Visual Servoing of the Orientation of an Earth Observation Satellite

Maxime Robic

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Automatique, productique, robotique*

Par
Maxime ROBIC

Visual Servoing of the Orientation of an Earth Observation Satellite

Thèse présentée et soutenue à Rennes, le 12 décembre 2023
Unité de recherche : Centre Inria de l'Université de Rennes

Rapporteurs avant soutenance :

Viviane CADENAT — Maître de conférences à l'Université Paul Sabatier
Pascal VASSEUR — Professeur à l'Université Picardie Jules Verne

Composition du Jury :

Président :	Paolo ROBUFFO GIORDANO	Directeur de recherche CNRS, IRISA
Rapporteurs :	Viviane CADENAT	Maître de conférences à l'Université Paul Sabatier
	Pascal VASSEUR	Professeur à l'Université Picardie Jules Verne
Examinateurs :	Guillaume ALLIBERT	Professeur à l'Université Côte d'Azur
Dir. de thèse :	François CHAUMETTE	Directeur de recherche Inria, IRISA
Co-dir. de thèse :	Eric MARCHAND	Professeur à l'Université de Rennes
Invité :	Renaud FRAISSE	Ingénieur Airbus D&S

REMERCIEMENTS

On y est ! Avant de commencer ces remerciements, il m'est important de souligner que la thèse est avant tout une aventure humaine, et que je suis si heureux d'avoir choisi ce chemin pour toutes ces rencontres qui m'auront fait grandir, d'un point de vue scientifique bien sûr, mais aussi d'un point de vue plus personnel. Pour être honnête, ces trois ans n'ont pas été un long fleuve tranquille, mais le fait d'avoir été entouré de personnes bienveillantes et vraies, a rendu la traversée plus qu'agréable.

Je tiens avant tout à remercier Éric et François, mes encadrants de luxe, deux grands scientifiques mais surtout deux très belles personnes qui m'auront énormément appris. Je réalise la chance que j'aie eue d'avoir pu les déranger chaque jour avec de nouvelles questions plus ou moins pertinentes. Ce fut un réel honneur et un grand plaisir pour moi d'avoir pu marcher à leurs côtés. Je tiens également à remercier Renaud, mon encadrant à Airbus D&S, pour son accessibilité et pour m'avoir mis des étoiles dans les yeux.

Je souhaite remercier mon jury Viviane, Pascal, et Guillaume pour avoir évalué mes travaux de doctorat et s'être déplacé pour partager ce moment avec moi. Merci enfin à Paolo pour avoir présidé ce jury, et avoir été un chef d'équipe au top, avec qui j'ai pu échanger énormément, notamment sur Star Wars...

J'aimerais d'ailleurs remercier cette équipe, la meilleure équipe de l'univers, j'ai nommé la Rainbow Team. Je pense ne surprendre personne lorsque je dis que j'ai vraiment passé des moments incroyables au sein de cette équipe. Chacun de ses membres m'a un jour apporté quelque chose et je les en remercie tellement. Je ne peux citer tout le monde (à l'oral, c'était plus simple !) ils ne m'en voudront pas, cependant j'aimerais tout de même remercier Fabien, Claudio, et Hélène pour leur gentillesse, leur accessibilité et pour l'aide, les conseils et le soutien qu'ils m'auront offerts en continu. Merci également à Vincent, Alexandre et Marie qui m'auront toujours accordé des discussions agréables et positives. Enfin, merci à Marco pour son dynamisme pour continuer à faire vivre cette chouette équipe.

J'ai également pu rencontrer dans cette équipe des personnes qui ont partagé mon quotidien pendant et au-delà des heures de labo, j'aimerais remercier mes co-bureaux Thomas, Lev et Pierre (le king de l'impression), mais également Samuel qui était par moments plus dans ce bureau que dans le sien, à ma rescousse pour debugger certaines de mes créations douteuses...

Je souhaiterais bien évidemment remercier Nicola, avec qui j'ai dû passer le plus de temps dans cette équipe, que ce soit au labo, au bar, au Tiffany's (surtout au Tiffany's en fait) et un peu au foot aussi. Je voudrais remercier Antonio qui est vite devenu un incontournable de mon quotidien, un pilier sur lequel je me suis beaucoup appuyé. J'aimerais aussi remercier Esteban aka Dr. Petit Monsieur, qui n'est d'ailleurs pas si petit que ça, il faudrait penser à arrêter cette blague à un moment donné, tout ça à cause de Salva et Tommasso... Merci à Maxime, Thibault et John, avec qui j'aurai vécu le Japon, et des moments que je n'aurais jamais pensé vivre durant ma thèse. Merci les gars.

J'avais dit que je ne citerai pas tout le monde mais bon comment ne pas le faire ? Merci Marco, merci Elodie, merci Alberto, merci Katja, merci Ali, merci Fouad, merci Erwan, merci Lendy, merci Inès, , merci Louise, merci Diane, merci Joudy, merci Lorenzo, merci Lisheng, merci Olivier, merci Mandela, merci Danilo, merci Massi, merci Francesca, merci Riccardo... merci à tous ceux qui auront partagé cette aventure avec moi, de près ou de loin, plus ou moins longtemps, vous faites partie de moi maintenant, je ne vous oublierai pas.

Je voudrais également remercier la Beauce Team au complet, qui aura suivi mon périple jusqu'au bout, je ne leur dis pas souvent mais je ne sais pas ce que je ferais sans eux. Parmi eux, mon pote Vincent, qui sait déjà au combien je suis reconnaissant de l'avoir à mes côtés. Je souhaite également remercier la team ENSMA (dont je tairai le véritable nom) et Loric qui aura vécu sa propre aventure à Toulouse, mais auquel j'avais toujours une pensée.

Enfin, j'aimerais remercier ma famille, en commençant par ma petite Marie-Valentine qui aura vécu cette aventure en première ligne avec moi, qui m'aura soutenu tout du long et qui m'aura très souvent fait sortir la tête de l'eau, et enfin avec qui j'aurais pu profiter de tous les moments cool de cette thèse, qui n'auraient jamais été aussi joyeux sans elle. J'aimerais également remercier Caroline et Olivier, ses parents, qui ont été d'une gentillesse et générosité sans égale avec moi. J'aimerais remercier toute ma famille, des Robic au Niel, mon papa, ma grand-mère, mon petit-frère, et aussi papi, mamie et Mimi auxquelles je pense tous les jours,

mes oncles et mes tantes, et plus spécialement à Marie-Laurence qui m'aura beaucoup aidé et soutenu, et à qui j'envoie énormément de force, et enfin à mes cousins. Mais bien sûr, je souhaite remercier également les Thion, et plus spécialement mon beau-père Christian, qui m'aura offert un cadre chaleureux auquel je pouvais me réfugier quand je le souhaitais.

Et bien sûr, merci à ma petite maman, mon phare dans le brouillard, la nuit, la tempête, qui aura été mon carburant principal pour accomplir tout ce chemin, qui a été présente à chaque moment, qui le sera toujours. Merci d'avoir construit tout cet environnement bienveillant autour de moi, de m'avoir transmis toutes ces belles valeurs qui m'auront finalement permis d'approcher les étoiles. Tout ça c'est notre victoire. Merci Maman.

RÉSUMÉ

Depuis le début de l'ère spatiale et le lancement des premiers satellites en orbite, les satellites artificiels se sont multipliés en nombre, diversifiés dans leurs applications et perfectionnés dans leurs performances. Suite au lancement de *Spoutnik 1* en 1957 pour démontrer la capacité de l'humanité à accéder au domaine orbital de la Terre et à en tirer parti, plus de 10 000 engins spatiaux sont maintenant en orbite autour de notre planète. Ces satellites sont dotés de missions variées et passionnantes, allant de la démonstration scientifique à la télécommunication, de la géolocalisation à l'observation. Cette dernière catégorie comprend des satellites équipés de capteurs à longue portée pour mesurer et enregistrer des informations de la surface de la Terre, fournissant des données à grande échelle, permettant un suivi global et précis de la planète.

Les satellites d'observation de la Terre bénéficient de leur orbite, une trajectoire induite autour de la Terre qui est souvent considérée comme leur principal propulseur naturel. Elle est déterminée en grande partie lors du lancement et permet aux satellites de balayer une partie de la surface terrestre à un moment spécifique. Les missions nécessitant une capacité de revisite rapide, c'est-à-dire d'une durée faible entre deux passages d'une même zone d'intérêt, se tourneront vers des orbites relativement basses et sont appelées satellites en orbite terrestre basse (LEO). Parmi eux, les satellites d'observation équipés d'un capteur de vision, ou satellites imageurs, tirent avantages de l'orbite basse en acquérant des images de la Terre avec une fréquence élevée et une grande résolution. Par exemple, Pléiades HR, une constellation de deux satellites d'observations sur une orbite quasicirculaire héliosynchrone (695 km), acquiert des images de la Terre avec une résolution de 70 cm et un intervalle de revisite inférieur à 24 heures pour chaque satellite. Le système d'acquisition d'images Pléiades utilise une technologie de balayage linéaire ou "push-broom", une ligne de capteurs permet l'acquisition d'une bande colorée qui est ensuite reconstituée en image.

L'acquisition nécessite de pointer précisément le capteur du satellite vers la surface de la Terre, ce qui est fait par la commande de l'orientation du satellite, communément appelé contrôle d'attitude. Lors d'une mission d'observation, un ensemble désiré d'attitudes est opérée par le satellite grâce à une loi de commande embarquée. Cependant, cette tâche peut être assez difficile en orbite basse, car le mouvement orbital du satellite et d'autres perturbations externes deviennent plus marquantes près de la Terre, ce qui entraîne une dérive de l'attitude et des in-

exactitudes de pointage. De plus, ces attitudes désirées, également appelées guidage d'attitudes, ne sont pas supposées changer pendant l'acquisition, car étant souvent calculées au préalable par un segment sol. La commande de l'orientation d'un satellite visant à effectuer un pointage d'une zone précise sur Terre est donc généralement un processus en boucle ouverte.

Cependant, un guidage en temps réel de l'attitude à partir des informations d'un capteur visuel pourrait être pertinent pour corriger la propagation d'incertitudes de pointage. De plus, il pourrait offrir plus de fonctionnalités aux satellites d'observation de la Terre, telles que la modification rapide de l'emplacement cible et surtout la définition de la mission d'observation en termes d'information visuelle, offrant encore plus de flexibilité et permettant des missions plus complexes, par exemple le suivi d'une cible mobile depuis le satellite, qui reste encore un problème ouvert.

Cette thèse s'inscrit dans le cadre du projet LICHIE, une collaboration scientifique avec Airbus Defence & Space visant à développer des satellites reposant sur un nouveau système d'acquisition appelé "Starer", traduit littéralement par "Regard fixe". L'instrument est constitué d'une matrice de capteurs permettant l'acquisition en temps réel d'images. Airbus DS prévoit d'incorporer cette nouvelle technologie dans la future constellation LION, composée de plusieurs satellites vidéos agiles. Ainsi, la "caméra" du satellite pourra être utilisée en tant que capteur d'entrée pour que le guidage d'attitude puisse être effectué en temps réel, en utilisant des images satellite d'une résolution exceptionnelle (jusqu'à 50 cm au sol). Dans cette thèse, nous démontrerons qu'une commande précise de l'orientation du satellite peut être conçue par asservissement visuel.

L'asservissement visuel est une approche de commande basée vision en boucle fermée qui résulte de l'association entre l'automatique et la vision par ordinateur. Il est désormais très présent dans le domaine de la robotique et a pu prouver son efficacité et sa robustesse en étant largement utilisé pour des tâches de manipulation et pour du suivi visuel. Finalement, un satellite LION peut être modélisé par une caméra commandable en rotation et intégrée dans un véhicule extrêmement rapide dont le mouvement n'est pas controlé, suivant un objet mobile. Si nous pouvons ensuite définir sa mission d'observation comme une tâche d'asservissement visuel, il devient possible de commander précisément son orientation. C'est l'hypothèse principale de notre travail.

En robotique spatiale, l'asservissement visuel a déjà été utilisé pour accomplir des missions en orbite, mais son application pour les satellites d'observations pose encore de nombreuses problématiques. Parmi celles-ci, nous traitons le mouvement relatif entre le satellite et la cible qui est bien plus conséquent que lors de l'interaction entre deux satellites sur une même orbite.

Vis-à-vis des quelques stratégies existantes pour les satellites d'observations, celles-ci considèrent des résolutions au sol moins importantes qui ne les confrontent pas aux mêmes ordres de précisions et par conséquent pas aux mêmes solutions pour leur commande. De plus, nous envisageons le fait que la localisation pointée puisse changer au cours de la mission, soit par la redéfinition de cette localisation, soit par le suivi d'un objet en mouvement. Dans les deux cas, la cible est définie à l'aide de mesures dans l'image, ce qui représente également une différence notable par rapport aux satellites classiques de pointage terrestre. Les contraintes mécaniques et dynamiques du satellite ne sont pas non plus négligées, cette thèse vise à proposer des solutions personnalisées pour les gérer dans la loi de commande.

En parlant de contraintes, l'une des plus importantes lorsque l'on travaille avec un satellite imageur, où chaque acquisition est importante, est celle de la qualité de l'image. Mais comment la qualité d'image peut-elle être quantifiable ? Une caractéristique remarquable est la netteté de l'image, qui peut être estimée à travers des métriques spécifiques. Dans notre contexte, la netteté est dégradée par le flou de mouvement induit par le mouvement de la caméra et celui des objets en mouvement dans la scène que nous souhaitons observer.

Cette perte de qualité n'est pas souhaitable du point de vue de l'utilisateur désireux d'exploiter ces images, mais aussi pour la robustesse de la commande, car les commandes visuelles s'appuient sur des mesures dans l'image. Dans cette thèse, nous proposons de gérer le flou de bougé pendant le mouvement de la caméra en considérant la netteté comme une contrainte de la commande. Cela est rendu possible grâce à des techniques avancées de commande sous contraintes, qui peuvent être appliquées pour l'asservissement visuel. Notre travail se concentre sur la conception de cette contrainte de netteté que nous pensons être un problème intrigant et intéressant.

Structure de la thèse

La thèse débute par une vue d'ensemble sur l'asservissement visuel. Le chapitre 1 présente un système caméra avec les bases mathématiques nécessaires pour expliquer et représenter sa position et son mouvement dans l'espace cartésien. Le modèle de caméra sténopé est également présenté, car il nous permet d'introduire les équations de perspective sur lequel repose la plupart des lois de commande que nous développerons. Ensuite, des contrôleurs visuels sont présentés, avec une attention particulière pour l'asservissement visuel. Enfin, nous mettons l'accent sur les perturbations causées par les mouvements externes à la caméra et à la manière de les traiter en asservissement visuel, dans l'espoir d'appliquer ces techniques à un satellite.

Le chapitre 2 se concentre sur l'étude d'un satellite en orbite terrestre basse, tout d'abord en décrivant son mouvement de translation autour de la Terre sous certaines hypothèses adéquates, mais aussi le mouvement d'un objet situé sur la surface terrestre, ce qui nous permet de créer un modèle réaliste d'une mission d'observation. Les contrôleurs d'attitude traditionnels sont ensuite présentés à partir de la littérature, ainsi que les différentes applications de l'asservissement visuel dans la robotique spatiale. Nous rappelons les principaux objectifs des lois de commande visuelle que nous visons à concevoir au regard des références existantes, et nous les développerons dans le prochain chapitre.

Dans le chapitre 3, nous présentons nos contributions à la fois dans le domaine de la robotique spatiale et de l'asservissement visuel. Nous développons des lois d'asservissement visuel visant à orienter précisément un satellite pour qu'il pointe vers un objet terrestre spécifique, en tenant compte des mouvements orbitaux, et en explorant les solutions disponibles afin de créer un contrôleur robuste, rapide et fiable. Cette contribution a été publiée dans [ROBIC, FRAISSE, MARCHAND & CHAUMETTE, 2022]. Dans la deuxième partie de ce chapitre, nous avons inclus la dynamique interne du satellite dans le modèle pour une représentation plus réaliste. Nous avons également détaillé différentes commandes basées sur un algorithme de saturation prédictive. Cette approche a été présentée à la communauté astronautique dans [ROBIC, FRAISSE, LAGADEC et al., 2022]. Ces lois de commande sont ensuite testées avec plusieurs expérimentations à différents niveaux, en commençant par des simulations avec un simulateur de trajectoire de satellite avec des caractéristiques visuelles simulées, puis avec des images à l'échelle réelle, jusqu'à leur mise en œuvre sur une plateforme robotique.

Dans le chapitre 4, nous introduisons un contrôleur visuel visant à limiter la création de flou de mouvement. Nous commençons par examiner ce flou et les techniques de traitement d'image traditionnellement utilisées pour l'estimer et restaurer des images dégradées. Nous discutons ensuite des métriques utilisées dans le domaine de la vision par ordinateur pour mesurer la netteté des images. Puis, nous sélectionnons la norme du gradient de l'image comme une métrique adéquate car sensible au flou de mouvement. Nous explorons ensuite les techniques d'asservissement visuel contraint, en particulier celles qui peuvent gérer des contraintes complexes, nous conduisant à envisager des méthodes d'optimisation. Nous proposons un contrôleur d'asservissement visuel basé sur la programmation quadratique (QP) capable de gérer les contraintes mécaniques et dynamiques du satellite, ainsi que d'autres contraintes de vitesse. La contrainte de netteté est ensuite conçue en fonction de la norme des gradients de l'image, et la Jacobienne de cette métrique est développée analytiquement pour déterminer et injecter une contrainte de vitesse dans le contrôleur. Enfin, des expériences sont menées avec les mêmes simulateurs de

trajectoire satellitaire. Ce travail a été proposé dans un article de journal dans [Robic et al, Soumis en Sept. 2023].

Enfin, nous concluons ce manuscrit en rappelant nos contributions dans un premier temps, puis nous discutons des limites de nos différentes stratégies et des perspectives potentielles de cette thèse. Parmi elles, nous détaillons la généralisation de notre approche à une constellation de satellites.

Publications

M. Robic, R. Fraisse, E. Marchand, F. Chaumette. "Vision-based rotational control of an agile observation satellite," In : *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

M. Robic, R. Fraisse, K. Lagadec, E. Marchand, F. Chaumette. "Visual servoing of an Earth observation satellite of the LION constellation," In : *73rd International Astronautical Congress (IAC)*, 2022.

Soumissions

M. Robic, R. Fraisse, E. Marchand, F. Chaumette. "QP-based Visual Servoing Under Motion Blur-Free Constraint," Submitted in Sept. 2023.

TABLE OF CONTENTS

INTRODUCTION

Since the beginning of the space age and the launch of the first satellite into orbit, artificial satellites have proliferated in number, diversified in applications, and advanced in performance. Indeed, starting by *Spoutnik 1* in 1957 to demonstrate humanity's ability to access the Earth's orbital domain and to take advantage of it, more than 10,000 spacecraft are now in orbit around the Earth, aimed with diverse and appealing missions, ranging from scientific demonstration to telecommunications, from geolocation to observation. This last category encompasses satellites equipped with long-range sensors to measure and record information from the Earth's surface, providing data in a big scale, allowing for a global and accurate monitoring of the planet.

Earth observation satellites benefit from their orbits, an induced trajectory around the Earth that is often considered as the main natural thruster, and mostly determined during the satellite launch. This asset allows satellites to scan a certain location on the Earth's surface at a specific moment. Missions requiring a fast revisit capability use relatively low orbits, they are called Low Earth Orbit (LEO) satellites. Among them, observation satellites equipped with a vision sensor, or imaging satellites, take advantage of LEO to frequently acquire images of the Earth with high resolution. For instance, Pléiades HR, a constellation of two imagery satellites on a Sun-synchronous quasicircular orbit (695 km), acquires Earth images with 70-cm resolution, with a revisit interval of less than 24 hours for each satellite. Pléiades image acquisition system uses push-broom technology (see Figure 1), i.e., a sensor line allows scanning of a 20 km high-quality image swath and reconstructs the image afterwards.

The acquisition requires to precisely point the satellite's sensor toward the Earth's surface, which is done through the control of the orientation of the satellite, called satellite attitude. To operate an acquisition mission, a desired set of attitude is maintained thanks to an onboard attitude control law. However, this task can be quite challenging in LEO, as the extreme orbital motion of the satellite, and other external perturbations becoming more salient near the Earth, leads to attitude drifting and to pointing inaccuracies. Furthermore, the requested set of attitudes is not supposed to change during the acquisition, as it is often computed offline by a ground segment, which classifies the global pointing control as an open-loop process.

However, real-time attitude guidance through the feedback of a vision sensor might be of relevance to correct the propagation of uncertainties. Moreover, it could offer more features

to Earth-pointing satellites, such as the fast redefinition of the target location and especially the definition of the mission in terms of sensor information. This powerful asset would allow for more flexibility and more complex missions, e.g. the tracking of a moving target from the satellite, which remains an open problem.

This thesis is part of the LICHIE project, a scientific collaboration with Airbus Defence & Space to develop a new satellite system with a "Starer" acquisition principle (see Figure 1). Acquisition is carried out using a sensor matrix that allows real-time acquisition of images. Airbus DS plans to incorporate this new technology into the future LION constellation composed of several agile video satellites. This way, the satellite "camera" can be used as an input sensor for real-time attitude guidance process using satellite images of rare resolution (up to 50 cm). In this thesis, we will show that precise control of the satellite orientation can be achieved using visual servo-control schemes.

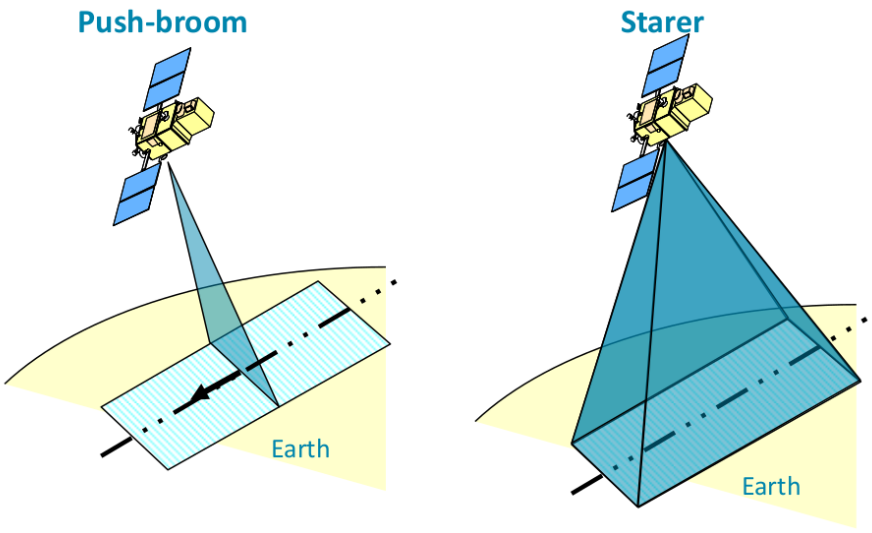


Figure 1 – Push-broom and starer technologies

Visual servoing is a vision-based control approach that results from the association between robotic control and computer vision. It is now widely used in the robotics field because of its appealing features for operating precision tasks using data obtained from camera feedback. As a closed-loop control, it is resilient to error propagation and is widely used for manipulation tasks, mobile robotics, and visual tracking. In fact, a LION satellite can be seen as a pan-tilt-roll camera embedded in a fast vehicle and subject to its uncontrolled motion, tracking a mobile object. This is the main consideration of our work.

In space robotics, visual servoing has already been used to solve on-orbit tasks, but its poten-

tial application to high-resolution observation satellites yields unique challenges. Among them, we deal with the relative motion between the satellite and the target, which is far more consequent compared to on-orbit servicing scenarios. Also, compared to the few existing strategies for observation satellites, we consider tracking a moving reference either through the redefinition of a fixed target location on the Earth's surface or through the tracking of a moving object. In both cases, the target is defined with image measurements, which also represents a noticeable difference from classical Earth-pointing satellites. The mechanical and dynamical constraints of the satellite are also not left behind. This thesis aims to propose personalized solutions to handle them within the control.

Speaking about constraints, one of the most important ones when working with an imaging satellite, where every output is important, is the one of image quality. But how can image quality be quantifiable? A noticeable characteristic is the sharpness of the image, which can be estimated through specific metrics. In our context, the sharpness is threatened by motion blur induced by the motion of the camera and by the potential motion of the object in the scene we intend to observe (see Figure 2).

This loss of quality is not desirable from the point of view of the customer but also for the robustness of the control, as visual controllers rely on image measurements. In this thesis, we propose to handle motion blur during the camera motion by considering sharpness as a constraint in the control scheme. This is made possible through advanced constraint management techniques that can be applied for visual servoing. Our work is focused on the design of this sharpness constraint, from what we believe to be an intriguing and interesting issue.

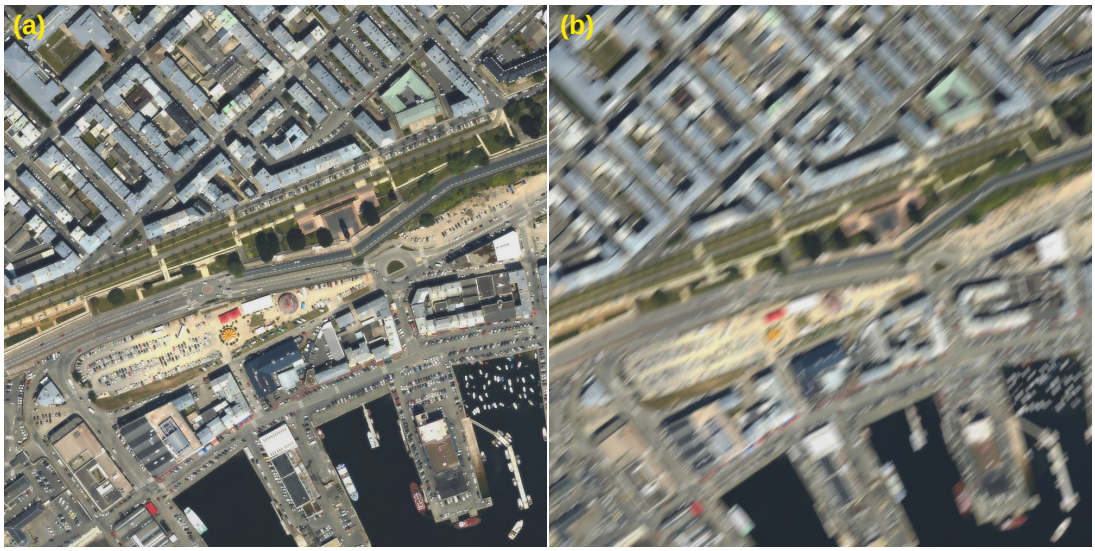


Figure 2 – Satellite image of 50-cm resolution of Brest (a) Sharp (b) Motion-blurred

Thesis structure

The thesis begins with an overview of visual servoing. Chapter 1 presents a camera system with the mathematical background needed to explain and represent its motion in Cartesian space. The pinhole camera model will also be presented, as it allows us to introduce the image frame on which most of the control we will develop is based. Then, visual controllers will be presented, with a special emphasis on visual servoing. Finally, particular attention will be paid to disturbances from external motions and how to address them in visual control, with the goal of applying these techniques to a satellite.

Chapter 2 focuses on the control of a Low Earth Orbit satellite, first by describing its translational motion around the Earth under adequate hypotheses, and also the motion of an object lying on the Earth's surface, which will help us to create our realistic model in Chapter 3. Traditional attitude controllers will be presented from the literature, and then the different applications of visual servoing for space robots. We will finally point out what challenges an Earth observation satellite controlled through its camera would face, yielding the different specifications that have to be respected by the visual control laws we aim to design. We will delve into their conception in the next chapter.

In Chapter 3, we present our contributions to both space control and visual servoing. We develop control laws to accurately orient a satellite to point at a specific terrestrial object of interest, taking into account orbital motions and all the visual control capabilities to create a robust, fast, and reliable controller. This contribution has been published in [Robic, Fraisse, Marchand, & Chaumette, 2022]. In the second part of this work, we have included the inner dynamic of the satellite in the model for a more realistic representation. We have also detailed different controllers based on a predictive saturation algorithm. This approach has been presented to the astronautic community in [Robic, Fraisse, Lagadec, et al., 2022]. These control laws are then tested with several experiments at different stakes, from projective simulations, with a satellite path simulator with real-scaled images, and finally implemented in a robotics platform.

In Chapter 4, we introduce a visual controller to limit motion blur creation. We begin by examining motion blur and traditional image processing techniques used to estimate it and restore motion-blurred images. We then discuss the metrics used in the computer vision field to measure image sharpness, selecting the norm of the image gradient as a sensitive one to motion blur. We then explore constrained visual servoing techniques, particularly those that can handle complex constraints, leading us to consider optimization frameworks. We propose a QP-based

visual servoing controller that can manage the satellite's mechanical and dynamical constraints, as well as other velocity constraints. The sharpness constraint is then designed based on the norm of the image gradients, and the Jacobian with respect to the control input is determined to inject a velocity constraint into the controller. Finally, experiments are conducted with the same satellite path simulator. This work has been proposed to a journal paper in [Robic et al, Submitted in Sept. 2023].

Publications

M. Robic, R. Fraisse, E. Marchand, F. Chaumette. "Vision-based rotational control of an agile observation satellite," In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

M. Robic, R. Fraisse, K. Lagadec, E. Marchand, F. Chaumette. "Visual servoing of an Earth observation satellite of the LION constellation," In: *73rd International Astronautical Congress (IAC)*, 2022.

Submitted

M. Robic, R. Fraisse, E. Marchand, F. Chaumette. "QP-based Visual Servoing Under Motion Blur-Free Constraint," Submitted in Sept. 2023.

CHAPTER 1

BASICS ON VISUAL SERVOING

1.1 Introduction

In this chapter, we propose to introduce some basics on visual servoing and present the mathematical tools that we will use throughout the thesis. We will begin with a first part on kinematics, specifically oriented for the representation of the physical world from a geometric point of view. Following this, we will introduce the classical model to represent the world as perceived through the lens of a digital camera. Subsequently, we will look at the techniques allowing a camera to navigate in this world based on its visual information, with a particular emphasis on visual servoing, our preferred control approach in this thesis. Finally, we will look at how the vision-based control community deals with disturbances caused by external motions from observed objects or from the system dynamics, a subject that will interest us for the following chapters.

1.2 Computer vision background

To fully understand visual servoing approaches, one needs to first have an overview of computer vision techniques [Jain et al., 1995][Stockman & Shapiro, 2001], to connect the physical 3D world and the world perceived by a camera. This is done through Euclidean geometry and adequate projection model that will be expressed in this section.

1.2.1 Rigid transformation

In order to express the position of an object in the world, we need to introduce the concept of frames, a geometrical structure defined by a specific point in space, its origin, and three orthogonal axes that parameterize a coordinate system. Let us introduce two frames F_A and F_B located and oriented differently in the world, we can now, for instance, express the position of a 3D point with respect to the frame F_A, denoted ${}^A\mathbf{X} = ({}^AX, {}^AY, {}^AZ)^\top$. Of course, it is

also possible to express this point with respect to F_B, giving ${}^B\mathbf{X}$, but we need to determine the transformation between F_A and F_B, also called the rigid transformation. It is made up of:

- a translation vector ${}^B\mathbf{t}_A \in \mathbb{R}^3$, representing the position of the origin of the frame F_A expressed in F_B,
- a rotation matrix ${}^B\mathbf{R}_A \in SO(3)$, expressing the orientation of the frame F_A with respect to F_B.

$SO(3)$ is the Special Orthogonal Group, defined by:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}\mathbf{R}^\top = \mathbf{I}_3,\ det(\mathbf{R}) = 1 \right\}. \tag{1.1}$$

From these, a 3D point ${}^A\mathbf{X}$ expressed in F_A can be transformed in F_B, giving ${}^B\mathbf{X}$, by the following:

$${}^B\mathbf{X} = {}^B\mathbf{R}_A\, {}^A\mathbf{X} + {}^B\mathbf{t}_A \tag{1.2}$$

1.2.2 Rotation

Let us have a look at the specific case of orientation in 3D space. We define an orientation by the rotation matrix $\mathbf{R}$ simply expressed by:

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{1.3}$$

The three lines and the three columns of $\mathbf{R}$ express a basis in the 3D space. If we state $\mathbf{R}$ to be a valid rotation, i.e., $\mathbf{R} \in SO(3)$, the matrix is orthonormal and its determinant is equal to 1, according to definition (1.1). One resulting property is that the inverse matrix is equal to the transpose. For example, taking ${}^B\mathbf{R}_A$ the previous matrix that rotates the frame F_A to the frame F_B, its inverse corresponds to:

$${}^B\mathbf{R}_A{}^{-1} = {}^B\mathbf{R}_A{}^\top = {}^A\mathbf{R}_B \tag{1.4}$$

which gives us the rotation matrix that rotates F_B to F_A. Moreover, rotation matrices can be composed to obtain new rotations. Taking into account a third frame F_C, the orientation of F_C with respect to F_A is given by:

$${}^A\mathbf{R}_C = {}^A\mathbf{R}_B\, {}^B\mathbf{R}_C = {}^B\mathbf{R}_A{}^{-1}\, {}^B\mathbf{R}_C = {}^B\mathbf{R}_A{}^\top\, {}^B\mathbf{R}_C \tag{1.5}$$

In addition, a rotation matrix is generally parameterized under three possible representations.

1.2.2.1 Euler angles

One classical representation, especially used in the astronautic community to define orbital parameters, is the **Euler angles representation**, or more precisely the consideration of a triplet (Ψ, Θ, Φ), where each angle parametrizes one rotation matrix that is successively multiplied to attain an infinite possibility of orientation. This succession is called an Euler angle sequence, which can be, for instance:

$$\begin{aligned}\mathbf{R} &= \mathbf{R}_x(\Phi)\mathbf{R}_y(\Theta)\mathbf{R}_z(\Psi) \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{pmatrix} \begin{pmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{pmatrix} \begin{pmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{pmatrix}\end{aligned} \tag{1.6}$$

where $\mathbf{R}_i(j) \in SO(3)$ is the rotation matrix along axis $i \in \{x, y, z\}$ of an angle $j \in \{\Phi, \Theta, \Psi\}$. The multiplication order and the rotation axes have meaning, and each sequence induces a different parameterization. The main issue of this representation is that a singularity occurs if two rotational axes align with each other during successive rotations, resulting in a loss of a degree of freedom called *gimbal lock*. Moreover, the inverse operation (obtaining the triplet from a matrix) is sometimes a tricky computation, which will not be expressed here.

1.2.2.2 Axis-angle

The **axis-angle representation** is a compact representation of a rotation extensively used in the robotics community, where $\mathbf{u}$ is a unit vector describing the axis around which to rotate and θ is the amplitude in radians of the rotation around $\mathbf{u}$. The function that maps the rotation matrix of $SO(3)$ with the axis-angle in $\mathbb{R}^3$ is called the logarithmic map, and its inverse function

is called the exponential map. They are defined as follows:

$$\begin{aligned} &\text{log: } SO(3) \rightarrow \mathbb{R}^3 \\ &\theta\mathbf{u} = \log \mathbf{R} \\ &\theta = \cos^{-1}\left(\frac{Tr(\mathbf{R})-1}{2}\right) \\ &\mathbf{u} = \frac{1}{2\sin\theta}\begin{pmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{pmatrix} \end{aligned} \tag{1.7}$$

with $Tr(\mathbf{R})$ the trace of $\mathbf{R}$. Then, the exponential map is the Taylor expansion given by:

$$\begin{aligned} &\text{exp: } \mathbb{R}^3 \rightarrow SO(3) \\ &\mathbf{R} = \exp(\theta\mathbf{u}) = \sum_{i=0}^{\infty} \frac{[\theta\mathbf{u}]_\times^i}{i!} = \mathbf{I}_3 + [\theta\mathbf{u}]_\times + \frac{1}{2!}[\theta\mathbf{u}]_\times^2 + \ ... \end{aligned} \tag{1.8}$$

with $[\theta\mathbf{u}]_\times$ the skew-symmetric matrix of $\theta\mathbf{u}$ defined by:

$$[\theta\mathbf{u}]_\times = \theta\begin{pmatrix} 0 & -u_x & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \tag{1.9}$$

A closed-form solution of Eq.(1.8) is obtained using Rodrigues formula:

$$\mathbf{R} = \mathbf{I}_3 + \frac{\sin\theta}{\theta}[\theta\mathbf{u}]_\times + \frac{1-\cos\theta}{\theta^2}[\theta\mathbf{u}]_\times^2 \tag{1.10}$$

1.2.2.3 Quaternion

Another useful representation, which is now frequently used in the astronautic community to avoid the gimbal lock singularity evoked previously, is the **quaternion** $\mathbf{q}$. Quaternions are an extension of complex numbers $\mathbb{C}$, represented as $\mathbf{q} = a + bi + cj + dk$, or also commonly by the 4D vector $\mathbf{q} = (a, b, c, d)$ with the basis $(1, i, j, k)$, where $a, b, c, d \in \mathbb{R}$ are the real parts, and $i, j, k \in \mathbb{I}$ the imaginary parts. In this representation, we have $i^2 = j^2 = k^2 = ijk = -1$, and $\mathbf{q}$ represents a valid rotation if $a^2 + b^2 + c^2 + d^2 = 1$. In summary, it offers more mathematical

assets, and a rotation matrix can be obtained from a quaternion such as:

$$\mathbf{R} = \begin{pmatrix} 2(a^2+b^2)-1 & 2(bc-ad) & 2(bd+ac) \\ 2(bc+ad) & 2(a^2+c^2)-1 & 2(cd-ab) \\ 2(bd-ac) & 2(cd+ab) & 2(a^2+d^2)-1 \end{pmatrix} \tag{1.11}$$

The inverse operation is easily obtained through axis-angle representation using the logarithmic map. Then $\mathbf{q}$ is obtained by taking $a = \cos\frac{\theta}{2}$ and $(b,c,d) = \sin\frac{\theta}{2}\mathbf{u}$ with θ and $\mathbf{u}$ defined in equation (1.7).

