation:

$$x = \begin{bmatrix} q_{i-SC} \\ \omega_{i-SC}^{SC} \end{bmatrix}, \quad \dot{x} = f(x), \quad F(x) = \frac{\partial f(t,x)}{\partial x} \tag{C.80}$$

Reminding the notation introduced in Sec.A.2 and Eqs.(A.12),(A.13),(A.14), it is possible to write:

$$F(x) = \frac{1}{2} \begin{bmatrix} \mathbb{J}_{Q_{prod}(q_{i-SC}, w^{SC}_{i-SC})/q_{i-SC}} & \mathbb{J}_{Q_{prod}(q_{i-SC}, w^{SC}_{i-SC})/w^{SC}_{i-SC}} \\ \emptyset_{3\times4} & \emptyset_{3\times3} \end{bmatrix} \tag{C.81}$$

where $w^{SC}_{i-SC} = [0;\, \omega^{SC}_{i-SC}]$. The analytical formulation of these Jacobians is provided in Sec.A.2.

C.3.2 Dynamic model for the absolute rotational dynamics propagation

For objects characterized by relatively high rotation rate, it is necessary to include the angular momentum equation in order to enable the estimation of both the attitude and the rotation rate. It is convenient to express the angular momentum equation in a frame that is attached to the rotating body, because the inertia matrix of the body is constant if expressed in such a frame (i.e., $\dot{I}^{SC}_{SC} = 0$, without taking into account time-dependent variations due to events such as propellant consumption, solar array deployment and so on). To do so, it is necessary to recall that the derivatives of a vector v taken in two different frames A and B, with B in relative motion with respect to A are related by the following:

$$\frac{\mathrm{d}^A(v)}{\mathrm{d}t} = \frac{\mathrm{d}^B(v)}{\mathrm{d}t} + \omega_{A-B} \times v \tag{C.82}$$

where $\frac{\mathrm{d}^A(v)}{\mathrm{d}t}$ indicates the derivative of the vector v taken in frame A, and $\frac{\mathrm{d}^B(v)}{\mathrm{d}t}$ indicates the derivative of the vector v taken in frame B. From the second law of the dynamics, it is known that the derivative of the angular momentum Γ taken in the inertial frame i is equal to the summation of the external torques T acting on the system:

$$\frac{\mathrm{d}^i(\Gamma)}{\mathrm{d}t} = T \tag{C.83}$$

From Eq.(C.82) it is also possible to write:

$$\frac{\mathrm{d}^i(\Gamma)}{\mathrm{d}t} = \frac{\mathrm{d}^{SC}(\Gamma)}{\mathrm{d}t} + \omega_{i-SC} \times \Gamma \tag{C.84}$$

The angular momentum Γ, expressed in target frame, is given by $\Gamma = I^{SC}_{SC}\omega^{SC}_{i-SC}$. As the inertia matrix expressed in the spacecraft body frame is constant, equalling Eq.(C.84) with Eq.(C.83) it is possible to write:

$$I^{SC}_{SC}\dot{\omega}^{SC}_{i-SC} + \omega^{SC}_{i-SC} \times I^{SC}_{SC}\omega^{SC}_{i-SC} = T^{SC} \tag{C.85}$$

In order to lighten the notation, we will use only one index for the inertia matrix, based on the assumption that in this thesis we will always express the inertia of a body in the body axis, i.e., $I_{SC}^{SC} = I_{SC}$. Eq.(C.85) can be written both for the chaser and the target. The torques acting on the SCs are the perturbing torques for both the satellites, and the control torques T_{ctrl} for the chaser. However, only control torques are modeled, while the orbital disturbance torques that affect the spacecraft dynamics will be considered in the systems as process noises:

$$\begin{aligned} I_{ch}\dot{\omega}_{i-ch}^{ch} + \omega_{i-ch}^{ch} \times I_{ch}\omega_{i-ch}^{ch} &= T_{ctrl}^{ch} \\ I_{tg}\dot{\omega}_{i-tg}^{tg} + \omega_{i-tg}^{tg} \times I_{tg}\omega_{i-tg}^{tg} &= 0 \end{aligned} \tag{C.86}$$

The model governing the rotational dynamics of the target is therefore:

$$\begin{cases} \dot{q}_{i-tg} &= \frac{1}{2} q_{i-tg} \otimes \begin{bmatrix} 0 \\ \omega_{i-tg}^{tg} \end{bmatrix} \\ \dot{\omega}_{i-tg}^{tg} &= -I_{tg}^{-1} \left(\omega_{i-tg}^{tg} \times I_{tg}\, \omega_{i-tg}^{tg} \right) \end{cases} \tag{C.87}$$

The navigation filter modeled according to this non-linear propagation model is referred to as *dynamic filter*. In order to implement the filter equations, the Jacobian of the state function have to be computed:

$$F(x) = \frac{1}{2} \begin{bmatrix} \mathbb{J}_{Q_{prod}(q_{i-tg}, w_{i-tg}^{tg})/q_{i-tg}} & \mathbb{J}_{Q_{prod}(q_{i-tg}, w_{i-tg}^{tg})/w_{i-tg}^{tg}} \\ \emptyset_{3\times 4} & \mathbb{J}_{\dot{\omega}_{i-tg}^{tg}/\omega_{i-tg}^{tg}} \end{bmatrix} \tag{C.88}$$

The sub-matrix corresponding to the quaternion derivative lines are equal to the ones of the kinematic filter in Eq.(C.81), whose analytical formulation is provided in Sec.A.2. The computation of the new term $\mathbb{J}_{\dot{\omega}_{i-tg}^{tg}/\omega_{i-tg}^{tg}}$ is provided in the following. Reminding that the general cross product between two vectors $v \times w$ can be written in matrix form as:

$$v \times w = [v]_\times w = -[w]_\times v \tag{C.89}$$

the Jacobian of the cross product can be derived with the formula:

$$\mathbb{J}(v \times w) = [v]_\times \mathbb{J}(w) - [w]_\times \mathbb{J}(v) \tag{C.90}$$

Using Eq.(C.89) the expression of ω_{i-tg}^{tg} can be written as:

$$\dot{\omega}_{i-tg}^{tg} = -I_{tg}^{-1} \left([\omega_{i-tg}^{tg}]_\times I_{tg}\omega_{i-tg}^{tg} \right) = -I_{tg}^{-1} \left(-[I_{tg}\omega_{i-tg}^{tg}]_\times \omega_{i-tg}^{tg} \right) \tag{C.91}$$

and the Jacobian with respect to ω_{i-tg}^{tg} becomes:

$$\mathbb{J}_{\dot{\omega}_{i-tg}^{tg}/\omega_{i-tg}^{tg}} = -I_{tg}^{-1} \left([\omega_{i-tg}^{tg}]_\times \frac{\partial (I_{tg}\omega_{i-tg}^{tg})}{\partial \omega_{i-tg}^{tg}} - [I_{tg}\omega_{i-tg}^{tg}]_\times \frac{\partial \omega_{i-tg}^{tg}}{\partial \omega_{i-tg}^{tg}} \right) = -I_{tg}^{-1} \left([\omega_{i-tg}^{tg}]_\times I_{tg} - [I_{tg}\omega_{i-tg}^{tg}]_\times \right). \tag{C.92}$$

C.4 Relative rotational dynamics

In this section we will derive the model of the relative attitude q_{ch-tg} dynamics. Even if this model is not used in the navigation function because of its high non-linearity, it is of interest to derive the equation governing the relative dynamics. According to the convention introduced in Sec.A.1, it is possible to derive the expression of the relative quaternion q_{ch-tg} as a function of the absolute attitudes of the chaser and the target:

$$\begin{cases} x^{ch} & = q^*_{i-ch} \otimes x^i \otimes q_{i-ch} \\ & = q^*_{i-ch} \otimes q_{i-tg} \otimes x^{tg} \otimes q^*_{i-tg} q_{i-ch} \\ & = q_{ch-tg} \otimes x^{tg} \otimes q^*_{ch-tg} \end{cases} \tag{C.93}$$

and:

$$q_{ch-tg} \quad = q^*_{i-ch} \otimes q_{i-tg} \tag{C.94}$$

The derivative of the relative quaternion responds to the general rule introduced in Sec.A.1:

$$\dot{q}_{ch-tg} \quad = \frac{1}{2} q_{ch-tg} \otimes \omega^{tg}_{ch-tg} = \frac{1}{2} \omega^{ch}_{ch-tg} \otimes q_{ch-tg}. \tag{C.95}$$

where, in order to lighten the notation, when expressing the derivative of a quaternion q_{a-b} we use the expression ω^b_{a-b} instead of $[0; \omega^b_{a-b}]$. The same result can been obtained using the chain rule for derivation, noting that for the generic angular velocity ω^{SC}_{i-SC} it is possible to write that $(\omega^{SC}_{i-SC})^* = -\omega^{SC}_{i-SC}$:

$$\begin{aligned} \dot{q}_{ch-tg} = \quad & \frac{\mathrm{d}}{\mathrm{d}t}(q^*_{i-ch} \otimes q_{i-tg}) = \dot{q}^*_{i-ch} \otimes q_{i-tg} + q^*_{i-ch} \otimes \dot{q}_{i-tg} \\ = \quad & \frac{1}{2}\left(q_{i-ch} \otimes \omega^{ch}_{i-ch}\right)^* \otimes q_{i-tg} + \frac{1}{2} q^*_{i-ch} \otimes \left(q_{i-tg} \otimes \omega^{tg}_{i-tg}\right) \\ = \quad & \frac{1}{2}\left(-\omega^{ch}_{i-ch}\right) \otimes q^*_{i-ch} \otimes q_{i-tg} + \frac{1}{2}\left(q_{ch-tg} \otimes \omega^{tg}_{i-tg} \otimes q^*_{ch-tg}\right) \otimes q_{ch-tg} \\ = \quad & \frac{1}{2}\left(-\omega^{ch}_{i-ch}\right) \otimes q_{ch-tg} + \frac{1}{2}\omega^{ch}_{i-tg} \otimes q_{ch-tg} \\ = \quad & \frac{1}{2}\left(-\omega^{ch}_{i-ch} + \omega^{ch}_{i-tg}\right) \otimes q_{ch-tg} \\ = \quad & \frac{1}{2}\omega^{ch}_{ch-tg} \otimes q_{ch-tg} \end{aligned} \tag{C.96}$$

The derivation of the relative angular acceleration requires more steps. First, reminding Eq.(C.82), it is possible to write:

$$\frac{\mathrm{d}^i(\omega_{ch-tg})}{\mathrm{d}t} = \frac{\mathrm{d}^{ch}(\omega_{ch-tg})}{\mathrm{d}t} + (\omega_{i-ch} \times \omega_{ch-tg}). \tag{C.97}$$