1.2.3 Homogeneous transformation

The rigid transformation expressed in the affine equation (1.2) is traditionally expressed linearly by considering homogeneous transformation. For that, ${}^{A}\mathbf{X}$ and ${}^{B}\mathbf{X}$ are expressed with homogeneous coordinates, such as $\bar{\mathbf{X}} = (wX, wY, wZ, w)$ with $w \in \mathbb{R}$, and equation (1.2) can be mapped as:

$${}^{B}\bar{\mathbf{X}} = {}^{B}\mathbf{T}_{A}\,{}^{A}\bar{\mathbf{X}} \tag{1.12}$$

with ${}^{A}\mathbf{T}_{B} \in \mathbb{R}^{4\times4}$ the homogeneous transformation from the frame F_B to the frame F_A, expressed by:

$${}^{B}\mathbf{T}_{A} = \begin{pmatrix} {}^{B}\mathbf{R}_{A} & {}^{B}\mathbf{t}_{A} \\ \mathbf{0} & 1 \end{pmatrix} \tag{1.13}$$

Once the homogeneous coordinates ${}^{B}\bar{\mathbf{X}}$ are computed, they can be converted back into their 3D representation by dividing ${}^{B}\bar{\mathbf{X}}$ by w and removing the fourth coordinate. $\mathbf{T}$ is a homogeneous matrix. Homogeneous matrices are widely used to represent frame transformations, they are more generally defined such as:

$$\mathbf{T} \in \left\{ \mathbf{T} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix} \mid (\mathbf{R}, \mathbf{t}) \in SE(3) \right\}. \tag{1.14}$$

with $SE(3) = SO(3) \times \mathbb{R}^3$ the Special Euclidean Group. They have interesting mathematical properties, among them the composition of homogeneous matrices to obtain new frame transformations. For example, taking a frame F_C, the transformation between frames F_C and F_A is

given by ${}^{A}\mathbf{T}_{C} = {}^{A}\mathbf{T}_{B}{}^{B}\mathbf{T}_{C}$. We can invert ${}^{A}\mathbf{T}_{B}$ to obtain:

$$ {}^{A}\mathbf{T}_{B}{}^{-1} = {}^{B}\mathbf{T}_{A} = \begin{pmatrix} {}^{A}\mathbf{R}_{B}{}^{-1} & -{}^{A}\mathbf{R}_{B}{}^{-1}{}^{a}\mathbf{t}_{b} \\ \mathbf{0} & 1 \end{pmatrix} = \begin{pmatrix} {}^{A}\mathbf{R}_{B}{}^{\top} & -{}^{A}\mathbf{R}_{B}{}^{\top}{}^{a}\mathbf{t}_{b} \\ \mathbf{0} & 1 \end{pmatrix} \tag{1.15} $$

These several transformations can be seen in Figure 1.1.

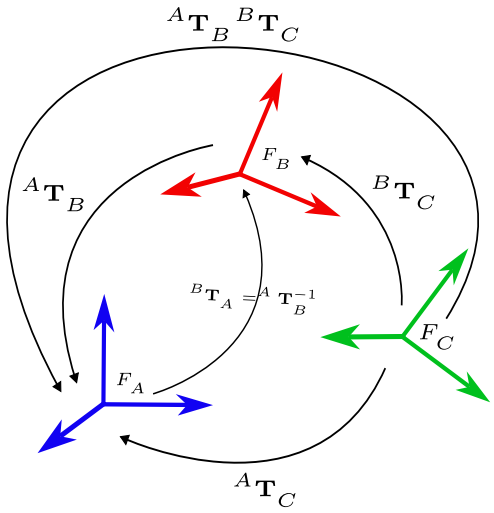


Figure 1.1 – Frame transformations

Transformation with a null translation

In this case the transformation matrix from one frame to another is a pure rotation, and equation (1.2) is now linear without switching to homogeneous coordinates. In these cases, we will prefer to use the resulting relationship:

$$ {}^{B}\mathbf{X} = {}^{B}\mathbf{R}_{A}{}^{A}\mathbf{X} \tag{1.16} $$

1.2.4 Camera system

1.2.4.1 Camera pose

After having represented the 3D world with adequate Euclidean geometry, a classic camera system can be introduced by three frames of reference, the world frame F_w, the camera frame F_c and the object frame F_o. The world frame F_w refers to an arbitrary fixed frame in the world, the camera frame F_c is the frame attached to the camera, and F_o is the object frame, fixed to the object we intend to observe (see Figure 1.2). As previously seen, each frame can be positioned

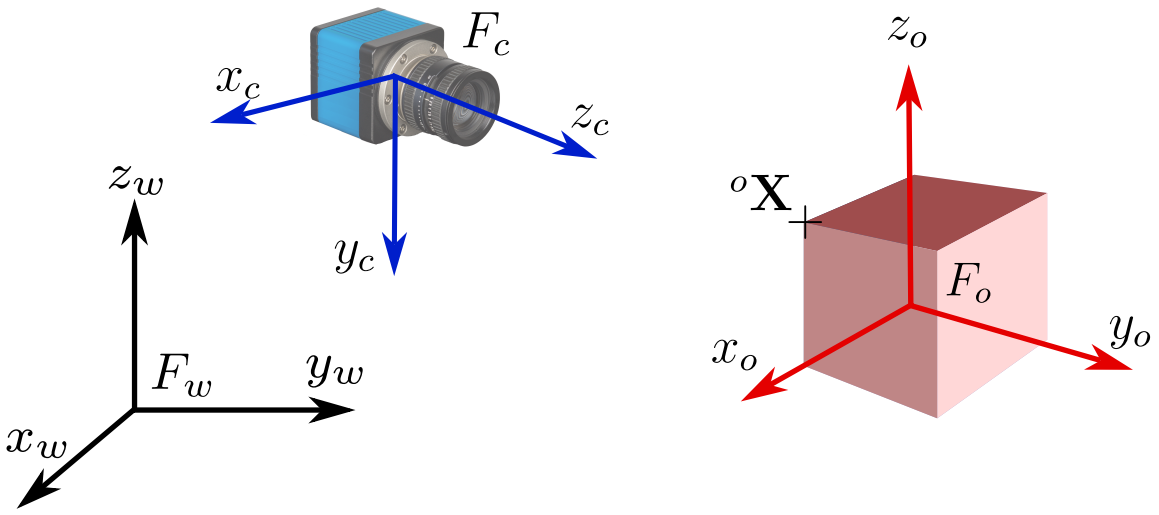


Figure 1.2 – Representation of the camera system. A camera referred to its camera frame F_c observes a point ${}^o\mathbf{X}$ expressed in the object frame F_o.

and oriented in the space with respect to another frame of reference thanks to the associated homogeneous matrix $\mathbf{T}$. In robotics, the term of **pose** is further used to represent the position and orientation of any frame with respect to another frame and is directly associated with a homogeneous matrix. One possible representation of this pose is a 6D vector $\mathbf{p}$ parameterized by the translation vector $\mathbf{t}$ of a given $\mathbf{T}$ and the axis-angle representation of the rotation $\mathbf{R}$ of this same $\mathbf{T}$, such as:

$$\mathbf{p} = \begin{pmatrix} \mathbf{t} \\ \theta\mathbf{u} \end{pmatrix} \tag{1.17}$$

Applied to our system, the pose ${}^w\mathbf{p}_c$ is the pose of the camera in the world frame, linked with the homogeneous matrix ${}^w\mathbf{T}_c$, and similarly, ${}^o\mathbf{p}_c$ is the pose of the camera with respect to the object frame associated to ${}^o\mathbf{T}_c$.

The camera pose is essential in visual servoing as it is the vector we intend to modify.

1.2.4.2 Camera model

Now that we have modeled the world and established where the camera is located, our aim is to observe an object of interest ${}^o\mathbf{X}$ through the lens of the camera. First, ${}^o\mathbf{X}$ is expressed in the camera frame as follows:

$${}^c\bar{\mathbf{X}} = {}^c\mathbf{T}_o\,{}^o\bar{\mathbf{X}} \tag{1.18}$$

so the geometrical position of the point with respect to the camera is known. Yet, a camera is unable to sense the complete 3D information of the world, especially for a classic camera (that does not acquire depth information), but captures a projection of the 3D world and maps it into a 2D digitized representation called an image.

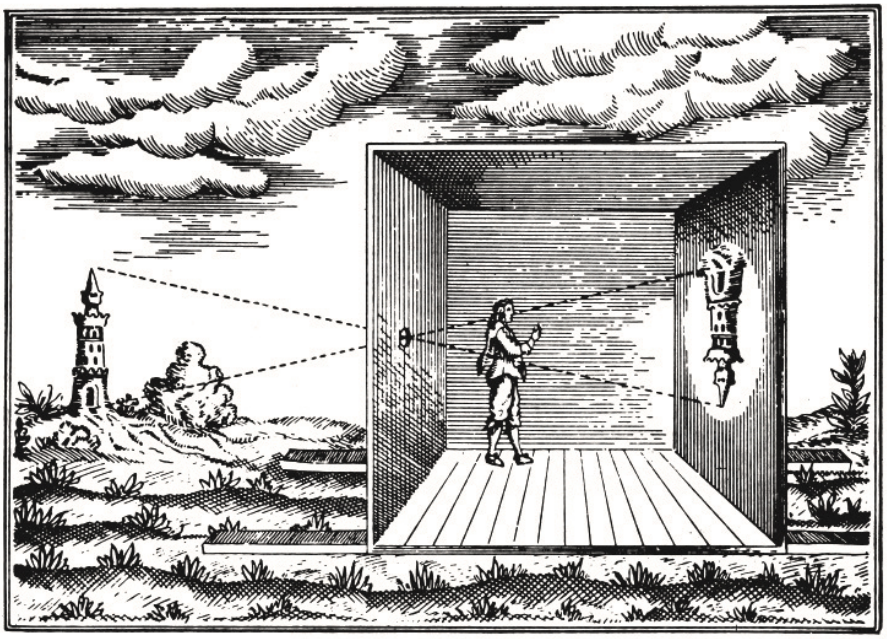


Figure 1.3 – The *camera obscura*: the light from the outside reaches a dark room through a minute aperture and produces an inverted image of the world.

The first step, the projection, is allowed by selecting an adequate camera model. The one used in this thesis, and commonly used by the computer vision community, is the one of a pinhole camera [Hartley & Zisserman, 2003], inspired by the *camera obscura* (Figure 1.3). Within this representation, light from the outside world passes through a pinhole and is then projected onto a plane in the dark chamber, which produces a two-dimensional representation of the environment. Current cameras are using the same principle to capture visual information. Applied to our situation, we will consider the projection plane to be in front of the projection point (the pinhole) in order to obtain an image that is not inverted by the geometry. This projection plane is also called the image plane. Hence, the 2D coordinates $\mathbf{x} = (x, y)$ of the object point ${}^c\mathbf{X} = ({}^cX, {}^cY, {}^cZ)$ expressed in the camera frame and projected into the image plane are given by:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{f}{{}^cZ} \begin{pmatrix} {}^cX \\ {}^cY \end{pmatrix} \tag{1.19}$$

with f the focal length of the lens of the camera, ${}^cZ$ also called the depth of the target in the camera frame, and (x, y) are also called the normalized coordinates of an image point. This equation can also be expressed with homogeneous coordinates $\bar{\mathbf{x}} = (x, y, 1)$ and ${}^c\bar{\mathbf{X}} = ({}^cX, {}^cY, {}^cZ, 1)$, such as:

$$\bar{\mathbf{x}} = \mathbf{A}\,{}^c\bar{\mathbf{X}} \qquad \text{with} \qquad \mathbf{A} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{1.20}$$

However, one last step is needed to obtain a digitized image.

1.2.4.3 Digital image

A digital image is basically the sampling of the image plane in a regular array of dimension $(N \times M)$ where N is the height of the image and M its width in pixels. Each pixel contains the amount of light received by the sensor at their location, which is referred to as an intensity value for grayscale images. From the image point $\mathbf{x} = (x, y)$ in normalized coordinates, its position

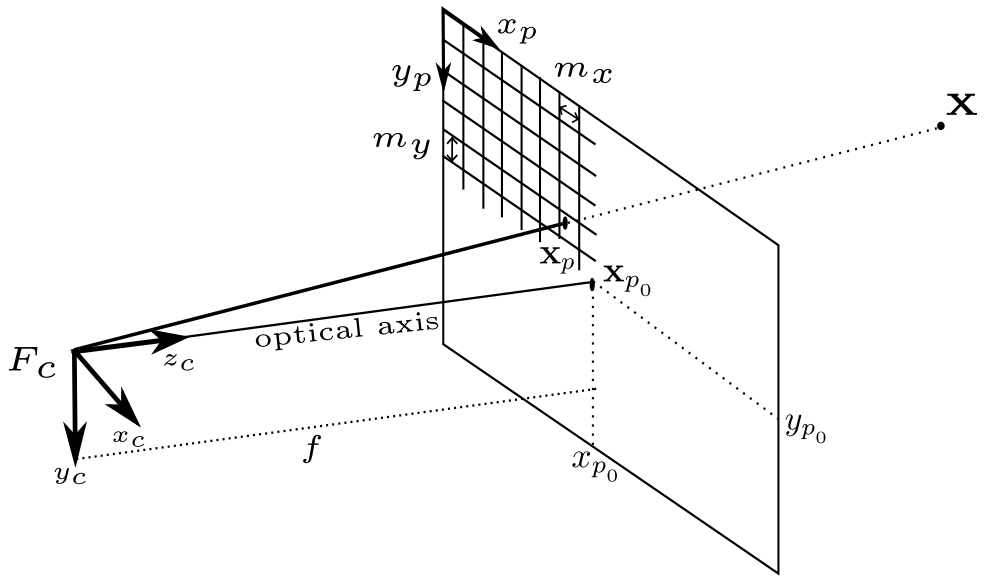


Figure 1.4 – A 3D point is projected and sampled in a digital image in pixel coordinates thanks to camera intrinsic parameters: f, (x_{p_0}, y_{p_0}), m_x and m_y.

in the array is given by its pixel coordinates $\mathbf{x}_p = (x_p, y_p)$. The conversion between $\mathbf{x}$ and $\mathbf{x}_p$ is an affine transformation if we neglect any distortion effect from the camera, it is expressed by

(see Figure 1.4):

$$\begin{pmatrix} x_p \\ y_p \end{pmatrix} = \begin{pmatrix} x_{p_0} \\ y_{p_0} \end{pmatrix} + \begin{pmatrix} \frac{1}{m_x} & 0 \\ 0 & \frac{1}{m_y} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{1.21}$$

with:

- (x_{p_0}, y_{p_0}) pixel coordinates of the principal point $\mathbf{x}_{p_0}$, the intersection point between the optical axis, aligned with the z_c-axis of the camera, and the image plane.
- m_x and m_y respectively the width and height in meters of a pixel on the sensor.

Once again, this affine equation can be expressed linearly using the homogeneous coordinates $\bar{\mathbf{x}}_p = (x_p, y_p, 1)$ and $\bar{\mathbf{x}}$, we obtain from equation (1.21):

$$\bar{\mathbf{x}}_p = \mathbf{K}'\bar{\mathbf{x}} \qquad \text{where} \qquad \mathbf{K}' = \begin{pmatrix} \frac{1}{m_x} & 0 & x_{p_0} \\ 0 & \frac{1}{m_y} & y_{p_0} \\ 0 & 0 & 1 \end{pmatrix} \tag{1.22}$$

with $\mathbf{K}'$ an affine transformation matrix.

Finally, we can now combine equation (1.20) and equation (1.22) to obtain the complete perspective model that follows:

$$\text{with} \qquad \mathbf{K} = \begin{pmatrix} p_x & 0 & x_{p_0} \\ 0 & p_y & y_{p_0} \\ 0 & 0 & 1 \end{pmatrix}$$

$$\bar{\mathbf{x}}_p = \mathbf{K}'\mathbf{A}^c\bar{\mathbf{X}} = \mathbf{K}\mathbf{\Pi}^c\bar{\mathbf{X}} \tag{1.23}$$

$$\text{and} \qquad \mathbf{\Pi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

with $p_x = \frac{f}{m_x}$ and $p_y = \frac{f}{m_y}$. $\mathbf{K}$ is called the intrinsic parameter matrix of the camera and $\mathbf{\Pi}$ is the projection matrix. We can go deeper and express the 3D point in its original frame in equation (1.23) which gives us:

$$\bar{\mathbf{x}}_p = \mathbf{K}\mathbf{\Pi}^c\mathbf{T}_o{}^o\bar{\mathbf{X}} \tag{1.24}$$

The previous equation provides us the complete mapping of a 3D point expressed in its object frame into its pixel position in a digitized image, and highlights the influence of the camera pose through the homogeneous matrix ${}^c\mathbf{T}_o$. This equation is the basis for many applications, such as camera calibration (determination of $\mathbf{K}$) [Remondino & Fraser, 2006], image simulation (de-

termination of $\bar{\mathbf{x}}_p$)[Kang, 1998][S. Chen & Williams, 2023], 3D localization (determination of ${}^c\mathbf{T}_o$)[Wu et al., 2018], 3D reconstruction (determination of ${}^o\bar{\mathbf{X}}$)[Zollhöfer et al., 2018][Aharchi & Ait Kbir, 2020], and, of course, vision-based control.

1.3 Vision-based control

Once we know how the world is represented, how the camera is located in it, and how the surrounding and, for instance, a specific object, are perceived from the camera, one may consider to move in this world. In this section, we focus on the techniques to move the camera with respect to visual information, more precisely, the types of control that consider vision with a high-level implication in their design.

Controlling the pose of a camera to observe an object of interest (and in some cases interact with it) has always been an appealing subject of research. An extensive survey proposed by [Corke, 1993] relates the blooming of the visual control approach. At this point, two categories of visual controllers stand out, the *look and move* approach, an open-loop control using visual information, and *visual servoing*, a closed-loop visual control approach.

1.3.1 Open-loop visual control

Open-loop visual control is a control strategy in which a camera first acquires visual information from the object of interest (*look*), then a pose or a sequence of poses (trajectory) are computed in advance before being executed (*move*) and without being adjusted due to camera feedback (or other sensor feedback) during the motion. This behavior can be likened to a camera head that takes in visual information, plans its next move, pauses the acquisition, and then resumes it once the control is achieved. The next action is determined directly from the perception of the surrounding environment at the moment the information is acquired. For that, this control approach can be really fast, as the next pose to reach is known, and the robot can operate a quick motion (a saccade for angular control) to attain it. Therefore, open-loop control can be effective in an environment that is known, static or predictable, in this case the computed pose will be relatively accurate. However, it may be less robust in an unknown or dynamic environment and induce a consequent positioning error in the worst cases as it does not adapt to changes in real-time.

Consequently, open-loop controls still present a lot of interest in devices that are prioritizing a fast motion, or reorientation for angular control, over a reasonable loss of precision. For

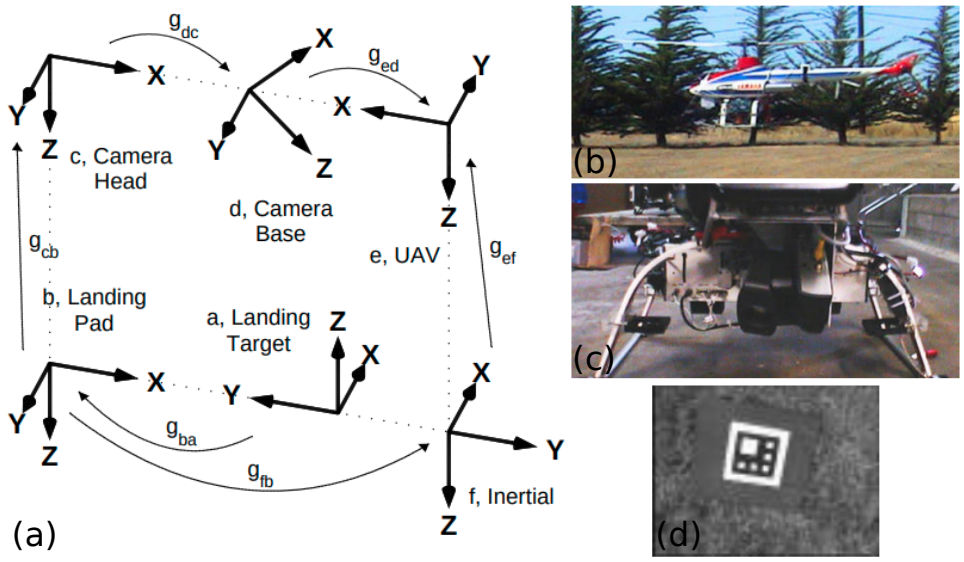


Figure 1.5 – (a) Frame transformations, experimental platform (b-c) and camera view (d) taken from [Sharp et al., 2001], the camera head can be oriented thanks to a pan-tilt unit to keep the landing target in the field of view of the camera.

example, camera surveillance systems [Lim et al., 2003] and [Senior et al., 2005], or pan-tilt cameras for general observations [Stolle & Rysdyk, 2003], which do not necessarily need to precisely focus on a specific object, but also vision systems for landing assistance [Sharp et al., 2001] where the camera is used to estimate the position of the landing target and only needs to occasionally adjust its orientation to keep it in its field of view (see Figure 1.5).

Nevertheless, the drawback in accuracy led the community to develop techniques that use real-time camera feedback to address more complex scenarios.

1.3.2 Visual servoing

Visual servoing (VS) is a closed-loop visual control approach that uses computer vision data as feedback to control and adjust the motion of a camera, and more generally of a robot, in real time [Espiau et al., 1992][Chaumette & Hutchinson, 2008]. Visual servoing tackles the main drawback of the previous approach, proposing controls of high accuracy and reactive to changes so particularly efficient in dynamic environments. It is based on visual features, a set of specific visual data designed from image measurements. These features can be expressed in many ways, directly from 2D image information (coordinates of a set of image points, global metric of an image, etc.) to 3D data (camera pose..) derived from image measurements.

From the beginning, a visual servoing control scheme aims to regulate to 0 an error $\mathbf{e}(t)$ that

is designed according to the task to perform and the number of degrees of freedom to control. It is defined by:

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^*(t) \tag{1.25}$$

where $\mathbf{s}(t) \in \mathbb{R}^k$ represents a set of k visual features and $\mathbf{s}^*(t)$ are their desired values. The dimension of $\mathbf{e}(t)$ is at least the number of degrees of freedom (DOF) that we want to control, i.e., $k \geq m$ to control the m degrees of freedom of a single camera with $m \leq 6$. The visual error vector, by itself, represents a significant deviation from open-loop approaches. Instead of directly calculating the camera pose variation according to the geometry model, we aim for specific features to reach predefined values (see Figure 1.6). Two major approaches to visual servoing can be detailed:

— Image-Based Visual Servoing (IBVS): the visual features vector $\mathbf{s}(t)$ is expressed in terms of 2D image information, generally the image coordinates of a set of points, but also more complex geometrical shapes [Chaumette et al., 1993], image moments [Chaumette, 2004], photometric data [Collewet & Marchand, 2011], etc. In the usual IBVS cases, $k \geq m$.

— Pose-Based Visual Servoing (PBVS): the visual feature vector is expressed in terms of camera pose computed from image measurements using 3D localization. In the usual PBVS cases, $k = m$.

Both methods have pros and cons. IBVS has the advantage of expressing the visual features only in the image plane without requiring the complete 3D model of the object, except for the depth of the target (which can be roughly estimated). However, the focus is on the image trajectory, and the camera may have a complex trajectory in the 3D space when controlling the 6DOF. PBVS offers more tools for specifying the trajectory of the camera in Cartesian space. Yet, it requires the complete 3D model of the target object, and since the control is not designed through direct image data, the target used to compute the camera pose may exit its field of view, leading to control failure. 2-1/2-D visual servoing [Malis et al., 1999] proposes to combine image features with camera pose in the visual feature vector, which can tackle some drawbacks of IBVS and PBVS, ensuring, for instance, a convenient camera and image trajectory.

Once the error vector is designed, it must be related to the camera motion. The most common approach is to associate this vector with the instantaneous camera velocity $\mathbf{v}_c = (\boldsymbol{\upsilon}_c, \boldsymbol{\omega}_c) \in \mathfrak{se}(3)$ (with $\mathfrak{se}(3)$ the Lie algebra of $SE(3)$), also called the camera kinematic screw, expressed in the camera frame. The components of $\mathbf{v}_c$ are $\boldsymbol{\upsilon}_c$ the instantaneous translational velocity of the camera and $\boldsymbol{\omega}_c$ the instantaneous angular velocity of the camera. Depending on how many degrees of freedom are controllable (and want to be controlled), the camera velocity is often

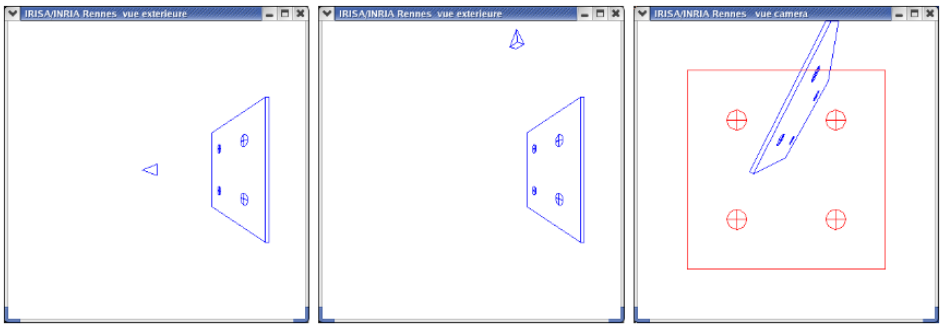


Figure 1.6 – Example of a visual servoing task taken from [Chaumette & Hutchinson, 2008], (a) represents the desired camera pose wrt. a target, (b) is the initial camera pose and (c) is the initial (blue) and desired (red) image of the target from the camera.

reduced ($e.g.$ for a pan-tilt-roll camera, only the instantaneous angular velocity is considered, i.e., up to 3 DOF). The relationship between the camera velocity $\mathbf{v}_c$ and the time-variation of the error is expressed by [Chaumette & Hutchinson, 2008]:

$$\dot{\mathbf{e}} = \mathbf{L}_\mathbf{e}\mathbf{v}_c \tag{1.26}$$

where $\mathbf{L}_\mathbf{e} \in \mathbb{R}^{k\times m}$ is the interaction matrix related to the error, which depends on the chosen visual features.

Finally, to design a velocity controller, i.e., $\mathbf{v}_c$ being the control input, we also need to specify the desired evolution of the error $\mathbf{e}$. The most simple is to consider an exponential decay of the error, such as:

$$\dot{\mathbf{e}} = -\lambda\mathbf{e} \tag{1.27}$$

with $\lambda > 0$ a scalar gain. The velocity controller is then obtained from equations (1.26) and (1.27) giving:

$$\mathbf{v}_c = -\lambda\mathbf{L}_\mathbf{e}^+\mathbf{e} \tag{1.28}$$

with $\mathbf{L}_\mathbf{e}^+ \in \mathbb{R}^{m\times k}$ the Moore-Penrose pseudo-inverse matrix of $\mathbf{L}_\mathbf{e}$. In theory, the interaction matrix is designed analytically through the derivative of visual features, its pseudo-inverse is then computed and used to obtain the control input. In practice, neither $\mathbf{L}_\mathbf{e}$ nor $\mathbf{L}_\mathbf{e}^+$ is known accurately, especially when $\mathbf{L}_\mathbf{e}$ contains 3D parameters that must be evaluated at each iteration and are not available or, at most, barely known. Therefore, estimations of these matrices are used to compute the control, one may then consider, respectively, $\widehat{\mathbf{L}_\mathbf{e}}$ and $\widehat{\mathbf{L}_\mathbf{e}^+}$ the approximations of

$\mathbf{L_e}$ and $\mathbf{L_e^+}$, providing the following control law:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_\mathbf{e}^+} \mathbf{e} \tag{1.29}$$

This control law is the basis of visual servoing controllers. Of course, it can be refined according to the scenario, but a lot of control laws are derived from equation (1.29). In the following, we present one example of the IBVS scheme, as it is one of the concepts on which this thesis relies.

1.3.3 Image-Based Visual Servoing with Image Points Coordinates

Image-Based Visual Servoing (IBVS) is a visual servoing approach whose control law is designed through the consideration of image features. The most common image feature are the 2D coordinates of an image point $\mathbf{x} = (x, y)$ expressed with its normalized coordinates, captured from a 3D point ${}^c\mathbf{X} = (X, Y, Z)$. It is possible to relate the time variation of $\mathbf{x}$ with the camera velocity through the well-known relationship [Chaumette & Hutchinson, 2008]:

$$\dot{\mathbf{x}} = \mathbf{L}_\mathbf{x} \mathbf{v}_c \tag{1.30}$$

with $\mathbf{L_x}$ the interaction matrix of a 2D point, whose full expression is given by:

$$\mathbf{L}_\mathbf{x} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \tag{1.31}$$

Usually, x and y are obtained from image measurements of a pixel point $\mathbf{x}_p = (x_p, y_p)$, mapped into its normalized coordinates with the camera intrinsic parameters, and Z is the depth of the target in the camera frame. To control the 6 DOF of the camera and operate a positioning task, we need k visual features with $k \geq 6$. When using image point features, the case $k = 6$, i.e., 3 points, is practically not used in IBVS because there exist 4 possible camera pose leading to $\mathbf{e} = \mathbf{0}$, which does not allow to reach a unique specified desired pose. To have it, 4 points are traditionally used at minima, the visual error $\mathbf{e} \in \mathbb{R}^8$ and its complete interaction matrix $\mathbf{L_e} \in \mathbb{R}^{8\times 6}$ are thus:

$$\mathbf{e} = \begin{pmatrix} \mathbf{x}_1 - \mathbf{x}_1^* \\ \mathbf{x}_2 - \mathbf{x}_2^* \\ \mathbf{x}_3 - \mathbf{x}_3^* \\ \mathbf{x}_4 - \mathbf{x}_4^* \end{pmatrix} \quad \text{and} \quad \mathbf{L_e} = \begin{pmatrix} \mathbf{L}_{\mathbf{x}_1} \\ \mathbf{L}_{\mathbf{x}_2} \\ \mathbf{L}_{\mathbf{x}_3} \\ \mathbf{L}_{\mathbf{x}_4} \end{pmatrix} \tag{1.32}$$

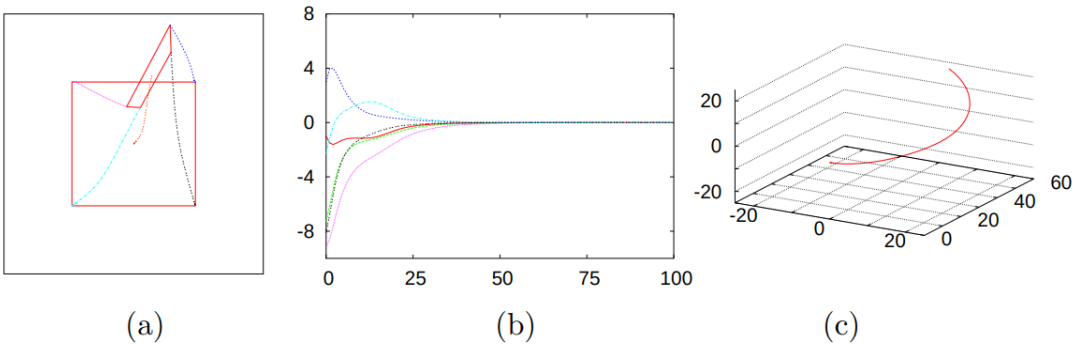


Figure 1.7 – Example of an IBVS behavior from [Chaumette & Hutchinson, 2008] when $\widehat{\mathbf{L}_\mathbf{e}^+} = \mathbf{L}_\mathbf{e}^+$ (a) are the image trajectory of the four points and also the center of the square that is not used in the control law (b) are the control inputs (in cm/s and deg/s) (c) 3D trajectory of the camera (in cm). Each image point follows an almost straight-line trajectory, whereas the camera trajectory describes a circular arc.

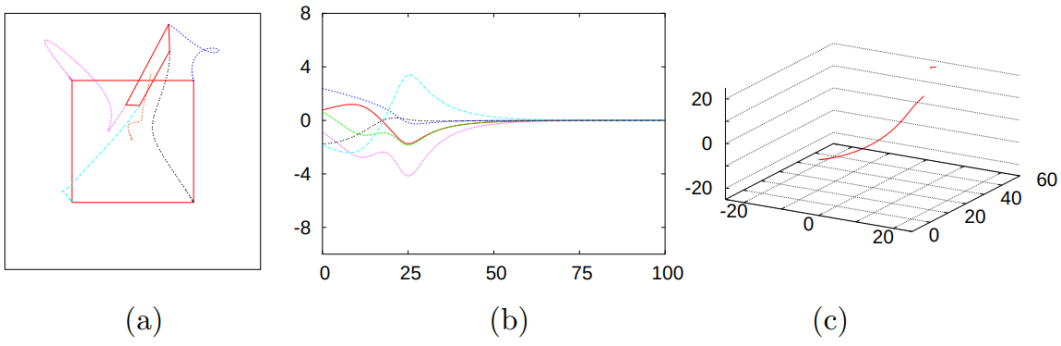


Figure 1.8 – Example of an IBVS behavior from [Chaumette & Hutchinson, 2008] when $\widehat{\mathbf{L}_\mathbf{e}^+} = \mathbf{L}_{\mathbf{e}^*}^+$. Image and camera trajectories are more complex and less foreseeable, yet, the convergence of the control law is not impacted.

The control law is then obtained through equation (1.29). Once again, the parameters of $\mathbf{L_e}$ (normalized coordinates and depth of each point at each iteration of the control) are rarely perfectly established and the interaction matrix is often approximated under certain hypotheses (camera intrinsic parameters roughly set, constant depth, constant interaction matrix computed at a known instant, etc.), which induces a different behavior of the control. Some of these behaviors are shown in Figure 1.7 for a perfect estimation of the interaction matrix ($\widehat{\mathbf{L}_\mathbf{e}^+} = \mathbf{L}_\mathbf{e}^+$) and in Figure 1.8 for an approximation considering that the matrix is constant and evaluated at the desired image positions with the corresponding desired depths($\widehat{\mathbf{L}_\mathbf{e}^+} = \mathbf{L}_{\mathbf{e}^*}^+$ with $\mathbf{e}^* = (\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^*, \mathbf{x}_4^*)$ and $Z_1 = Z_1^*, Z_2 = Z_2^*$, etc.). Indeed, the choice of the approximation for the interaction matrix is essential to ensure or not the expected behavior and preserve the stability of the system.

Special importance is given to the estimation or approximation of the depth Z for each point considered, which is the only 3D parameter involved when considering image point features, but has a noticeable effect on the trajectories. However, the depth Z does not influence the accuracy of the scheme.