Developing Eq.(C.97), and expressing all the vectors in the chaser RF, we can write:

$$\frac{\mathrm{d}^{ch}(\omega^{ch}_{ch-tg})}{\mathrm{d}t} = -\frac{\mathrm{d}^i(\omega^{ch}_{i-ch})}{\mathrm{d}t} + \frac{\mathrm{d}^i(\omega^{ch}_{i-tg})}{\mathrm{d}t} - (\omega^{ch}_{i-ch} \times \omega^{ch}_{ch-tg}). \tag{C.98}$$

At this point, an important property of the rotation rate needs to be introduced. Given the angular rate ω_{a-b}, from Eq.(C.82) we have:

$$\begin{aligned} \frac{\mathrm{d}^a(\omega_{a-b})}{\mathrm{d}t} &= \frac{\mathrm{d}^b(\omega_{a-b})}{\mathrm{d}t} + (\omega_{a-b} \times \omega_{a-b}) \\ \frac{\mathrm{d}^a(\omega_{a-b})}{\mathrm{d}t} &= \frac{\mathrm{d}^b(\omega_{a-b})}{\mathrm{d}t} \end{aligned}, \tag{C.99}$$

and therefore:

$$\begin{cases} \dfrac{\mathrm{d}^i(\omega_{i-ch})}{\mathrm{d}t} = \dfrac{\mathrm{d}^{ch}(\omega_{i-ch})}{\mathrm{d}t} \\ \dfrac{\mathrm{d}^i(\omega_{i-tg})}{\mathrm{d}t} = \dfrac{\mathrm{d}^{tg}(\omega_{i-tg})}{\mathrm{d}t} \end{cases}. \tag{C.100}$$

Reminding the angular momentum equations Eq.(C.86) for the chaser (i.e., expressed in chaser coordinates), and for the target (i.e., expressed in target coordinates) it is possible to write:

$$\begin{cases} \dfrac{\mathrm{d}^i(\omega_{i-ch}^{ch})}{\mathrm{d}t} = -I_{ch}^{-1}\left(\omega_{i-ch}^{ch} \times I_{ch}\omega_{i-ch}^{ch} - T_{ctrl}^{ch}\right) \\ \dfrac{\mathrm{d}^i(\omega_{i-tg}^{tg})}{\mathrm{d}t} = -I_{tg}^{-1}\left(\omega_{i-tg}^{tg} \times I_{tg}\omega_{i-tg}^{tg}\right) \end{cases}. \tag{C.101}$$

The second equation in the previous Eq.(C.101) can be expressed in the chaser frame according to:

$$\frac{\mathrm{d}^i(\omega_{i-tg}^{ch})}{\mathrm{d}t} = q_{ch-tg} \otimes \left[-I_{tg}^{-1}\left(\omega_{i-tg}^{tg} \times I_{tg}\omega_{i-tg}^{tg}\right)\right] \otimes q_{ch-tg}^{*}. \tag{C.102}$$

Then, using Eqs.(C.101) and (C.102), Eq.(C.98) becomes:

$$\dot{\omega}_{ch-tg}^{ch} = -I_{ch}^{-1}\left(\omega_{i-ch}^{ch} \times I_{ch}\omega_{i-ch}^{ch} - T_{ctrl}^{ch}\right) - q_{ch-tg} \otimes \left[-I_{tg}^{-1}\left(\omega_{i-tg}^{tg} \times I_{tg}\omega_{i-tg}^{tg}\right)\right] \otimes q_{ch-tg}^{*} - (\omega_{i-ch}^{ch} \times \omega_{ch-tg}^{ch}). \tag{C.103}$$

Appendix D

Insight on the tracking algorithm

Contents

D.1 Masking algorithm

In this section the code developed to project a 3D geometric model into the image frame in order to determine the visibility of its edges is presented. The code has two main functions. First, given the pose parameters, it projects the model in the image frame. Then, it determines a set of control point belonging to the model visible edges. In computer vision, there are two main families of algorithms used to determine the visibility of the objects in a scene, the "*Ray Tracing*" algorithms [Glassner, 1989], and the "*z-buffer*" algorithms [Greene et al., 1993]. The code developed for this thesis is derived from the second family of algorithms: the 3D model of the observed object at a given pose is projected in the image frame using the pinhole camera model, and the z components of the projected points are checked to determine visible and masked parts.

D.1.1 Projection of the model

The geometry of the 3D object to be projected, that we will refer to as target for sake of simplicity, is declared using a matrix V collecting the 3D coordinates of each vertex of the model, and a matrix S which assigns to each surface of the model the vertex that belong to it:

- Matrix V has size $n_{vertex} \times 3$, with n_{vertex} the number of vertex in the 3D model. The line i of matrix V corresponds to the coordinates of vertex v_i, expressed in the target reference frame. Vertex are defined by their position in matrix V, i.e., the vertex whose coordinates are in the 4^{th} row is referred to as vertex 4.

- Matrix S has size $n_{surf} \times v_{max}$, with n_{surf} the number of surfaces in the model, and v_{max} is the maximum number of vertex that any surface of the model has. For example, for a pentagonal prism, $v_{max} = 5$ since the basis have 5 vertex. The element S_{ij} is the j^{th} vertex of surface i. There is no rules in selecting the first vertex of a surface (i.e., S_{i1}). However, once that the first vertex has been selected, the following vertex must be declared following the perimeter of the surface, in the sense that defines the exiting normal to the surface according to the right hand rule. If a surface j has a number of vertex n_{v_j} lower than v_{max} vertex, the element S_{i1} is repeated $v_{max} - n_{v_j} + 1$ times. Surfaces are defined by their position in matrix S.

The code receives as input the last update of relative camera-target pose, which is composed by the estimated attitude quaternion q_{cam-tg} and the relative position of the target COM with respect to the camera, i.e., tr^{cam}_{cam-tg}. The projection of the 3D model is easily made by applying the following transformations to each vertex X_i^{tg} in target RF Firstly, each point is roto-translated in camera RF, according to:

$$X_i^{cam} = \begin{bmatrix} x_i^{cam} \\ y_i^{cam} \\ z_i^{cam} \end{bmatrix} = tr^{cam}_{cam-tg} + q_{cam-tg} \otimes X_i^{tg} \otimes q^*_{cam-tg} \tag{D.1}$$

Then the points are projected in image RF according to the pinhole projection model, i.e., Eq.(3.6):

$$m_i = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x \dfrac{x_i^{cam}}{z_i^{cam}} + c_x \\ \\ f_y \dfrac{y_i^{cam}}{z_i^{cam}} + c_y \\ \\ z_{scale}\, z_i^{cam} \end{bmatrix} \tag{D.2}$$

The component z_i^{cam} is multiplied by factor z_{scale} in order to make the order of magnitude of the w component equal to the order of magnitude of u and v. At this point, the 3D distorted coordinates of the model vertex are obtained and the geometry can be built. First, the normal n_i of the surface i is computed. Calling j, k, l three consecutive vertex appertaining to surface i, and being e_{jk} the edge relying point j to point k and e_{kl} the edge relying point k to point l, the normal can be computed as follows:

$$n_i = \frac{\text{cross}\left(e_{kl}, -e_{jk}\right)}{\left|\text{cross}\left(e_{kl}, -e_{jk}\right)\right|} \tag{D.3}$$

where the operator *cross*() denotes the cross product.

Then, all the edges belonging to the model are declared. Edges belonging to different surfaces and relying the same two points are counted once. A n_{edges} x 2 matrix E is defined, where n_{edges} is the total number of edges in the model. The element E_{i1} is the number corresponding to first vertex of edge i, and the element E_{i2} is the number corresponding to the second vertex (end) of edge i. Edges are defined by their position in matrix E. Then, using matrix V, S, and E, three matrix are built:

- SV, a matrix of size $n_{surf} \times n_{vertex}$ where the element SV_{ij} is equal to 1 if vertex j belongs to surface i and 0 otherwise.
- SE, a matrix of size $n_{surf} \times n_{edges}$ where the element SE_{ij} is equal to 1 if edge j belongs to surface i and 0 otherwise.
- EV, a matrix of size $n_{edges} \times n_{vertex}$ where the element EV_{ij} is equal to 1 if vertex j belongs to edge i and 0 otherwise.

D.1.2 Computation of visible edges

The logic of the algorithm is to compare a surface i with all the other surfaces in order to determine which edge of surface i is masked (totally or partially) by the other surfaces.

Since the 3D model has already been distorted according to the pinhole model, the visibility of the object is studied as it were hit by a parallel beam of light directed as z (i.e., the optical axis of the camera). Only the surfaces exposed to the “flow” are checked, i.e., those surfaces for which the condition dot(n_i, $[0,\ 0,\ 1]^T$) < 0 is valid, with n_i the normal vector of the surface.

The surfaces that satisfy this condition are the meshed. Two different meshes are done: a bi-dimensional mesh along the surface, and a mono-directional mesh along the perimeter of the surface. The set of points belonging to the mono-directional mesh is the set of candidate control points whose visibility has to be determined in order to use them in the

optimization process to determine the new pose parameters. The points belonging to the surface mesh are the points that can potentially mask the points of the first set.
At this point the real masking computation starts. Surface i is compared with each one of the j surfaces going from 1 to $i-1$. The aim is to understand if surface i is masking part of the edges of surface j and vice-versa, and this is done by checking the w coordinates of the points that are superposed in the plane uv.
First of all, Matlab function "*inpolygon*" is used to determine four families of points:

- pe_i, the points of the edges belonging to surface i that are inside the projected perimeter defined by the vertex of surface j.
- pe_j, the points of the edges belonging to surface j that are inside the projected perimeter defined by the vertex of surface i.
- ps_i, the meshed points of surface i that are inside the projected perimeter defined by the vertex of surface j.
- ps_j, the meshed points of surface j that are inside the projected perimeter defined by the vertex of i.

Four different situations can happen:

- All the families are empty: there is no conflict between surface i and j.
- The family ps_i (and pe_i) is composed only by points that are on the projected perimeter defined by the vertex of j (and vice versa for the families ps_j and pe_j): the surfaces share an edge (or part of it), and this do not generate any kind of masking
- One of the two surfaces is completely behind the other. This happens when the maximum w coordinate of one family (ps_i or pe_i) is lower than (or equal to) the minimum w coordinate of the other family (ps_j or pe_j):

$$\begin{cases} \max\Big(w(ps_i)\Big) & \leq \text{mix}\Big(w(ps_j)\Big) \quad i \text{ is masking } j \\ \max\Big(w(ps_j)\Big) & \leq \min\Big(w(ps_i)\Big) \quad j \text{ is masking } i \end{cases} \tag{D.4}$$

- None of these previous condition is satisfied, so there is a complex conflict between the two families and it is not possible to state that one family is completely behind the other. The check has to be done point by point.