1.3.4 Dealing with external motions

After this introduction on visual servoing and the example of an IBVS control scheme, we consider the case of visual control of a camera in the presence of external motions from the scene. These external motions can arise from two causes, either due to the inherent motion of the observed object or as a result of the motion of the robot or vehicle to which the camera is attached, and may disturb the regulation of the visual error. In this case, the equation (1.26) is completed with an additional term [Chaumette & Hutchinson, 2008]:

$$\dot{\mathbf{e}} = \mathbf{L}_{\mathbf{e}}\mathbf{v}_c + \frac{\partial \mathbf{e}}{\partial t} \tag{1.33}$$

where $\frac{\partial \mathbf{e}}{\partial t}$ is the variation of $\mathbf{e}$ due to unknown external motions. Consequently, from Equation (1.33), the velocity controller taking into account external motions and still designed to ensure an exponential decay of the error is expressed by:

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}_{\mathbf{e}}^{+}}\mathbf{e} - \widehat{\mathbf{L}_{\mathbf{e}}^{+}}\widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{1.34}$$

where $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ is the estimation of $\frac{\partial \mathbf{e}}{\partial t}$ to be compensated in the control law. With respect to the source and the characteristic of the motion, different approaches are proposed to compute $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ and thus improve the behavior of the system.

1.3.4.1 Target tracking

We consider the target to have its own unknown velocity. It is indeed a well-known issue, as it generally induces a tracking error on the control using the basic control scheme (equation (1.29)), i.e., the visual features never reach their desired values. This phenomenon becomes even more significant when the target moves fast, the camera's frequency is low (resulting in the target image drifting further), and the target's velocity is non-linear.

The easiest case appears when the target velocity is known. In this context, this issue is addressed by compensating for the target motion directly in the control law, i.e., we have $\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \frac{\partial \mathbf{e}}{\partial t}$,

and the control law can be obtained directly from equation (1.38). For example, a challenging scenario is presented in [Keipour et al., 2022], where a UAV is operating an autonomous landing on a moving vehicle whose motion is supposed to be estimated before visual control.

If the target velocity is unknown, a classic approach in automatic control is to consider that $\frac{\partial \mathbf{e}}{\partial t}$ can be compensated by introducing an integral term into the control law. Taking $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ the estimation of the perturbation, the expression of the "visual" integrator is given by [Chaumette et al., 1991]:

$$\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \mu \sum_{j=0}^{k-1} \mathbf{e}(j) \tag{1.35}$$

where μ is the integral gain and k is the current iteration of the control loop. In case the target moves with a constant velocity, the tracking error is fully regulated after several iterations. This technique has been used, for example, in underwater robotics [E. Marchand et al., 2001] where the pan-tilt orientation of a camera is adjusted to always maintain in the camera field of view the end effector of a manipulator, whose exact motion is not specified in the camera control law. This is an effective way to address tracking errors without having to use more complex estimation methods.

Another approach presented in [Chaumette & Santos, 1993] computes the estimation $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ from equation (1.33) such as:

$$\widehat{\frac{\partial \mathbf{e}}{\partial t}} = \hat{\dot{\mathbf{e}}} - \widehat{\mathbf{L}_\mathbf{e}} \widehat{\mathbf{v}_c} \tag{1.36}$$

and discretizes the quantity $\hat{\dot{\mathbf{e}}} = [\mathbf{e}(t) - \mathbf{e}(t - \Delta t)]/\Delta t$ with $\Delta t = 1/F$, F being the frequency of the control loop. Equation (1.36) shows that the estimation is based on velocity measurements from the image and from the output of the controller. The estimation $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ is thus sensitive to noise and can be improved through filtering methods.

Among these, the Kalman filter [Welch & Bishop, 1995][S. Chen, 2011] is widely used to measure, estimate, and predict certain states of a robotic system. In our scenario, the Kalman filter is used to measure and predict $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ through equation (1.36). It can also be used more generally to track the motion of the target. In this context, linear Kalman filters handle constant velocity and constant acceleration models, while extended Kalman filters allow considering nonlinear motions through a local linearization process. In [Bensalah & Chaumette, 1995], the authors propose to refine linear Kalman filters by detecting jumps in velocity and acceleration that improve the convergence of the estimation. Other filtering approaches can be used to estimate complex target motion, for example, particle filters [Arulampalam et al., 2002][X. Wang et al., 2017], or learning-based filters [Yaqi et al., 2022]. Moreover, if some information on target

motion is available, it can be used to improve the estimation, as in [Allen et al., 1993] where the authors propose to estimate the 3D rigid motion of the target whose repetitive trajectory is barely known and obtain the different motion parameters using several low-pass filters.

Visual servoing together with visual target tracking is now extensively present in the literature, some examples can be found in [C. Liu et al., 2012] and [Zhao et al., 2019] where a Kalman filter is explicitly used in the controller.

1.3.4.2 Perturbations from the system dynamics

The variation of $\frac{\partial \mathbf{e}}{\partial t}$ can also be caused by the dynamics of the system, or more precisely the motion of the robot or vehicle to which the camera is attached. An adequate example is that of a pan-tilt-roll (PTR) camera embedded in a moving vehicle (see Figure 1.9). In this configuration, the frame F_b is free-flying in the world and induces its motion to its end effector frame F_e and by transition to the camera frame F_c, rigidly attached to F_e. In fact, from the point of view of

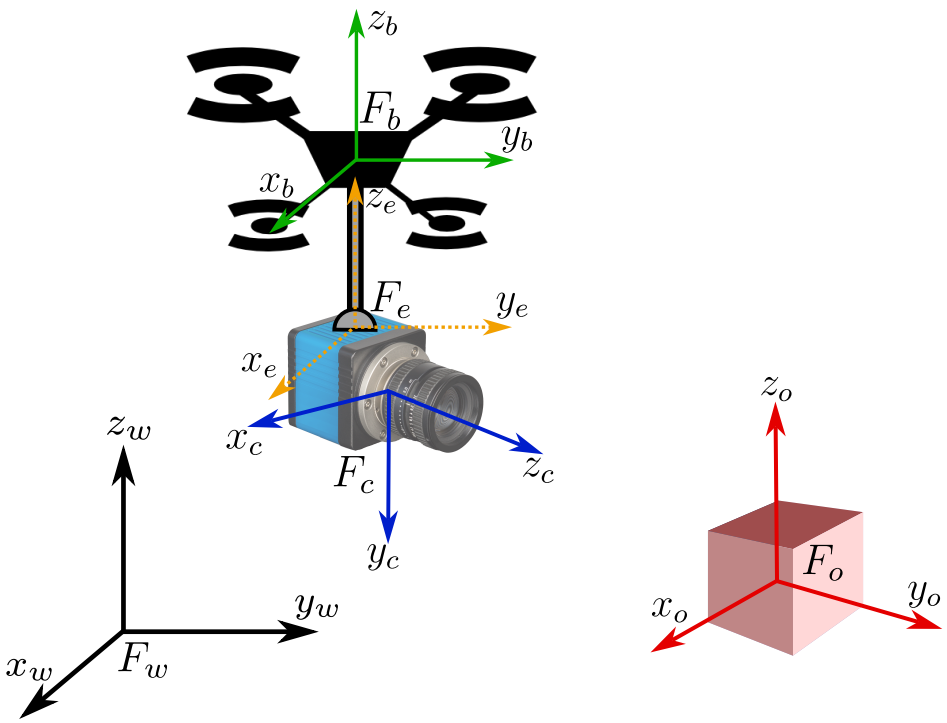


Figure 1.9 – Camera embedded on a moving vehicle (here a drone). Among the common frames F_w, F_c and F_o, the frame F_b is attached to the body of the drone and the frame F_e corresponds to the end effector frame which is here a pan-tilt-roll unit.

the camera, the motion of the vehicle is external, and it is a priori impossible to decorrelate the motion of the target from the motion of the vehicle in the image. We are in the same situation

as expressed by equation (1.33). For example, in [Crétual & Chaumette, 2000] the image-based visual servoing of a pan-tilt camera mounted on an underwater robot is subject to unknown uncontrolled motions, either the one of the target or the one of the submarine, summarized as the motion of the scene. This joint motion is then estimated by a Kalman filter and compensated in the control law. A similar approach is used for the control of an embedded gimbal in a drone [X. Liu et al., 2020] where a Kalman filter is used to estimate and compensate for both the motion of the UAV and the target.

Still, when system dynamics can be established, i.e., when uncontrolled motions are known or very well estimated, it is adequate to incorporate these motions directly in the visual control law [Hashimoto & Kimura, 1995]. It is indeed the same situation as when tracking an object with a known velocity. Let us consider $\mathbf{v}_b$ the velocity of the vehicle in its body frame, equation (1.33) can be rewritten as:

$$\dot{\mathbf{e}} = \mathbf{L}_\mathbf{e}\mathbf{v}_c + \mathbf{L}_\mathbf{e}{}^c\mathbf{V}_b\mathbf{v}_b + \frac{\partial \mathbf{e}_t}{\partial t} \qquad \text{where} \qquad {}^c\mathbf{V}_b = \begin{pmatrix} {}^c\mathbf{R}_b & [{}^c\mathbf{t}_b]_\times {}^c\mathbf{R}_b \\ \mathbf{0}_3 & {}^c\mathbf{R}_b \end{pmatrix} \tag{1.37}$$

with ${}^c\mathbf{V}_b$ the velocity twist matrix from frame F_b to F_c, and $\frac{\partial \mathbf{e}_t}{\partial t}$ the variation of $\mathbf{e}$ due to the unknwon motion of the target only. Consequently, to obtain a controller that compensates for the motion of its inner system (and which is still designed for inducing an exponential decay of the error), we isolate $\mathbf{v}_c$ from equation (1.37) and get:

$$\mathbf{v}_c = -\lambda\widehat{\mathbf{L}_\mathbf{e}^+}\mathbf{e} - \widehat{\mathbf{L}_\mathbf{e}^+}\widehat{\mathbf{L}_\mathbf{e}}\widehat{{}^c\mathbf{V}_b}\widehat{\mathbf{v}_b} - \widehat{\mathbf{L}_\mathbf{e}^+}\widehat{\frac{\partial \mathbf{e}_t}{\partial t}} \tag{1.38}$$

with $\widehat{\mathbf{v}_b}$ the estimation or approximation of the vehicle's velocity.

This approach has been used for different mobile robots that embed a camera controlled in rotations, for instance, for exploration robots [Ma et al., 2020], and in aerial robotics for auto-landing [C. Chen et al., 2021], where the landing target is defined as an image point to center in the image frame of the camera using an IBVS scheme.

1.4 Conclusion

This first chapter allowed us to present the common tools on visual servoing that we will require throughout this thesis. We first introduced some computer vision concepts and presented the main mathematical objects that we will manipulate. Thus, we have discussed homogeneous transformations between frames and defined important notions and properties of these matrices.

We then introduced a camera system by defining the concept of pose, which is widely used in the vision-based control community, and also the pinhole camera model and its associated perspective equations. We then presented the vision-based control approaches, focusing on visual servoing, which is the main scheme we will use in this thesis. With the same intention, we presented in more details the case of IBVS, but also control laws to manage disturbances due to external movements to the system. With this background in robotics and visual control in mind, we can now delve into the main system considered in this thesis: a Low Earth Orbit satellite.

CHAPTER 2

CONTROL OF A LOW EARTH ORBIT SATELLITE

2.1 Introduction

Our focus in this chapter is on the control of Low Earth Orbit (LEO) satellite. A Low Earth Orbit satellite, as the name suggests, follows an elliptical path around the Earth at an altitude below 2000 km. LEO orbits have gained significant interest in recent times as a result of several advantages in terms of cost, dynamics, and performance.

Firstly, placing satellites in LEO requires less energy compared to other orbit types, resulting in lower costs and relatively affordable access to space. Then, LEO satellites can complete more than 10 orbits per day, depending on their altitude, allowing for frequent flybys of the Earth. This frequent coverage is particularly appealing for Earth observation satellites, allowing for regular acquisitions of Earth data, especially images, in the case of imagery satellites.

However, Earth observation satellites face a challenge when it comes to pointing their instruments toward a precise location on the ground. The consequent orbital motion of the satellite requires the use of an agile attitude control system to compensate for this motion and maintain a precise pointing accuracy. Indeed, the actuation of such systems plays an essential role but is subject to orbital constraints, which necessitates the consideration of appropriate technologies. In the pursuit of controlling a satellite to map precise locations on the ground, these actuators must be accompanied by precise commands.

In this chapter, we will present the principle laws of mechanics to describe celestial motions. The case of a satellite in a circular orbit will be studied, together with the motion of a static object on the Earth's surface. Then, we will explore how the satellite attitude is typically controlled, allowing us to explain the architecture of the Attitude Control Subsystem. Later, this will lead us to consider vision-based control as a popular and modern approach for satellite guidance. Finally, we will conclude this study and discuss the objectives of the project this thesis belongs to and the directions we assume to take to design an agile video satellite.

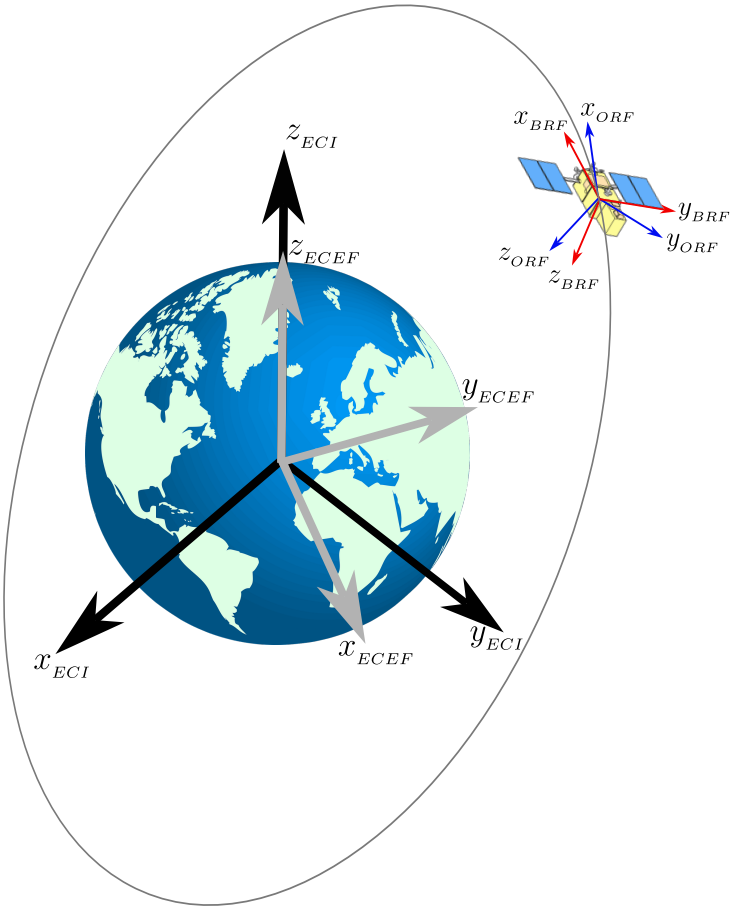


Figure 2.1 – Reference frames, Earth-Centered Inertial (ECI) frame in black, Earth-Centered Earth-fixed (ECEF) frame in gray, Body Reference Frame (BRF) in red, and Orbit Reference Frame (ORF) in blue.

2.2 Orbital mechanics

This section recalls orbital mechanics laws for satellites in a low circular Earth orbit and the spherical motion of an object lying on the Earth's surface. First, we define the reference frames, then, we simplify Kepler's law of motion in the case of a circular orbit satellite, and we express the satellite's position relatively to the Earth-Centered Inertial frame. Finally, we express the position of any particular point on Earth with respect to the same reference frame.

2.2.1 Reference frames

Within the astronautical community, orbital and terrestrial motions are expressed in different reference frames, defined in Figure 2.1. They are explained below:

— ECI: the Earth-Centered Inertial coordinate frame has its origin at the center of mass of

the Earth and is fixed with respect to the stars. Therefore, this frame does not rotate with the Earth and is particularly useful for describing orbital motions

— ECEF: the Earth-Centered Earth-Fixed coordinate frame has its origin located in the center of mass of the Earth similarly to the inertial frame. However, it rotates with the Earth, and this frame is often used to express terrestrial locations with latitude and longitude.

— BRF: the Body Reference Frame, has its origin in the center of mass of the satellite. Its axes are fixed within the satellite body, typically aligned with the satellite's rotational axes.

— ORF: the Orbit Reference Frame, shares the same origin as the BRF, with its z-axis pointing toward the center of the Earth, and its x-axis tangent to the orbit in the direction of the motion.

Each of these frames has a specific interest, especially in expressing orbital and terrestrial trajectories.

2.2.2 Satellite trajectory

Let us now consider the few laws that allow to describe satellite movements around the Earth based on [Curtis, 2013]. In this subsection, we focus only on point mechanics, i.e., we focus on the natural translations of a satellite around the Earth, summarized by its center of mass $\mathbf{X}_s$, which will eventually be expressed in the ECI frame: ${}^{ECI}\mathbf{X}_s \in \mathbb{R}^3$. We decide to use the same notation as used in [Curtis, 2013] as they are common in the astronautic area. Yet, these notations might be confusing with other notations of the robotics community, so, we will consider them in this chapter only. Furthermore, we only study circular orbit, first because it is faster to express and secondly because the satellite that we will consider is supposed to have a circular orbit.

First, we know that, according to Kepler's first law or the law of orbits, every object in orbit around the Earth describes an elliptical motion. Let us consider that the satellite is at a distance $r = ||{}^{ECI}\mathbf{X}_s||$ from the Earth center, we have $r = R_E + z_{alt}$ with R_E and z_{alt} respectively the radius of the Earth and the altitude of the satellite. It is commonly said that the satellite has an orbit of z_{alt} km. The expression of Kepler's first law is given by:

$$r = \frac{h^2}{\mu} \frac{1}{1 + e\cos\theta(t)} \tag{2.1}$$

with:

— h the relative angular momentum of the satellite per unit of mass, which measures the

revolution of the satellite around the Earth, it will be specified later,
— μ the geocentric gravitational constant, which represents the strength of the Earth's gravitational attraction,
— e the eccentricity of the orbit, it measures the shape of an orbit, basically $e = 0$ for a perfect circular orbit and $e = 1$ for a perfect ellipse.
— θ is the true anomaly, it is the angle position of the satellite in its perifocal frame.

Since we consider a circular orbit, $e = 0$ and r is constant with respect to time, we have directly:

$$r = \frac{h^2}{\mu} \tag{2.2}$$

Let us consider the plane $(\mathbf{X}_s, u_r, u_\perp)$, which is the plane where the motion of the satellite is contained with u_r the radial unit vector and $u_\perp$ the azimuthal unit vector. When considering circular orbits, u_r is perpendicular to the orbital trajectory and oriented outward while $u_\perp$ is tangent to the orbital motion and oriented in its direction. They are shown in Figure 2.2. The velocity $\boldsymbol{\upsilon}_s \in \mathbb{R}^2$ of the satellite in this plane is $\boldsymbol{\upsilon}_s = (\upsilon_r, \upsilon_\perp)^T$ with $\upsilon_\perp$ the azimuth component of the satellite's velocity in its orbit. In a circular orbit, the second component of the velocity, the radial velocity $\upsilon_r = \dot{r} = 0$ because r is constant. The angular momentum h is expressed as follows:

$$h = ||\mathbf{r} \times \boldsymbol{\upsilon}_s|| \tag{2.3}$$

with $\mathbf{r}$ the translation vector representing the distance of the satellite to the Earth expressed in terms of radial and azimuthal components. In a circular orbit, this vector is expressed by $\mathbf{r} = (r, 0)$. Therefore, we can simplify the angular momentum by $h = r\upsilon_\perp$. By substituting equation (2.3) into equation (2.2) we obtain the expression of the azimuth velocity of the satellite in a circular orbit:

$$\upsilon_\perp = \sqrt{\frac{\mu}{r}} \tag{2.4}$$

$\upsilon_\perp$ is constant, so we can calculate the period of a circular orbit, which is the ratio of the trajectory circumference to the speed:

$$T = \frac{2\pi r}{\upsilon_\perp} = \frac{2\pi}{\sqrt{\mu}} r^{\frac{3}{2}} \tag{2.5}$$

Now we will express the position of the satellite in its perifocal plane P_P, i.e., the plane (O, x_P, y_P) with O the center of the Earth, x_P and y_P are lying in the P_{ORF}, with x_P directed towards the periapsis of the orbit (i.e., the position of the satellite when $\theta(t) = 0$) and y_P is 90

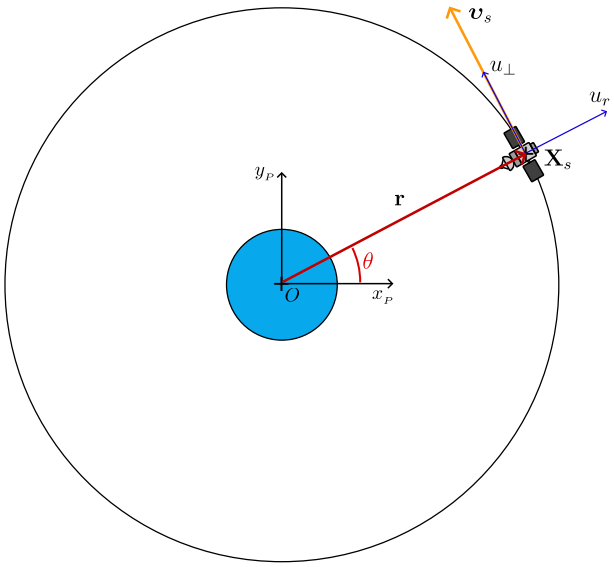


Figure 2.2 – Satellite trajectory on its perifocal plane

degrees counterclockwise (see Figure 2.2). We consider the parameter that varies the position of the satellite in its orbit, the true anomaly: $\theta(t)$. We wish to relate this parameter to time t to be able to express the position of the satellite at any moment. We can use another expression for $v_\perp$ based on the tangential velocity of a circular motion, expressed by:

$$v_\perp = r\dot{\theta} \tag{2.6}$$

which then gives:

$$h = r^2\dot{\theta} \Leftrightarrow \frac{d\theta}{dt} = \frac{h}{r^2} \tag{2.7}$$

It follows, after using equation (2.2) in equation (2.7) and separating the variables, that:

$$d\theta = \frac{\mu^2}{h^3}dt \tag{2.8}$$

We integrate the previous expression between t_p and t where t_p is the time at the passage of the periapsis, and t is the current time.

$$\theta(t) - \theta_p = \frac{\mu^2}{h^3}(t - t_p) \tag{2.9}$$

with θ_p the true anomaly of the periapsis which is by definition equal to zero. Substituting h

using equation (2.2) in equation (2.9) we obtain:

$$t = \frac{r^{\frac{3}{2}}}{\sqrt{\mu}}\theta(t) + t_p = \frac{T}{2\pi}\theta(t) + t_p \tag{2.10}$$

The position of the satellite in its perifocal plane can now be expressed as a 2D translation vector ${}^{P}\mathbf{x}_s \in \mathbb{R}^2$ such as:

$${}^{P}\mathbf{x}_s = \begin{pmatrix} r\cos\theta(t) \\ r\sin\theta(t) \end{pmatrix} \tag{2.11}$$

We finally aim to express the coordinates of the satellite in the Earth-Centered Inertial frame $(x_{ECI}, y_{ECI}, z_{ECI})$ (see Figure 2.3). As the ECI and the perifocal frame have the same origin, the transition between the perifocal frame and the ECI is simply a rotation matrix, traditionally expressed by the classical Euler angle sequence $\mathbf{R}_z(\Omega)\mathbf{R}_x(i)\mathbf{R}_z(\omega)$. These angles are called orbital parameters, they are defined by:

— i is the orbit inclination, $i \in [0, 180]°$;
— Ω is the longitude of the ascending node, it determines where the two intersections of the orbit with the equatorial plane will be located, $\Omega \in [0, 360]°$;
— ω is the argument of periapsis, $\omega \in [0, 360]°$.

However, in the case of a circular orbit, we can assume that the periapsis has no direct meaning here, and it is often placed on the ascending node, so $\omega = 0$ and $\mathbf{R}_z(\omega) = \mathbf{I}_3$. The two arguments allowing us to define the position of the circular orbit according to the Earth-Centered Inertial frame are thus the inclination i and the longitude of the ascending node Ω. We now introduce ${}^{ECI}\mathbf{R}_P \in SO(3)$ the rotation matrix between the perifocal plane and the ECI, given by:

$${}^{ECI}\mathbf{R}_P = \mathbf{R}_z(\Omega)\mathbf{R}_x(i) = \begin{pmatrix} \cos\Omega & -\sin\Omega & 0 \\ \sin\Omega & \cos\Omega & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos i & -\sin i \\ 0 & \sin i & \cos i \end{pmatrix} \tag{2.12}$$

Then we obtain:

$${}^{ECI}\mathbf{R}_P = \begin{pmatrix} \cos\Omega & -\sin\Omega\cos i & \sin\Omega\sin i \\ \sin\Omega & \cos\Omega\cos i & -\cos\Omega\sin i \\ 0 & \sin i & \cos i \end{pmatrix} \tag{2.13}$$

Finally, after converting the 2D vector ${}^{P}\mathbf{x}_s$ into a 3D vector ${}^{P}\mathbf{X}_s \in \mathbb{R}^3$ such that ${}^{P}\mathbf{X}_s =$

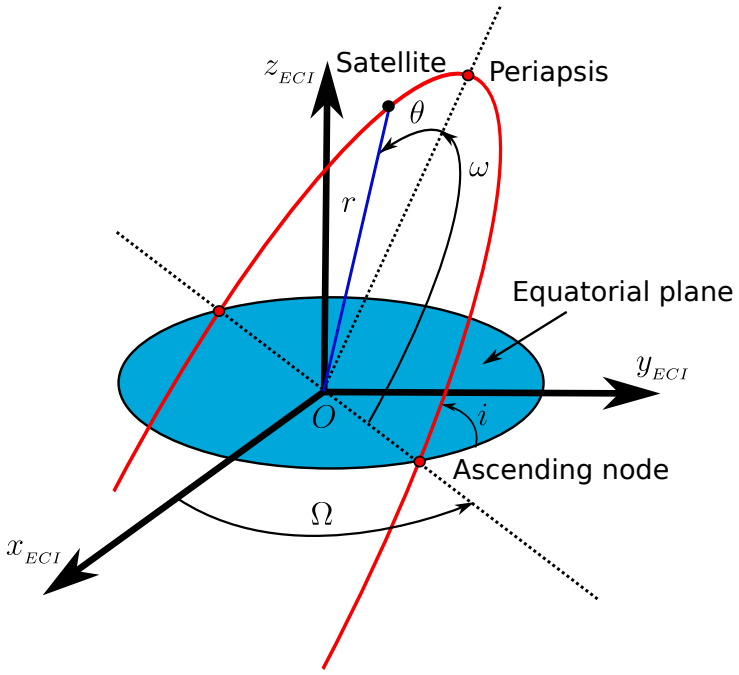


Figure 2.3 – Earth-Centered Inertial frame with orbital parameters

$({}^{P}\mathbf{x}_s, 1)$, the satellite position is given by:

$$ {}^{ECI}\mathbf{X}_s = {}^{ECI}\mathbf{R}_P \, {}^{P}\mathbf{X}_s = \begin{pmatrix} r\cos\theta(t)\cos\Omega - r\sin\theta(t)\sin\Omega\cos i \\ r\cos\theta(t)\sin\Omega + r\sin\theta(t)\cos\Omega\cos i \\ r\sin\theta(t)\sin i \end{pmatrix} \tag{2.14} $$

which determines the position of the satellite in the ECI frame at any time t. As the point $\mathbf{X}_s$ is located in the center of mass of the satellite, it is also the origin of the BRF. We directly have ${}^{ECI}\mathbf{t}_{BRF} = {}^{ECI}\mathbf{X}_s$.

2.2.3 Terrestrial object trajectory

The satellite targets a terrestrial object, we thus want to express the position of an object point o in the Earth-Centered Inertial frame ${}^{ECI}\mathbf{X}_o \in \mathbb{R}^3$.

Traditionally, a location on the Earth is expressed by its longitude $\alpha \in [-180, 180]^\circ$ and latitude $\delta \in [-90, 90]^\circ$ which are fixed wrt. to the Earth-Centered Earth-Fixed frame. The position of an object in the ECEF frame can be expressed using the projection of the spherical

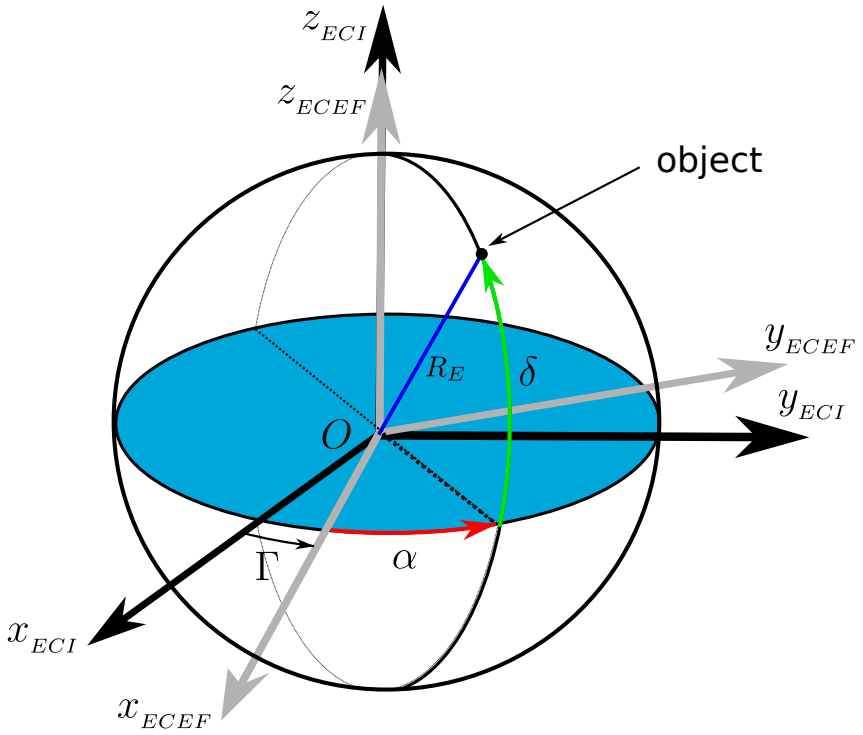


Figure 2.4 – Object in the Earth-Centered Earth-Fixed frame

coordinates (see Figure 2.4) with the hypothesis that the Earth is a perfect sphere:

$$^{ECEF}\mathbf{X}_o = \begin{pmatrix} R_E \cos\delta \cos\alpha \\ R_E \cos\delta \sin\alpha \\ R_E \sin\delta \end{pmatrix} \tag{2.15}$$

The transition matrix between the ECI and the ECEF frames can be hypothesized to be simply a rotation matrix around the z_{ECEF} if we assume that z_{ECEF} and z_{ECI} perfectly coincides. Under this hypothesis, we introduce the rotation matrix $^{ECI}\mathbf{R}_{ECEF} \in SO(3)$. The argument of this rotation is $\Gamma(t) \in [0, 360]°$, the Greenwich Sidereal Angle, a time-dependent argument that will be detailed later. Thus, $^{ECI}\mathbf{R}_{ECEF} = \mathbf{R}_3(\Gamma(t))$ and we have:

$$^{ECI}\mathbf{R}_{ECEF} = \begin{pmatrix} \cos\Gamma(t) & -\sin\Gamma(t) & 0 \\ \sin\Gamma(t) & \cos\Gamma(t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.16}$$

Then, the position of the Earth object in the ECI is given by:

$$^{ECI}\mathbf{X}_o = {}^{ECI}\mathbf{R}_{ECEF}\ {}^{ECEF}\mathbf{X}_o \begin{pmatrix} R_E \cos\delta(\cos\alpha\cos\Gamma(t) - \sin\alpha\sin\Gamma(t)) \\ R_E \cos\delta(\cos\alpha\sin\Gamma(t) + \sin\alpha\cos\Gamma(t)) \\ R_E \sin\delta \end{pmatrix} \tag{2.17}$$

which can be simplified by:

$$^{ECI}\mathbf{X}_o = \begin{pmatrix} R_E \cos\delta\cos(\alpha + \Gamma(t)) \\ R_E \cos\delta\sin(\alpha + \Gamma(t)) \\ R_E \sin\delta \end{pmatrix} \tag{2.18}$$

The Greenwich Sidereal Angle $\Gamma(t)$ is generally computed through a deeper process that allows expressing the orientation of the ECEF at any moment t, expressed in a certain time reference. We are not interested in expressing perfectly this quantity, we assume:

$$\Gamma(t) = \omega_E t + \Gamma_0 \tag{2.19}$$

with $\omega_E = 7.29217 \times 10^{-}5$ rad/s the Earth's rotation speed, and Γ_0 is the original angle.

Assumption on t_p and Γ_0

In our work, the time at the passage of the periapsis t_p and Γ_0 are supposed arbitrarily set to attain any initial configuration. For instance, if we put $t_p = 0, \Gamma_0 = 0$, at $t = 0$ the satellite is initially in the equatorial plane, and the ECEF is oriented from the latitude α wrt. to the ECI.

However, $^{ECI}\mathbf{X}_o$ is only correct for a static object on the Earth's surface. Indeed, if the Earth object is a vehicle, then α and δ are time-dependent and vary according to the object's motion.

2.3 Control of the satellite attitude

The term *control* can have different significations. Indeed, the *control* in the robotics community encompasses all the techniques to govern the behavior of a robot, from high-level strategies (perception, planning) to low-level ones (execution). In the astronautical community, the *control*, and especially the *attitude control* refers primarily to the low-level controller. Indeed, the full modification of the satellite attitude is handled by two classically decorrelated processes.

— The attitude **guidance**, which aims to compute a desired attitude from the user specifi-

cations (for instance, a certain position on the Earth). The attitude guidance represents the mission of the satellite.

— The attitude **control**, which aims to actuate the satellite in order to obtain a certain orientation specified by the attitude guidance subsystem. It encompasses the study of the actuators, the dynamics of the satellite, and the external torques that might be applied to the spacecraft. It relies on attitude measurements to estimate and adjust the current attitude. The attitude control is the execution of the satellite, it converts a command into an orientation.

With these definitions, we propose two categories of satellite control:

— Classical control: the attitude guidance is an open-loop process, the attitude control reacts to a pre-computed trajectory.

— Real-time sensor-based control: the attitude guidance is a closed-loop process, the desired attitude is adjusted continuously according to sensor feedback. The attitude control responds to a trajectory computed online.

Classical control is widely studied in the literature because it is the conventional way to control Earth-pointing satellites, and we will briefly present it in the following. Real-time sensor-based control is an interesting solution in order to enhance the pointing accuracy of observation satellites and automate on-orbit servicing. Vision-based control, which is a real-time sensor-based control, will be explored in the next subsection.

The satellite attitude is the orientation of the satellite body frame (BRF) with respect to a reference frame. In the literature, the frame of reference for the orientation of the satellite can vary between the ECI and the orbital frame of reference (ORF). In our case, we will consider the orientation wrt. the ECI, because taking a rotating frame as the reference frame assumes that this rotation is also naturally operated by the satellite, which is most of the time not the case for LEO satellites, where the orientation is generally fully controlled.

2.3.1 Classical control

Let us present the traditional attitude control and attitude guidance subsystems, i.e., the attitude control is reacting to a precomputed trajectory provided by the attitude guidance, which defines the satellite mission.

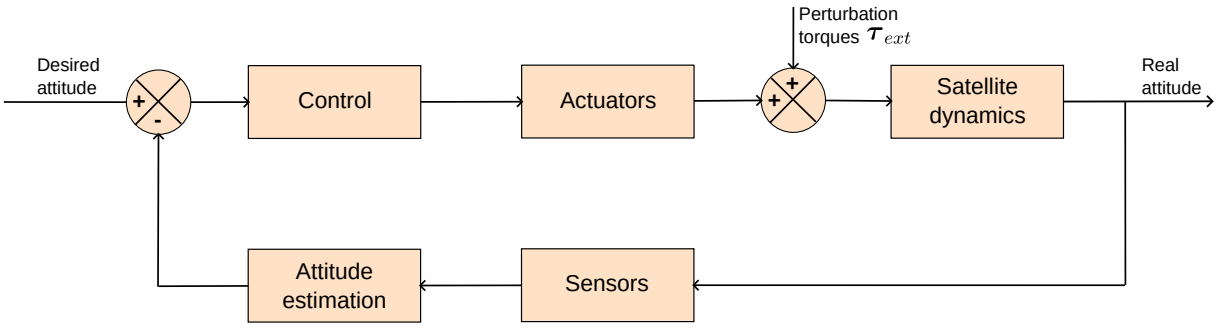


Figure 2.5 – Attitude Control Subsystem

2.3.1.1 Attitude control

The classical and overseen techniques to control a satellite is the consideration of the ACS: Attitude Control Subsystem. It is in charge of controlling the 3 axes of the satellite attitude by receiving a command and operating it through actuators. Figure 2.5 show an example of a typical ACS. Such subsystems are prevalent in satellites accross all orbits, yet, they can be specified for LEO satellites.