The logic used to solve such a complex conflict is the following. For each point of pe_i the two closest neighbour in plane uv appertaining to ps_j are searched and their average w

coordinate is computed. If the w coordinate of pe_i point is lower, it means that the two ps_j neighbours are are masked by it. If the w coordinate of pe_i point is greater, the latter is masked. This procedure is repeated for all the pe_i points, and then for all pe_j points (which are checked with respect to the ps_i family).

For some complex geometry (e.g., geometries having very small surfaces that have to be compared to big ones) it is possible that one of the family ps_i or ps_j has less than 2 points, so that the search of the two closest neighbours would provide no solution: in these cases the search of the nearest neighbour is extended to the whole set of points of the meshed surface.

At the end of the iteration on the surfaces s_i, all the visible edges of the model have been detected and sampled. Their coordinates 3D in target RF and the corresponding 2D coordinates (u and v) in the image frame are known ad ready to be used in the tracking algorithm. However, in order to perform the matching (see Sec. D.2), also the bi-dimensional vector normal to the projected edge must be known. Since the endpoints of each edge are known, assuming that a is the first endpoint and b the second one, the normal assigned to all the points i belonging to that edge will be:

$$n_i = \begin{bmatrix} v_a - v_b \\ u_b - u_a \end{bmatrix} \tag{D.5}$$

The normal of the endpoints is set to be null since they belong to more than on edge. This means that vertex will not be used in the optimization procedure.

For a complex object having lot of surfaces, this algorithm can be very computationally heavy and it represents more than a half of the total time required for the whole tracking algorithm. The computation time strongly depends on the number of points processed, and therefore on the mesh density and the apparent dimension of the object (and therefore to its real dimensions and its relative distance with respect to the camera). Given a certain mesh density, the smaller and the further the target is, the fastest is the algorithm. In the same way, for a given distance and target dimension, the less dense is the mesh, the fastest is the algorithm. On the other hand, the denser is the mesh, the highest is the number of control points to be used in the optimizations procedure. Even if the performance of the tracking algorithm does not depend directly on the number of control points (since together with the number of control points also the number of possible outliers increases), it is always better to have a big set of control points ([Drummond and Cipolla, 2002] suggest to use at least 400 points). For a given distance (and target geometry) the goal is therefore to find a trade off between the latency of the masking algorithm and the performance of the optimization, and therefore the mesh density that ensure this trade off. An offline mesh density “scheduling” should be done in order to compute offline the optimal value of this parameter as the target-chaser distance varies.

D.1.3 Computation of the silhouette perimeter

An algorithm capable of detecting only the external perimeter of the target (the space/object boundaries) has been implemented to enable the tracking of the target silhouette perimeter.

The code relies on the same preliminary algorithm discussed in section D.1.1. The main difference with respect to the code discussed in section D.1.2 is the fact that surfaces will not be meshed, and only the mono-dimensional mesh along the edges of the surfaces satisfying the condition $\mathrm{dot}(n_i, [0,\ 0,\ 1]^T) < 0$ is done. If some points of the edges in surface i is superposed in the plane uv to some points of surface j (or vice-versa), it is not necessary to determine whether the edge is behind or ahead of surface j. In fact, it certainly do not belong to the external perimeter of the object, since its projection is "contained" in the projection of surface j. This matter of fact allows to avoid the two dimensional mesh of the surfaces and the resulting check point by point, which was the cause of the high computational load of the former masking algorithm. The algorithm will then proceed as follows:

Surface i is compared with each one of the j surfaces going from 1 to $i-1$, and "*inpolygon*" is used to determine two families of points:

- pe_i, the meshed points of the edges belonging to surface i that are inside the projected perimeter defined by the vertex of surface j
- pe_j, the meshed points of the edges belonging to surface j that are inside the projected perimeter defined by the vertex of surface i

As already explained, all the points belonging to families pe_i and pe_j are surely not on the external perimeter of the object, and then they are discarded. At the end of the iteration, only the points belonging to the external perimeter will be left, and they will be selected as control points.

D.2 Matching algorithm

The output of the masking algorithm is a set of 3D visible points in target reference frame (i.e., X_i^{tg}), their corresponding 2D coordinates projected in the image reference frame (i.e., m_i), and the corresponding two components normal to the edge containing the 2D control point (i;e., n_i). In order to match the points m_i with the edged extracted on the current image, only the 2D coordinates u_i, v_i of the projected points m_i and their corresponding normal vector n_i will be used.

The image used for the matching is the output of the Canny edge extraction on the grey-scale image (if all the visible geometrical edges are considered) or alternatively on the silhouette image (if only the edges belonging to the external perimeter of the target are considered). The image is in the form of a $p_u \times p_v$ binary matrix that we denotes I_c, where p_u is the number of pixels along u direction and p_v is the number of pixels along v direction. The elements of the matrix are equal to 1 where an edge has been found and 0 elsewhere. The matching of a model point m_i with a measured point m_i^{meas} in the image is found moving from m_i along n_i till a value equal to 1 is found. The main difficulty in the matching procedure is to deal with the discrete domain of search of the image, which goes in u direction from 1 to p_u and in v direction from 1 to p_v.

Let's take a projected control point m_i having coordinates:

$$m_i = \begin{bmatrix} u_i \\ v_i \end{bmatrix} \tag{D.6}$$

Its pixel coordinates in the measured image frame are:

$$m_0 = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \text{ceil}(u) \\ \text{ceil}(v) \end{bmatrix} \tag{D.7}$$

The normal vector n corresponding to this point is:

$$n = \begin{bmatrix} n_u \\ n_v \end{bmatrix} \tag{D.8}$$

As already anticipated in Section D.1, points belonging to corners (which could therefore be associated with more than one edge), are not taken into account and therefore their normal is set to $n = [0, 0]^T$. Once that the algorithm has checked that the control point is not a corner, the strategy distinguishes three different cases:

- $n_u \sim 0$, $n_v \neq 0$: vertical normal
- $n_u \neq 0$, $n_v \sim 0$: horizontal normal
- $n_u \neq 0$, $n_v \neq 0$: any other case

The first and the second cases are very easy to solve. In the first case the match is searched by keeping the value of $u = u_0$ constant and changing the value of v, which will be therefore $v = v_0 \pm j$ whit j positive integer satisfying the condition imposed by the tolerance d_{toll}:

$$\begin{aligned} j \in \mathbb{Z}^+ : &\ \frac{|(v_0 + j) - v_0| + |(v_0 - j) - v_0|}{2} < d_{toll} \\ : &\ j < d_{toll} \end{aligned} \tag{D.9}$$

The search is stopped if a matching is found (i.e $I_c(u, v) = 1$), if the condition on the tolerance is no more satisfied, or if the search exceed the limit of the image ($1 \leq u \leq p_u$, $1 \leq v \leq p_v$).

The same consideration can be done for the second case: the value of v is kept constant to $v = v_0$ and the search is performed along u axis ($u = u_0 \pm k$ whit k positive integer satisfying tolerance constraint) and it is stopped according to the same criteria already discussed.

$$\begin{aligned} k \in \mathbb{Z}^+ : &\ \frac{|(u_0 + k) - u_0| + |(u_0 - k) - u_0|}{2} < d_{toll} \\ : &\ k < d_{toll} \end{aligned} \tag{D.10}$$

The case with a generic normal which is not parallel to any axis is more complex. The straight line described by the normal vector is introduced. Its angular coefficient m_{uv} and offset q will be:

$$\begin{cases} m_{uv} & = \dfrac{n_v}{n_u} \\ q_{uv} & = v_0 - m_{uv} u_0; \end{cases} \tag{D.11}$$

The search is then performed starting from $(u, v) = (u_0,\, v_0)$ in the following way:

- at each iteration j, v is increased (or decreased) of the positive integer j: $v_p = v_0 + j$, $v_m = v_0 - j$
- for the image row defined by a value of v_p, all the value of u that are crossed by the straight line defined by m_{uv} and q_{uv} are searched to find a matching. For the value $v_p = v_0 + j$, if $1 \leq v_p \leq p_v$, two value of u are defined:

  $$\begin{cases} u_{p_1} & = \text{ceil}\left(\dfrac{v_p - 1 - q_{uv}}{m_{uv}}\right) \\ u_{p_2} & = \text{ceil}\left(\dfrac{v_p - q_{uv}}{m_{uv}}\right) \end{cases} \tag{D.12}$$

  The matching will be searched then for all the $u \in \mathbb{Z}$ satisfying the following conditions:

  $$\begin{cases} \min\left(u_{p_1}, u_{p_2}\right) \leq u_p \leq \max\left(u_{p_1}, u_{p_2}\right) \\ 1 \leq u_p \leq p_u \end{cases} \tag{D.13}$$

  For each couple $(u_p,\, v_p)$, the distance from the original point $(u_0,\, v_0)$ is computed in order to see if the constraint on the tolerance is satisfied:

  $$\sqrt{(u_p - u_0)^2 + (v_p - v_0)^2} < d_{toll} \tag{D.14}$$

The same search is done for $v_m = v_0 - j$. Two value of u are defined:

$$\begin{cases} u_{m_1} & = \text{ceil}\left(\dfrac{v_m - 1 - q}{m}\right) \\ u_{m_2} & = \text{ceil}\left(\dfrac{v_m - q}{m}\right) \end{cases} \tag{D.15}$$

The matching will be searched then for all the $u_m \in \mathbb{Z}$ satisfying:

$$\min\left(u_{m_1}, u_{m_2}\right) \leq u_m \leq \max\left(u_{m_1}, u_{m_2}\right) \tag{D.16}$$

under the constraints:

$$\begin{cases} \sqrt{(u_m - u_0)^2 + (v_m - v_0)^2} < d_{toll} \\ 1 \leq v_m \leq p_v \\ 1 \leq u_m \leq p_u \end{cases} \tag{D.17}$$

If, for the same value of j, a match is found both for the positive values v_p and for the negative values v_m, only the closest to u_0, v_0 is keept.

- if no match has been found, and the average distance $0.5\sqrt{(u_p - u_m)^2 + (v_p - v_m)^2}$ is lower than d_{toll} a new iteration start with j increased of a unit.

The points that will be used in the optimization are the control points that have found a matching. Therefore the number of points usable will be lower than the set of visible points provided by the masking algorithm.