First, the external torques applied to the satellite in LEO are the following: $\boldsymbol{\tau}_{gravity}$ the torque due to the Earth's gravity gradient, $\boldsymbol{\tau}_{aero}$ the aerodynamic torque due to the atmospheric drag, $\boldsymbol{\tau}_{magnetic}$ the torque due to magnetic field and $\boldsymbol{\tau}_{solar}$ the torque due to the solar radiation pressure [Won, 1999]. The three firsts are more present in LEO than in geostationary orbit (GEO) because of their proximity to the Earth. These torques can be grouped as external perturbations, denoted:

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{gravity} + \boldsymbol{\tau}_{aero} + \boldsymbol{\tau}_{magnetic} + \boldsymbol{\tau}_{solar} \tag{2.20}$$

To set the satellite in motion, the most common actuators for LEO satellites are either reaction wheels (RW) or control momentum gyroscopes (CMG). These actuators use electricity obtained from the solar panels to exert a torque on the rotational axes of the satellite from a motorized wheel. For the reaction wheels, the motor alters the angular speed of the wheel which provides a torque to the whole system. To control the 3 rotational axes, the same amount of RWs are commonly used, one per axis of rotation. The positions of the RW with respect to the BRF induce different rotational axes and, consequently, different performances. Still, it overstands other passive controllers, such as gravity gradient stabilization or magnetic control, in terms of pointing accuracy because of their dependency on either the gravitational field or the geomagnetic field [Ismail & Varatharajoo, 2010]. For the control momentum gyroscope (CMG) one single wheel is used, which now rotates at a constant speed. The wheel is then tilted by a gimbal

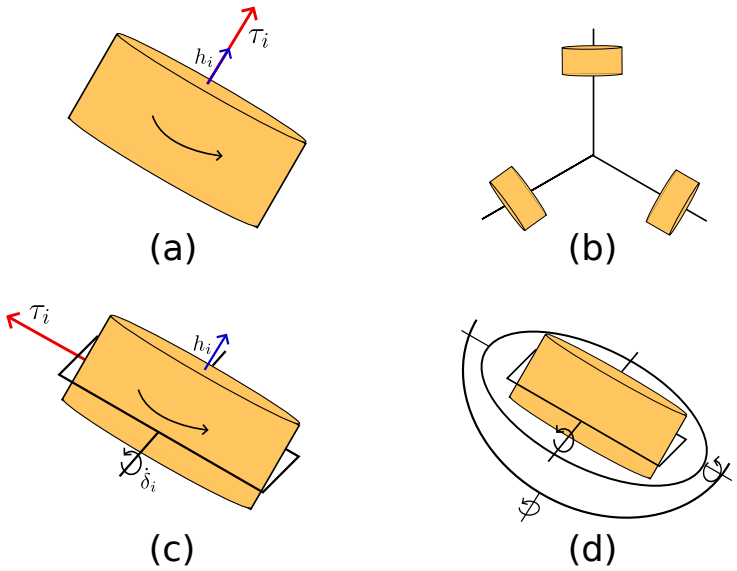


Figure 2.6 – Principles of reaction wheel and control moment gyroscope, (a) RW exerting a torque $\boldsymbol{\tau}_i = \mathbf{h}_i$ with $\mathbf{h}_i$ the angular momentum of the rotating wheel, (b) Composition of RWs to control 3 rotational axes, (c) CMG exerting a torque $\boldsymbol{\tau}_i = \dot{\boldsymbol{\delta}}_i \times \mathbf{h}_i$ with $\dot{\boldsymbol{\delta}}_i$ the angular speed of the gimbal, (d) CMGs: a 3 DOF gimbal to orientate the rotating wheel and so the resulting torque.

to generate a torque whose possible direction relies on the number of degrees of freedom of the gimbal (Figure 2.6). These electric actuators are widely used as they do not require onboard fuel and provide consequent torques. In fact, CMGs can provide considerably more torque than RWs for less power [Votel & Sinclair, 2012]. Consequently, CMGs become more appealing because of the agility they can provide to a satellite. They offer more torque for less energy, enabling higher angular accelerations and the ability to operate at higher velocity levels compared to RWs, which can achieve similar results but require more power and additional mass [Lappas et al., 2002][Thieuw & Marcille, 2007]. In the end, these technologies can provide a torque $\boldsymbol{\tau}_{act}$ to actuate up to three rotational axes of the satellite.

Finally, the control law designs the torque $\boldsymbol{\tau}_{act}$ to be sent to the actuators. This controller is designed based on satellite dynamics. One possible model for it can be obtained considering the rotation of a rigid body, the Euler's rotation equations give us:

$$\mathbf{J}\dot{\boldsymbol{\omega}}_s = \boldsymbol{\omega}_s \times \mathbf{J}\boldsymbol{\omega}_s + \boldsymbol{\tau}_{ext} + \boldsymbol{\tau}_{act} \tag{2.21}$$

with $\mathbf{J}$ the inertial matrix and $\boldsymbol{\omega}_s = {}^{ECI}\boldsymbol{\omega}_{BRF}$ the angular velocity of the satellite, expressed in the inertial frame here.

The torque controller $\boldsymbol{\tau}_{act}$ is then chosen to induce a particular behavior of the system. The

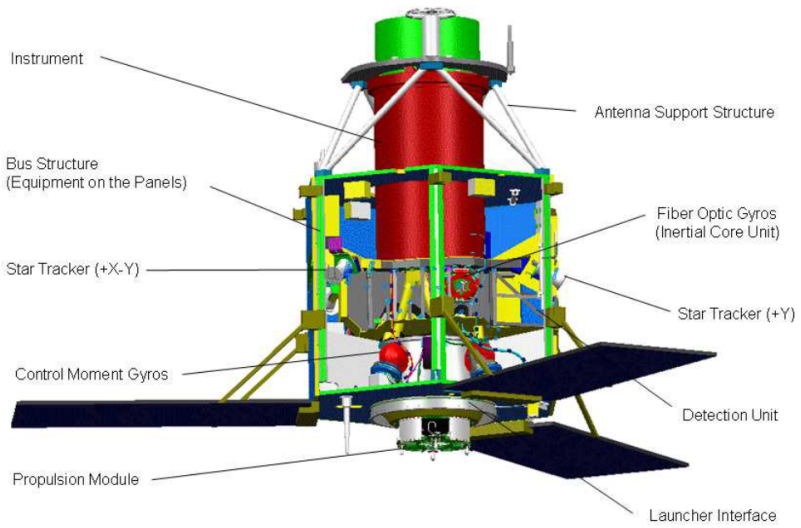


Figure 2.7 – Architecture of the Pléiades satellite from [Gleyzes et al., 2012]. The CMGs are located near the center of mass of the satellite.

most well-known and most used in the ACS is a Proportional Derivative (PD) controller considering that the external torques are negligibles, inducing a first-order system on the angular velocity [Won, 1999]. The advantages of such controllers is the design simplicity and their robustness. [X. Chen et al., 2000] proposes a Proportional Integral Derivative (PID) controller with the same specifications that also takes into account the external disturbances of the system, together with a quaternion formulation in order to avoid the singularity of Euler's angle representation. More intricate control approaches have also been investigated, for instance, [Kristiansen & Nicklasson, 2005] proposes a nonlinear control based on quaternion representation and an integrator backstepping. [Won, 1999] also considers the resolution of the attitude tracking as an optimal control problem, through the minimization of the state of the system (attitude and angular velocity) and the control (the actuator torques) with respect to the satellite dynamics. More recently, an optimal control approach using a nonlinear model-predictive control is presented in [Elbeltagy et al., 2018].

Now that the ACS has been detailed, it can be considered as an independent process that reacts to an attitude request and orientates the satellite. In terms of control, we will consider it as the low-level controller.

2.3.1.2 Attitude guidance

Classical satellite control, as previously defined, involves adjusting the satellite's orientation based on a pre-computed attitude profile provided by the attitude guidance subsystem,

and maintaining it. For Earth observation satellites, this profile consists of the set of orientations required to align the satellite's instruments with a specific location on the Earth's surface. Typically, this desired orientation is determined by using kinematics to calculate the angular difference necessary to align the sensor's line of sight with the desired terrestrial location $\mathbf{X}^*$. To account for the Earth's rotation, the desired location $\mathbf{X}^*$ is traditionally expressed in a fixed reference frame, commonly the Earth-Centered Inertial (ECI) frame.

Initially, the sensor's line of sight crosses the surface of the Earth at the focus point $\mathbf{X}$, which is standardly the Nadir point when the sensor line is aligned with the z-axis of the ORF, which always points toward the center of the Earth. It is then transformed into the satellite's body frame (BRF) to account for the ORF's rotation, compensating for the satellite's orbital translation at Nadir. Finally, it is expressed in the ECI frame through the satellite's orbital parameters, which encapsulate the satellite's translational motion around the center of the Earth.

Then, if the satellite is missioned to map the surface of the Earth, the attitude profile will not be adjusted, so the focus point travels on the surface of the Earth to acquire an image swath [Gleyzes et al., 2012][Soubirane, 2019]. If the satellite then needs to point and hold an exact fixed location on the ground, then the attitude profile will always compensate for the orbital translation of the satellite, and the rotational motion of the Earth [X. Chen et al., 2000][Elbeltagy et al., 2018]. In [Gleyzes et al., 2012], the satellite has also a "video" mode to obtain more images from approximately the same location.

However, the classical control approach has some limitations. First, because the computation of the attitude guidance is made offline, the open-loop approach can induce tracking error for many reasons including sensor drift and internal/external perturbations not considered in the model. Secondly, if the desired spot on the Earth's surface is an object whose location on the ECEF is barely known (for instance, a static vehicle located by GPS), the classical control can only operate within the given input parameters. Finally, if the specific location is moving in the ECEF, i.e., a mobile object on the Earth's surface, it becomes increasingly challenging, and in some cases, even impossible to precisely track the target.

In these situations, the orientation of the satellite would then need to be adjusted with additional information becoming available while the mission is operated. In simple terms, it would be interesting to consider a closed-loop attitude guidance. This allows us to introduce real-time sensor-based control as a solution to these different issues. One particular and appealing real-time sensor-based control is vision-based control, which is now addressed.

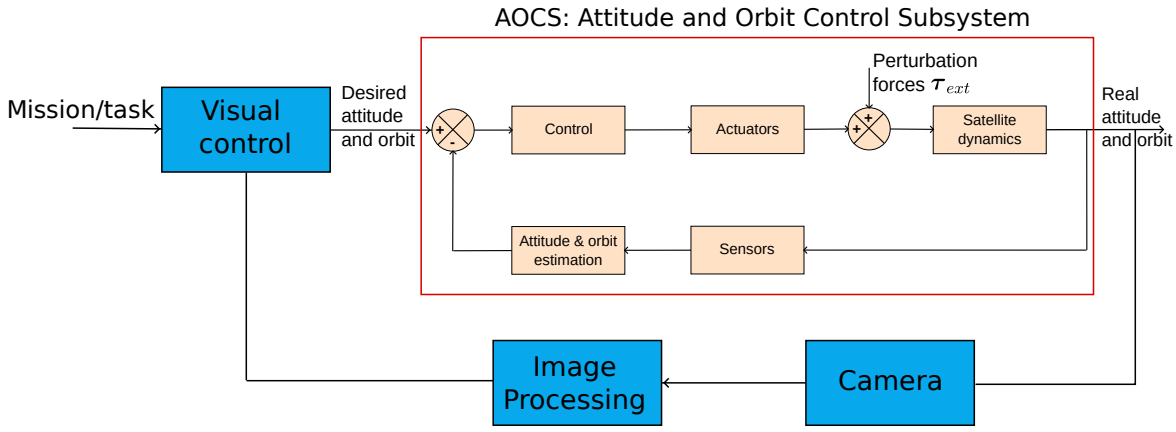


Figure 2.8 – Satellite vision-based control loop

2.3.2 Vision-based control

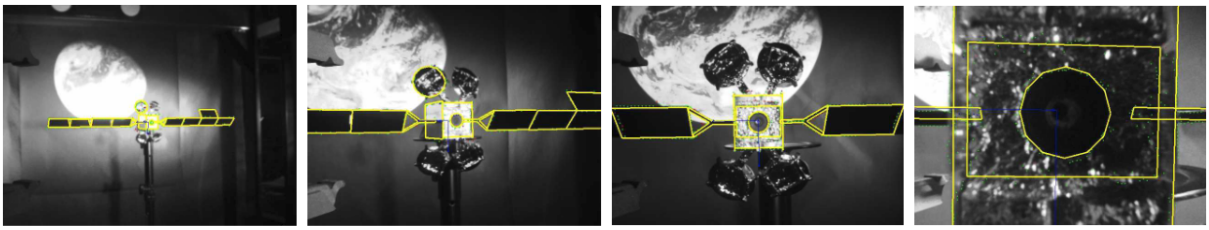
Satellite vision-based control is the consideration of a closed-loop attitude guidance that refines in real-time the attitude profile sent to the low-level controller based on image data obtained from an embedded camera. In the literature, satellite vision-based control strategies are not only considering the control of the orientation of the satellite but at times its translations. In this case, we speak about attitude and orbit guidance, and attitude and orbit control subsystem (AOCS) for the low-level controller and, of course, other actuators are considered in order to alter the translation of the satellite, i.e., propulsion systems. They will not be investigated in this thesis, but one may find some references in [Myers et al., 1994][Tummala & Dutta, 2017]. A possible vision-based control scheme is given by Figure 2.8. Vision-based attitude and orbit guidance are almost all designed with visual servoing.

2.3.2.1 On-orbit tasks

Visual servoing, employed for satellite attitude and orbit guidance, has seen extensive application, as it provides autonomous and efficient control for solving precision on-orbit tasks.

One challenging and thorough application is satellite rendezvous, or capture. In this scenario, the visual servoing scheme aims to approach the chaser satellite to the target satellite, then align with it, and finally operate the rendezvous. [M. Liu & Jasiobedzki, 2002] proposes a PBVS to control the 6 degrees of freedom (DOF) of the chaser satellite through the aforementioned tasks using a model-based pose estimation to compute the 3D-position of the target and

Figure 2.9 – Camera view during visual servoing from [Inaba et al., 2003]. The end-effector is perfectly converging to the grasping handle thanks to the visual positioning task.



(a) $d = 1.56m$ (b) $d = 0.62m$ (c) $d = 0.43m$ (d) $d = 0.14m$

Figure 2.10 – Camera view during visual servoing from [Petit et al., 2011]. The satellite is approaching the target satellite while keeping its centroid in the center of the image frame wrt. to different distance d from the target.

the chaser [Jasiobedzki et al., 2002]. Indeed, PBVS is well-qualified to operate in such situations where the trajectory of the chasing satellite must be smooth and accurate, moreover, the 3D model of the satellite target is commonly perfectly known. In [Inaba et al., 2003], an IBVS scheme is then presented to control both the motion of the chasing satellite and the motion of its gripper in different modes. Visual markers, composed of 2 circles, are plugged into the satellite target (see Figure 2.9). From these 2 circles, 5 visual information are extracted (2 centroids coordinates and the area ratio of the two circles) which allows controlling up to 5 DOF. The control law is also refined with the estimation of the constant motion of the satellite target.

Hybrid visual servoing has been also considered by [Petit et al., 2011], together with a 3D model-based tracker. The satellite target model is made up of lines which provides good robustness to common space acquisition constraints. The error vector is made up of 2D 1/2 visual features (i.e., image data and pose) and provides control over the 6 DOF of the satellite (Figure 2.10).

Naturally, consequent works have been pursued on space manipulators. [Alepuz et al., 2016] presents a direct 6-DOF IBVS to control a robotic arm through a camera mounted on an eye-in-hand configuration. Visual features rely on 4 points observable on the target, whose desired

positions in the image frame are not necessarily fixed, inducing the tracking of complex image trajectory. The term *direct* means that the control is expressed directly as a torque to be applied to the robot joint, to take into account its dynamics in the control loop. A similar approach is presented in [H. Wang et al., 2017], either in a planar or in a three-dimensional situation. In this case, the desired position of the feature points remains fixed, but the target is moving with a nonlinear motion which is estimated through the design of an observer. In [Abdul Hafez et al., 2017], the authors broaden the study with a multi-arm robot capable of up to 14 DOF that may be controlled by direct IBVS.

Indeed, visual servoing is now widely used for space manipulation, yet, the approaches presented above are specific to use cases where the satellite and the target are really close to each other (in the same orbit) and where the relative velocity between them is almost null.

After having taken a look at the applications of visual servoing for space interaction, let us focus on the applications of observation satellites. An interesting and growing application at the moment is space situational awareness, i.e., space-to-space observation for space debris monitoring. In fact, the research in this field is mainly focused on the tracking of space bodies through the estimation of their motion [Ramos et al., 2018]. It is still an open problem as visual tracking of general debris, without previous knowledge of the target, is a tricky task. Still, if the position in the image of the space target can be extracted and tracked, it is possible to operate visual servoing on the observation satellite and increase the duration of tracking, and thus, obtain more information about the target. For instance, [Mithun et al., 2018] assumes that the space debris is often tumbling and describes an elliptical motion in the image frame, they consequently design their control law with elliptical features in order to improve the control behavior. Recent research presented in [M. Wang et al., 2022] proposes a PD controller to track a space target using only a projected point as a visual feature with some optimal hypotheses, such that the image processing and the target recognition part are not considered, and thus with a high camera frequency and image refresh.

These various applications highlight the different possibilities of visual servoing in space-related applications. A particular focus is now addressed on the specific Earth observation satellite scenario, which is the core of the present thesis.

2.3.2.2 Earth observation satellite

Finally, we redirect our attention towards visual servoing for Earth observation, Earth-pointing satellites, where only the visual control of the satellite orientation is considered, since the satellite's translation is induced by its orbit. As previously evoked, the visual servoing, in this

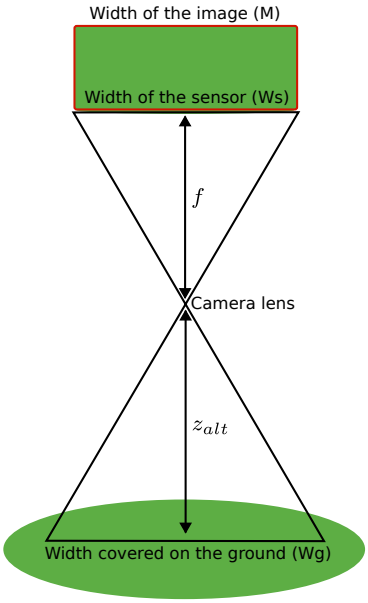


Figure 2.11 – Computation of the GSD: Ground Sampling Distance

case, will focus on improving the ground-target tracking of a motionless location on the Earth, through real-time attitude adjustment. We introduce a crucial parameter for imagery satellites: the ground sampling distance (GSD) expressed in meters/pixel. It basically refers to the rate that converts an image resolution to the corresponding spatial resolution. The higher the GSD value of the image, the lower the spatial resolution of the image. It is computed as (see Figure 2.11):

$$\text{GSD} = \frac{z_{alt} W_s}{f M} = \frac{z_{alt} m_x}{f} = \frac{z_{alt}}{p_x} \tag{2.22}$$

with z_{alt} the attitude of the satellite, W_s the width of the sensor, f the focal lens of the camera, M the width of the image, m_x the width of a pixel in the sensor in meter and $p_x = \frac{f}{m_x}$. Why this parameter is so important in visual servoing for observation satellites? First, because the GSD defines the accuracy of the satellite, as respecting a 1-pixel accuracy for a GSD of 10m/px is not comparable with the same accuracy but with a GSD of 1m/px. Secondly, because the GSD is directly responsible of stronger influences of 3D motions in the image. For instance, let us virtually consider that a particular image point is moving at 7.5 km/s, i.e., the average orbital speed of an LEO satellite. In 1 second, with a GSD of 20 m/px (which corresponds to some of the highest GSDs for Earth observation satellites), this point travels 375 px in the image if there are no compensations of the orbital motion with attitude adjustment, but with a GSD of 1 m/px (which corresponds to one of the lowest GSD), this point now travels 7500 px in 1 second

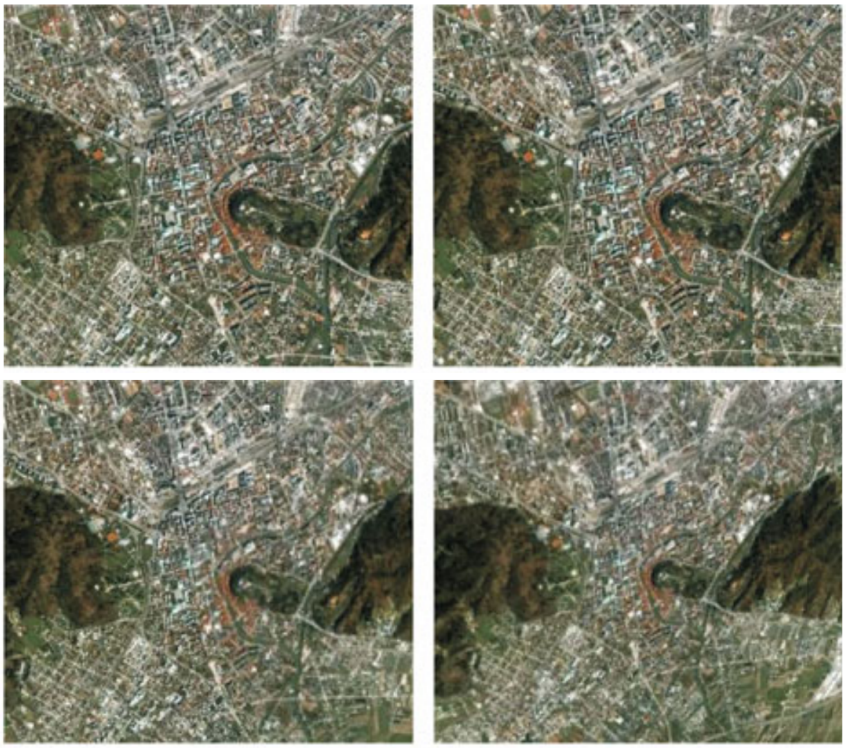


Figure 2.12 – Sequence of images acquired during satellite motion from [Klančar et al., 2012] at every 25 seconds. The reference zone is tracked with low positional error. A shrinkage of the reference zone is observed because the satellite moves away from the zenith.

(again, without any compensations). Indeed, for a fixed image resolution, a lower GSD induces more influences of 3D motions such as the aforementionned orbital speed of the satellite, but also the translation induced by the Earth's rotation, and eventually the own motion of the target. Thus, a lower GSD also induces a more challenging situation for a vision-based control scheme.

In [Klančar et al., 2012], the authors proposed an image-based control scheme for the attitude control of observation satellites, similar to an IBVS but with different formulations and control laws, under the following hypotheses:

— The GSD is of 15 m/px for an image resolution of 320x280 px^2 with a camera frequency of 10Hz, embedded on a satellite in a 600 km orbit;
— The Orbital Reference Frame (ORF) of the satellite is modified such as its z-axis point the desired tracking zone on the Earth, now called the reference zone;
— The attitude angles are expressed wrt. the ORF, so the present approach aims to regulate to 0 these angles to always follow the desired position on the Earth's surface.

The reference zone is selected during a flyby of the Earth, and a reference image is acquired

(see Figure 2.12). For each next image obtained by the camera, a SIFT algorithm is operated to match a specific number of interest points between the current and the reference images, from which a visual error vector composed of three visual features is computed: the global horizontal and vertical image position errors, and the global orientation error. This vector is of size 3 and allows controlling the three attitude angles of the satellite through a PD controller. When the reference zone is selected and the control begins, the initial tracking error corresponds to the drift in the next image, due to the Earth's rotation and the satellite's orbital motion, which is then fully regulated and maintained to 0.

A similar recent approach is presented in [Dauner et al., 2023], the authors use the same expression for visual features, which are extracted and followed from interest points selected by a SURF algorithm and tracked by a KLT tracker. The main difference is that the error vector is directly expressed in terms of orientation computed from the visual features, which classifies this approach into a 3-DOF PBVS. The GSD is not specified by the authors, but given the altitude of the satellite and the field of view of the camera, it is expected to be in the same range as [Klančar et al., 2012]. The same conclusion as before can be made, as the motion induced by the kinematics of the system creates an initial attitude error which is then compensated by the visual control law, to always point to the reference zone. A distributed control law is also studied to control the attitude of a follower satellite through the transfer of visual features from the leader satellite.

These two approaches show that visual servoing can be efficiently used to control Earth observation satellites to lock a specific fixed location on the Earth's surface. The impact of all kinematics motion (mainly the orbital speed of the satellite) initially generates a noticeable, but weak, tracking error due to the high GSD, easily compensated by the control without specifying a stronger target tracking strategy, through image motion estimation. Finally, in terms of control, these approaches are more focused on establishing visually the attitude error of the satellite, and regulating it through direct moment inputs.

2.4 Objectives

Now that we presented the different approaches to control a LEO satellite using vision-based control, and especially the case of Earth observation satellites, we recall the context of the thesis. First, let us recall the two acquisition principles used for Earth observation satellites for imagery or optical missions.

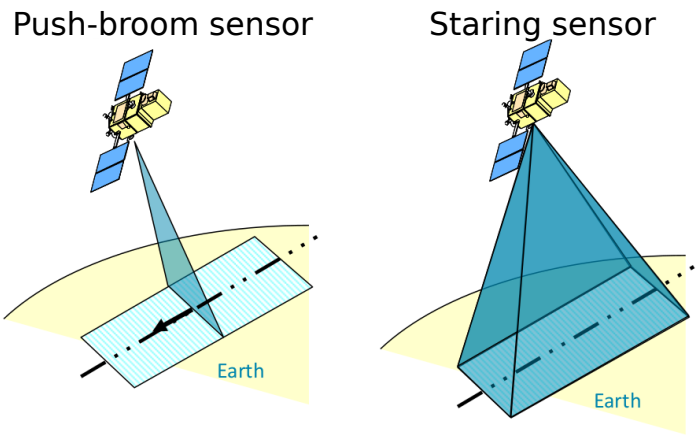


Figure 2.13 – Push-broom and staring sensors.

Push-broom and Staring sensors

The two main sensors used for Earth observation missions are the push-broom and the staring sensor (Figure 2.13). They have the following specifications:

- The push-broom sensor: also called line scanners, it is a technology that uses a sensor line aligned along the track of the satellite that continuously scans the surface of the Earth, acquiring an image swath with a high spatial resolution which is then reconstructed in several snapshots of the covered area. It is better used for high-resolution imaging.
- The staring sensor: consists in a two-dimensional array of pixels that is comparable to a digital camera. Unlike the push-broom, the staring sensor does not continuously scan the Earth's surface, but acquires directly an entire image of a specific geographic location with regularly spaced intervals. It is better used for multispectral imaging that captures data on multiple bands, for Earth's surface monitoring, thus considering large areas and lower resolutions.

As already said in the introduction, this work belongs to the LICHIE project. It aims to launch the next Airbus constellation LION (see Figure 2.14), composed of small agile satellites in LEO equipped with a "Starer" acquisition principle, a new technology comparable to a staring sensor but with enhanced features. The sensor would allow a video mode acquiring images in real-time with a frequency of up to 5 Hz, and most importantly with a resolution of 50 cm, which is comparable to the best resolutions obtained with push-broom sensors. Airbus plans to integrate this new technology so that a "Starer" satellite could be capable of adjusting its attitude in real-

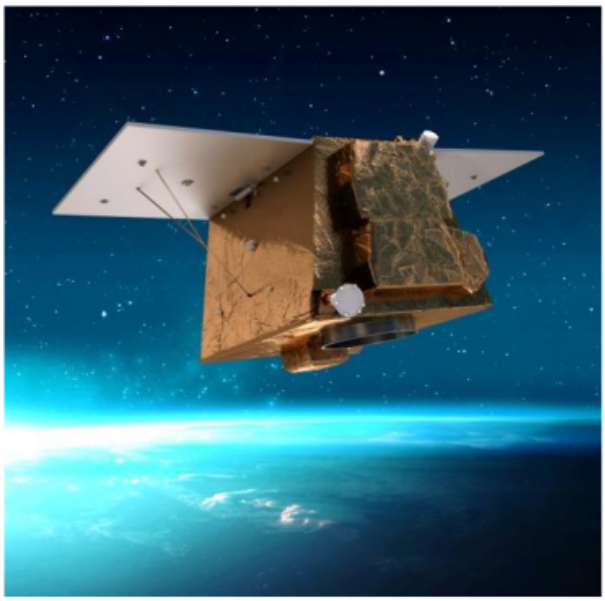


Figure 2.14 – Artistic representation of a LION satellite.

time using the camera as an input sensor for feedback control using images of high resolutions. This constellation aims to complete the current Airbus constellation Pléiades HR [Gleyzes et al., 2012], and Pléiades Néo [Soubirane, 2019], whose image acquisition systems use push-broom technology to cover both defense and civilian observation needs. LION aims to ensure the continuity and improve current systems with the asset of real-time reorientation that might have relevant interest for ground targeting (e.g. to track in real-time a moving target, or to improve the 3D reconstruction of Earth's surface using satellite images [Bernard et al., 2012]...). For our work, we will consider the following specifications:

— Kinematics: the satellite moves in a circular sunsynchroneous orbit of 500 km, i.e., its known orbital parameters are $e = 0$, $z_{alt} = 500km$, $i = 98°$, $\omega = 0°$ and Ω is arbitrarily selected when the satellite is launched.

— Dynamics: the satellite is first supposed to freely rotate on its 3 degrees of freedom (the low-level controller is an integrator), but the dynamics will be refined afterward. The satellite has also velocity and acceleration constraints:
Velocity: $\boldsymbol{\omega}_{max} = (3, 3, 1.2)$deg/s;
Acceleration $\boldsymbol{\gamma}_{max} = (0.6, 0.6, 0.25)$deg/s².

It is equipped with a camera fixed wrt. the satellite, with the following characteristics:

— The frequency of the camera is fixed at 5 Hz.

— The GSD is 50 cm, thus particularly of high resolution (see Figure 2.15).

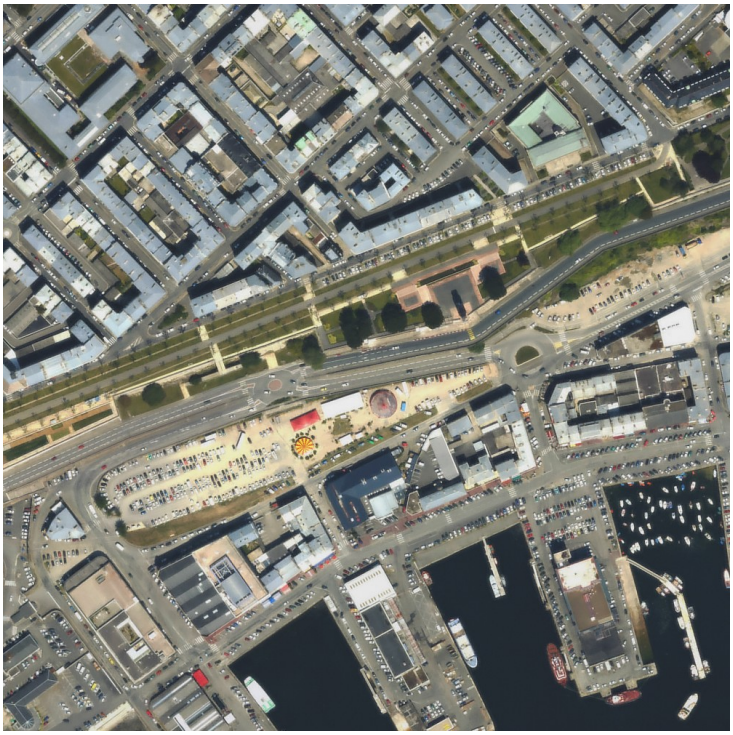


Figure 2.15 – Typical acquired images, with a GSD of 50 cm.

Finally, the objectives of the satellite are to:

- propose a fast mode, i.e., a redefinition of the targetted location, and a fast reorientation to this location.
- lock a specific location on the Earth's surface, defined by visual information.
- track terrestrial moving targets, from slow motions (cars, boats) to high-speed objects (planes).
- respect its mechanical and dynamical constraints.
- acquire an image of high quality in any selected mode, as it is the main mission of the satellite.

Regarding these different specifications and requirements, the literature proposes some approaches to address the locking mode satisfactorily ([Klančar et al., 2012], and recently [Dauner et al., 2023]) but with a very important GSD corresponding to the pointing accuracy of the satellite that could be improved with a smaller GSD. Furthermore, when considering an important GSD, orbital motions of the satellite and translational motions induced by the Earth's rotation are much less significant than they are with a small GSD, and need to be precisely considered. The other points have not yet been effectively addressed in Earth observation satellites, the approaches discussed previously do not consider redefining the target location, and also do not

consider that the target can move on the Earth's surface. We will propose different visual servoing control laws to address all the stated issues. We will consider features directly linked to the target that will be defined more locally (a road, a building, a vehicle, etc.) rather than by an entire city, granting a higher level of tracking accuracy and an important flexibility for the selection of the target. Consequently, we will pay special attention to compensate for all the external motions of the system directly in the control law and focus on proposing a smooth trajectory of the target in the image. We will also propose strict respect for the mechanical constraints of the satellite, also considered in the control input design. Finally, we will show that we can address challenging scenarios and undergo deeper studies, such as guaranteeing image quality while operating the control.

2.5 Conclusion

In this chapter, Low Earth Orbit satellites have been introduced. First, satellite and terrestrial object trajectories have been expressed thanks to orbital mechanics under some adequate hypotheses. Then, the control of the satellite attitude has been investigated, starting with some definitions and classifications. The Attitude Control Subsystem (ACS), responsible for the actuation of the satellite, was detailed and will be considered in the following as the low-level controller of the satellite. Vision-based control has finally been addressed as the guidance of the system, and several approaches using visual servoing to operate on-orbit tasks, including Earth-imaging satellites, have been considered. Finally, we recalled the context of our study and the main objectives we aim to tackle in this thesis.

In the next chapter, we delve into the core of our work by designing a visual servoing for an observation satellite.

CHAPTER 3

VISUAL SERVOING OF AN AGILE OBSERVATION SATELLITE

3.1 Introduction

In the previous chapter, we studied the different approaches to control a low-orbit satellite, including vision-based control schemes. Actually, one of the main differences between the aforementioned studies and our application is that they either do not have to deal with large velocity differences between the target and the sensor when the satellite and its target are moving in almost the same orbit, or they are considering a higher GSD inducing a lower influence of these 3D motions in the image frame. Indeed, in our context, the orbital velocity of the satellite and the velocity induced by the Earth's rotation are the main issues for performing an accurate pointing of a precise location on the Earth's surface. Furthermore, if the target is a mobile vehicle, its motion cannot be left behind. In Chapter 1, we saw that there exist many relevant techniques to achieve precise control tasks while handling perturbations induced by external motions from a moving object or from the system dynamics, and we propose to use these techniques to design our controller.

This chapter is thus focused on establishing a visual control law to precisely control a LION satellite attitude using images provided by the starer sensor. The goal is to perform acquisition missions devoted to precisely focus on an object of interest, motionless or not, and defined with visual information. The sensor is fixed with respect to the satellite, which is supposed to travel on a sun-synchronous circular orbit of 500 km. We have full control over the three rotational degrees of freedom of the satellite subject to velocity and acceleration constraints, while the satellite is moving in an orbit that only influences its position (that is not controlled by our control law). We will show that this precise control of the satellite can be achieved using a closed-loop visual servoing (VS) control scheme.

We will start by modeling the visual servoing system and designing a velocity controller based on this model. Then, we will evaluate its feasibility and stability. This controller is de-

signed by first discussing the selection of adequate visual features to control the 3 rotational degrees of freedom of the satellite and operate an image positioning task. We then introduce the orbital velocity compensations directly in the control law using previous celestial models. For the perturbations induced by the motion of a terrestrial object, we propose to separate its velocity by the known part induced by the Earth's rotation and its own residual and supposedly unknown motion. We then present several methods to improve our control law without changing its physical nature and allowing us to respect the mechanical constraints of the satellite. We validate this control law by designing a satellite simulator based on projections of real images. We then improve our model by adding the internal dynamics of the satellite and propose a predictive saturation algorithm to respect the same aforementioned constraints while minimizing the alteration of the servoing behavior. Finally, we perform additional simulations and validate our approach on our robotic platform.

3.2 Design of the control law

In this section, our objective is to design a visual servoing control scheme to precisely orient a satellite to focus on a terrestrial object of interest.

3.2.1 System modeling

Traveling between astronautic and robotic notation is a tricky task. This is why we introduce the model of our system in Figure 3.1, and consider only these frames for the rest of the thesis. In this representation, the world frame F_w is equivalent to the Earth-Centered Inertial frame, i.e., its origin is in the center of Earth and it does not rotate with the Earth. The object frame F_o is located at an arbitrary location on the Earth's surface and represents the target we intend to track, its orientation is supposed to be fixed, but its translations are altered by the Earth's rotation and by its eventual own motion. The frame F_s represents the inertial frame of the satellite, its origin is in the center of mass (COM) of the satellite, its orientation is fixed with respect to the time when the satellite is launched but its translations are induced by its orbit. Finally, the camera frame F_c, fixed to the camera with its origin at the optical center and its z_c-axis aligned with the optical axis, is also supposed to represent the body frame of the satellite. In that case, and for the sake of clarity, we will consider the optical center of the camera to coincide with the center of mass of the satellite. Thus, this frame is fixed in translations with respect to the satellite inertial frame, but its orientation can be modified by controllers.

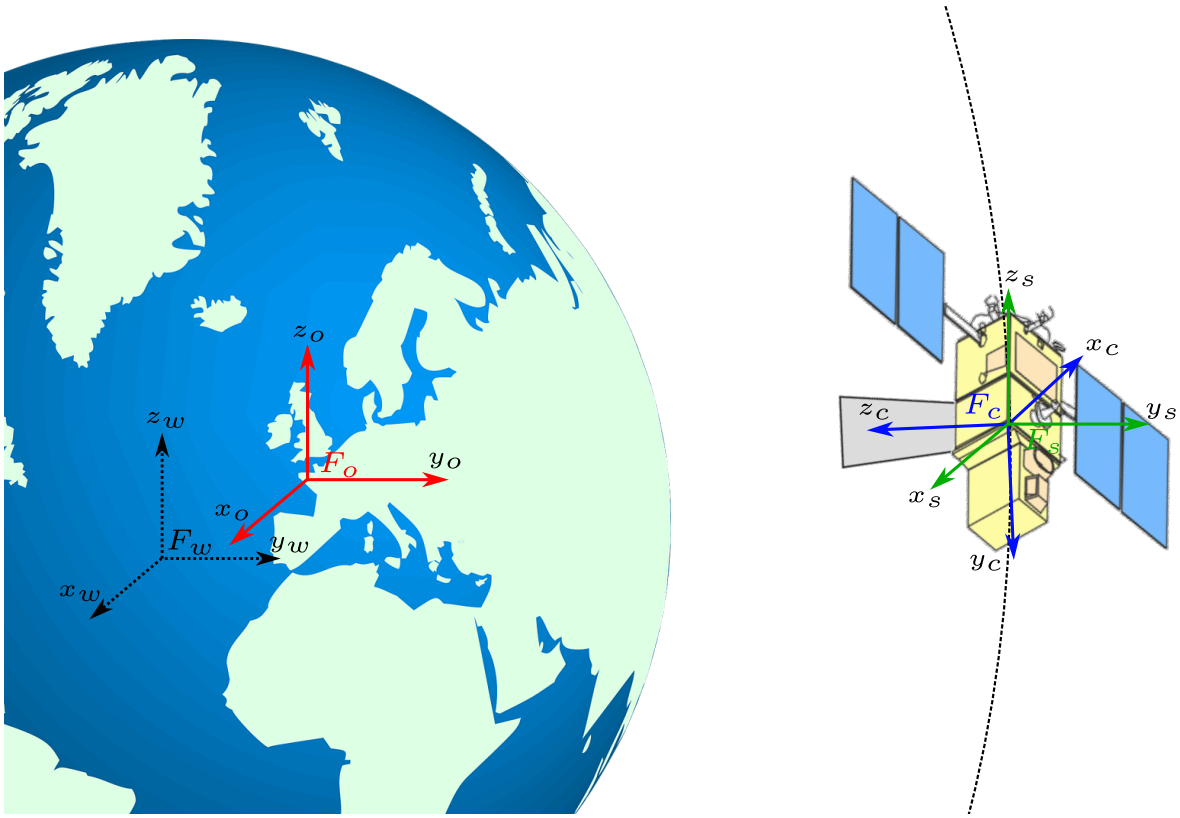


Figure 3.1 – Modeling of the system: a camera embedded on a satellite F_c is directed by the uncontrolled orbital translations induced by its orbit at the center of mass of the satellite F_s, and controlled in rotations to track a terrestrial object of interest F_o. The world frame F_w corresponds to the Earth-Centered Inertial frame of reference.

3.2.2 Visual servoing in space environment

We propose to control the satellite orientation with visual servoing. According to [Chaumette & Hutchinson, 2008], and what we recalled in Chapter 1, the goal of a visual servoing control scheme is to regulate to 0 an error function $\mathbf{e}$ that can be defined by:

$$\mathbf{e}(t) = \mathbf{s}(t) - \mathbf{s}^* \tag{3.1}$$

where $\mathbf{s}(t)$ represents a set of k visual features derived from image measurements, and $\mathbf{s}^*$ is their desired value, considered constant in our case. In our application, the camera is embedded in an agile satellite whose full orientation can be controlled. We propose to design a velocity controller to control the 3 rotational velocity components. The correct theoretical model for

such a situation is [Chaumette & Hutchinson, 2008]:

$$\dot{\mathbf{e}} = \mathbf{L}_{\boldsymbol{\omega}_c} \boldsymbol{\omega}_c + \frac{\partial \mathbf{e}}{\partial t} \tag{3.2}$$

where $\mathbf{L}_{\boldsymbol{\omega}_c}$ is the $k \times 3$ interaction matrix that links the features time-derivative to the rotational velocity $\boldsymbol{\omega}_c$. The term $\frac{\partial \mathbf{e}}{\partial t}$ represents the feature variation due to the other motions. In our case, first, the satellite is moving on a circular trajectory around the Earth called orbit, with a significant orbital speed far superior to any terrestrial or aerial vehicle. Second, a terrestrial target, or object of interest, has a motion induced by Earth's rotation. Finally, the target may have its own velocity in case it is a mobile vehicle. Therefore, $\frac{\partial \mathbf{e}}{\partial t}$ is expressed by:

$$\frac{\partial \mathbf{e}}{\partial t} = \mathbf{L}_s \, {}^c\boldsymbol{\upsilon}_s - \mathbf{L}_o \, {}^c\boldsymbol{\upsilon}_{o_E} + \frac{\partial \mathbf{e}_t}{\partial t} \tag{3.3}$$

where

— ${}^c\boldsymbol{\upsilon}_s$ is the satellite translational velocity (order of magnitude 7 km/s) expressed in camera frame F_c induced by orbital motion, and $\mathbf{L}_s$ its $k \times 3$ interaction matrix,

— ${}^c\boldsymbol{\upsilon}_{o_E}$ is the target translational velocity expressed in camera frame F_c induced by Earth's rotation (order of magnitude 0.4 km/s), and $\mathbf{L}_o$ its $k \times 3$ interaction matrix,

— $\frac{\partial \mathbf{e}_t}{\partial t}$ is an additional term induced by target -generally unknown- own motion (order of magnitude less than 0.3 km/s for terrestrial targets).

From equation (3.2), a velocity controller is obtained by specifying a desired evolution of the error $\mathbf{e}$. In our case, we will consider the classic exponential decay of the error function, such as:

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{3.4}$$

with $\lambda > 0$. Then, taking the Moore-Penrose pseudo-inverse of the interaction matrix $\mathbf{L}_{\boldsymbol{\omega}_c}$ and isolating $\boldsymbol{\omega}_c$ in equation (3.2) provide the following control law:

$$\boldsymbol{\omega}_c = -\lambda \widehat{\mathbf{L}^+_{\boldsymbol{\omega}_c}} \mathbf{e} - \widehat{\mathbf{L}^+_{\boldsymbol{\omega}_c}} \widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{3.5}$$

where $\widehat{\mathbf{L}}$ represents an estimation of the true expression of $\mathbf{L}$ and $\mathbf{L}^+$ the pseudo-inverse of $\mathbf{L}$. We now question the stability of such control in space environment. For that, we consider the closed-loop behavior of the system by injecting equation (3.5) in equation (3.2):

$$\dot{\mathbf{e}} = -\lambda \mathbf{L}_{\boldsymbol{\omega}_c} \widehat{\mathbf{L}^+_{\boldsymbol{\omega}_c}} \mathbf{e} - \mathbf{L}_{\boldsymbol{\omega}_c} \widehat{\mathbf{L}^+_{\boldsymbol{\omega}_c}} \widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t} \tag{3.6}$$

To validate the stability of our system, we will use Lyapunov analysis. For that, we consider the candidate Lyapunov function $\mathcal{L}(\mathbf{e}) = \frac{1}{2}||\mathbf{e}||^2$ which is the squared error norm of the error function. To obtain the global asymptotic stability of the system, we need to prove that the derivative $\dot{\mathcal{L}}$ of the candidate Lyapunov function $\mathcal{L}(\mathbf{e})$ is negative definite, i.e., $\dot{\mathcal{L}} < 0$. $\dot{\mathcal{L}}$ is given by:

$$\dot{\mathcal{L}} = \mathbf{e}^\top \dot{\mathbf{e}} \tag{3.7}$$

Substituting $\dot{\mathbf{e}}$ with its expression when considering the closed-loop behavior of the system in equation (3.6), we obtain:

$$\dot{\mathcal{L}} = -\lambda \mathbf{e}^\top \mathbf{L}_{\omega_c} \widehat{\mathbf{L}^+_{\omega_c}} \mathbf{e} + \mathbf{e}^\top \mathbf{L}_{\omega_c} \widehat{\mathbf{L}^+_{\omega_c}} \widehat{\frac{\partial \mathbf{e}}{\partial t}} - \mathbf{e}^\top \frac{\partial \mathbf{e}}{\partial t} \tag{3.8}$$

The control scheme is thus globally asymptotically stable in the sense of Lyapunov if:

1. $\widehat{\mathbf{L}^+_{\omega_c}}$ is such that $\mathbf{L}_{\omega_c} \widehat{\mathbf{L}^+_{\omega_c}} > 0$.
2. $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ is well estimated.

This analysis shows that the stability of the visual servoing control scheme is first based on having a sufficiently good estimation of the interaction matrix $\mathbf{L}_{\omega_c}$ and, therefore, of its pseudo-inverse $\mathbf{L}^+_{\omega_c}$. In practice and for our application, we will see that it is possible to perfectly estimate $\widehat{\mathbf{L}^+_{\omega_c}}$ so that $\mathbf{L}_{\omega_c} \widehat{\mathbf{L}^+_{\omega_c}} = \mathbf{I}_k$, which of course ensures $\mathbf{L}_{\omega_c} \widehat{\mathbf{L}^+_{\omega_c}} > 0$ and allows to be as near as possible of the expected behavior $\dot{\mathbf{e}} = -\lambda \mathbf{e}$. Then, the estimation $\widehat{\frac{\partial \mathbf{e}}{\partial t}}$ must be accurate enough so that it compensates for all the perturbations induced by external motions and allows one to cancel any tracking error. In our case, we will see that this term can be correctly estimated using a combination of analytical and experimental methods.

In the following, we will discuss the visual features $\mathbf{s}$ that we have chosen and the corresponding interaction matrices. Then we will present how the translational velocities in equation (3.6) are estimated using orbital mechanics, and how to compensate for the residual motion induced by the target's own motion.

3.2.3 Image-Based Visual Servoing

We can now start to select our visual features. The first important question is to state what type of feature family can help us specify an angular positioning task of the satellite according to our primary goal: acquire high-resolution images of an object of interest and track it through these flows of images. In our scenario, an adequate task is thus to focus on this specific target and do not let it exit from the field of view of the camera. Moreover, we assume that the main

interest is in the image trajectory. A solution is to consider a visual feature attached to the target, and in this case, image-based geometric visual features are good candidates. Therefore, we choose an image-based visual servoing control scheme for our work.

We aim to control the 3 rotational degrees of the system, thus, we only need 3 independent visual features. The primary task of the satellite is to focus on an object of interest, which can be defined simply by a position in the image. The first visual features are thus simply the 2D image coordinate $\mathbf{x} = (x, y)$ of a point $\mathbf{X}$ belonging to the target, typically its centroid. The desired values of these features are $\mathbf{x}^* = (x^*, y^*)$, which represents the desired image position of the target. This first set of visual features defines a *focusing* task and a *centering* task if $x^* = y^* = 0$, in which case the target must appear in the center of the image plane of the camera. Note that if $\dot{\mathbf{e}} = -\lambda \mathbf{e}$ is ensured, then the trajectory in the image of point $\mathbf{X}$ is a straight line from its initial position to the desired one. This characteristic usually serves as a simple way to check that interaction matrices and perturbative compensation terms have been accurately estimated.

A secondary task can be designed to control the last degree of freedom of the satellite. In fact, the primary task, which will now be called the focusing task, will mainly involve the *roll* and the *pitch* axes of the satellite through its respective rotational velocity ω_x and ω_y. Thus, the secondary task will mostly involve controlling the *yaw* rotational axis through its velocity component ω_z, which rotates around the optical axis of the camera. An adequate task could be to specify the desired orientation of the image during acquisition. Therefore, a third visual feature α, corresponding to an angle in the image, is selected for a second task called the *orientation* task, but this angle must be defined.

The simplest solution is to arbitrarily consider a second point $\mathbf{X}'$ through its image point $\mathbf{x}' = (x', y')$, and the visual feature α is defined as $\arctan(\frac{y-y'}{x-x'})$, that is, the angle of $[\mathbf{x}, \mathbf{x}']$, the projected segment $[\mathbf{X}, \mathbf{X}']$ on the image plane, with the horizontal axis. A more detailed choice for $\mathbf{X}'$ can also be desired, for example, if we want to acquire images with the sky on the top and the ground on the bottom. For that, we can select a second point $\mathbf{X}'$ above the target, i.e., with a different altitude (the top of a building, of a mountain, of a volcano..), and the corresponding segment $[\mathbf{X}, \mathbf{X}']$ must appear as vertical in the image, so α^* is set as $\frac{\pi}{2}$ so that the desired segment is vertical (see Figure 3.2). But with this configuration, which will be called the *relief* configuration, α is well-defined when the satellite is not at the zenith of the target, i.e., when the length of the projected segment $l = \sqrt{(x-x')^2 + (y-y')^2}$ is not null. In these situations, the orientation task is disabled until α can be computed again, or we may prefer to consider a *plane* configuration when $\mathbf{X}$ and $\mathbf{X}'$ have the same altitude but different longitude and latitude, ensuring l never to be null.

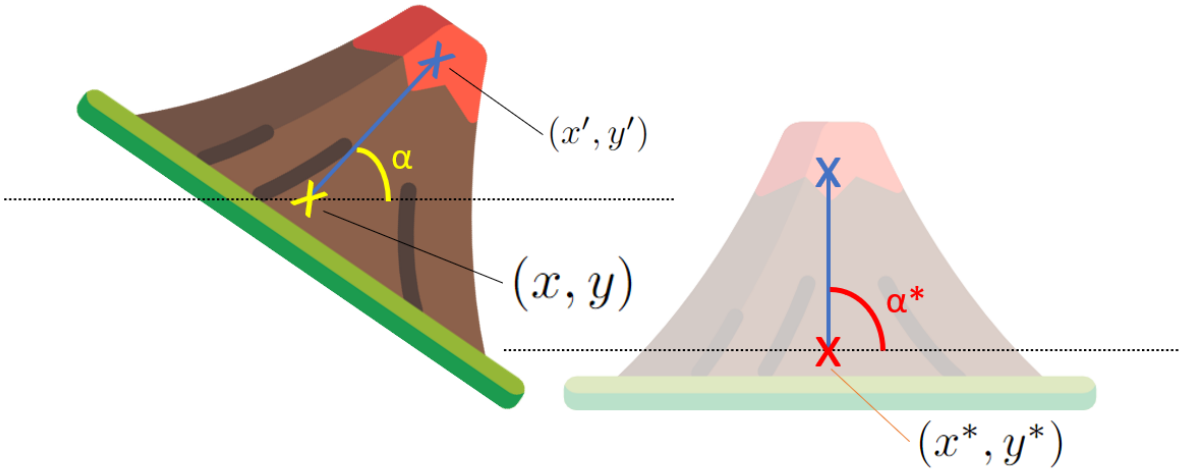


Figure 3.2 – The three visual features considered in our IBVS scheme, the coordinates (x, y) belonging to the target and its desired position (x^*, y^*) defining a focusing task, and the angle α and its associated desired value α^* defining an orientation task.

Now that the visual features are selected, we obtain the visual error $\mathbf{e}$ to be regulated to 0 combining both the focusing and the orientation task:

$$\mathbf{e} = \begin{pmatrix} x - x^* \\ y - y^* \\ \alpha - \alpha^* \end{pmatrix} \tag{3.9}$$

The full interaction matrix $\mathbf{L_e} \in \mathbb{R}^{3\times 6}$ related to this visual error function is the stacking of the interaction matrix of a 2D point $\mathbf{L_x}$ with the interaction matrix of an angle of a projected segment $\mathbf{L}_\alpha$. They are now well-established in the literature [Chaumette et al., 1993]. $\mathbf{L_e}$ is expressed by:

$$\mathbf{L_e} = \begin{pmatrix} \mathbf{L_x} \\ \mathbf{L}_\alpha \end{pmatrix} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -1-x^2 & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \\ -\frac{D}{l}s & \frac{D}{l}c & \frac{D}{l}(xs-yc) & -xs^2+ycs & -yc^2+xcs & -1 \end{pmatrix} \tag{3.10}$$

with $s = \sin\alpha$, $c = \cos\alpha$, and $D = 1/Z' - 1/Z$ where Z and Z' are respectively the depth of $\mathbf{X}$ and $\mathbf{X}'$ in the camera frame. From the complete interaction matrix, we obtain the 3 DOF

interaction matrix in rotation $\mathbf{L}_{\boldsymbol{\omega}_c}$ by taking only the rotational part of $\mathbf{L}_\mathbf{e}$:

$$\mathbf{L}_{\boldsymbol{\omega}_c} = \begin{pmatrix} xy & -1-x^2 & y \\ 1+y^2 & -xy & -x \\ -xs^2+ycs & -yc^2+xcs & -1 \end{pmatrix} \tag{3.11}$$

All the parameters involved in $\mathbf{L}_{\boldsymbol{\omega}_c}$ are directly available as image measurements. The interaction matrix $\mathbf{L}_{\boldsymbol{\omega}_c}$ is always invertible and it is possible to use $\widehat{\mathbf{L}_{\boldsymbol{\omega}_c}} = \mathbf{L}_{\boldsymbol{\omega}_c}$ in equation (3.5), so that $\mathbf{L}_{\boldsymbol{\omega}_c}\widehat{\mathbf{L}^+_{\boldsymbol{\omega}_c}} = \mathbf{L}_{\boldsymbol{\omega}_c}\mathbf{L}^{-1}_{\boldsymbol{\omega}_c} = \mathbf{I}_3$, which induces the first condition for stability to be ensured.

If the third visual feature α is not considered in the control scheme, e.g., when:

— α can not be computed because the satellite is near the zenith of the target,

— the target is a moving object, in this case, the orientation task has no special relevance,

the orientation task is disabled by removing $\alpha - \alpha^*$ from $\mathbf{e}$ and by switching from $\mathbf{L}_{\boldsymbol{\omega}_c}$ to $\mathbf{L}_{\boldsymbol{\omega}_{xy}}$ with:

$$\mathbf{L}_{\boldsymbol{\omega}_{xy}} = \begin{pmatrix} xy & -1-x^2 \\ 1+y^2 & -xy \end{pmatrix} \tag{3.12}$$

so that only the roll and pitch of the satellite are controlled. Once again, all the parameters of the interaction matrix are available, $\widehat{\mathbf{L}_{\boldsymbol{\omega}_c}} = \mathbf{L}_{\boldsymbol{\omega}_{xy}}$ and $\mathbf{L}_{\boldsymbol{\omega}_c}\widehat{\mathbf{L}^+_{\boldsymbol{\omega}_c}} = \mathbf{L}_{\boldsymbol{\omega}_{xy}}\mathbf{L}^{-1}_{\boldsymbol{\omega}_{xy}} = \mathbf{I}_2$, and we fall into the traditional case of a pan-tilt camera observing a point. In fact, it is also possible to control the roll axis that would lead the interaction matrix to be a 2×3 matrix with the last column related to the z axis, but it does not have a special interest, especially when we consider a centering task, we would have the third column null when $x = y = 0$, i.e., when we have converged.

Note that we do not consider exactly the same features as in [Klančar et al., 2012]: in their studies, as they are considering tracking a wider area, the visual feature vector is the global horizontal and vertical image position errors and the global orientation error between a reference image and the current one. The visual feature vector is then automatically computed from the extraction and tracking of several points of interest between both images. In our case, the feature vector is directly linked to the precise location where we want to point the satellite, which we believe provides more flexibility for the selection of the target.

3.2.4 Motion compensation

We have designed our visual features so that the control operates a focusing task and eventually an orientation task, and we have shown that the associated matrices can be directly estimated from image measurements. The last stability condition relies on the estimation of $\frac{\partial \mathbf{e}}{\partial t}$. In

this subsection, we define and compute the external motions induced by the entire system and explain how to compensate for them in equation (3.5).

3.2.4.1 Satellite motion

The studied satellite is a high-speed system whose uncontrolled motion is induced by its orbit. As demonstrated in Chapter 2, its translational position is determined by its orbital parameters, i.e., knowing these parameters allows us to compute an accurate estimation of the satellite translation speed.

Reminder of Chapter 2 : orbital parameters

- $r = R_E + z_{alt}$ with R_T the radius of the Earth and z_{alt} the orbit altitude.
- i is the orbit inclination, $i \in [0, 180]°$;
- Ω is the longitude of the ascending node, it determines where the two intersections of the orbit with the equatorial plane will be located, $\Omega \in [0, 360]°$;
- ω is the argument of periapsis, $\omega \in [0, 360]°$.

We recall our hypothesis that the satellite is supposed to have a Sun-synchronous circular orbit of 500 km. In this case, the argument of periapsis is set to 0, and we can express the translational position of the satellite with respect to the world frame F_w, given in equation (2.14) of Chapter 2:

$$ {}^{w}\mathbf{t}_s = \begin{pmatrix} r\cos\theta(t)\cos\Omega - r\sin\theta(t)\sin\Omega\cos i \\ r\cos\theta(t)\sin\Omega + r\sin\theta(t)\cos\Omega\cos i \\ r\sin\theta(t)\sin i \end{pmatrix} \tag{3.13} $$

where $\theta(t)$ is the true anomaly with respect to time t. For a circular orbit $\theta = \frac{\mu_e^2}{h^3}(t - t_p)$ with h the angular momentum, μ_e the geocentric gravitational constant and t_p the time at the passage of the periapsis. We recall that this parameter is arbitrarily set to attain any initial satellite position on its orbit. Furthermore, the hypotheses given by Airbus D&S fix the orbital parameters $z_{alt} = 500$ km and $i = 98°$, but the parameter Ω is not specified, and so, as t_p, it is arbitrarily set to obtain different orbits that induce a flyby of a different location on Earth.

Let us note that for a given satellite orbit, all these parameters are perfectly known through on-board GPS receivers or Star trackers [Liebe, 1995]. The translational velocity ${}^{w}\boldsymbol{v}_s$ is directly

deduced from equation (3.13), and we obtain:

$$ {}^{w}\boldsymbol{v}_s = \begin{pmatrix} r\dot{\theta}(-\sin\theta(t)\cos\Omega - \cos\theta(t)\sin\Omega\cos i) \\ r\dot{\theta}(-\sin\theta(t)\sin\Omega + \cos\theta(t)\cos\Omega\cos i) \\ r\dot{\theta}\cos\theta(t)\sin i \end{pmatrix} \tag{3.14} $$

This velocity is expressed with respect to the word frame, we need to express it in the camera frame to be injected in equation (3.6). It is classically done with the use of the velocity twist matrix between two frames, expressed in equation (1.37) in Chapter 1, but in our case we only consider the translational part of the velocity, thus, the velocity twist matrix is only the rotation matrix between the two frames. We have:

$$ {}^{c}\boldsymbol{v}_s = {}^{c}\mathbf{R}_w \, {}^{w}\boldsymbol{v}_s \tag{3.15} $$

with ${}^{c}\mathbf{R}_w = {}^{s}\mathbf{R}_c^T \, {}^{w}\mathbf{R}_s^T$ where:

— ${}^{s}\mathbf{R}_c$, the orientation of the camera or satellite body with respect to the satellite inertial frame, which is controlled by visual servoing, so this matrix is perfectly known.

— ${}^{w}\mathbf{R}_s$ is the orientation of the satellite inertial frame with respect to the world frame F_w, fixed with respect to time and arbitrarily set. We chose to define this matrix with orbital parameters, it can be simply set to ${}^{w}\mathbf{R}_s = \mathbf{R}_z(\Omega)\mathbf{R}_x(i)$ defined in equation (2.12) of Chapter 2 so that the plane (x_s, y_s) corresponds to the perifocal plane where the orbital trajectory of the satellite is contained.

Then, the interaction matrix $\mathbf{L}_s$ is the 3 dof interaction matrix for translations considering the three selected visual features, so it is the translational part of $\mathbf{L_e}$ expressed in equation (3.10). It is given by:

$$ \mathbf{L}_s = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} \\ 0 & -\frac{1}{Z} & \frac{y}{Z} \\ -\frac{D}{l}s & \frac{D}{l}c & \frac{D}{l}(xs - yc) \end{pmatrix} \tag{3.16} $$

In case we remove the visual feature α, we obtain $\mathbf{L}_s = \mathbf{L}_{\boldsymbol{v}}$ the translational part of the interaction matrix $\mathbf{L_x}$ expressed by:

$$ \mathbf{L}_{\boldsymbol{v}} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} \\ 0 & -\frac{1}{Z} & \frac{y}{Z} \end{pmatrix} \tag{3.17} $$

These matrices include 3D parameters Z and eventually Z' through D, which have to be estimated as they are not obtained from image measurements. An easy choice is to consider Z fixed with $Z = z_{alt}$ (which is true when the satellite is at the zenith of the target and the target at sea

level) and Z' selected according to the configuration, but we will see in Section 3.2.5.3 that it is possible to have a more accurate estimate of depth considering projective simulations.

3.2.4.2 Earth rotational motion

The target is considered a terrestrial object visible in a scene of interest, and represented by the object frame F_o. The position ${}^w\mathbf{t}_o$ can be determined by spherical coordinates, as long as we hypothesize that the Earth is spherical and not an actual geoid. Its translational motion has been given in equation (2.18) of Chapter 2, and it is expressed by:

$$ {}^w\mathbf{t}_o = \begin{pmatrix} R_E \cos\delta_l \cos(\alpha_l + \Gamma(t)) \\ R_E \cos\delta_l \sin(\alpha_l + \Gamma(t)) \\ R_E \sin\delta_l \end{pmatrix} \tag{3.18} $$

where δ_l is the latitude of the origin of the frame, α_l its longitude and $\Gamma(t)$ is the Greenwich Sidereal Angle, we recall its expression from (2.19):

$$ \Gamma(t) = \omega_E t + \Gamma_0 \tag{3.19} $$

with $\omega_E = 7.29217 \times 10^{-}5$ rad/s Earth's rotation speed, and Γ_0 the Greenwich angle at the origin of time, which is also supposed arbitrarily set to attain any initial configuration of an Earth object. The translational velocity induced by Earth's rotation is obtained by the time-derivative of equation (3.18) considering equation (3.19):

$$ {}^w\boldsymbol{v}_{o_E} = \begin{pmatrix} -\omega_E R_E \cos\delta_l \sin(\alpha_l + \Gamma(t)) \\ \omega_E R_E \cos\delta_l \cos(\alpha_l + \Gamma(t)) \\ 0 \end{pmatrix} \tag{3.20} $$

Once again, this translational velocity is expressed in the camera frame F_c using:

$$ {}^c\boldsymbol{v}_{o_E} = {}^c\mathbf{R}_w \, {}^w\boldsymbol{v}_{o_E} \tag{3.21} $$

Finally, $\mathbf{L}_o$ has the same definition as $\mathbf{L}_s$, and we have $\mathbf{L}_o = \mathbf{L}_s$.

In practice, the longitude and latitude of the object frame are considered known from the moment it is selected in the scene of interest, we can obtain its 3D parameters through the intersection of the line of sight with the surface of the Earth's and deduce its longitude and latitude.

3.2.4.3 Target own motion

In the case where the target has an unknown motion, the term $\frac{\partial \mathbf{e}_t}{\partial t}$, unlike previous velocities, can not be determined analytically. This term can be estimated by various approaches, as recalled in Chapter 1, but if the target moves with a constant speed, a simple solution to compensate for that motion is to consider an visual integrator in the control law, which will eventually eliminate the drag error induced by a motion with constant velocity. So, re-calling equation (3.3), $\frac{\partial \mathbf{e}_t}{\partial t}$ will be compensated by:

$$\widehat{\frac{\partial \mathbf{e}_t}{\partial t}} = \mu \sum_{j=0}^{k-1} \mathbf{e}(j) \tag{3.22}$$

where μ is the integral gain. It is generally fixed and tuned according to the amplitude of the tracking error. In the following Section 3.2.5.1, we propose a variation that we think is more adapted to the situation. Of course, we could also have considered a Kalman filter to accomplish this task. It will be used in Chapter 4.

3.2.5 Improving control behavior

In this subsection, we present some control tools to obtain what we believe to be an adequate behavior of our visual servoing scheme with respect to the task to be achieved by the satellite.

3.2.5.1 Adaptive gain

The gain λ associated with the exponential decrease of the error $\mathbf{e}$ is usually constant. However, λ can be modulated to decrease high velocities when visual error is high and increase convergence speed when it is low [Kermorgant & Chaumette, 2013]. An adaptive gain is then introduced, λ becoming error-dependent:

$$\lambda(||\mathbf{e}||) = (\lambda_0 - \lambda_\infty) e^{-\frac{\lambda'_0}{\lambda_0 - \lambda_\infty}||\mathbf{e}||} + \lambda_\infty \tag{3.23}$$

where

— λ_0 is the gain tuned for small values of $||\mathbf{e}||$,
— λ_∞ is the gain for high values of $||\mathbf{e}||$,
— λ'_0 is the slope of λ at $||\mathbf{e}|| = 0$.

Consideration of an adaptive gain increases the convergence speed of the system together with its stability when the error is high. Furthermore, in case that the target has its own motion, the

integrator gain μ in equation (3.22) is tuned in the same way, so that the integrator becomes an adaptive visual integrator, which will have a stronger influence when it comes to reducing only the tracking error induced by a potentially moving target.

3.2.5.2 Velocity constraints

A satellite is a critical system and its rotational axes, commonly called pitch, roll, and yaw as previously said, are limited to a certain maximal speed and acceleration:

— On x and y axes, speed is limited to 3 deg/s and acceleration to 0.6 deg/s²,

— On the z axis, speed is limited to 1.2 deg/s and acceleration to 0.25 deg/s².

These constraints raise the possibility that the z axis can be decoupled from the others, and a different gain can be used on this axis. We define $\mathbf{\Lambda}$ by:

$$\mathbf{\Lambda} = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda_\alpha \end{pmatrix} \tag{3.24}$$

with λ and λ_α adaptive gains following equation (3.23) with λ_α tuned with significantly below parameters than λ to cope with stronger constraints around z axis. Equation (3.5) becomes:

$$\boldsymbol{\omega}_c = -\mathbf{L}^+_{\boldsymbol{\omega}_c} \mathbf{\Lambda} \mathbf{e} - \mathbf{L}^+_{\boldsymbol{\omega}_c} \widehat{\frac{\partial \mathbf{e}}{\partial t}} \tag{3.25}$$

For ensuring $\dot{\mathbf{e}} = -\mathbf{\Lambda}\mathbf{e}$ induced by equation (3.25), if the velocity or the instant acceleration becomes too high, a velocity reduction is operated on all components of $\boldsymbol{\omega}_c$ for not altering as much as possible the nature of the image trajectory, i.e., a straight line. When a constraint is activated, a reduction factor is computed on the axis concerned. Reduction factors $(r^\omega_x, r^\omega_y, r^\omega_z)$ for velocity constraints and $(r^\gamma_x, r^\gamma_y, r^\gamma_z)$ for instant acceleration constraints are computed according to the following rules:

— $\forall i \in [x, y, z]$, if $|\omega_i| > \omega_{max_i}$ then $r^\omega_i = \frac{\omega_{max_i}}{|\omega_i|}$ else $r^\omega_i = 1$,

— $\forall i \in [x, y, z]$, if $|\gamma_i| > \gamma_{max_i}$ where $\gamma_i = \frac{\omega_i(t) - \omega_i(t-dt)}{dt}$ then $r^\gamma_i = \frac{\gamma_{max_i}}{|\gamma_i|}$ else $r^\gamma_i = 1$.

Then, since the z axis has different constraints than the others, and to not penalize the time to convergence of the *focusing* task, we adopt the following strategy:

— raising velocity or instant acceleration constraints on x or y axis induces a velocity reduction on all axes, i.e., $r_x = r_y = r_z$,

— raising constraints on z axis induces a velocity reduction on z axis only, i.e., $r_x = r_y = 1$,

Raising constraints simultaneously on multiple axes induces that $r_x = r_y = min(r_x, r_y)$ and $r_z = min(r_x, r_y, r_z)$. Finally, when a velocity constraint is raised, the rotational velocity $\boldsymbol{\omega}_{sat}$ sent to the low-level controller is given by:

$$\boldsymbol{\omega}_{sat} = \begin{pmatrix} r_{xy}^{\omega} & 0 & 0 \\ 0 & r_{xy}^{\omega} & 0 \\ 0 & 0 & r_{z}^{\omega} \end{pmatrix} \boldsymbol{\omega}_c \tag{3.26}$$

while, for an acceleration constraint:

$$\boldsymbol{\omega}_{sat} = \begin{pmatrix} r_{xy}^{\gamma} & 0 & 0 \\ 0 & r_{xy}^{\gamma} & 0 \\ 0 & 0 & r_{z}^{\gamma} \end{pmatrix} \boldsymbol{\omega}_c + \begin{pmatrix} 1-r_{xy}^{\gamma} & 0 & 0 \\ 0 & 1-r_{xy}^{\gamma} & 0 \\ 0 & 0 & 1-r_{z}^{\gamma} \end{pmatrix} \boldsymbol{\omega}_c(t-dt) \tag{3.27}$$

During a saturation on x or y axis, we have $\boldsymbol{\omega}_{sat} = r\boldsymbol{\omega}_c$ for a velocity constraint and $\boldsymbol{\omega}_{sat} = r\boldsymbol{\omega}_c + (1-r)\boldsymbol{\omega}_c(t-dt)$ for an acceleration constraint. In both cases, the time-variation of the error is given by:

$$\dot{\mathbf{e}} = -r\boldsymbol{\Lambda}\mathbf{e} + \mathbf{K} \tag{3.28}$$

with $\mathbf{K} = -r\widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t}$ for velocity saturation, and $\mathbf{K} = -r\widehat{\frac{\partial \mathbf{e}}{\partial t}} + \frac{\partial \mathbf{e}}{\partial t} + (1-r)\mathbf{L}_{\boldsymbol{\omega}_c}\boldsymbol{\omega}_c(t-dt)$ for acceleration saturation. The term $\mathbf{K}$ in equation (3.28) shows that $\frac{\partial \mathbf{e}}{\partial t}$ is compensated partially by $-r\widehat{\frac{\partial \mathbf{e}}{\partial t}}$, which necessitates the residual external motion $(1-r)\frac{\partial \mathbf{e}}{\partial t}$ to be compensated by the feedback term $\boldsymbol{\Lambda}\mathbf{e}$ and thus reduces the time-to-convergence of the visual tasks. However, this constraint management allows the servoing scheme to keep a straight-line trajectory of the target point in the image during a possible saturation of axis x or y. Indeed, from equation (3.28) and assuming $\mathbf{K}$ is constant during the saturation, we can express $x(t)$ and $y(t)$ components of $\mathbf{e}$ by:

$$\begin{pmatrix} x(t)-x^* \\ y(t)-y^* \end{pmatrix} = \begin{pmatrix} x(0)e^{-r\lambda t} + \frac{K_1}{r\lambda} \\ y(0)e^{-r\lambda t} + \frac{K_2}{r\lambda} \end{pmatrix} \tag{3.29}$$

which leads to:

$$\frac{x(t)}{x(0)} - \frac{y(t)}{y(0)} - \frac{\frac{K_1}{r\lambda} + x^*}{x(0)} + \frac{\frac{K_2}{r\lambda} + y^*}{y(0)} = 0 \tag{3.30}$$

showing that the image trajectory of point $\mathbf{X}$ is a straight line. However, as soon as $K_1 \neq 0$ or $K_2 \neq 0$, which occurs during a saturation, the image point does not converge anymore to its desired position (x^*, y^*) but to $(x^* + \frac{K_1}{r\lambda}, y^* + \frac{K_2}{r\lambda})$. This means that saturation on x or y will make the trajectory deviate to a set of successive segments. That is why such saturation must

not occur during all the servo, and, for that, λ is tuned so that constraint raising happens rarely. Finally, note that the straight-line trajectory is also lost when the z axis is saturated. That is why λ_α must be tuned with the right balance between image trajectory and convergence speed. Note that if saturation of the axis z appears at the end of the focusing task, it will not be seen on the trajectory of point $\mathbf{X}$ once this point is already at its desired position.