This kind of matching procedure is not perfect and, even when the match is found in correspondence of a true image edge (and therefore the matched point is considered to be an inlier), three sources of errors are always present. First, errors intrinsic to the mono-directional search exist. Secondly, there is an error due to the differences between the 3D *a-priori* model implemented on the algorithm and the real geometry of the target. Thirdly, the discrete domain of the image introduces an error. This latter source of error is due to the fact that the coordinates of a control point can assume any value in the real domain, while the matched point has coordinates expressed in the integer domain. This means that it will be useless to exceed in the quantity of control points since two control points very close in the image frame will be likely matched to the same pixel.

D.3 Physical principle behind RAPiD linearization

Let's introduce the same notation used in 3.3.1, with tr the relative position camera-target in camera frame and R the target relative attitude rotation matrix at an instant t, and

X_i^{tg} the coordinates of a target point expressed in target RF:

$$tr = \begin{bmatrix} tr_x \\ tr_y \\ tr_z \end{bmatrix}, \quad R\,X_i^{tg} = \begin{bmatrix} R_{x_i} \\ R_{y_i} \\ R_{z_i} \end{bmatrix} \quad X_i^{tg} \begin{bmatrix} x_i^{tg} \\ y_i^{tg} \\ z_i^{tg} \end{bmatrix} \tag{D.18}$$

Let's now suppose that we also know the first derivatives of the pose parameters, i.e., the relative translational velocity vr_{cam-tg}^{cam}, named vr in the sequel, and the relative camera-target rotation rate in camera axis, i.e., ω_{cam-tg}^{cam}, named ω in the sequel:

$$vr = \begin{bmatrix} vr_x \\ vr_y \\ vr_z \end{bmatrix}, \quad \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{D.19}$$

Reminding Eq.(3.7), at any instant t the points can be projected in image reference frame:

$$\begin{cases} u(t) & = f_x \dfrac{tr_x + R_{x_i}}{tr_z + R_{z_i}} + c_x \\ v(t) & = f_y \dfrac{tr_y + R_{y_i}}{tr_z + R_{z_i}} + c_y \end{cases} \tag{D.20}$$

with tr and R that are function of time. Assuming that the same X_i^{tg} model point is tracked from the frame at instant t_0 to the frame at instant $t_0 + \delta t$, the trajectory of the point in the image frame at the instant $t_0 + \delta t$ can be approximated at the first order:

$$\begin{cases} u(t_0 + \delta t) = & u(t_0) + \left.\dfrac{\mathrm{d}u}{\mathrm{d}t}\right|_{t=t_0} \delta t \\ v(t_0 + \delta t) = & v(t_0) + \left.\dfrac{\mathrm{d}v}{\mathrm{d}t}\right|_{t=t_0} \delta t \end{cases} \tag{D.21}$$

where the first time derivative can be written as:

$$\begin{cases} \dfrac{\mathrm{d}u}{\mathrm{d}t} = \dfrac{\partial u}{\partial tr_x}\dfrac{\mathrm{d}tr_x}{\mathrm{d}t} + \dfrac{\partial u}{\partial tr_y}\dfrac{\mathrm{d}tr_y}{\mathrm{d}t} + \dfrac{\partial u}{\partial tr_z}\dfrac{\mathrm{d}tr_z}{\mathrm{d}t} + \dfrac{\partial u}{\partial R\,X_i^{tg}}\dfrac{\mathrm{d}R\,X_i^{tg}}{\mathrm{d}t} \\ \dfrac{\mathrm{d}v}{\mathrm{d}t} = \dfrac{\partial v}{\partial tr_x}\dfrac{\mathrm{d}tr_x}{\mathrm{d}t} + \dfrac{\partial v}{\partial tr_y}\dfrac{\mathrm{d}tr_y}{\mathrm{d}t} + \dfrac{\partial v}{\partial tr_z}\dfrac{\mathrm{d}tr_z}{\mathrm{d}t} + \dfrac{\partial v}{\partial R\,X_i^{tg}}\dfrac{\mathrm{d}R\,X_i^{tg}}{\mathrm{d}t} \end{cases} \tag{D.22}$$

Let's focus on the derivation of the u component. The computation for the partial derivative with respect to the translational terms is straightforward:

$$\begin{cases} \dfrac{\partial u}{\partial tr_x} & = f_x \dfrac{1}{tr_z + R_{z_i}} \\ \dfrac{\partial u}{\partial tr_x} & = 0 \\ \dfrac{\partial u}{\partial tr_z} & = -f_x \dfrac{tr_x + R_{x_i}}{(tr_z + R_{z_i})^2} \end{cases} \tag{D.23}$$

and the time derivative of vector tr is:

$$\begin{cases} \dfrac{\mathrm{d}tr_x}{\mathrm{d}t} &= vr_x \\ \dfrac{\mathrm{d}tr_y}{\mathrm{d}t} &= vr_y \\ \dfrac{\mathrm{d}tr_z}{\mathrm{d}t} &= vr_z \end{cases} \tag{D.24}$$

The derivation of the terms concerning the rotation is more complex. The vector $\dfrac{\partial u}{\partial R\,X_i^{tg}}$ is the 3 vector:

$$\frac{\partial u}{\partial R\,X_i^{tg}} = \left[\frac{\partial u}{\partial R_{x_i}}, \frac{\partial u}{\partial R_{y_i}}, \frac{\partial u}{\partial R_{z_i}}\right] \tag{D.25}$$

where, from Eq.(D.20), we compute:

$$\begin{cases} \dfrac{\partial u}{\partial R_{x_i}} &= f_x \dfrac{1}{tr_z + R_{z_i}} \\ \dfrac{\partial u}{\partial R_{y_i}} &= 0 \\ \dfrac{\partial u}{\partial R_{z_i}} &= -f_x \dfrac{tr_x + R_{x_i}}{(tr_z + R_{z_i})^2} \end{cases} \tag{D.26}$$

Reminding the expression of the derivative of a rotation matrix in Eq.(A.11) and the definition of the cross product matrix in Eq.(A.4), the time derivative of $R\,X_i^{tg}$ becomes:

$$\frac{\mathrm{d}R\,X_i^{tg}}{\mathrm{d}t} = \frac{\mathrm{d}R}{\mathrm{d}t}X_i^{tg} + R\frac{\mathrm{d}X_i^{tg}}{\mathrm{d}t} = [\omega]_\times RX_i^{tg} + R\,\emptyset_{3\times 1} \tag{D.27}$$

$$\frac{\mathrm{d}R\,X_i^{tg}}{\mathrm{d}t} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} R_{x_i} \\ R_{y_i} \\ R_{z_i} \end{bmatrix} = \begin{bmatrix} -\omega_z R_{y_i} + \omega_y R_{z_i} \\ -\omega_x R_{z_i} + \omega_z R_{x_i} \\ -\omega_y Px + \omega_x R_{y_i} \end{bmatrix} \tag{D.28}$$

The complete expression of the time derivative of u is:

$$\begin{aligned} \frac{\mathrm{d}u}{\mathrm{d}t} &= f_x \frac{1}{tr_z + R_{z_i}} vr_x - f_x \frac{tr_x + R_{x_i}}{(tr_z + R_{z_i})^2} vr_z + f_x \frac{-\omega_z R_{y_i} + \omega_y R_{z_i}}{(tr_z + R_{z_i})^2} - f_x \frac{(tr_x + R_{x_i})(-\omega_y R_{x_i} + \omega_x R_{y_i})}{(tr_z + R_{z_i})^2} \\ &= \frac{f_x}{(tr_z + R_{z_i})^2}\left[(tr_z + R_{z_i})(vr_x - \omega_z R_{y_i} + \omega_y R_{z_i}) - (tr_x + R_{x_i})(vr_z - \omega_y R_{x_i} + \omega_x R_{y_i})\right] \end{aligned} \tag{D.29}$$

The same computations can be done for the v component. The derivative with respect to the translation is:

$$\begin{cases} \dfrac{\partial v}{\partial tr_x} &= 0 \\ \dfrac{\partial v}{\partial tr_x} &= f_y \dfrac{1}{tr_z + Pz} \\ \dfrac{\partial v}{\partial tr_z} &= -f_y \dfrac{tr_y + R_{y_i}}{(tr_z + R_{z_i})^2} \end{cases} \tag{D.30}$$

and with respect to the rotation, being $\dfrac{\partial v}{\partial R\, X_i^{tg}}$ the 3×1 vector:

$$\frac{\partial v}{\partial R\, X_i^{tg}} = \left[\frac{\partial v}{\partial R_{x_i}}, \frac{\partial v}{\partial R_{y_i}}, \frac{\partial v}{\partial R_{z_i}} \right] \tag{D.31}$$

is:

$$\begin{cases} \dfrac{\partial v}{\partial R_{x_i}} & = 0 \\ \dfrac{\partial v}{\partial R_{y_i}} & = f_y \dfrac{1}{tr_z + Pz} \\ \dfrac{\partial v}{\partial R_{z_i}} & = -f_y \dfrac{tr_y + Py}{(tr_z + Pz)^2} \end{cases} \tag{D.32}$$

resulting in:

$$\begin{aligned} \frac{\mathrm{d}v}{\mathrm{d}t} &= f_y \frac{1}{tr_z + R_{z_i}} vr_y - f_y \frac{tr_y + R_{y_i}}{(tr_z + R_{z_i})^2} vr_z + f_y \frac{-\omega_x R_{z_i} + \omega_z R_{x_i}}{(tr_z + R_{z_i})^2} - f_x \frac{(tr_y + R_{y_i})(-\omega_y R_{x_i} + \omega_x R_{y_i})}{(tr_z + R_{z_i})^2} \\ &= \frac{f_y}{(tr_z + R_{z_i})^2} \left[(tr_z + R_{z_i})(vr_y - \omega_x R_{z_i} + \omega_z R_{x_i}) - (tr_y + R_{y_i})(vr_z - \omega_y R_{x_i} + \omega_x R_{y_i}) \right] \end{aligned} \tag{D.33}$$

Now, assuming for simplicity that at the instant t_0 the state variables tr_0, vr_0, R_0 and ω_0 are written as $[tr_x,\ tr_y,\ tr_z,\ vr_x,\ vr_y,\ vr_z, R,\ \omega_x,\ \omega_y,\ \omega_z]$, the linearized expression for $u(t_0 + \delta t)$ and $v(t_0 + \delta t)$ becomes:

$$\begin{cases} u(t_0 + \delta t) = & u(t_0) + \dfrac{f_x}{(tr_z + R_{z_i})^2} \left[(tr_z + R_{z_i})(vr_x - \omega_z R_{y_i} + \omega_y R_{z_i}) - (tr_x + R_{x_i})(vr_z - \omega_y R_{x_i} + \omega_x R_{y_i}) \right] \delta t \\ v(t_0 + \delta t) = & v(t_0) + \dfrac{f_y}{(tr_z + R_{z_i})^2} \left[(tr_z + R_{z_i})(vr_y - \omega_x R_{z_i} + \omega_z R_{x_i}) - (tr_y + R_{y_i})(vr_z - \omega_y R_{x_i} + \omega_x R_{y_i}) \right] \delta t \end{cases} \tag{D.34}$$

Moreover, noting that

$$\begin{cases} \delta tr_x = & vr_x \delta t \\ \delta tr_y = & vr_y \delta t \\ \delta tr_z = & vr_z \delta t \\ \delta \theta_x = & \omega_x \delta t \\ \delta \theta_y = & \omega_y \delta t \\ \delta \theta_z = & \omega_z \delta t \end{cases}, \tag{D.35}$$

the terms in Eq.(D.34) can be rearranged and, reminding the definition of the parameter δp and the coefficient matrix C_i from Eq.(3.14), one obtains:

$$\begin{bmatrix} u(t_0 + \delta t) \\ v(t_0 + \delta t) \end{bmatrix} = \begin{bmatrix} u(t_0) \\ v(t_0) \end{bmatrix} + C_i \delta p \tag{D.36}$$

From Eqs.(D.34) and (D.36) it is easy to notice that the linearization introduced in RAPiD algorithm (see Sec.3.3.1) corresponds to a linearization of the projected trajectory of the points X_i^{tg} in the image frame.

D.4 Levenberg-Marquardt algorithm

The Levenberg-Marquardt (LM) algorithm is an iterative technique that locates the minimum of a cost function expressed as the sum of the squares of non-linear real-valued functions. The LM algorithm has become a standard technique for non-linear least-squares problems, widely adopted in a broad spectrum of disciplines. The Levenberg-Marquardt algorithm is a combination of two optimization methods, the *gradient descent method*, where the sum of the squared errors is reduced by updating the parameters in the steepest-descent direction, and the *Gauss-Newton method*, where the cost function is assumed to be locally quadratic. In this section we will provide details of version of LM algorithm developed for the scope of this thesis.

Given a fitting function $\hat{y}(p)$ of a vector of N unknown parameters p, and a set of M data points y_i, it is convenient to minimize the sum of the weighted squares of the errors between the measured data y_i and the fitting function $\hat{y}(p)$. The resulting scalar cost function is called *chi-squared error criterion*:

$$\begin{aligned}\chi^2(p) &= \sum_i^m \left[\frac{y_i - \hat{y}(p)}{w_i}\right]^2 \\ &= (Y - \hat{Y}(p))^T W (Y - \hat{Y}(p)) \\ &= Y^T W Y - 2Y^T W \hat{Y} + \hat{Y}^T W \hat{Y}\end{aligned} \tag{D.37}$$

with W the diagonal weighting matrix where the element W_{ii} is equal to $1/\sigma^2_{y_i}$, i.e., the inverse of the square measurement error y_i. Being $\hat{Y}$ non linear, the minimization is done iteratively.

The *gradient descent method* searches the minima by moving in the direction opposite to the gradient of the objective function, which is the steepest direction. The gradient of the $\chi^2(p)$ function is:

$$\begin{aligned}\frac{\partial \chi^2(p)}{\partial p} &= -2(Y - \hat{Y}(p))^T W \frac{\partial \hat{Y}(p)}{\partial p} \\ &= -2(Y - \hat{Y}(p))^T W \mathbb{J}\end{aligned} \tag{D.38}$$

where $\mathbb{J}$ is the $M \times N$ Jacobian matrix of the function $\hat{Y}$ with respect to the parameter p. Starting from an initial guess p_0, at the iteration k the parameter update h_k is computed by moving in the direction of the steepest descent:

$$\begin{aligned}p_k &= p_{k-1} + h_k \\ h_k &= -\alpha \left(\frac{\partial \chi^2(p)}{\partial p}\right)^T \Bigg|_{p_{k-1}} \\ &= \alpha \mathbb{J}^T W (Y - \hat{Y}(p_{k-1}))\end{aligned} \tag{D.39}$$

where α is a positive scalar.

On the other side, the *Gauss-Newton method* approximates the cost function $\chi^2(p)$ around the optimal parameter $\hat{p}$ with a quadratic function. Using a first order Taylor expansion it is possible to write:

$$\begin{aligned} \hat{Y}(p+h) &\sim \hat{Y}(p) + \left[\frac{\partial \hat{Y}}{\partial p}\right] h \\ &\sim \hat{Y} + \mathbb{J}h \end{aligned} \tag{D.40}$$

Replacing the approximation $\hat{Y}(p+h) \sim \hat{Y} + \mathbb{J}h$ in the expression of $\chi^2(p+h)$:

$$\begin{aligned} \chi^2(p+h) &\sim Y^TWY - 2Y^TW\hat{Y}(p+h) + \hat{Y}(p+h)^TW\hat{Y}(p+h) \\ &\sim Y^TWY - 2Y^TW\hat{Y}(p) - 2Y^TW\mathbb{J}h + \hat{Y}(p)^TW\hat{Y}(p) + 2\hat{Y}(p)^TW\widehat{\mathbb{J}h} + h^T\mathbb{J}^TW\mathbb{J}h \\ &\sim Y^TWY + \hat{Y}(p)^TW\hat{Y}(p) - 2Y^TW\hat{Y}(p) - 2(Y-\hat{Y}(p))^TW\mathbb{J}h + h^T\mathbb{J}^TW\mathbb{J}h \end{aligned} \tag{D.41}$$

the expression of $\chi^2(p+h)$ becomes a quadratic function of the perturbation h, having approximated Hessian matrix equal to $\mathbb{J}^TW\mathbb{J}$. The value of h that minimizes the cost function is found by setting the first derivative of the approximated $\chi^2(p+h)$ to zero:

$$\frac{\partial \chi^2(p)}{\partial h} \sim \quad -2(Y-\hat{Y}(p))^TW\mathbb{J} + 2h^T\mathbb{J}^TW\mathbb{J} = 0 \tag{D.42}$$

which results in

$$\begin{aligned} h_k = &\quad \left((Y-\hat{Y}(p_{k-1}))^TW\mathbb{J}(\mathbb{J}^TW\mathbb{J})^{-1}\right)^T \\ = &\quad \mathbb{J}^TW(Y-\hat{Y}(p_{k-1}))(\mathbb{J}^TW\mathbb{J})^{-1} \end{aligned} \tag{D.43}$$

and

$$p_k = \quad p_{k-1} + h_k \tag{D.44}$$

It is possible to notice that the increment in the Gauss-Newton method is equal to the increment in the Gradient Descent method when the quantity $(\mathbb{J}^TW\mathbb{J})^{-1}$ is equal to $\alpha I_{N\times N}$, where $I_{N\times N}$ is the $N \times N$ identity matrix.

The LM algorithm is capable of adaptively varying the parameter update between the one provided by the Gauss-Newton method and the Gradient Descent method thanks to the λ parameter:

$$h = \mathbb{J}^TW(Y-\hat{Y}(p_{k-1}))(\mathbb{J}^TW\mathbb{J} + \lambda\mathbf{I})^{-1} \tag{D.45}$$

If λ, which is called *damping parameter*, is equal to 0, the LM update will be equal to the Gauss-Newton update, while if the value of λ is large, the LM update will tend to the Gradient Descent update. The value of λ is increased only if an iteration results in a worst fitting of the cost function, i.e., $\chi(p+h) > \chi(p)$. Otherwise, as solution improves, the damping parameter is decreased, approaching to the Gauss-Newton method.

The algorithm used in this work implements a version of LM proposed by [Gavin, 2013] and [Nielsen et al., 1999], where the quantity $\rho_k(h)$, called metric, is introduced:

$$\rho_k(h) \quad = \frac{\chi^2(p) - \chi^2(p+h)}{h^T(\lambda h + \mathbb{J}^T W(Y - \hat{Y}(p))} \tag{D.46}$$

The metric is a measure of the improvement the update h gives to χ^2. If ρ_k is greater than a positive threshold ϵ_4, it means that the updated parameter $p_k + h_k$ is sufficiently better than p_k, so that p_{k+1} is set to $p_k + h_k$ and the value of λ is decreased. Otherwise the parameter p is left unchanged (i;e., $p_{k+1} = p_k$) and λ is increased.
The convergence criteria are the ones suggested by [Gavin, 2013]. The iterations are stopped if one of the following conditions is satisfied:

o Convergence of the gradient: $\max|\mathbb{J}WY - \hat{Y})| < \epsilon_1$

o Convergence of the parameters: $\max|h_i/p_i| < \epsilon_2$

o Convergence of the reduced cost function: $\chi^2/(M - N + 1) < \epsilon_3$

where M is the size of the measurements vector and N the size of the parameters vector. The values of ϵ_1, ϵ_2, ϵ_3, ϵ_4 are user-specified. If no criterion is satisfied, the algorithm stops when the iterations have reached a maximal threshold.

Bibliography

[Abu-Mostafa and Psaltis, 1984] Abu-Mostafa, Y. S. and Psaltis, D. (1984). Recognitive aspects of moment invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:698–706.

[Alexander, 1991] Alexander, H. L. (1991). State estimation for distributed systems with sensing delay. In *Data Structures and Target Classification*, volume 1470, pages 103–111. International Society for Optics and Photonics.

[Allen et al., 2008] Allen, A. C., Langley, C., Mukherji, R., Taylor, A. B., Umasuthan, M., and Barfoot, T. D. (2008). Rendezvous lidar sensor system for terminal rendezvous, capture, and berthing to the international space station. In *Sensors and Systems for Space Applications II*, volume 6958, page 69580S. International Society for Optics and Photonics.

[Anselmo and Pardini, 2008] Anselmo, L. and Pardini, C. (2008). Space debris mitigation in geosynchronous orbit. *Advances in Space Research*, 41(7):1091–1099.

[Arbter et al., 1990] Arbter, K., Snyder, W. E., Burkhardt, H., and Hirzinger, G. (1990). Application of affine-invariant fourier descriptors to recognition of 3-d objects. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):640–647.

[Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

[Bar-Itzhack and Oshman, 1985] Bar-Itzhack, I. and Oshman, Y. (1985). Attitude determination from vector observations: Quaternion estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 1:128–136.

[Barczak et al., 2011] Barczak, A. L., Gilman, A., Reyes, N. H., and Susnjak, T. (2011). Analysis of feature invariance and discrimination for hand images: Fourier descriptors versus moment invariants. In *International Conference Image and Vision Computing New Zealand IVCNZ2011*.