3.2.5.3 Computation of Z

As mentioned before, the interaction matrix $\mathbf{L}_s$ related to the translational velocity induced by the orbital motion of the satellite and by the rotation of the Earth requires the depth Z of the target in the camera frame. A rough approximation of Z is possible by setting it constant and equal to the satellite's altitude as previously mentioned, which will not impede the stability of the control but potentially non-straight trajectories in the image and the conservation of a residual tracking error (if no integrator is considered) if the actual Z diverges greatly from its estimation. Another solution is to take advantage of the model, by computing the depth Z.

Indeed, our application is focused on the acquisition part when the object of interest is in the camera field of view. As previously said, it is possible to obtain its longitude and latitude when it is selected, and so we know from equation (3.18) the 3D information of the target with respect to the world frame, denoted ${}^w\mathbf{X}$. The target point can be expressed in the camera frame with homogeneous coordinates $\bar{\mathbf{X}} = (\mathbf{X}, 1)$ such as:

$$ {}^c\bar{\mathbf{X}} = {}^c\mathbf{T}_w \, {}^w\bar{\mathbf{X}} \tag{3.31} $$

where ${}^c\mathbf{T}_w = {}^s\mathbf{T}_c{}^\top \, {}^w\mathbf{T}_s{}^\top$, with:

— ${}^s\mathbf{T}_c = \begin{pmatrix} {}^s\mathbf{R}_c & {}^s\mathbf{t}_c \\ \mathbf{0} & 1 \end{pmatrix}$, with ${}^s\mathbf{R}_c$ the orientation of the camera or satellite body, controlled by visual servoing, and ${}^s\mathbf{t}_c = \mathbf{0}$,

— ${}^w\mathbf{T}_s = \begin{pmatrix} {}^w\mathbf{R}_s & {}^w\mathbf{t}_s \\ \mathbf{0} & 1 \end{pmatrix}$ with ${}^w\mathbf{R}_s = \mathbf{R}_z(\Omega)\mathbf{R}_x(i)$ arbitrarily set, and ${}^w\mathbf{t}_s$ given by equation (3.13).

We obtain ${}^c\mathbf{X}$ by dropping the fourth component. The third component of ${}^c\mathbf{X}$ corresponds to the depth Z of the target in the camera frame. Taking equation (3.31) to compute Z is a much better estimate than a constant depth, as it can be updated for each iteration of the control considering that equation (3.31) is time-varying.

3.2.5.4 Prediction of $\mathbf{L}_s$

An issue appears when considering the terms $\mathbf{L}_s({}^c\boldsymbol{\upsilon}_s - {}^c\boldsymbol{\upsilon}_{o_E})$ which corresponds to the compensation of the variation of the error due to known external motions. In the way we designed our controller, $\mathbf{L}_s$ is computed with the current information of the system, $i.e.$, at an instant k, where ${}^c\boldsymbol{\upsilon}_s - {}^c\boldsymbol{\upsilon}_{o_E}$ corresponds to the translational velocity between the satellite and the scene at an instant $k+1$. Generally in visual servoing, taking $\mathbf{L}_s(k)$ is sufficient to obtain a precise compensation of these motions because we consider a control frequency synchronized with the camera frequency, which is usually of at least 30 Hz, meaning that between two iterations the parameters of the matrix would not change so much. In our case, we recall that the frequency of the camera and of the control loop is fixed to 5 Hz, which is not an issue when considering "normal" external motions inducing small variations of the matrix parameters between two iterations. However, we are not in a "normal" case and the external motions have an order of magnitude of 7 km/s which induces a noticeable variation of the parameters of the matrix during two iterations. In summary, it seems more interesting to estimate the approximate values of these parameters at instant $k+1$ so that the compensation term is as effective as possible. To do so, as we do not know in advance what the next image will be and what value will have the features, we propose to consider an internal simulation loop that will iteratively update the parameters at a higher frequency, based on current 3D motions, to approach their possible value at instant $k+1$ which should be a better estimate than taking their value at instant k. This prediction is obtained with the following sequence for each iteration k:

— Measure the velocity screw $\mathbf{v}(k)$ of the system,

— In an inner loop of a frequency $F_i > F$ update the visual features and the depth thanks to their complete interaction matrices:

$$\mathbf{s}_{i+1} = \frac{1}{F_i}\mathbf{L}_{\mathbf{e}_i}\mathbf{v}(k) + \mathbf{s}_i \qquad \text{where } \mathbf{L}_{\mathbf{e}_i} = \begin{pmatrix} \mathbf{L}_{\mathbf{x}_i} \\ \mathbf{L}_{\alpha_i} \end{pmatrix} \tag{3.32}$$

$$Z_{i+1} = \frac{1}{F_i}\mathbf{L}_{Z_i}\mathbf{v}(k) + Z_i \qquad \text{where } \mathbf{L}_{Z_i} = \begin{pmatrix} 0 & 0 & -1 & -y_i Z_i & x_i y_i Z_i & 0 \end{pmatrix} \tag{3.33}$$

with $\mathbf{L}_{Z_i}$ the depth interaction matrix, and $\mathbf{s}_i$ initialized with $\mathbf{s}(k)$ and the same for all parameters.

— When the inner loop ends, update $\mathbf{L}_s$ taking the translational part of the last $\mathbf{L}_{\mathbf{e}_i}$ computed.

This sequence is also valid to compute a prediction of $\mathbf{L}_{\boldsymbol{v}}$ considering $\mathbf{L}_{\mathbf{x}}$ instead of $\mathbf{L}_{\mathbf{e}}$. We will compute our compensation terms using this technique.

3.2.5.5 Initial pointing

As mentioned above, we assume that the satellite is already directed toward an area of interest that encompasses the target. This initial pointing is considered to be accomplished through an open-loop attitude controller compensating for satellite and Earth motions, and this controller achieves the following angular rates:

$$\boldsymbol{\omega}_c(t) = \mathbf{L}^{+}_{\boldsymbol{\omega}_{xy}}(\mathbf{X}_0)\mathbf{L}_{\boldsymbol{v}}(\mathbf{X}_0)({}^c\boldsymbol{v}_s(t) - {}^c\,\boldsymbol{v}_{o_E}(t)) \tag{3.34}$$

with $\mathbf{L}_{\boldsymbol{\omega}_{xy}}(\mathbf{X}_0)$ and $\mathbf{L}_{\boldsymbol{v}}(\mathbf{X}_0)$ here computed with the parameters taken from the projection of the initial pointed target $\mathbf{X}_0$ (as we consider the visual servoing to not have started yet), i.e., with the parameters $x = {}^cX_0/{}^cZ_0$, $y = {}^cY_0/{}^cZ_0$ and $Z = {}^cZ_0$. Finally, we initialize the very first angular rates of the visual loop with the value $\boldsymbol{\omega}_c(t_0)$ at its initial time t_0.

3.2.5.6 Control law

At the end of the day, the 3 dof IBVS scheme aiming to control an Earth-pointing observation satellite in rotation is given by:

$$\boldsymbol{\omega}_c = -\mathbf{L}^{+}_{\boldsymbol{\omega}_c}(\boldsymbol{\Lambda}\mathbf{e} - \mathbf{L}_s({}^c\boldsymbol{v}_s - {}^c\,\boldsymbol{v}_{o_E}) + \mu \sum_{j=0}^{k-1} \mathbf{e}(j)) \tag{3.35}$$

and proposed in a 2 dof pan-tilt configuration:

$$\boldsymbol{\omega}_c = -\mathbf{L}^{+}_{\boldsymbol{\omega}_{xy}}(\lambda\mathbf{e} - \mathbf{L}_{\boldsymbol{v}}({}^c\boldsymbol{v}_s - {}^c\,\boldsymbol{v}_{o_E}) + \mu \sum_{j=0}^{k-1} \mathbf{e}(j)) \tag{3.36}$$

Note that the visual integrator is only enabled when target tracking is considered. This velocity $\boldsymbol{\omega}_c$ is then sent to the velocity saturation process described previously before being applied by the low-level attitude controller of the system (see Figure 3.3). For now, we consider that the dynamic response of the satellite to the attitude controller is a simple integrator. Additional remarks can be made on equation (3.35) and equation (3.36), if $\mathbf{e} = \mathbf{0}$, the positioning task is completed and we have operated a reorientation to point the target, still, the control law continue to compensate for the external motions so the target is locked in the center of the image.

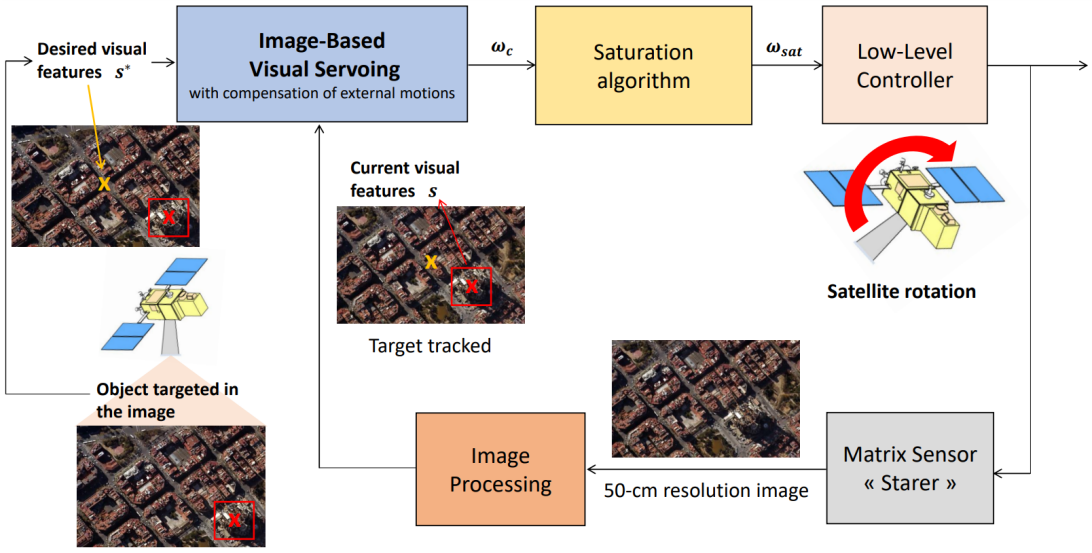


Figure 3.3 – Satellite control loop: an object is targeted in a scene of interest, current and desired visual features are initialized and fed to the IBVS scheme which computes angular rates ω_c, subject to the saturation algorithm, and finally sent to the low-level controller which makes the satellite rotates. The satellite acquires new images, and an image processing algorithm tracks and updates visual features for the control scheme.

Moreover, these laws are designed considering the pseudo-inverse of the interaction matrices, but we could also propose another controller, for example, based on optimization frameworks. This will be discussed in Chapter 4.

3.3 Validation of the control law

In this section, we aim to validate the previous control law. Two approaches will be proposed:

— geometric simulations, we simulate all the 3D motions of the system thanks to the motion laws described in Chapter 2, but we do not consider the image processing part. Visual features are acquired through projections of 3D points in the camera frame, thanks to the geometry that is considered perfectly known.

— image-based simulations, once again all the 3D motions of the system are simulated, and we use these 3D motions to simulate a continuous camera view of real-scaled satellite images, set in their real 3D locations on the Earth's surface, and projected into the camera frame to obtain simulated images.

The control law is implemented using the ViSP framework [É. Marchand et al., 2005].

3.3.1 Geometric simulations

3.3.1.1 Simulated features

A camera attached to a satellite is simulated in free-floating mode around the Earth thanks to its relative pose to the world frame ${}^{w}\mathbf{T}_{c} = {}^{w}\mathbf{T}_{s}{}^{s}\mathbf{T}_{c}$ allowing us to accurately describe a satellite path. A scene of interest is defined from an object frame whose position is determined by ${}^{w}\mathbf{T}_{o}$. A 3D point $\mathbf{X}$ belonging to the object of interest is expressed in its object frame with the coordinates ${}^{o}\mathbf{X}$ supposedly known exclusively in this case where we have no image information, this knowledge replaces the vision sensor. The acquisition is simulated by projecting the point $\mathbf{X}$ into the camera image plane considering a pinhole camera model as presented in Chapter 1, that is:

$$x = {}^{c}X/{}^{c}Z,\ y = {}^{c}Y/{}^{c}Z,\ Z = {}^{c}Z \tag{3.37}$$

with Z the depth of the target in the camera frame perfectly known, as we assume that we know the position ${}^{o}\mathbf{X}$. We consider that the camera acquisition frequency F is synchronized with the control frequency, with $F = 5$ Hz, specified by Airbus D&S. For every configuration (until the end of this thesis in fact), the desired set of visual features are $\mathbf{e}^{*} = (0, 0, 90°)$, corresponding to a centering task so that the target appears vertical in the image.

3.3.1.2 Simulation results

In these first tests, our objective is to validate the control law by verifying that it performs its two tasks, the centering task and the orientation task. We will then look at the influence of saturation on control and study the case of the centering of a mobile object.

A nominal relief configuration is shown in Figure 3.4, a 3D point is positioned on the Earth's surface and a second point defining the orientation task is set 500 m above it. The visual error is perfectly regulated to 0 (Figure 3.4a) which can be observed on the image trajectory (Figure 3.4c) with the first point, the target point, performing the centering task in 4 s with a perfectly straight line trajectory showing that the external motions of the system are accurately compensated. The orientation point is then positioned on the vertical line that crosses the center of the image frame, this task is slower (approximately 40 s) as far as we have decoupled the exponential decay of the z axis from the others and have tuned the completion of this task with lower gain λ_{α}. These tasks are accomplished with the consequent orbital motions shown in Figure 3.5 which induces a predominant influence of the angular velocity needed to compensate for these motions in ω_x and ω_y (Figure 3.4b) before the orientation task ends, and then only

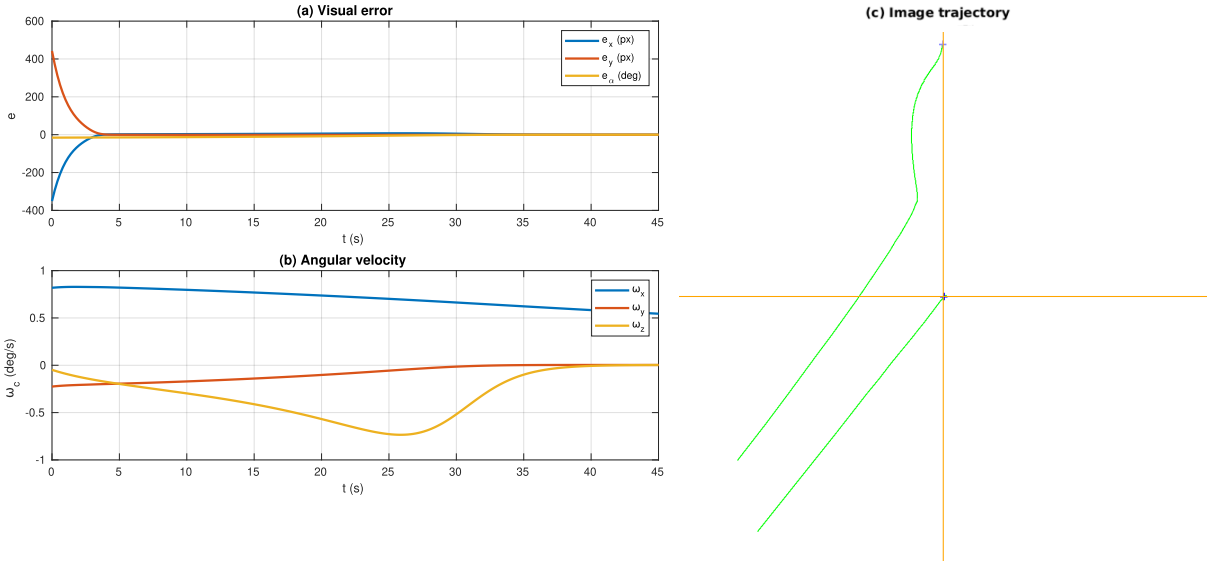


Figure 3.4 – Simulation results of the control law applied in a relief configuration. Gains are tuned so that no saturation occurs during the control.

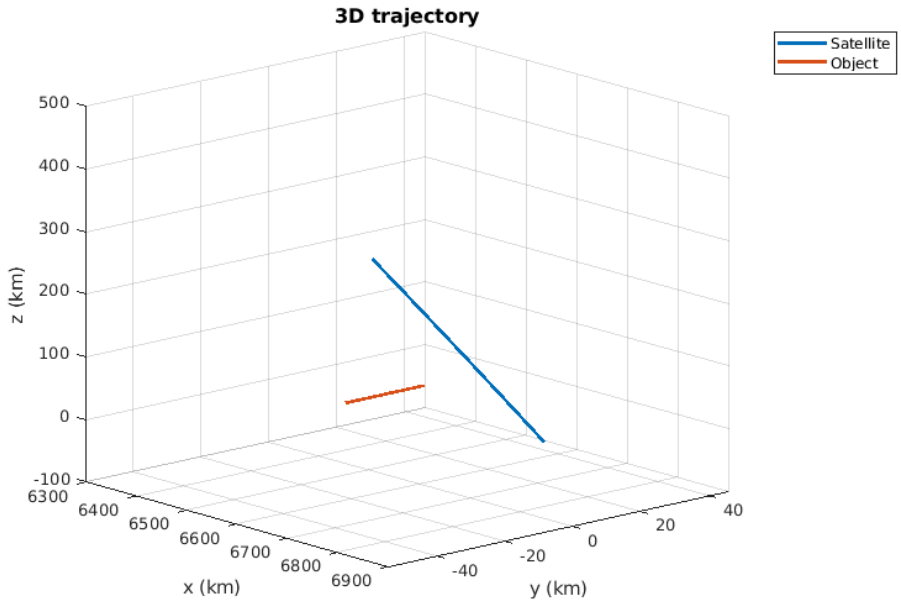


Figure 3.5 – 3D trajectory of the satellite and of the object for Figure 3.4 and Figure 3.6 with respect to the world frame. The satellite travels more than 300 km in 45 s.

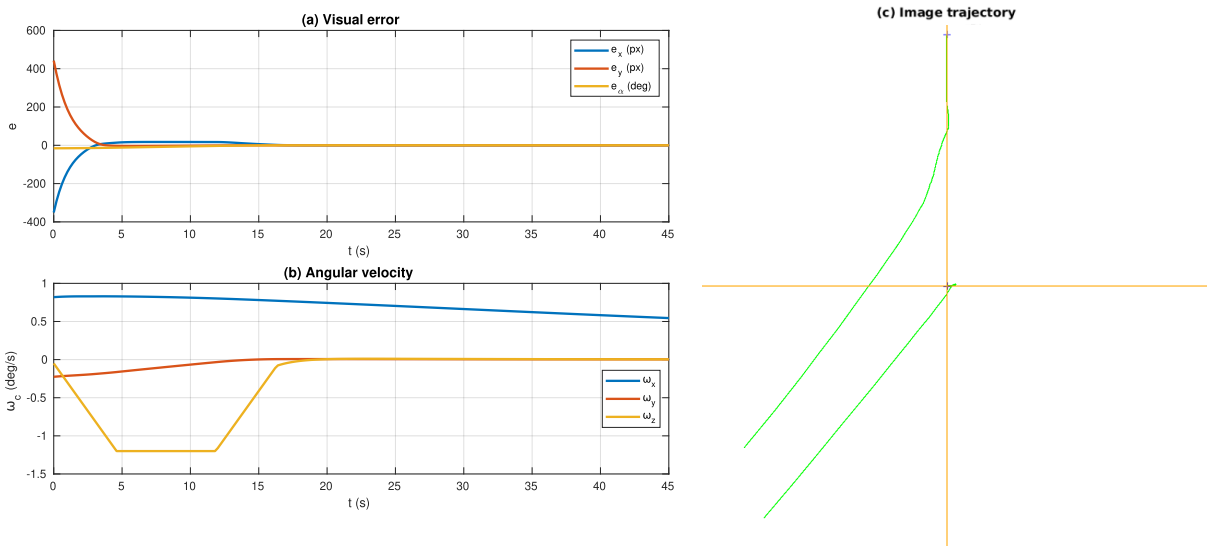


Figure 3.6 – Same configuration as Figure 3.4 but λ_α is increased so that it triggers an acceleration and then a velocity saturation on the z axis. As a result, the point $\mathbf{X}$ is not perfectly centered at first, which is then corrected when the saturations end.

by ω_x (the satellite is now oriented in z so that the orbital motion influences only ω_x in this configuration). ω_z is fully activated by the orientation task, and as it is quite slow in relation to the centering task, we increase the gain λ_α in a second configuration presented in Figure 3.6.

The orientation task is now accomplished faster, in approximately 18 s (Figure 3.6a), but axis z is saturated both in acceleration (linear increase and decrease of ω_z with a slope of γ_{max_z}) at the beginning of the orientation task and at its end, and in velocity from 4.8 s to 12 s (constant velocity at ω_{max_z})(Figure 3.6b), inducing a small deviation of the centering task that is almost not seen in the image trajectory (Figure 3.6c). Thus, it is convenient to adjust λ_α with the right balance between the convergence speed and the camera trajectory. Note that after reaching its desired angle, the orientation segment $[\mathbf{x}, \mathbf{x}']$ increases because the orientation point is rising on the vertical axis due to the change of point of view of the satellite with respect to time, another influence of the orbital trajectory.

A second configuration is proposed in Figure 3.7 where the target is a vehicle moving either at 300 km/h or supposed motionless for the comparison. When dealing with target tracking, the orientation task is disabled, and we consider the control law expressed in equation (3.36). If the target is fixed, the centering task is once again directly completed with a perfect straight-line trajectory (Figure 3.7c) and we can, by the way, observe the behavior of the angular velocity subtracted from the compensations of external motions in Figure 3.7b, which is composed of local decreasing exponential laws due to the adaptive gain. If the target has its own motion (300

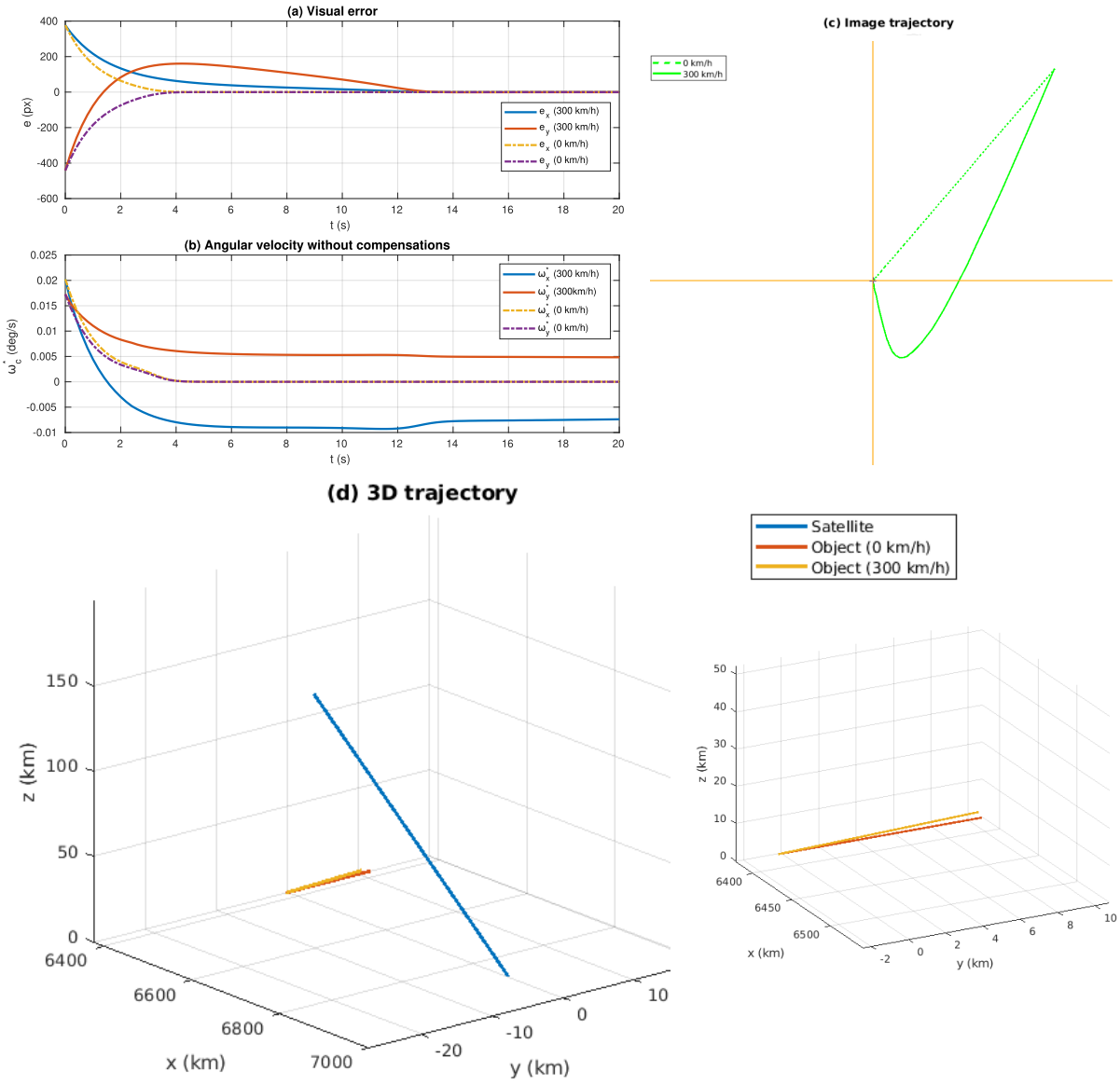


Figure 3.7 – Simulation results in a pan-tilt configuration of the satellite, tracking a vehicle moving at 300 km/h. The trajectory in the dot line represents the image trajectory if the vehicle is fixed. The 3D trajectory on the right shows that the Earth's rotation has more influence on the object's trajectory than its own motion. Angular velocity are shown without the compensations of orbital and Earth's rotational motions.

km/h but supposedly unknown by the control scheme), the trajectory deviates from the center of the image frame and reaches an inflection point corresponding to the convergence point of the control scheme if there was no integrator in the control loop (thus, in approximately 4s, same convergence time as with a fixed target), this residual error is the tracking error. Then, the influence of the adaptive integrator becomes preponderant and smoothly compensates for the rest of the error. Then it is possible to obtain an approximation of the target velocity from the constant threshold obtained in Figure 3.7b, but this will not be expressed in this thesis. An interesting outcome of the adaptive integrator is that it barely alters the velocity induced by the exponential decay of the error and acts similarly as a secondary task, which is why the trajectory of the moving object describes two distinct (almost) straight-line trajectories.

To conclude these first simulations, the control law succeeds in performing an accurate positioning task despite the satellite trajectory and the motion of the object with respect to the world frame. Given that simulations that consider simulated visual features are conclusive, we can now try them with real images.

3.3.2 Image-based simulations

3.3.2.1 Simulated images

The control law is now validated considering real satellite images provided by Airbus Defence and Space. To do so, the same approach as geometric simulations is used to simulate 3D motions of the camera. The main difference is that we consider the visualization of a simulated image. A simulated image is obtained such as:

— We have a satellite image of 50 cm-resolution of the Earth, generally corresponding to a real surface below 1km$\times$1km of dimensions.

— The four corners of the image are positioned in their approximate real locations on the Earth through 3D points, which define a 3D plane that corresponds to a colored point cloud.

— The simulation is started, the scene and the camera move thanks to their pose with respect to the world frame, respectively obtained through ${}^{w}\mathbf{T}_{o}$ and ${}^{w}\mathbf{T}_{c}$.

— The 3D plane is expressed in the camera frame thanks to ${}^{c}\mathbf{T}_{o} = {}^{w}\mathbf{T}_{c}{}^{-1}{}^{w}\mathbf{T}_{o}$, and projected in the image plane with the pinhole camera model.

— The image is digitized, and each pixel corresponds to the intensity of projected objects in that pixel position.

We use the ViSP vpImageSimulator class [É. Marchand et al., 2005] to perform this task.

3.3.2.2 Template tracking

These images are simulated at the same frequency $F = 5$ Hz and we need to track the visual features inside. In fact, we have no prior knowledge of the target and we only know that it is contained in the image. Thus, it seems difficult to set up 3D model-based tracking, which is very effective in machine vision, but requires knowledge of the 3D model of the object to be tracked. Another possibility is to consider template tracking, which is an approach that seeks to match a reference area called a template, selected directly in the initial image, with the areas of the current image, and to estimate the transformation between the reference template and its current position.

To proceed with template tracking, a model of motion needs to be selected that will characterize the transformation of the template along images, together with a function of similarity and a minimization method to detect the transformed template in current images with respect to the reference template. More precisely, we selected to track a homography, which is a generic motion model that adequately describes the motion of our simulated images with respect to time, using a SSD (Sum of Squared Difference) as a similarity function and an inverse compositional approach as the minimization process [Baker & Matthews, 2004] [É. Marchand et al., 2005].

Then, an initial template is selected manually since object detection is not within the scope of this thesis. It is then tracked by the template tracking algorithm. The visual features are calculated from the template: its centroid is the target point $\mathbf{x}_p = (x_p, y_p)$ in pixel coordinates, while the middle of its first segment defines the second point from which the angle α is computed. To obtain normalized coordinates $\mathbf{x} = (x, y)$, a frame change is performed through the intrinsic parameters of the simulated camera expressed in equation (1.23) of Chapter 1, such as:

$$x = \frac{x_p - x_{p_0}}{p_x}, \; y = \frac{y_p - y_{p_0}}{p_y} \tag{3.38}$$

We consider a camera image of 1000×1000px², and set $x_{p_0} = y_{p_0} = 500$ px. For p_x and p_y, they are fixed and expressed by equation (2.22) of Chapter 2:

$$p_x = p_y = \frac{z_{alt}}{\text{GSD}} \tag{3.39}$$

as $z_{alt} = 500$ km and GSD$= 50$ cm, $p_x = p_y = 1e6$.

3.3.2.3 Simulation results

In this part, our aim is to validate the control law when considering real-scale images with simulated satellite acquisition. The results are presented in Figure 3.8. First, we are in a plane configuration where the focusing point (or centering, as the goal is to center the target in the image frame) and the orientation point are considered with the same altitude. The template is selected in the image that defines the visual features to be tracked and extracted. The centering task is then performed with a straight-line trajectory that shows once again that the external motions are well compensated (Figure 3.8a&d). The orientation task is still completed later due to a more consequential initial error in angle and a lower proportional gain. Still, an acceleration saturation is triggered on z axis at the very beginning of the control, which is almost not seen in the image trajectory. When both visual tasks are fully accomplished (Figure 3.8a), the control scheme succeeds in keeping the target at the center of the image frame with precision below the pixel, and with the desired orientation, in spite of satellite and scene translations (Figure 3.8c), that can be observed with the sequence of images displayed in Figure 3.8d. In this configuration, it is easy to see that the satellite performs a fast reorientation to a newly defined target and holds it precisely for more than 2 minutes, which is more than expected for a video satellite in a 500 km circular orbit. As the pointing is defined in terms of image information, it is impossible to deviate from the mission as soon as the visual control law is converging properly.

To conclude, the control law designed to provide an accurate pointing of a precise zone on the Earth's surface and to acquire images in a specific orientation performs adequately in the proposed configurations. Now, we propose to add the satellite's inner dynamics to the model and make sure that the control law works just as well.

3.4 Incorporation of the satellite dynamics

Now that a control law has been designed for an Earth observation satellite whose dynamic response to the attitude controller was a simple integrator, we can propose a second model provided by Airbus D&S which is considered to represent more accurately the dynamic response of the satellite to the low-level attitude controller. This will induce some changes in our control law.

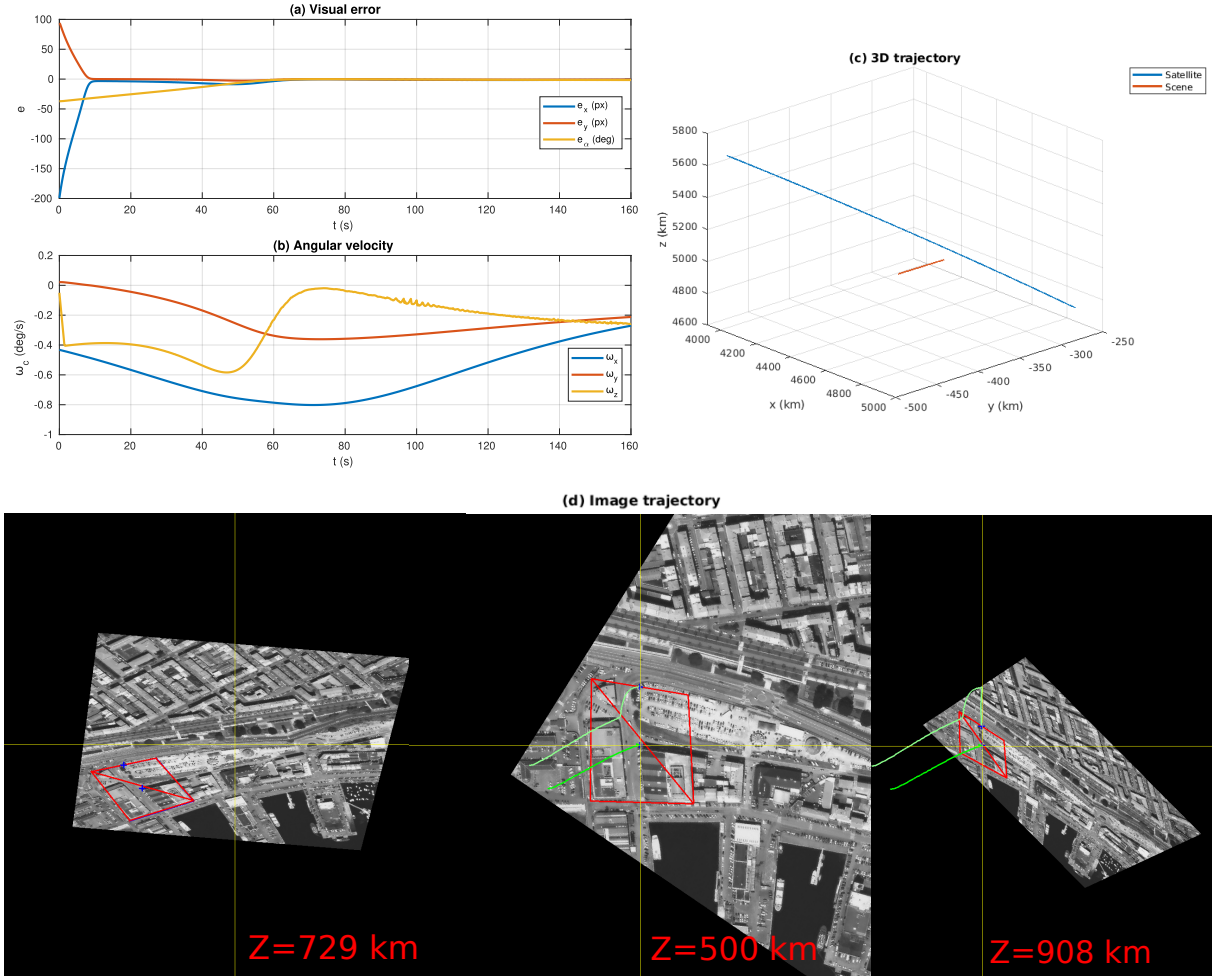


Figure 3.8 – Simulation results considering an object of interest located in a real satellite image of the port of Brest. A video illustrating the results is available here. The selected images are represented at three different instants, the definition of the object of interest by the center of the reference template, the completion of the centering and of the orientation task, and the holding of the precise location when the satellite moves away. The depth of the scene in the camera frame is displayed and is consistent with the satellite motion, as the satellite first gets closer to the target and reaches its zenith, the image is enlarged. Then, the distance to the target increases and the image shrinks.