[Barrow et al., 1977] Barrow, H. G., Tenenbaum, J. M., Bolles, R. C., and Wolf, H. C. (1977). Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, pages 21–27. Science Applications, Inc Arlington, VA.

[Benedict, 2013] Benedict, B. L. (2013). Rationale for need of in-orbit servicing capabilities for geo spacecraft. In *AIAA SPACE 2013 Conference and Exposition*, page 5444.

[Benninghoff et al., 2014] Benninghoff, H., Rems, F., and Boge, T. (2014). Development and hardware-in-the-loop test of a guidance, navigation and control system for on-orbit servicing. *Acta astronautica*, 102:67–80.

[Bombardelli and Pelaez, 2011] Bombardelli, C. and Pelaez, J. (2011). Ion beam shepherd for contactless space debris removal. *Journal of guidance, control, and dynamics*, 34(3):916–920.

[Bonnal et al., 2013] Bonnal, C., Ruault, J.-M., and Desjean, M.-C. (2013). Active debris removal: Recent progress and current trends. *Acta Astronautica*, 85:51–60.

[Breuers, 1999] Breuers, M. G. (1999). Image-based aircraft pose estimation using moment invariants. In *Automatic Target Recognition IX*, volume 3718, pages 294–304. International Society for Optics and Photonics.

[Carmi and Oshman, 2007] Carmi, A. and Oshman, Y. (2007). On the covariance singularity of quaternion estimators. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6814.

[Carpenter and D'Souza, 2018] Carpenter, J. R. and D'Souza, C. N. (2018). Navigation filter best practices. -.

[Chang and Ghosh, 2000] Chang, K.-y. and Ghosh, J. (2000). Three-dimensional model-based object recognition and pose estimation using probabilistic principal surfaces. In *Applications of Artificial Neural Networks in Image Processing V*, volume 3962, pages 192–203. International Society for Optics and Photonics.

[Chen et al., 2004] Chen, Q., Petriu, E., and Yang, X. (2004). A comparative study of fourier descriptors and hu's seven moment invariants for image recognition. In *Canadian conference on electrical and computer engineering 2004 (IEEE Cat. No. 04CH37513)*, volume 1, pages 103–106. IEEE.

[Chen and Ho, 1991] Chen, Z. and Ho, S.-Y. (1991). Computer vision for robust 3d aircraft recognition with fast library search. *Pattern recognition*, 24(5):375–390.

[Christian and Cryan, 2013] Christian, J. A. and Cryan, S. (2013). A survey of lidar technology and its use in spacecraft relative navigation. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, page 4641.

[CONFERS, 2018] CONFERS (2018). Satellite servicing safety framework technical and operational guidance document. https://www.satelliteconfers.org/publications/. Online; [retrieved 06 December 2019].

[Conseil et al., 2007] Conseil, S., Bourennane, S., and Martin, L. (2007). Comparison of fourier descriptors and hu moments for hand posture recognition. In *2007 15th European Signal Processing Conference*, pages 1960–1964. IEEE.

[Crassidis and Junkins, 2011] Crassidis, J. L. and Junkins, J. L. (2011). *Optimal estimation of dynamic systems*. CRC press.

[Curtis, 2013] Curtis, H. D. (2013). *Orbital mechanics for engineering students*. Butterworth-Heinemann.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.

[Drummond and Cipolla, 2002] Drummond, T. and Cipolla, R. (2002). Real-time visual tracking of complex structures. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):932–946.

[Dudani et al., 1977] Dudani, S. A., Breeding, K. J., and McGhee, R. B. (1977). Aircraft identification by moment invariants. *IEEE transactions on computers*, 100(1):39–46.

[D'Amico et al., 2013] D'Amico, S., Ardaens, J.-S., Gaias, G., Benninghoff, H., Schlepp, B., and Jørgensen, J. (2013). Noncooperative rendezvous using angles-only optical navigation: system design and flight results. *Journal of Guidance, Control, and Dynamics*, 36(6):1576–1595.

[E Roberts and Harkness, 2007] E Roberts, P. C. and Harkness, P. G. (2007). Drag sail for end-of-life disposal from low earth orbit. *Journal of Spacecraft and Rockets*, 44(6):1195–1203.

[ESA, 2003] ESA (2003). Geostationary servicing, https://www.esa.int/enabling_support/space_engineering_technology/automation_and_robotics/geostationary_servicing. Online; accessed 10-December-2019.

[ESA, 2012] ESA (2012). Conventional celestial reference system, https://gssc.esa.int/navipedia/index.php/conventional_celestial_reference_system. Online; accessed 30-September-2020.

[ESA, 2015a] ESA (2015a). Esa space debris mitigation compliance verification guidelines, issue 1.

[ESA, 2015b] ESA (2015b). Mitigating space debris generation, https://www.esa.int/safety_security/space_debris/mitigating_space_debris_generation. Online; accessed 10-December-2019.

[ESA, 2016] ESA (2016). Multispectral sensing for relative navigation, https://www.esa.int/enabling_support/space_engineering_technology/shaping_the_future/multispectral_sensing_for_relative_navigation. Online; accessed 04-Novembre-2020.

[ESA, 2017] ESA (2017). What is a launching state, http://blogs.esa.int/cleanspace/2017/06/13/what-is-a-launching-state. Online; accessed 2-December-2019.

[ESA, 2018] ESA (2018). Esa's e.deorbit debris removal mission reborn as servicing vehicle, https://www.esa.int/safety_security/clean_space/esa_s_e.deorbit_debris_removal_mission_reborn_as_servicing_vehicle. Online; accessed 10-December-2019.

[ESA, 2019a] ESA (2019a). Esa's annual space environment report. -.

[ESA, 2019b] ESA (2019b). From adr to in-orbit servicing: The shifted trajectory of e.deorbit, part three, http://blogs.esa.int/cleanspace/2019/01/09/from-adr-to-in-orbit-servicing-the-shifted-trajectory-of-e-deorbit-part-three. Online; accessed 20-December-2019.

[Evans, 1990] Evans, R. (1990). Kalman filtering of pose estimates in applications of the rapid video rate tracker. In *BMVC*, pages 1–6.

[Falkenhayn, 1988] Falkenhayn, JR, E. (1988). Multimission modular spacecraft (mms). In *Space Programs and Technologies Conference*, page 3513.

[Fehse, 2003] Fehse, W. (2003). Autonomous rendezvous and docking of spacecraft.

[Flegel et al., 2009] Flegel, S., Gelhaus, J., Wiedemann, C., Vorsmann, P., Oswald, M., Stabroth, S., Klinkrad, H., and Krag, H. (2009). The master-2009 space debris environment model. In *Fifth European Conference on Space Debris*, volume 672.

[Flusser, 2000] Flusser, J. (2000). On the independence of rotation moment invariants. *Pattern recognition*, 33(9):1405–1410.

[Flusser and Suk, 2006] Flusser, J. and Suk, T. (2006). Rotation moment invariants for recognition of symmetric objects. *IEEE Transactions on Image Processing*, 15(12):3784–3790.

[Flusser et al., 2009] Flusser, J., Zitova, B., and Suk, T. (2009). *Moments and moment invariants in pattern recognition.* John Wiley & Sons. pp 7,13-30,186-197.

[Frey and Dueck, 2007] Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *science*, 315(5814):972–976.

[Frogerais et al., 2012] Frogerais, P., Bellanger, J.-J., and Senhadji, L. (2012). Various ways to compute the continuous-discrete extended kalman filter. *IEEE Transactions on Automatic Control*, 57(4):1000.

[Fu and Sun, 2017] Fu, T. and Sun, X. (2017). The relative pose estimation of aircraft based on contour model. In *International Conference on Optical and Photonics Engineering (icOPEN 2016)*, volume 10250, page 102502T. International Society for Optics and Photonics.

[Gandía and Casas, 2003] Gandía, F. and Casas, A. G. (2003). Fast spacecraft pose estimation based on zernike moments. In *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Nara, Japan*, pages 19–23.

[Gavin, 2011] Gavin, H. (2011). The levenberg-marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, pages 1–15.

[Gavin, 2013] Gavin, P. H. (2013). *The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems.* Duke University.

[Gelb, 1974] Gelb, A. (1974). *Applied optimal estimation.* MIT press.

[Gibbs, 2011] Gibbs, B. P. (2011). *Advanced Kalman filtering, least-squares and modeling: a practical handbook.* John Wiley & Sons.

[Gishkori and Mulgrew, 2019] Gishkori, S. and Mulgrew, B. (2019). Pseudo-zernike moments based sparse representations for sar image classification. *IEEE Transactions on Aerospace and Electronic Systems*, 55(2):1037–1044.

[Glais and Ayoun, 1994] Glais, T. and Ayoun, A. (1994). Image-based air target identification. In *Applications of Digital Image Processing XVII*, volume 2298, pages 540–551. International Society for Optics and Photonics.

[Glassner, 1989] Glassner, A. S. (1989). *An introduction to ray tracing*. Elsevier.

[Gopalakrishnan et al., 2011] Gopalakrishnan, A., Kaisare, N. S., and Narasimhan, S. (2011). Incorporating delayed and infrequent measurements in extended kalman filter based nonlinear state estimation. *Journal of Process Control*, 21(1):119–129.

[Graham and Kingston, 2015] Graham, A. R. and Kingston, J. (2015). Assessment of the commercial viability of selected options for on-orbit servicing (oos). *Acta Astronautica*, 117:38–48.

[Greene et al., 1993] Greene, N., Kass, M., and Miller, G. (1993). Hierarchical z-buffer visibility. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 231–238. ACM.

[GSFC, 2010] GSFC (2010). On-orbit satellite servicing study project report. *NASA Project Report NP-2010-08-162-GSFC, Greenbelt, MD.*

[Harris and Stennett, 1990] Harris, C. and Stennett, C. (1990). Rapid-a video rate object tracker. In *BMVC*, pages 1–6.

[He et al., 2018] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2018). Mask r-cnn. arXiv:1703.06870v3.

[Heinrich et al., 2015] Heinrich, S., Leglise, F., Harrison, L., Renard, F., and Nold, O. (2015). Trade-off atmospheric re-entry: Design for demise vs controlled re-entry. In *Space Safety is No Accident*, pages 423–435. Springer.

[Hibbard, 1996] Hibbard, R. L. (1996). Satellite on-orbit refueling: A cost effectiveness analysis. Technical report, Naval Postgraduate School Monterey CA.