3.4.1 Low-level controller

The Laplace transfer function given by Airbus D&S that models the inner reaction of the satellite is:

$$F(p) = \frac{2z_c\omega_0 p + \omega_0^2}{p^2 + 2z_c\omega_0 p + \omega_0^2} \tag{3.40}$$

which is a classic 2nd order system with $z_c = \frac{\sqrt{2}}{2}$ the damping coefficient corresponding to a fast pseudo periodic regime, and $\omega_0 = 2\pi F_o$ the natural frequency of the controller, which is set to $\omega_0 = \pi$ with $F_o = 0.5$ Hz, that is, in fact, corresponding to the same bandwidth of the Pléiades CMGs [Thieuw & Marcille, 2007]. Thanks to the knowledge of this transfer function, it is possible to predict the effect of sending a new $\boldsymbol{\omega}_c$ at each iteration of the VS loop, and with a first look at this function, it slows down the execution of the control law. If the consideration of this function does not change the way the control is designed, the saturation algorithm developed previously is too tight as we constrain the control input, while the mechanical constraints of the satellite are specified for the attitude control output. Thus, we choose to propose a second saturation algorithm based on the prediction of the output velocity.

3.4.2 Predictive saturator

We recall the velocity and acceleration constraints subject to the rotational axes of the satellite:

— On x and y axes, speed is limited to 3 deg/s and acceleration to 0.6 deg/s²,

— On the z axis, speed is limited to 1.2 deg/s and acceleration to 0.25 deg/s².

As mentioned before, the satellite rotation will be slowed down by its inner reaction, these constraints should not be respected by the IBVS output $\boldsymbol{\omega}_c$ but by $\boldsymbol{\omega}_r$, the real velocity generated by the low-level controller that will be operated by the satellite.

3.4.2.1 Predicted velocity

Typically, the real velocity $\boldsymbol{\omega}_r$ is unknown, but we can predict it by relying on the transfer function given by equation (3.40). In fact, it is difficult to obtain analytically the time response of the velocity through this transfer function, as the velocity we send to the low-level controller is nonlinear. Still, we can approximate the time response of the velocity considering that we send acceleration steps to the low-level controller. Indeed, the time-response $f(t)$ of the Laplace

function for a unit step is directly obtained from equation (3.40), for $t \geq 0$:

$$f(t) = 1 - e^{-z_c\omega_0 t}(\cos(z_c\omega_0 t) - \sin(z_c\omega_0 t)) \tag{3.41}$$

else if $t < 0$, $f(t) = 0$. Let us consider $\boldsymbol{\gamma}$ an acceleration step, the response $\boldsymbol{\gamma}_r$ of the low-level controller to this step is given by:

$$\boldsymbol{\gamma}_r(t) = \boldsymbol{\gamma} f(t) \tag{3.42}$$

To obtain a velocity $\boldsymbol{\omega}_c(t)$ designed by the control law as input of our system, we need to send to the low-level controller the correct acceleration step $\boldsymbol{\gamma}_c(t) = \frac{\omega_c(t) - \omega_c(t-dt)}{dt}$ only during the sampling time of the control loop $dt = 1/F = 200ms$. This is done by sending an acceleration step $\boldsymbol{\gamma}$ that contains $\boldsymbol{\gamma}_c$ and compensates for the last step sent so that the previous step also lasts during dt. If we now send successive acceleration steps to the controller at each iteration of the control, we obtain the following sequence:

Iteration 0

$\boldsymbol{\gamma}(0) = \boldsymbol{\gamma}_c(0)$

Iteration 1

$\boldsymbol{\gamma}_r(t) = \boldsymbol{\gamma}_c(0) f(t)$

$\boldsymbol{\gamma}(dt) = \boldsymbol{\gamma}_c(dt) - \boldsymbol{\gamma}_c(0)$

Iteration 2

$$\begin{aligned}\boldsymbol{\gamma}_r(t) &= (\boldsymbol{\gamma}_c(dt) - \boldsymbol{\gamma}_c(0)) f(t - dt) + \boldsymbol{\gamma}_c(0) f(t) \\ &= \boldsymbol{\gamma}_c(dt) f(t - dt) + \boldsymbol{\gamma}_c(0)(f(t) - f(t - dt))\end{aligned} \tag{3.43}$$

$\boldsymbol{\gamma}(2dt) = \boldsymbol{\gamma}_c(2dt) - \boldsymbol{\gamma}_c(dt)$

[...]

Iteration k+1

$$\begin{aligned}\boldsymbol{\gamma}_r(t) = \boldsymbol{\gamma}_c(kdt) f(t - kdt) &+ \boldsymbol{\gamma}_c((k-1)dt)(f(t - (k-1)dt) - f(t - kdt)) \\ &+ ... + \boldsymbol{\gamma}_c(0)(f(t) - f(t - dt))\end{aligned}$$

The last line of the iteration $k + 1$ can also be written:

$$\boldsymbol{\gamma}_r(t) = \boldsymbol{\gamma}_c(kdt)(f(t - kdt) - f(t - (k+1)dt) + ... + \boldsymbol{\gamma}_c(0)(f(t) - f(t - dt)) \tag{3.44}$$

with $f(t-(k+1)dt) = f(0) = 0$ as $t = (k+1)dt$ at iteration $k+1$. Thus equation (3.44) can also be written:

$$\boldsymbol{\gamma}_r(t) = \boldsymbol{\gamma}_c(t-dt)(f(dt) - f(0)) + ... + \boldsymbol{\gamma}_c(0)(f(t) - f(t-dt)) \tag{3.45}$$

Equation (3.45) is valid $\forall t$, and can be expressed using a sliding window. The acceleration output of the low-level controller is given by:

$$\boldsymbol{\gamma}_r(t) = \sum_{i=0}^{k_i} \Big[\boldsymbol{\gamma}_c(t-idt)\,(f(idt) - f((i-1)dt)) \Big] \tag{3.46}$$

with k_i is the length of the time window (k_i $\approx$10). The time window represents the number of acceleration steps considered to approach the real acceleration, and the selected value k_i provides a good approximation, considering that $f(k_i dt) - f((k_i - 1)dt \approx 0$. Note that the sum starts at $i = 0$ corresponding to the acceleration step sent at the current instant, but considering this step: $\boldsymbol{\gamma}_c(t)\,(f(0) - f(-dt)) = 0$ as $f(0) = 0$ and $f(-1) = 0$ by definition of the time-response $f(t)$. It is an expected result, as the acceleration step sent at instant t has no influence on the current acceleration output $\boldsymbol{\gamma}_r(t)$.

We can express equation (3.46) in terms of velocity since $\boldsymbol{\gamma}_r(t) = \frac{\omega_r(t) - \omega_r(t-dt)}{dt}$ and $\boldsymbol{\gamma}_c(t) = \frac{\omega_c(t) - \omega_c(t-dt)}{dt}$. We obtain:

$$\begin{aligned}
\boldsymbol{\omega}_r(t) &= \boldsymbol{\gamma}_r(t)dt + \boldsymbol{\omega}_r(t-dt) \\
&= \sum_{i=0}^{k_i} \Big[\left(\boldsymbol{\gamma}_c(t-idt)\right) \left(f(idt) - f((i-1)dt)\right) \Big] dt + \boldsymbol{\omega}_r(t-dt) \\
&= \sum_{i=0}^{k_i} \Big[\left(\frac{(\boldsymbol{\omega}_c(t-idt) - \boldsymbol{\omega}_c(t-(i+1)dt)}{dt} \right) \left(f(idt) - f((i-1)dt)\right) \Big] dt + \boldsymbol{\omega}_r(t-dt)
\end{aligned}$$

We finally have:

$$\boldsymbol{\omega}_r(t) = \sum_{i=0}^{k_i} \Big[\left((\boldsymbol{\omega}_c(t-idt) - \boldsymbol{\omega}_c(t-(i+1)dt)\right) \left(f(idt) - f((i-1)dt)\right) \Big] + \boldsymbol{\omega}_r(t-dt) \tag{3.47}$$

which can be written:

$$\boldsymbol{\omega}_r(t) = \sum_{i=0}^{k_i} \mathrm{g}(i,t) + \boldsymbol{\omega}_r(t-dt) \tag{3.48}$$

with $\mathrm{g}(i,t) = \left((\boldsymbol{\omega}_c(t-idt) - \boldsymbol{\omega}_c(t-(i+1)dt)\right) \left(f(idt) - f((i-1)dt)\right)$, $\boldsymbol{\omega}_r(t)$ is the real angular rate operated by the satellite at instant t. In the following, we will propose to refine

our previous saturation algorithm by saturation of $\boldsymbol{\omega}_c(t)$, the control input designed by visual servoing at instant t, if the real angular velocity at instant $t + dt$ would exceed the constraints. This is done by considering $\boldsymbol{\omega}_p(t + dt)$ the prediction of $\boldsymbol{\omega}_r$ at instant $t + dt$, given by:

$$\boldsymbol{\omega}_p(t + dt) = \sum_{i=0}^{k_i} \mathbf{g}(i, t + dt) + \boldsymbol{\omega}_r(t) \tag{3.49}$$

This prediction allows detecting if a limit would be breached by the sending of a new $\boldsymbol{\omega}_c(t)$, and to determine a saturated velocity $\boldsymbol{\omega}_s(t)$ at instant t so that it does not occur.

3.4.2.2 Saturation algorithm

A new saturation algorithm is then designed considering each component ω_{pi} of $\boldsymbol{\omega}_p$ expressed by (3.49). When a constraint is breached, a reduction factor is computed on the axis concerned, they are expressed by $(r_x^\gamma, r_y^\gamma, r_z^\gamma)$ for instant acceleration constraints and $(r_x^\omega, r_y^\omega, r_z^\omega)$ for angular rates constraints. These factors are computed according to the following rules:

— $\forall i \in [x, y, z]$, if $|\gamma_i| > \gamma_{max_i}$ where $\gamma_i = \frac{\omega_{pi}(t+dt) - \omega_{ri}(t)}{dt}$ then $r_i^\gamma = \frac{\gamma_{max_i}}{|\gamma_i|}$ else $r_i^\gamma = 1$.
— $\forall i \in [x, y, z]$, if $|\omega_{pi}| > \omega_{max_i}$ then $r_i^\omega = \frac{\omega_{max_i}}{|\omega_{pi}|}$ else $r_i^\omega = 1$.

Then, as the z axis has stronger constraints, and for not penalizing the time-to-convergence of the centering task, the same strategy of the previous algorithm is adopted:

Reminder of Section 3.2.5.2

— exceeding velocity or acceleration constraints on x or y axis induces a velocity reduction on all axes, *i.e.*, $r_x = r_y = r_z$.
— exceeding constraints on z axis induces a velocity reduction on z axis only, *i.e.*, $r_x = r_y = 1$.
— exceeding constraints simultaneously on multiple axes induces that $r_x = r_y = \min(r_x, r_y)$ and $r_z = \min(r_x, r_y, r_z)$.

This strategy has been selected to perturb the trajectory as little as possible while maintaining a straight-line trajectory of the target in case it is static on the Earth.

Then, the saturated velocity $\boldsymbol{\omega}_s$ depends on whether the crossed constraint is an acceleration constraint or a velocity constraint. When an acceleration constraint is breached, we have:

$$\mathbf{C}^\gamma \Big(\boldsymbol{\omega}_p(t + dt) - \boldsymbol{\omega}_r(t) \Big) = \sum_{i=0}^{k_i} \mathbf{g}(i, t + dt) \tag{3.50}$$

where:

$$\mathbf{C}^{\gamma} = \begin{pmatrix} r_{xy}^{\gamma} & 0 & 0 \\ 0 & r_{xy}^{\gamma} & 0 \\ 0 & 0 & r_{z}^{\gamma} \end{pmatrix} \tag{3.51}$$

We isolate the last computed velocity $\boldsymbol{\omega}_c(t)$ which, in case of a saturation, is written $\boldsymbol{\omega}_s(t)$, and we obtain:

$$\boldsymbol{\omega}_s(t) = \frac{\mathbf{C}^{\gamma}\Big(\boldsymbol{\omega}_p(t+dt) - \boldsymbol{\omega}_r(t)\Big) - \sum_{i=2}^{k_i} \mathbf{g}(i, t+dt)}{f(dt)} + \boldsymbol{\omega}_s(t-dt) \tag{3.52}$$

Note that $\boldsymbol{\omega}_s(t-dt) = \boldsymbol{\omega}_c(t-dt)$ if there was no acceleration saturation at instant $t-dt$. This equation means basically that if $\mathbf{C}^{\gamma} = \mathbf{I}_3$, no saturation occurs and $\boldsymbol{\omega}_s(t) = \boldsymbol{\omega}_c(t)$. Else if $\mathbf{C}^{\gamma} \neq \mathbf{I}_3$ (acceleration constraint crossed), a new prediction $\boldsymbol{\omega}_p$ must be computed with $\boldsymbol{\omega}_s$ in equation (3.52) to test if the new prediction exceeds the velocity constraints.

Besides, when a velocity constraint is breached, we have:

$$\mathbf{C}^{\omega}\boldsymbol{\omega}_p(t+dt) = \sum_{i=0}^{k_i} \mathbf{g}(i, t+dt) + \boldsymbol{\omega}_r(t) \tag{3.53}$$

where $\mathbf{C}^{\omega} = \begin{pmatrix} r_{xy}^{\omega} & 0 & 0 \\ 0 & r_{xy}^{\omega} & 0 \\ 0 & 0 & r_{z}^{\omega} \end{pmatrix}$. Once again we isolate the last computed velocity $\boldsymbol{\omega}_c(t)$ which, in case of a saturation, is written $\boldsymbol{\omega}_s(t)$, and we obtain:

$$\boldsymbol{\omega}_s(t) = \frac{\mathbf{C}^{\omega}\boldsymbol{\omega}_p(t+dt) - \boldsymbol{\omega}_r(t) - \sum_{i=2}^{k_i} \mathbf{g}(i, t+dt)}{f(dt)} + \boldsymbol{\omega}_s(t-dt) \tag{3.54}$$

Note that, again, $\boldsymbol{\omega}_s(t-dt) = \boldsymbol{\omega}_c(t-dt)$ if there was no velocity saturation at instant $t-dt$. In the end, we have $\boldsymbol{\omega}_s(t) = \boldsymbol{\omega}_c(t)$ if $\mathbf{C}^{\omega} = \mathbf{C}^{\gamma} = \mathbf{I}_3$ (that means no limit has been detected).

This saturation algorithm guarantees that each $\boldsymbol{\omega}_s(t)$ sent to the low-level controller implies $\boldsymbol{\omega}_r(t+dt)$ to ensure the mechanical constraints of the satellite.

3.5 Results of the control law

In this section, we show the results obtained for our control law considering the dynamic model specified by Airbus D&S. We first present our results in the satellite path simulator

and show that our control law works equally well in many scenarios. Finally, we propose an implementation of the law on a Cartesian robot simulating reduced satellite movements.

3.5.1 Satellite simulation results

The control law is still implemented using the ViSP framework [É. Marchand et al., 2005] with the same trajectory simulations presented in Section 3.3. The difference in the control parameters is that we now consider the predictive saturation algorithm and that we disable the prediction of $\mathbf{L}_s$ as it is harder to set up correctly when considering the inner dynamic of the satellite (and which, in the current state, does not improve the behavior of the control). However, we will see that the image trajectories are still very satisfactory.

3.5.1.1 Motionless object

We first consider that the target is a fixed location on the Earth's surface that is initially visible in the scene of interest.

A first nominal simulation is proposed in Figure 3.9 with exactly the same image target and gains as in Figure 3.8. We observe similar results in terms of convergence speed and angular velocity (Figure 3.8a&b). Indeed, both visual tasks are fully completed with a slower convergence for the orientation task (approximately 5 s for the centering task and 60 s for the orientation task) due to the significant initial error in α and the stricter constraints on the yaw axis, which leads to acceleration saturation at the beginning. However, the trajectory of the target is no longer a perfect straight line (Figure 3.8d) because the compensation terms of satellite orbital motion and terrestrial movements (Figure 3.8c) also undergo the dynamics of the satellite, resulting in a slightly curved trajectory at the beginning, intensified by the saturation of z axis , and slight jerks along the trajectory. Yet, it is still very satisfactory as long as there are no perturbative behaviors or oscillations.

Most of all, once the error is regulated to 0 and the target location is reached, it is precisely held (at a pixel accuracy) regardless of the 3D trajectories with respect to time, which are, for the satellite, corresponding to a distance of approximately 1200 km in 160 s. This last statement is comparable to the locking mode obtained in [Klančar et al., 2012] and recently [Dauner et al., 2023], although they consider a greater ground sampling distance (GSD), which induces much fewer perturbations of external motions.

In a second experiment presented in Figure 3.10, we decide to increase the gain λ_α so the orientation task has a reasonable convergence time, even if we reach an acceleration saturation

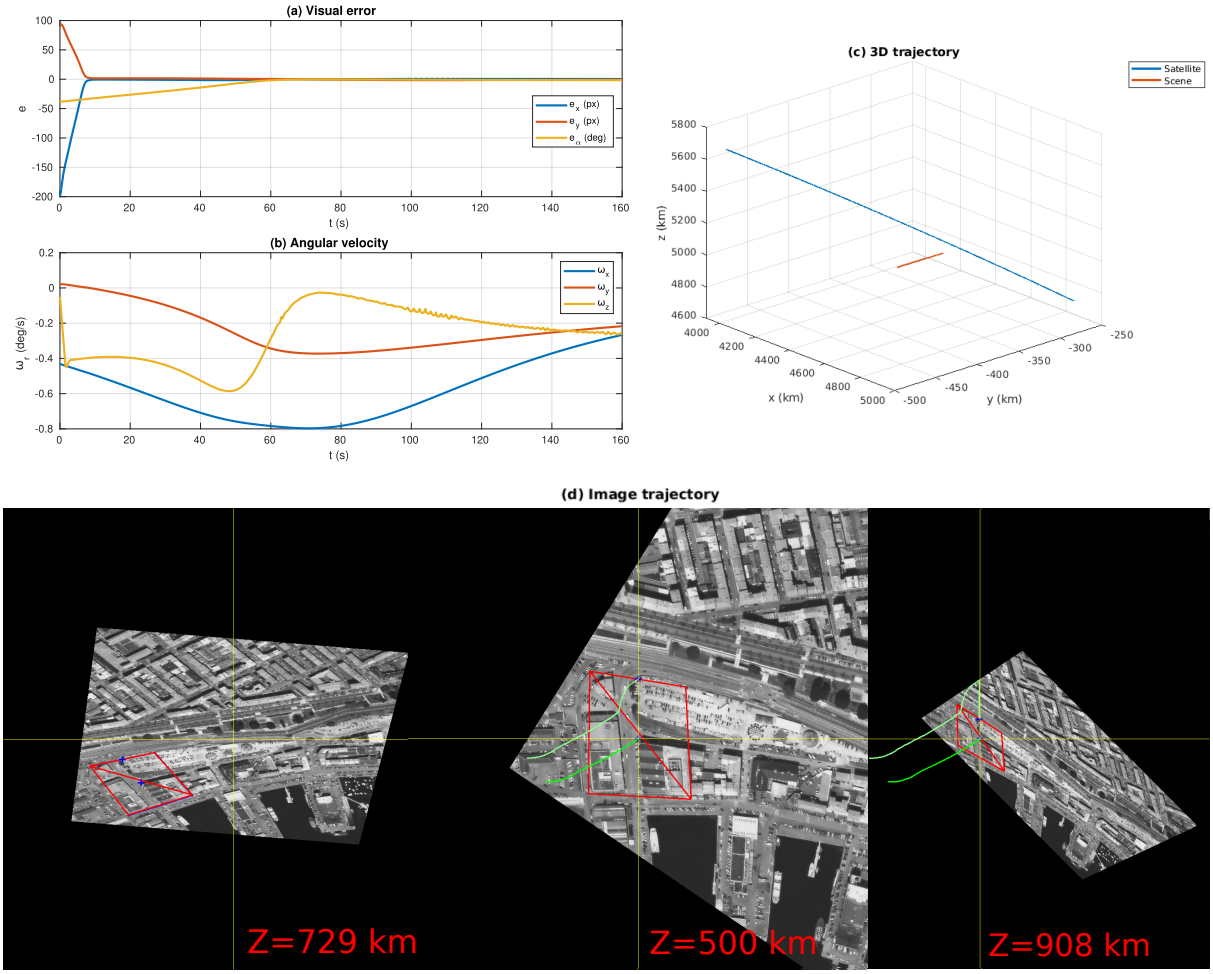


Figure 3.9 – Control law results considering the same object of interest as Figure 3.8 and with exactly the same gains. A video illustrating the results is available here. The results are similar in terms of convergence time and angular velocity, but the image trajectory is not a perfect straight line at the beginning.

at the beginning and an angular rate saturation from around 15 to 25 seconds (Figure 3.10(b)). This allows us to complete the orientation task two times faster with almost no loss in the image trajectory compared to the previous case. As a preliminary conclusion, we see that the inner dynamic of the satellite is well handled by our control law, first by respecting its mechanical constraints and secondly without visibly impacting either the convergence time or the angular velocity of the original controller. Without a complete demonstration about it, it is likely explainable by the nature of the control law, which is by design a feedback controller that induces a velocity input comparable as an exponential decrease function (or at least locally when considering an adaptive gain) whose amplitude depends on the amplitude of the error itself, which is thus adjusted at each iteration. If such controllers are slow down, it means that the error is not decreasing as fast as expected, and so the next input is computed considering a higher error function as it would.

We finally keep the same settings to operate diverse acquisitions in different areas of the globe and so with different influences of the orbital motions. The image trajectories during the acquisition of Brest, Barcelona, Tarragona and Vancouver are shown in Figure 3.11. In all these multiple cases, the control law successfully completes both tasks with an adequate image trajectory.

3.5.1.2 Target tracking

Now that we considered fixed target on the Earth's surface, we need to assess the tracking efficiency for moving targets. In this case we implement again the pan-tilt law expressed in equation (3.36). The targets are assumed to move at a constant velocity and the adaptive visual integrator is included in the control loop to compensate for the tracking error caused by this motion. We first considered fully simulated targets with different trajectories and velocities. First, Figure 3.12 illustrates the significant impact of the target motion and of the residual uncompensated orbital and terrestrial motions: during the first segment, corresponding to the initial iteration of the control scheme until $t = 0.2$s, the visual servoing has less effect due to the latency triggered by the low-level controller. Then, the control law operates the centering task but does not reach directly the center of the image plane. The bend present in each curve at $t = 4$ s would represent the final position of the target in the image if there was no integrator in the control loop, and where the tracking error induced by the target own motion and also by the residual motions of the satellite and of the Earth can be measured. Of course, the amplitude of this tracking error varies mostly according to the target's speed, as greater tracking errors occur for faster motion. From $t = 4$ s onward, the influence of the visual integrator becomes more

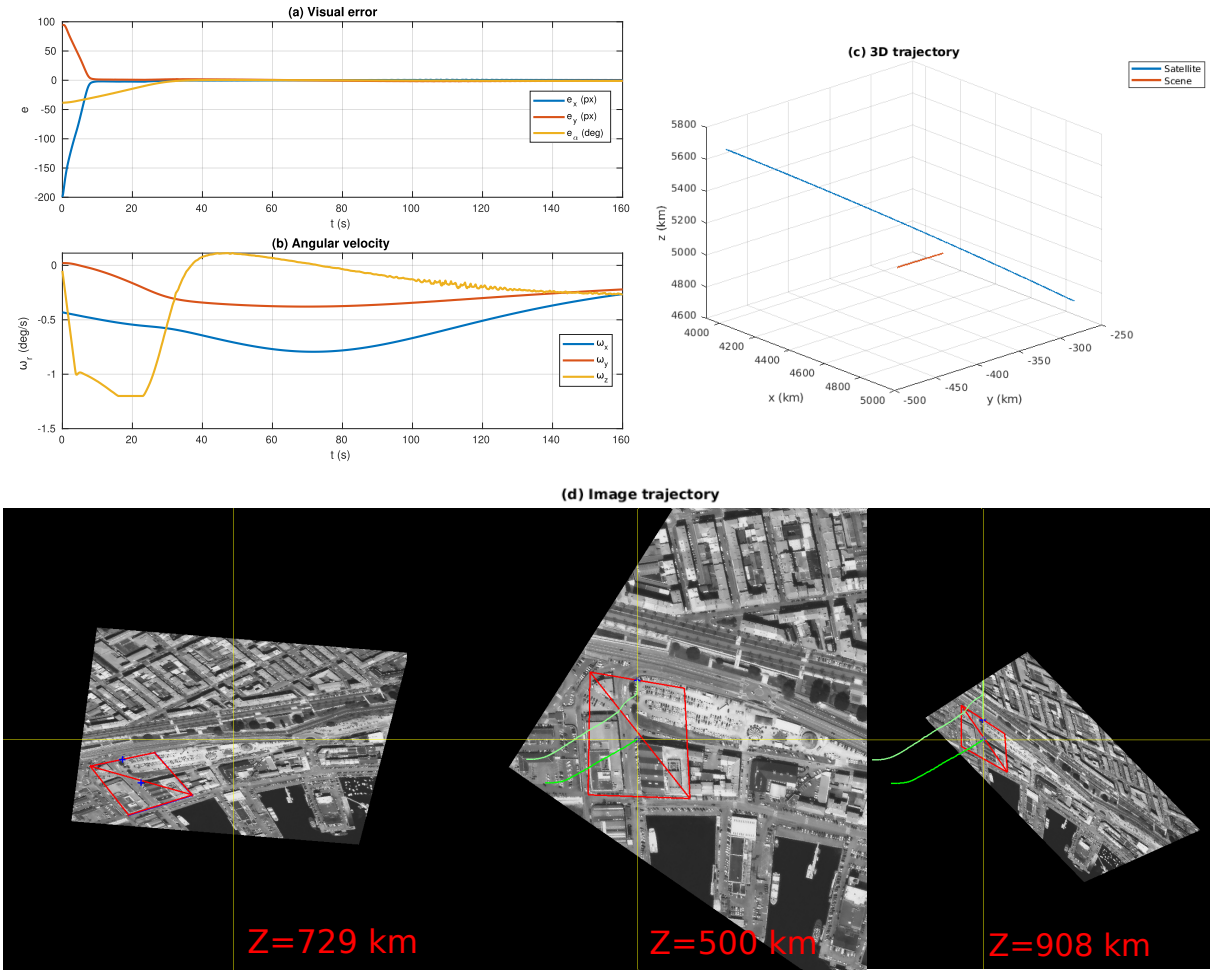


Figure 3.10 – Control law results in the same configuration as Figure 3.9 but the gain λ_α is increased, which enhances the convergence time of the orientation task against stronger saturations on the z axis, which have no visible impact on the image trajectory. A video illustrating the results is available here.

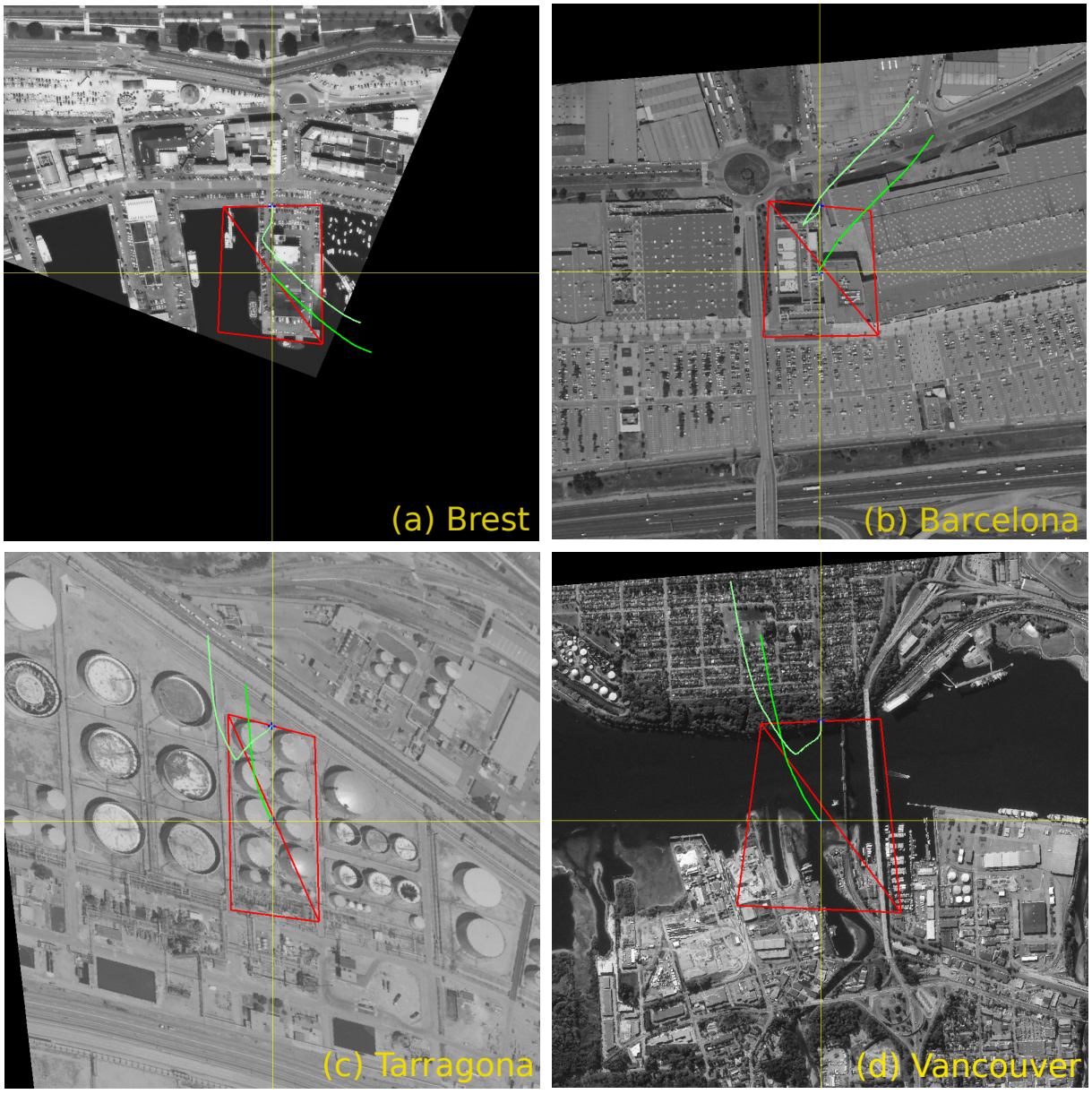


Figure 3.11 – Multiple image trajectories after the completion of the visual task in (a) the port of Brest, (b) a commercial center in Barcelona, (c) A refinery in Tarragona, and (d) a bridge in Vancouver. Images provides by Airbus D&S.

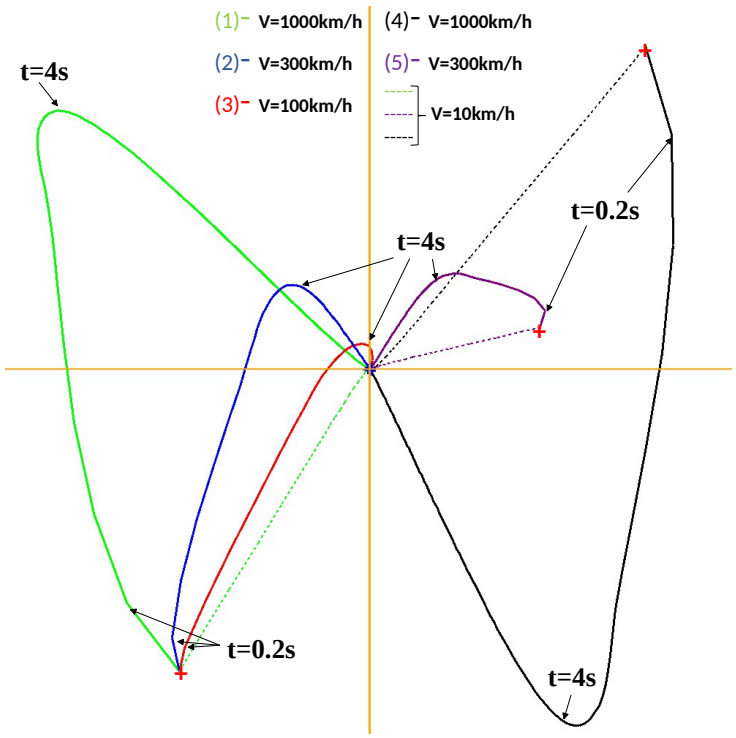


Figure 3.12 – Image trajectory of mobile targets with different configurations. Targets (1), (2) and (3) are starting from the same origin and moves with the same direction, and they are, respectively, moving at a constant speed of 1000 km/h, 300 km/h and 100 km/h. Targets (4) and (5) are moving at a constant speed of 1000 km/h and 300 km/h, but with different origins and directions. The dotted curves are image trajectories of the mobile targets if their speed was very low (10 km/h).

prominent, gradually reducing and eventually canceling out the tracking error. In Figure 3.13, the convergence time is observed, indicating the point at which all residual errors are completely eliminated for each configuration. As the integrator gain μ is adjusted to achieve smooth compensation of the tracking error, the convergence time directly depends on the target speed, which corroborates our last statement in Section 3.3 where the compensation of the tracking error is performed as a secondary task when considering an adaptive visual integrator. This use case demonstrates that our control law can track and center any terrestrial object moving at a constant speed, as long as the target remains within the image during the initial phase when the visual servoing effect is diminished by the satellite dynamics latency.

As a result, the same pan-tilt control law can be exerted on a real sequence of images with a moving object inside. For example, in Figure 3.14, the satellite is flying over a commercial center in Barcelona and the selected target is a car moving on the highway. In that context, the car appears white on a gray road and is tracked using a gray-level tracker implemented in the

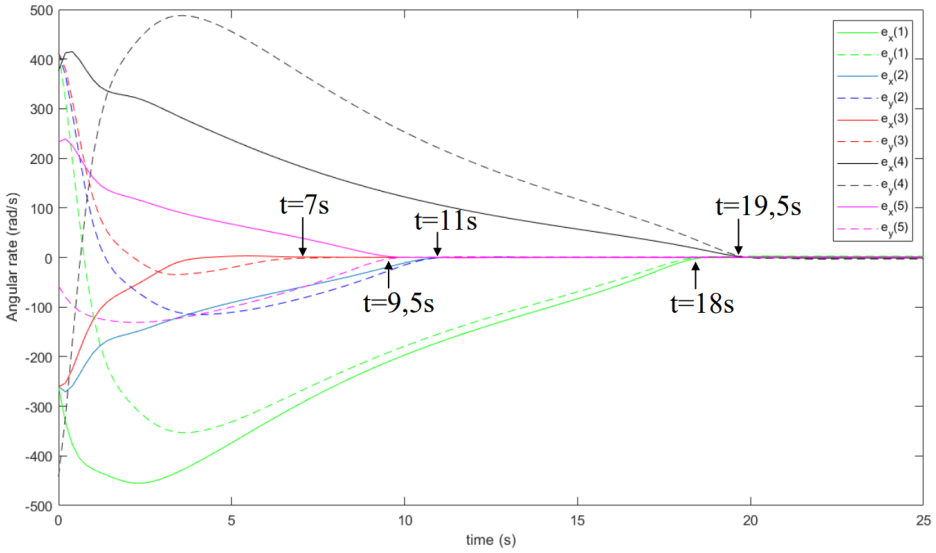


Figure 3.13 – Visual feature error during the tracking of fast-moving targets in different configurations.

ViSP class vpDot2, the feature point $\mathbf{x}$ being the centroid of the tracked area. The car is successfully tracked and centered in the image plane (Figure 3.14d) and similar comments as for fully simulated targets can be made. The influence of the feedback term is more consequent until 4 s, which describes a slightly curved trajectory, and then the visual integrator has more influence and completely centers the target in 7.6 s. Moreover, the trajectory obtained is comparable to the one presented in Figure 3.12. As the gains used for the adaptive visual integrator are similar, it is even possible to compare the convergence time with Figure 3.13 and guesses that the vehicle speed is therefore approximately 100 km/h, but this guess can be refined using the threshold attained in Figure 3.14b, that shows the angular velocity subtracted from the compensations of known external motions, and where the influence of the integrator can be measured.

With this use case, the control law succeeds in operating an accurate pointing of a moving car that appears as a small white dot in the image, and again, this pointing is made under the consequent motions induced by the satellite orbit and the Earth's rotation. With this last acquisition, the results provided by the visual servoing scheme are more than conclusive.