[Hinterstoisser et al., 2010] Hinterstoisser, S., Lepetit, V., Ilic, S., Fua, P., and Navab, N. (2010). Dominant orientation templates for real-time detection of texture-less objects. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2257–2264. IEEE.

[Hsiao et al., 1996] Hsiao, F.-H., Pan, S.-T., et al. (1996). Robust kalman filter synthesis for uncertain multiple time-delay stochastic systems. *Journal of dynamic system measurement and control-Transaction of the ASME*, 118(4):803–808.

[Hu, 1962] Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187.

[Huang and Leng, 2010] Huang, Z. and Leng, J. (2010). Analysis of hu's moment invariants on image scaling and rotation. In *2010 2nd International Conference on Computer Engineering and Technology*, volume 7, pages V7–476. IEEE.

[Hwang and Kim, 2006] Hwang, S.-K. and Kim, W.-Y. (2006). A novel approach to the fast computation of zernike moments. *Pattern Recognition*, 39(11):2065–2076.

[IADC, 2002] IADC (2002). Inter-agency space debris coordination committee space debris mitigation; 2002; revised 2007.

[ISU, 2007] ISU (2007). Doctor: Developing on-orbit servicing concepts. *International Space University, Summer Session Program.*

[Jasiobedski et al., 2001] Jasiobedski, P., Greenspan, M., and Roth, G. (2001). *Pose determination and tracking for autonomous satellite capture.* National Research Council of Canada.

[Jasiobedzki et al., 1999] Jasiobedzki, P., Abraham, M., Newhook, P., and Talbot, J. (1999). Model based pose estimation for autonomous operations in space. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pages 211–215. IEEE.

[Jasiobedzki et al., 2005] Jasiobedzki, P., Se, S., Pan, T., Umasuthan, M., and Greenspan, M. (2005). Autonomous satellite rendezvous and docking using lidar and model based vision. In *Spaceborne Sensors II*, volume 5798, pages 54–66. International Society for Optics and Photonics.

[Kaszkurewicz and Bhaya, 1996] Kaszkurewicz, E. and Bhaya, A. (1996). Discrete-time state estimation with two counters and measurement delay. In *Proceedings of 35th IEEE Conference on Decision and Control*, volume 2, pages 1472–1476. IEEE.

[Kelley, 2012] Kelley, R. (2012). Using the design for demise philosophy to reduce casualty risk due to reentering spacecraft. -.

[Kelsey et al., 2006] Kelsey, J. M., Byrne, J., Cosgrove, M., Seereeram, S., and Mehra, R. K. (2006). Vision-based relative pose estimation for autonomous rendezvous and docking. In *Aerospace Conference, 2006 IEEE*, pages 20–pp. IEEE.

[Kessler et al., 2010] Kessler, D. J., Johnson, N. L., Liou, J., and Matney, M. (2010). The kessler syndrome: implications to future space operations. *Advances in the Astronautical Sciences*, 137(8):2010.

[Khotanzad and Hong, 1990] Khotanzad, A. and Hong, Y. H. (1990). Invariant image recognition by zernike moments. *IEEE Transactions on pattern analysis and machine intelligence*, 12(5):489–497.

[Khotanzad and Liou, 1996] Khotanzad, A. and Liou, J.-H. (1996). Recognition and pose estimation of unoccluded three-dimensional objects from a two-dimensional perspective view by banks of neural networks. *IEEE Transactions on Neural networks*, 7(4):897–906.

[Kisantal et al., 2020] Kisantal, M., Sharma, S., Park, T. H., Izzo, D., Märtens, M., and D'Amico, S. (2020). Satellite pose estimation challenge: Dataset, competition design and results. *IEEE Transactions on Aerospace and Electronic Systems.*

[Klinkrad, 1991] Klinkrad, H. (1991). Discos-esa's database and information system characterising objects in space. *Advances in Space Research*, 11(12):43–52.

[Koay, 2011] Koay, C. G. (2011). Analytically exact spiral scheme for generating uniformly distributed points on the unit sphere. *Journal of computational science*, 2(1):88–91.

[Kreisel, 2002] Kreisel, J. (2002). On-orbit servicing of satellites (oos): its potential market & impact. In *proceedings of 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation 'ASTRA.*

[Kreisel, 2003] Kreisel, J. (2003). On-orbit servicing (oos): Issues & commercial implications. In *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, pages IAA–3.

[Kuffner, 2004] Kuffner, J. J. (2004). Effective sampling and distance metrics for 3d rigid body path planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 4, pages 3993–3998. IEEE.

[Kuhl and Giardina, 1982] Kuhl, F. P. and Giardina, C. R. (1982). Elliptic fourier features of a closed contour. *Computer graphics and image processing*, 18(3):236–258.

[Kulikov and Kulikova, 2014] Kulikov, G. Y. and Kulikova, M. V. (2014). Accurate numerical implementation of the continuous-discrete extended kalman filter. *IEEE Transactions on Automatic Control*, 59(1):273–279.

[Larbi et al., 2017] Larbi, B., Grzesik, B., Radtke, J., Trentlage, C., and Stoll, E. (2017). Active debris removal for mega constellations: Cubesat possible. In *9th international workshop on satellite constellations and formation flying, Boulder, CO, USA*, pages 19–21.

[Larsen et al., 1998] Larsen, T. D., Andersen, N. A., Ravn, O., and Poulsen, N. K. (1998). Incorporation of time delayed measurements in a discrete-time kalman filter. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171)*, volume 4, pages 3972–3977. IEEE.

[Larson and Wertz, 1992] Larson, W. J. and Wertz, J. R. (1992). Space mission analysis and design. Technical report, Torrance, CA (United States); Microcosm, Inc.

[Lentaris et al., 2019] Lentaris, G., Stratakos, I., Stamoulias, I., Soudris, D., Lourakis, M., and Zabulis, X. (2019). High-performance vision-based navigation on soc fpga for spacecraft proximity operations. *IEEE Transactions on Circuits and Systems for Video Technology*.

[Lepetit et al., 2005] Lepetit, V., Fua, P., et al. (2005). Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends® in Computer Graphics and Vision*, 1(1):1–89.

[Lepetit et al., 2009] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155.

[Letizia, 2016] Letizia, F. (2016). *Space debris cloud evolution in Low Earth Orbit.* PhD thesis, University of Southampton.

[Letizia et al., 2016] Letizia, F., Colombo, C., Lewis, H. G., and Krag, H. (2016). Assessment of breakup severity on operational satellites. *Advances in Space Research*, 58(7):1255–1274.

[Li et al., 2004] Li, R., Corripio, A. B., Henson, M. A., and Kurtz, M. J. (2004). On-line state and parameter estimation of epdm polymerization reactors using a hierarchical extended kalman filter. *Journal of process control*, 14(8):837–852.

[Lichter and Dubowsky, 2004] Lichter, M. D. and Dubowsky, S. (2004). State, shape, and parameter estimation of space objects from range images. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2974–2979. IEEE.

[Liou et al., 2013] Liou, J., Anilkumar, A., Virgili, B. B., Hanada, T., Krag, H., Lewis, H., Raj, M., Rao, M., Rossi, A., and Sharma, R. (2013). Stability of the future leo environment–an iadc comparison study. In *Proceedings of the 6th European Conference on Space Debris*, volume 723.

[Liou, 2011] Liou, J.-C. (2011). An active debris removal parametric study for leo environment remediation. *Advances in Space Research*, 47(11):1865–1876.

[Liou and Johnson, 2009] Liou, J.-C. and Johnson, N. L. (2009). A sensitivity study of the effectiveness of active debris removal in leo. *Acta Astronautica*, 64(2-3):236–243.

[Liou et al., 2010] Liou, J.-C., Johnson, N. L., and Hill, N. (2010). Controlling the growth of future leo debris populations with active debris removal. *Acta Astronautica*, 66(5-6):648–653.

[Liu et al., 2016] Liu, L., Zhao, G., and Bo, Y. (2016). Point cloud based relative pose estimation of a satellite in close range. *Sensors*, 16(6):824.

[Long et al., 2007] Long, A., Richards, M., and Hastings, D. E. (2007). On-orbit servicing: a new value proposition for satellite design and operation. *Journal of Spacecraft and Rockets*, 44(4):964–976.

[Losekamm et al., 2015] Losekamm, M. J., Hacker, J., Sardesi, N., Peculjic, A., and Vigneron, A. (2015). Legal and political implications of future on-orbit servicing missions. In *International Astronautical Congress, Jerusalem, Israel*.

[Lourakis and Zabulis, 2017] Lourakis, M. and Zabulis, X. (2017). Model-based visual tracking of orbiting satellites using edges. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3791–3796. IEEE.

[Lücking et al., 2013] Lücking, C., Colombo, C., and McInnes, C. R. (2013). Solar radiation pressure-augmented deorbiting: passive end-of-life disposal from high-altitude orbits. *Journal of Spacecraft and Rockets*, 50(6):1256–1267.

[Mallick, 2018] Mallick, S. (2018). Shape matching using hu moments. https://www.learnopencv.com/shape-matching-using-hu-moments-c-python/. Online; [retrieved 23 April 2020].

[Markley, 2003] Markley, F. L. (2003). Attitude error representations for kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317.

[Markley, 2004] Markley, F. L. (2004). Attitude estimation or quaternion estimation? *Journal of Astronautical Sciences*, 52(1):221–238.

[Markley and Crassidis, 2014] Markley, F. L. and Crassidis, J. L. (2014). *Fundamentals of spacecraft attitude determination and control*, volume 33. Springer.

[Martin et al., 2005] Martin, C., Cheeses, J., Sanchez-Ortiz, N., Klinkrad, H., Bunte, K., Hauptmann, S., Fritsche, B., and Lips, T. (2005). Introducing the esa drama tool. *Science and Technology Series*, 110:219–233.

[Mazzoni, 2008] Mazzoni, T. (2008). Computational aspects of continuous–discrete extended kalman-filtering. *Computational Statistics*, 23(4):519–539.

[Miravet et al., 2008] Miravet, C., Pascual, L., Krouch, E., and del Cura, J. M. (2008). An image-based sensor system for autonomous rendez-vous with uncooperative satellites. *arXiv preprint arXiv:0807.4478*.

[Nielsen et al., 1999] Nielsen, H. B. et al. (1999). *Damping parameter in Marquardt's method*. IMM.

[OpenCV development team, 2019a] OpenCV development team (2019a). Geometric transformations of images. https://docs.opencv.org/master/da/d6e/tutorial_py_geometric_transformations.html. Online; [retrieved 24 November 2020].

[OpenCV development team, 2019b] OpenCV development team (2019b). Shape matching using hu moments. https://docs.opencv.org/2.4/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html. Online; [retrieved 28 April 2020].

[Opromolla et al., 2015a] Opromolla, R., Fasano, G., Rufino, G., and Grassi, M. (2015a). A model-based 3d template matching technique for pose acquisition of an uncooperative space object. *Sensors*, 15(3):6360–6382.

[Opromolla et al., 2015b] Opromolla, R., Fasano, G., Rufino, G., and Grassi, M. (2015b). Uncooperative pose estimation with a lidar-based system. *Acta Astronautica*, 110:287–297.

[Opromolla et al., 2017a] Opromolla, R., Fasano, G., Rufino, G., and Grassi, M. (2017a). Pose estimation for spacecraft relative navigation using model-based algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 53(1):431–447.

[Opromolla et al., 2017b] Opromolla, R., Fasano, G., Rufino, G., and Grassi, M. (2017b). A review of cooperative and uncooperative spacecraft pose determination techniques for close-proximity operations. *Progress in Aerospace Sciences*, 93:53–72.

[Otiniano-Rodrıguez et al., 2012] Otiniano-Rodrıguez, K., Cámara-Chávez, G., and Menotti, D. (2012). Hu and zernike moments for sign language recognition. In *Proceedings of international conference on image processing, computer vision, and pattern recognition*, pages 1–5.

[Ott et al., 2011] Ott, T., Benoit, A., Van den Braembussche, P., and Fichter, W. (2011). Esa pointing error engineering handbook. In *8th International ESA Conference on Guidance, Navigation & Control Systems.*

[Persoon and Fu, 1977] Persoon, E. and Fu, K.-S. (1977). Shape discrimination using fourier descriptors. *IEEE Transactions on systems, man, and cybernetics*, 7(3):170–179.

[Pesce et al., 2019] Pesce, V., Opromolla, R., Sarno, S., Lavagna, M., and Grassi, M. (2019). Autonomous relative navigation around uncooperative spacecraft based on a single camera. *Aerospace Science and Technology*, 84:1070–1080.

[Petit, 2013] Petit, A. (2013). *Robust visual detection and tracking of complex objects: applications to space autonomous rendez-vous and proximity operations.* PhD thesis, Université Rennes 1.

[Petit et al., 2011] Petit, A., Marchand, E., and Kanani, K. (2011). Vision-based space autonomous rendezvous: A case study. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 619–624. IEEE.

[Petit et al., 2012] Petit, A., Marchand, E., and Kanani, K. (2012). Tracking complex targets for space rendezvous and debris removal applications. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4483–4488. IEEE.

[Petit et al., 2013] Petit, A., Marchand, E., and Kanani, K. (2013). Détection et suivi basé modèle pour des applications spatiales. In *Congrès francophone des jeunes chercheurs en vision par ordinateur, ORASIS'13*, pages 1–6.

[Pittelkau, 2003] Pittelkau, M. E. (2003). An analysis of the quaternion attitude determination filter. *The Journal of the astronautical sciences*, 51(1):103–120.

[Polites, 1998] Polites, M. E. (1998). An assessment of the technology of automated rendezvous and capture in space.

[Popova and Schaus, 2018] Popova, R. and Schaus, V. (2018). The legal framework for space debris remediation as a tool for sustainability in outer space. *Aerospace*, 5(2):55.

[Prasad et al., 2002] Prasad, V., Schley, M., Russo, L. P., and Bequette, B. W. (2002). Product property and production rate control of styrene polymerization. *Journal of Process Control*, 12(3):353–372.

[Prata and Rusch, 1989] Prata, A. and Rusch, W. (1989). Algorithm for computation of zernike polynomials expansion coefficients. *Applied Optics*, 28(4):749–754.

[Reeves et al., 1988] Reeves, A. P., Prokop, R. J., Andrews, S. E., and Kuhl, F. P. (1988). Three-dimensional shape analysis using moments and fourier descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):937–943.

[Rems et al., 2017] Rems, F., Risse, E., and Benninghoff, H. (2017). Rendezvous gnc-system for autonomous orbital servicing of uncooperative targets. In *Proceedings of the 10th International ESA Conference on Guidance, Navigation and Control Systems, Salzburg, Austria.*

[Rondao and Aouf, 2018] Rondao, D. and Aouf, N. (2018). Multi-view monocular pose estimation for spacecraft relative navigation. In *2018 AIAA Guidance, Navigation, and Control Conference*, page 2100.

[Rondao et al., 2020] Rondao, D., Aouf, N., Richardson, M. A., and Dubanchet, V. (2020). Robust on-manifold optimization for uncooperative space relative navigation with a single camera. *arXiv preprint arXiv:2005.07110.*

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Sharma et al., 2018a] Sharma, S., Beierle, C., and D'Amico, S. (2018a). Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. In *2018 IEEE Aerospace Conference*, pages 1–12. IEEE.

[Sharma and D'Amico, 2020] Sharma, S. and D'Amico, S. (2020). Neural network-based pose estimation for noncooperative spacecraft rendezvous. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–1.

[Sharma et al., 2016] Sharma, S. et al. (2016). Comparative assessment of techniques for initial pose estimation using monocular vision. *Acta Astronautica*, 123:435–445.

[Sharma et al., 2018b] Sharma, S., Ventura, J., and D'Amico, S. (2018b). Robust model-based monocular pose initialization for noncooperative spacecraft rendezvous. *Journal of Spacecraft and Rockets*, 55(6):1414–1429.

[Shen et al., 2000] Shen, D., Ip, H. H.-S., and Teoh, E. K. (2000). A novel theorem on symmetries of 2d images. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 3, pages 1002–1005. IEEE.

[Shepherd and Command, 2006] Shepherd, L. C. G. and Command, A. F. S. (2006). Space surveillance network. In *Proceedings of the Space Situational Awareness Conference, Colorado Springs, CO, USA*.

[Simon, 2006] Simon, D. (2006). *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons.

[Sola, 2015] Sola, J. (2015). Quaternion kinematics for the error-state kf. Technical report, Laboratoire d'Analyse et d'Architecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep.

[Spaceflight Now, 2020] Spaceflight Now (2020). Two commercial satellites link up in space for first time, https://spaceflightnow.com/2020/02/26/two-commercial-satellites-link-up-in-space-for-first-time/. Online; accessed 03-Novembre-2020.

[Sullivan et al., 2017] Sullivan, J., Grimberg, S., and D'Amico, S. (2017). Comprehensive survey and assessment of spacecraft relative motion dynamics models. *Journal of Guidance, Control, and Dynamics*, 40(8):1837–1859.

[Taati and Greenspan, 2005] Taati, B. and Greenspan, M. (2005). Satellite pose acquisition and tracking with variable dimensional local shape descriptors. In *Proc. IEEE/RSJ IROS 2005, Workshop Robot Vision for Space Applications*, pages 4–9.

[Tahri, 2004] Tahri, O. (2004). Application des moments a l'asservissement visuel et au calcul de pose. *These de doctorat, Université de Rennes*, 1.

[Teague, 1980] Teague, M. R. (1980). Image analysis via the general theory of moments. *JOSA*, 70(8):920–930.

[Teh and Chin, 1988] Teh, C.-H. and Chin, R. T. (1988). On image analysis by the methods of moments. *IEEE Transactions on pattern analysis and machine intelligence*, 10(4):496–513.

[Trawny and Roumeliotis, 2005] Trawny, N. and Roumeliotis, S. I. (2005). Indirect kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005.

[UN, 1967a] UN (1967a). Agreement on the rescue of astronauts, the return of astronauts and the return of objects launched into outer space; 672 unts 119; entered into force 3 december 1968.

[UN, 1967b] UN (1967b). Treaty on principles governing the activities of states in the exploration and use of outer space, including the moon and other celestial bodies;610 unts 205; entered into force on 10 october 1967.

[UN, 1972] UN (1972). Convention on international liability for damage caused by space objects; 961 unts 187; entered into force 1 september 1972.

[UN, 1974] UN (1974). Convention on registration of objects launched into outer space; 1023 unts 15; entered into force 15 september 1976.

[UN, 1979] UN (1979). Agreement governing the activities of states on the moon and other celestial bodies; 1363 unts 13; entered into force 11 july 1984.

[Wallace and Wintz, 1980] Wallace, T. P. and Wintz, P. A. (1980). An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing*, 13(2):99–126.

[Wallin and Kubler, 1995] Wallin, Å. and Kubler, O. (1995). Complete sets of complex zernike moment invariants and the role of the pseudoinvariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1106–1110.

[Wee et al., 2004] Wee, C.-Y., Paramesran, R., and Takeda, F. (2004). New computational methods for full and subset zernike moments. *Information Sciences*, 159(3-4):203–220.

[Wertz and Bell, 2003] Wertz, J. R. and Bell, R. (2003). Autonomous rendezvous and docking technologies: status and prospects. In *Space Systems Technology and Operations*, volume 5088, pages 20–31. International Society for Optics and Photonics.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[Wormnes et al., 2013] Wormnes, K., Le Letty, R., Summerer, L., Schonenborg, R., Dubois-Matra, O., Luraschi, E., Cropp, A., Krag, H., and Delaval, J. (2013). Esa technologies for space debris remediation. In *6th European Conference on Space Debris*, volume 1, pages 1–8. ESA Communications ESTEC, Noordwijk, The Netherlands.

[Xu et al., 2009] Xu, Y.-L., Horstman, M., Krisko, P., Liou, J.-C., Matney, M., Stansbery, E., Stokely, C., and Whitlock, D. (2009). Modeling of leo orbital debris populations for ordem2008. *Advances in Space Research*, 43(5):769–782.

[Yamanaka and Ankersen, 2002] Yamanaka, K. and Ankersen, F. (2002). New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of guidance, control, and dynamics*, 25(1):60–66.

[Zahn and Roskies, 1972] Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on computers*, 100(3):269–281.

[Zanetti and Bishop, 2006] Zanetti, R. and Bishop, R. (2006). Quaternion estimation and norm constrained kalman filtering. In *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, page 6164.

[Zhang et al., 2002] Zhang, D., Lu, G., et al. (2002). A comparative study of fourier descriptors for shape representation and retrieval. In *Proc. 5th Asian Conference on Computer Vision*, page 35. Citeseer.

[Zhou et al., 2012] Zhou, J., Jiang, Z., and Tang, G. (2012). A new approach for teleoperation rendezvous and docking with time delay. *Science China Physics, Mechanics and Astronomy*, 55(2):339–346.

[Zisserman, 1995] Zisserman, M. A. A. (1995). Robust object tracking. In *Proceedings of the Asian Conference on Computer Vision, Singapore*, pages 5–8.