3.5.2 Real robot experimentation results

Finally, our control law was applied to a Cartesian robot that mimics satellite movements in different configurations. The robot is a 6 DoF Gantry robot. Its end effector is equipped with a camera mounted with a pan-tilt-roll unit. The robot follows a downscaled trajectory similar to that of a satellite, while the camera's orientation is controlled through visual servoing. Con-

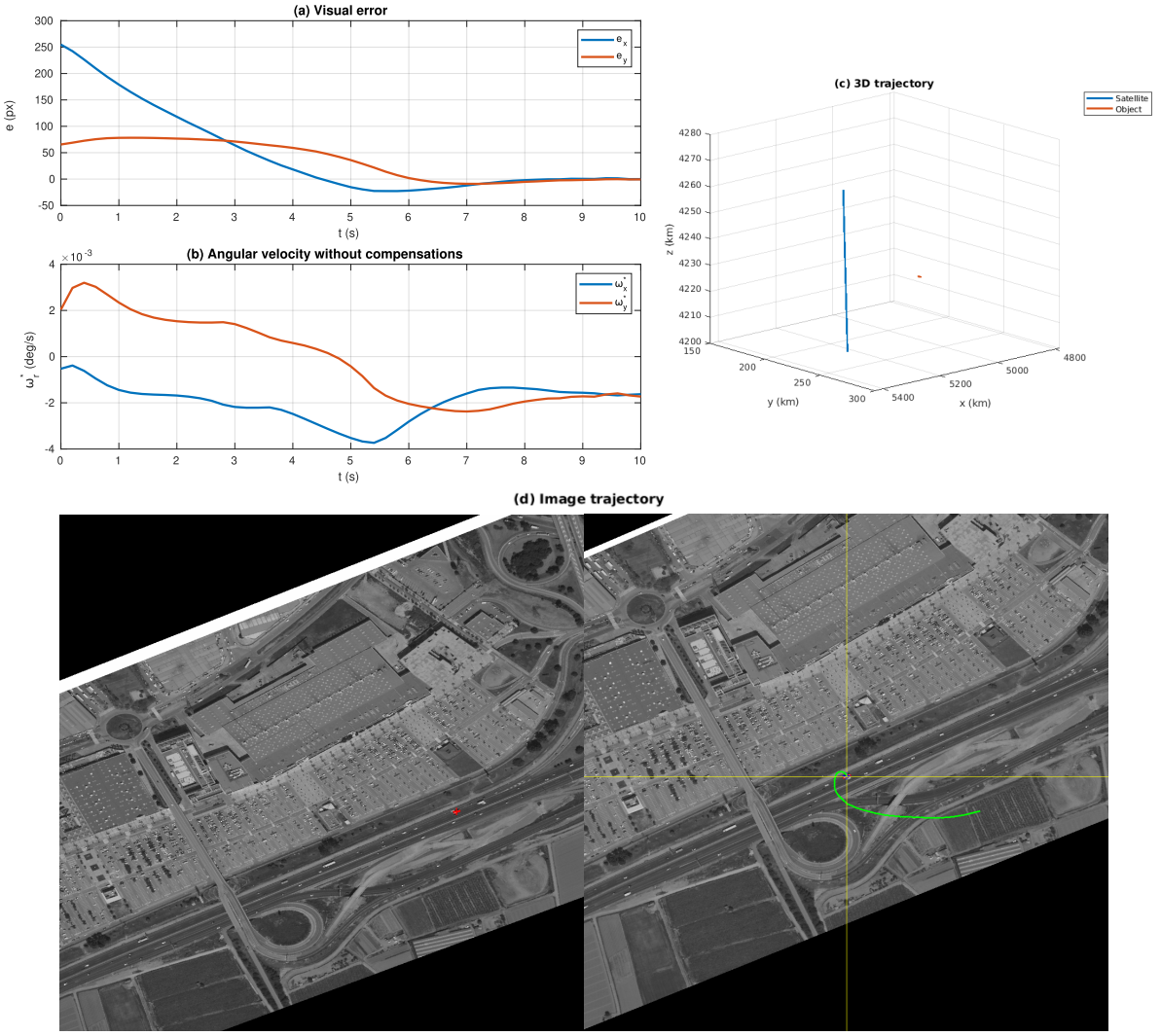


Figure 3.14 – Control law results considering a moving car in a sequence of images of Barcelona. A video illustrating the results is available here. The car is successfully tracked and centered thanks to the adaptive visual integrator that cancels the tracking error, visible from the first inflection point. The angular velocity is shown without the compensation of orbital and terrestrial motions, so the influence of the feedback term and of the integrator are highlighted.

sistent with previous sections, the robot also simulates the inner dynamic of the satellite. The camera is directed towards a scaled satellite image, which reflects our specific application scenario (see Figure 3.15). The robot undergoes translational movement along a typical satellite

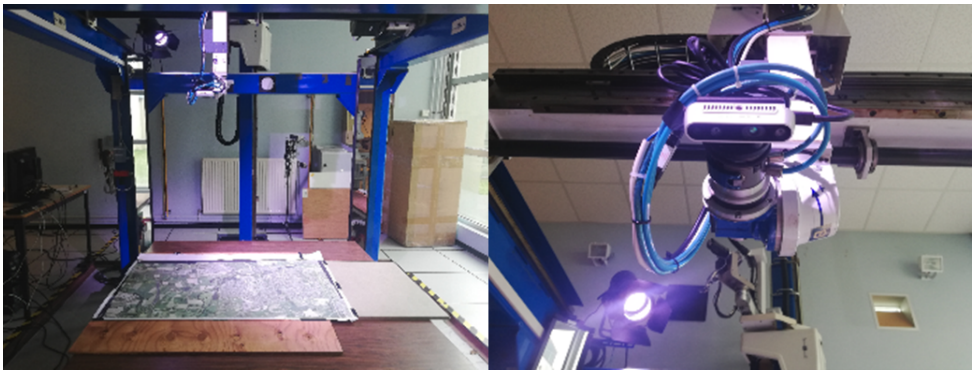


Figure 3.15 – AFMA Cartesian robot positioned above a satellite image, and its INTEL D435 camera mounted on a pan-tilt-roll unit.

trajectory, with its translational velocities scaled down by a factor S_c. This scaling factor corresponds to the ratio between the altitude of the simulated satellite (fixed at 1 meter) and the actual altitude of the satellite at its zenith, which is 500 km: $S_c = 2 \times 10^{-6}$. Furthermore, the target depth Z remains fixed at 1 meter for all configurations and camera inclinations, and we acquire visual features using the template tracking algorithm on the images acquired by the D435 camera at a frequency $F = 5$ Hz. The feature points are then expressed in their normalized coordinates using the camera's intrinsic parameters. Note that in this configuration, the external motions of the system have much less influence and that the camera has a larger field of view, which allows for operating a more consequent angular velocity. As a result, we consider that the robot's angular velocity is null at the instant t_0 when the control starts, and we can experience saturations on the axes x and y that were not observed in our simulations when considering a smaller field of view.

Figure 3.16 illustrates three different configurations: a plane configuration where the desired orientation is determined by two points at the same altitude, a relief configuration where the two points share the same longitude and latitude but have different altitudes, and a moving object configuration that focuses solely on the centering task for target tracking.

3.5.2.1 Plane configuration

In the plane configuration, the two points that determine the centering and orientation tasks are located at the same altitude. As shown in Figure 3.16a, the control law effectively centers the

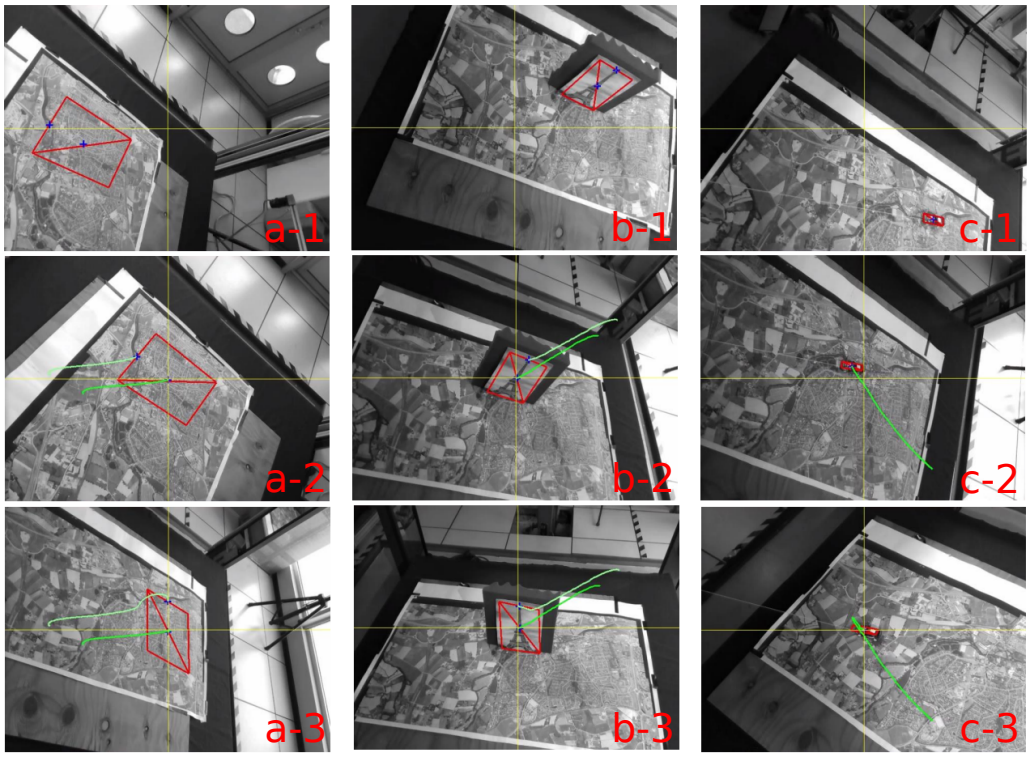


Figure 3.16 – Image trajectory induced by visual control law in plane (a), relief (b) and moving target (c) configurations: at the beginning (1), when centering errors are low (2), and at the end (c). A video illustrating the results is available here. In the first two situations, a nearly straight-line trajectory is observed while completing the centering task, then the target is kept in the center of the image while the orientation task ends. In the last configuration, a straight-line trajectory occurs while reaching the desired position of the car, and then during the compensation of the tracking error induced by its motion.

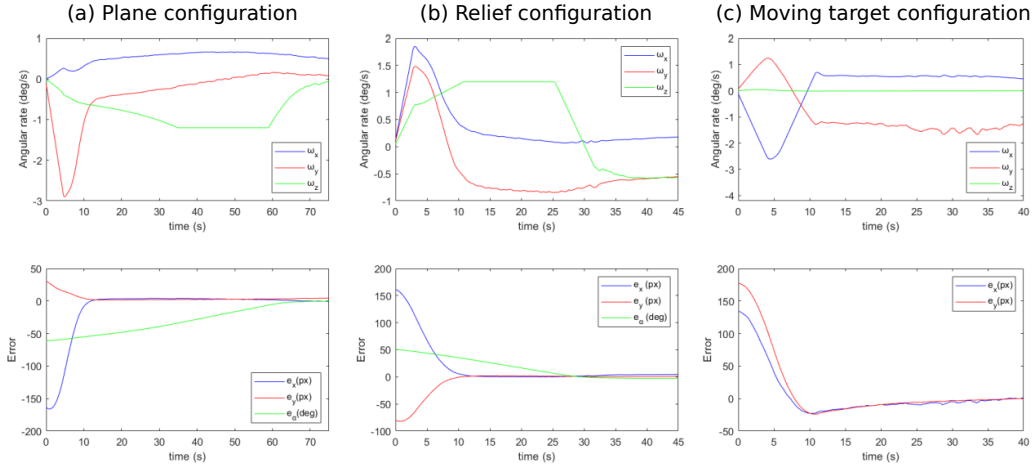


Figure 3.17 – Measured angular velocity ω_r and visual feature error for the visual control law in plane (on the left), relief (in the middle) and moving target (on the right) configurations.

target with a nearly perfect straight-line trajectory, except for the beginning, and successfully orients it without deviating from the center. The impact of our saturation algorithm can be observed in Figure 3.17a. Initially, the acceleration along the y axis is saturated, resulting in a linear decrease of ω_y with a slope of γ_{max_y}, which in turn causes the x and z axes to saturate as well but with different slopes as described previously in Section 3.4.2.2. The slight bending of the curve at the beginning is explained by the inner dynamic of the satellite and amplified by the saturation. When the centering task is achieved, the speed along the z axis becomes saturated (maintained at a constant ω_{max_z}). Finally, when both tasks have converged, the control law effectively compensates for the robot's translational motion, ensuring low visual feature errors.

3.5.2.2 Configuration with relief

In the relief configuration, the two points that determine the centering and orientation tasks have the same longitude and latitude, but different altitudes. To do so (see Figure 3.17b), template tracking is performed on an image of the Eiffel Tower, which is located out of the plane. In this configuration, the satellite traverses the scene without directly passing over the target's zenith, ensuring that the angle α remains well-defined. Similarly to the previous configuration, the control scheme achieves convergence with a quasi-straight-line trajectory of the target, followed by the completion of the orientation task. The same observations regarding saturation can be made, as seen in the plane configuration: initial acceleration saturation of the x axis and angular velocity saturation of the z axis, all while maintaining the straight-line trajectory.

3.5.2.3 Moving target configuration

In this last situation, only the pan-tilt configuration is considered to track a moving car with almost constant speed (see Figure 3.16c). As in the previous simulation results in which moving targets were considered, the target first moves in the image to a position with a tracking error, since the integrator has almost no influence at the beginning of the visual servoing (see Figure 3.17c). Then, the adaptive visual integrator has more influence and smoothly compensates for the tracking error. During the compensation of the residual error, angular rates are slightly oscillating because the template tracking is operated on a small area, which makes the target difficult to be tracked, and because the speed of the vehicle is not perfectly constant as it is manually displaced. Nevertheless, the control scheme succeeds in tracking the car in the image despite the car's own motion and the robot translational velocity.

These last results demonstrate the validity of our control law in a real robotic platform

mimicking a satellite test scenario. With this last part, we can conclude on the effectiveness of our approach.

3.6 Conclusion

In this chapter, we proposed a visual servoing control scheme to precisely adjust the orientation of a satellite subject to its orbital motion to track a terrestrial target with a potential partially unknown motion.

We first explored the feasibility of such control by stating its stability in the sense of Lyapunov, and saw that it mostly depends on how the visual features are designed and estimated. So, we considered an image-based visual servoing with image point coordinates as the main visual feature, and we demonstrated that the interaction matrices can be well-estimated because they are mainly based on image information directly available during the control, and the few 3D parameters can be estimated quite accurately. These features defined two visual tasks, a focusing and an orientation task, allowing one to control the full degrees of freedom of the satellite, if desired. Then, the external motion induced by the satellite orbit and the Earth's rotation were analytically established based on our model presented in Chapter 2, and directly included as compensation terms in the control law. To deal with the last residual motion induced by the possible own motion of the target, we implemented an adaptive visual integrator, which provides a satisfying behavior by sequentially canceling all the tracking errors smoothly. We proposed several solutions to improve the behavior of the control, such as the decoupling of the roll and pitch axes with the yaw axis because of stronger velocity and acceleration constraints on the last one. These constraints were handled by a saturation algorithm, which was also considered in a predictive configuration when considering the dynamic response of the satellite low-level controller.

After that, we first validated our visual servoing controller in scenarios considering that the low-level controller was a simple integrator and showed that the control law performed remarkably well either with fully simulated visual features or with features extracted from simulated acquisitions of real-scale satellite images. Then, we applied the control law considering the inner dynamic of the satellite and stated minor visible changes in the behavior of the control. We then successfully performed several simulations considering fixed terrestrial targets in various locations of the globe, but also challenging acquisitions of moving targets (up to 1000 km/h), and especially the tracking of a moving car along satellite images. All of these acquisitions were performed with a pointing accuracy at the pixel level, which corresponds to a ground accuracy

of 50 cm at the zenith. Finally, we implemented the control law on our robotic platform mimicking the path of a satellite. We validated the effectiveness of our approach and showed that this control law could be broadened to other robotic systems using an embedded camera mounted on a pan-tilt-roll unit.

Now that this control law is fully operational in several possible configurations, in the next chapter we will study the issue of image sharpness during camera movements and how to ensure it by reducing motion blur.

CHAPTER 4

QP-BASED VISUAL SERVOING UNDER MOTION BLUR-FREE CONSTRAINT

4.1 Introduction

In the previous chapter, we have developed new control laws to precisely control the orientation of an Earth observation satellite and demonstrated challenging acquisitions by accurately pointing an object of interest on the Earth's surface, that either corresponds to a fixed ground location, or to a target moving with an a priori unknown motion.

For all of the experiments we led, we considered that the image obtained was of high quality, since we simulated the point of view of the satellite through its camera considering post-processed satellite images of 50-cm on-ground resolution. However, some perturbations may cause the images not to be as good as expected. In satellite imagery, the capture can suffer from atmosphere occlusions, frame jumps, parallax effects, etc. But all of these perturbations are independent of the control (except the parallax effect, which occurs mainly when the satellite is consequently away from the zenith of the target). Another perturbation is motion blur, a common problem in image acquisition devices that is due to the camera and/or object motion during the integration time for acquiring an image. As high camera velocity or fast-moving objects can create motion blur in the image, observation satellites can suffer from this default due to their high orbital motion (7 km/s), their fast reorientation as seen in Chapter 3, and the motion of moving targets. Finally, a blurred image is not a good output for both the user and the visual controller, as the former wants to exploit it and the latter relies on features based on image measurements (see Figure 4.1).

Indeed, motion blur induces a degradation of the image quality, which in visual servoing may perturb the feature extraction step and eventually make the control fail. Motion blur is a known problem in the computer vision community, and there are many solutions to estimate the characteristics of blur and restore a blurry image. Still, these techniques are often not applicable to image feedback controllers, as they require too much computation time. Another approach

would be to measure the quality of the image thanks to a sharpness metric, and select a specific metric that can size the quantity of blur. If blur can be evaluated with a function applied to an image, it means that it can be taken as input for a feedback controller. More precisely, specifying that we want to limit the creation of motion blur during camera motion, it can be considered as a constraint on the velocity applied to the camera. Such constraints can be handled by specific techniques from constrained control theory that can be considered in a visual servoing context as soon as we can design a constraint based on the sharpness metric. This chapter is focused on the design of this sharpness constraint.

In this chapter, we will first briefly discuss the aforementioned new notions such as motion blur, sharpness metrics, and constrained control theory applied to visual servoing. Then, we will present a quadratic programming (QP) controller based on our previous IBVS scheme and present several models of the centering task with a pan-tilt configuration. After having selected the norm of the image gradient as a sharpness metric, we will discuss the parameters related to the motion blur kernel affecting our metric and analytically express its Jacobian. Then, we will express the sharpness constraint through a control barrier function so that it can be formulated as a velocity constraint to be injected in the QP, for several configurations of the system: fixed pan-tilt, satellite in pan-tilt mode subject to orbital motions, and target tracking. This constraint will also be proposed with a predictive step to improve the behavior of the system. Finally, we validate our motion blur-free controller for these different use cases in our previous satellite path simulator.

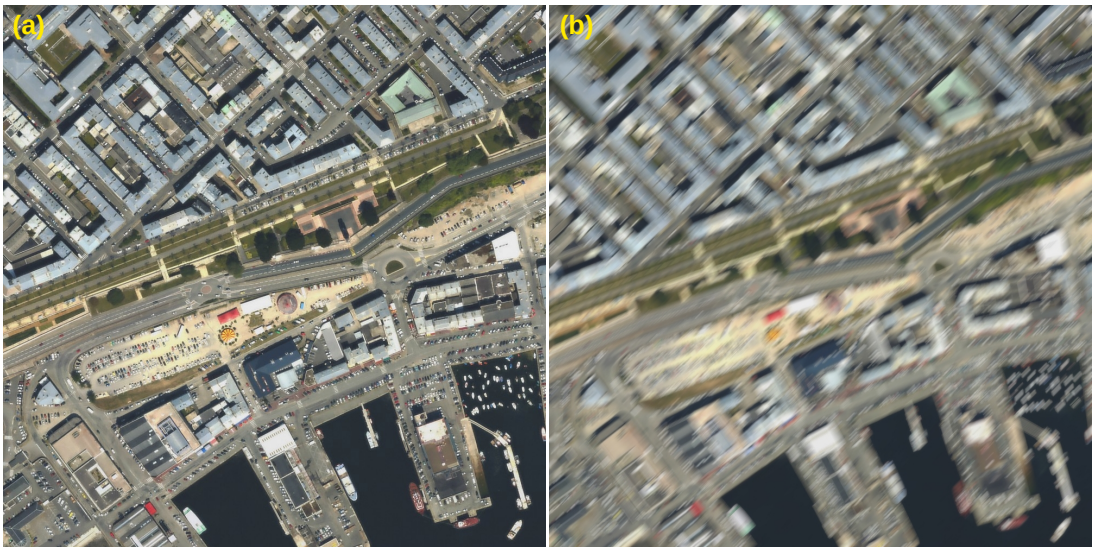


Figure 4.1 – Satellite image of 50-cm resolution of Brest (a) Sharp (b) Motion-blurred

4.2 Preliminaries

In this section, we will present different notions that are not so used in vision-based control, allowing one to understand properly this chapter.

4.2.1 Motion blur

Motion blur, by definition, is a type of blur that results from the projected movement of the camera or objects in the image during a certain integration time. As with every other blur, it induces a loss of information in the image, which appears less sharp.

Motion blur is characterized by its *point spread function* (PSF), or kernel, that can take various forms depending on the cause of the blur, and part of the literature is focused on how to implement a given PSF in image rendering [Potmesil & Chakravarty, 1983][Navarro & Gutierrez, 2011], either to enhance the perceived motion in a single snapshot or to increase the realism in movies or video games [McGuire et al., 2012], but also simply to create realistic acquisitions when considering simulated images. The most common point spread function h is the one used to characterize a linear motion blur, it is given by [Tiwari et al., 2013]:

$$h(a,b) = \begin{cases} 1/L \text{ if} \sqrt{a^2+b^2} \leq \frac{L}{2} \text{ and } \frac{b}{a} = -\tan\alpha \\ 0 \quad \text{else} \end{cases} \tag{4.1}$$

with L the motion amplitude in pixel, also called the length of motion, which corresponds to the intensity of the blurring, α the direction of the motion (NB: it is not correlated with the feature α of Chapter 3, that will not be used in this chapter) that indicates the orientation of the blurring effect, a and b are pixel coordinates that are used in the edge conditions for the repartition of the kernel. The convolution of such kernel with an image produces a motion-blurred image, or more generally, supposing h a motion blur kernel or PSF, I^* a sharp image, we can obtain I the motion-blurred image such as:

$$I = I^* \circledast h \tag{4.2}$$

with $\circledast$ the convolution operator.

Another major stake is to estimate the PSF of motion-blurred images and to recover sharp images through deconvolution, i.e., in a simplified approach, by determining h and then its inverse function h^{-1}. The parameters estimation is usually processed on a single-blurred image, where the length and the orientation of the blur are evaluated through image gradients and spectral approaches [Yitzhaky & Kopeika, 1997][Tiwari et al., 2013]. For example, in Figure 4.2,

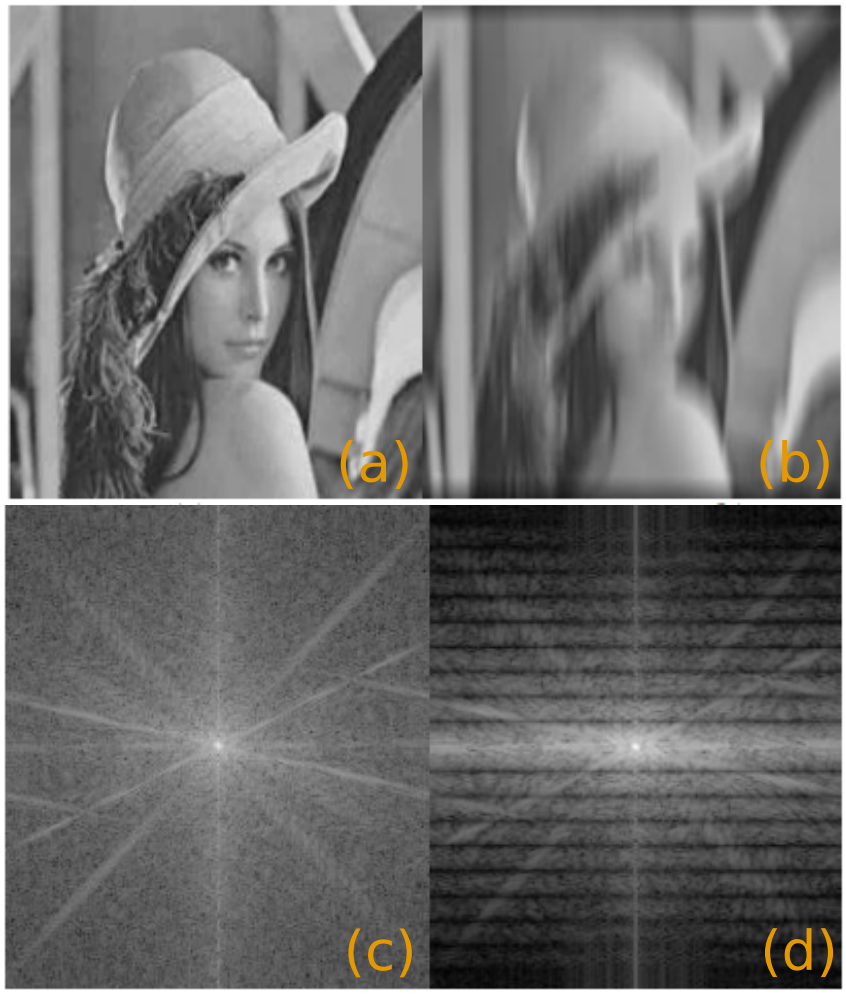


Figure 4.2 – Snapshot taken from [Tiwari et al., 2013] (a) the original image (b) blurred image with motion amplitude of 20 px and direction of motion 90° (c) Fourier spectrum of the original image and (d) Fourier spectrum of blurred image.

the motion amplitude is obtained by measuring the distance between the parallel lines obtained by the Fourier spectrum of the motion-blurred image. The direction of motion can also be stated since it is orthogonal to the same lines.

While these studies are often focused on linear motion blur, relevant works have also been made to handle more general motion types [Ji & Liu, 2008], such as nonlinear, rotational, or space-variant [Dai & Wu, 2008][Cho et al., 2007][Nayar & Ben-Ezra, 2004]. Then, motion-blurred image restoration has been extensively tackled with impressive results: for instance, while [Shan et al., 2008] presents a probabilistic model to deblur a single image, [Cho & Lee, 2009] slightly enhances the computation time by introducing a prediction step in the PSF es-

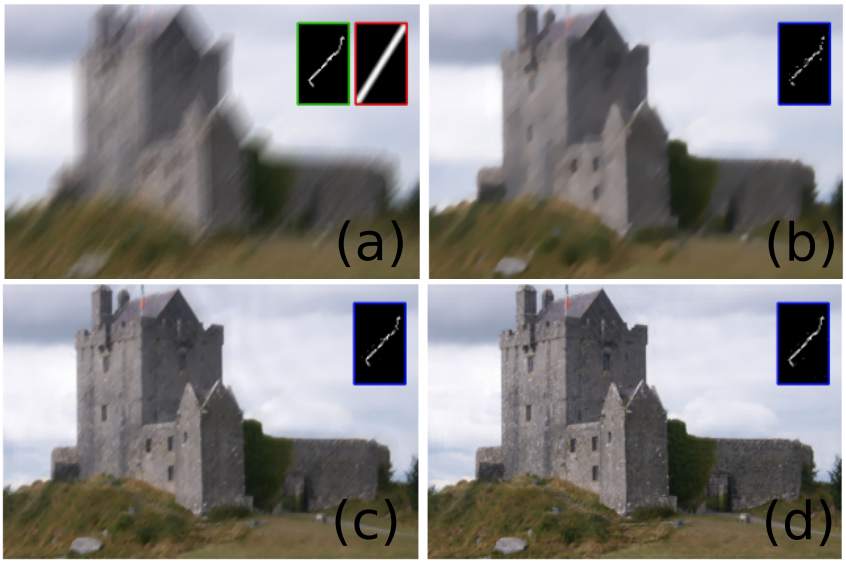


Figure 4.3 – Restored images obtained by [Shan et al., 2008] (a) is the image initially blurred, the green rectangle is the true PSF of motion blur and the red one is the initial guess of the PSF (b) iteration 1 of the deblurring process, the blue rectangle represents the current estimate of the PSF (c) iteration 6 (d) iteration 10, the PSF is perfectly estimated and the image deblurred.

timation and [Xu & Jia, 2010] proposes a robust framework refining the estimation process together with noise suppression during the image restoration step. Finally, learning-based approaches have also been proposed to deblur an image without any estimation of the PSF [Nah et al., 2017][Kupyn et al., 2018]. Although all these studies show restored images of high quality, they need high computational power and their current computational time is not compatible with real-time processing. Since visual servoing is a closed-loop control scheme based on measurements extracted on each image acquired, and since our ultimate goal is to embed all computations onboard, which induces a low-computational process in the case of satellites, the previous approaches can not be employed. So, our method is not based on deblurring the image but uses a measure of the sharpness directly in the control scheme to avoid blurring the image.

4.2.2 Sharpness metric

A sharpness metric is a function that takes as input local or global information of an image (often the intensity of an area of pixels or the spatial derivative of it), and gives a measurement of the quality of an image. These evaluations are hard to compare between them as they are not computed in the same way and do not provide the "same" quantity, but the gradient of their sensitivity to image noise can be observed. We are interested in metrics that are sensitive to blur.

Several sharpness metrics have been studied in the literature [Wee & Paramesran, 2007]: norm of the image gradient [Krotkov, 1988] [Subbarao & Tyan, 1998], norm of the image Laplacian [Krotkov, 1988], mutual information [Dame & Marchand, 2011], eigenvalues of the image covariance matrix [Wee & Paramesran, 2007], and other statistical functions like normalized variance, auto-correlation, or standard-deviation-based correlation [Sun & Duthaler, 2005]. Some comparisons between these metrics are proposed in Figure 4.4 and Figure 4.5.

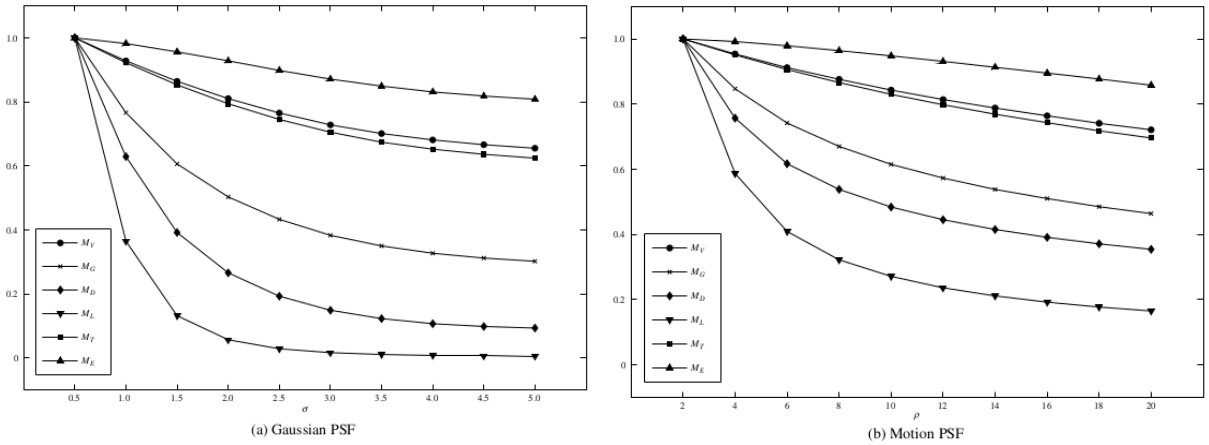


Figure 4.4 – Comparison of sharpness metrics from [Wee & Paramesran, 2007] for (a) Gaussian blur (with σ the standard deviation of the Gaussian) and (b) motion blur (with ρ the motion amplitude). The metrics studied are: M_v the gray-level variance, M_G the l_1-norm of the image gradient, M_D the l_1-norm of the second derivatives of the image, M_L energy of the Laplacian, M_T Tchebichef moments (also presented in [Wee & Paramesran, 2007]) and M_E eigenvalues.

In this study, we have selected the norm of the image gradient (see Figure 4.6), which has been shown to be sensitive to Gaussian and motion blur (see Figure 4.7) and is probably one of the simplest metrics to consider as it relies only on one-order image derivatives instead of the Laplacian one that is very promising as well. The norm of the image gradient S is traditionally expressed by:

$$S = \sum_{\mathbf{u} \in \text{ROI}} \left[\nabla I_u^2(u,v) + \nabla I_v^2(u,v) \right] \tag{4.3}$$

with ROI the region of interest considered to evaluate the metric and ∇I_u and ∇I_v are the spatial image gradient along axes u and v. The spatial image gradient can be computed with different techniques, the most common being to consider the convolution of the image with derivative filters. We choose to consider a smooth horizontal and vertical gradient by using

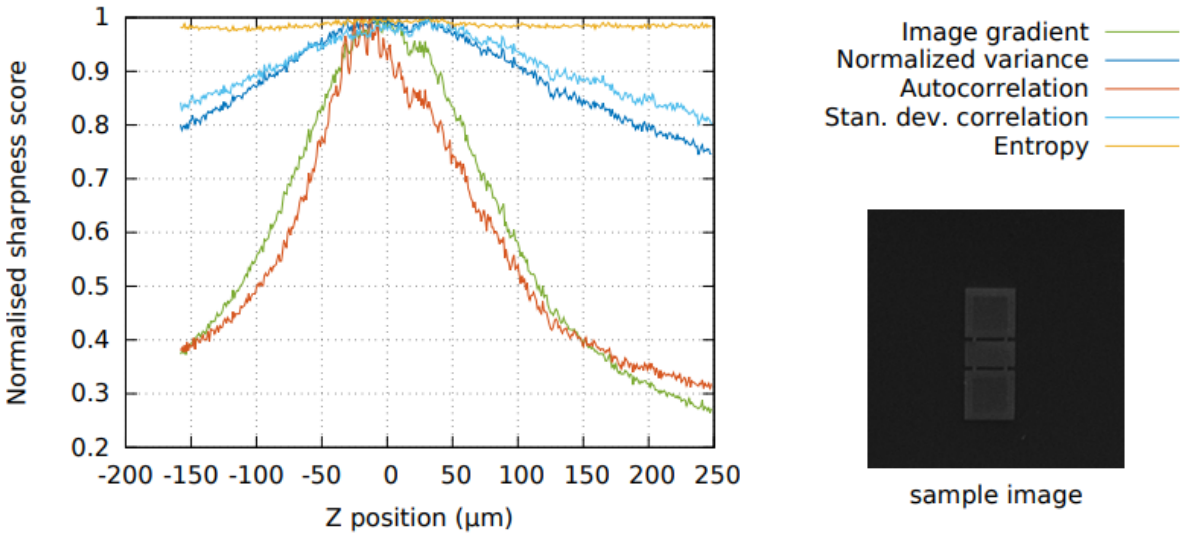


Figure 4.5 – Comparison of sharpness metrics on Gaussian blur for autofocus application with respect to the depth Z corresponding to the distance to the optimal position where the image is sharp ($Z = 0$ being the optimal position), taken from [Cui, 2016].

Gaussian derivative filters used in [É. Marchand et al., 2005], and expressed by:

$$\begin{aligned} \nabla I_u &= (G_u \circledast D_u) \circledast I \\ \nabla I_v &= (G_v \circledast D_v) \circledast I \end{aligned} \tag{4.4}$$

with G_u and G_v Gaussian filters, D_u and D_v derivative filters, respectively along axes u and v.

The norm of the image gradient has already been considered as a visual feature in visual servoing [É. Marchand, 2007][É. Marchand & Collewet, 2010], especially for autofocusing [Cui et al., 2015]. In these works, a single objective is considered that consists of maximizing the sharpness, but our control scheme also have to simultaneously consider the achievement of the primary visual task, that is, centering and tracking the target. As already mentioned, our approach is to limit motion blur during camera motion by considering it as a constraint in the control scheme.

4.2.3 Constrained control

When speaking of constrained control, we refer to the techniques used to modify the control input if they do not respect a formalized condition called constraint.

This constraint can be of many types, the most common that have been treated so far in this

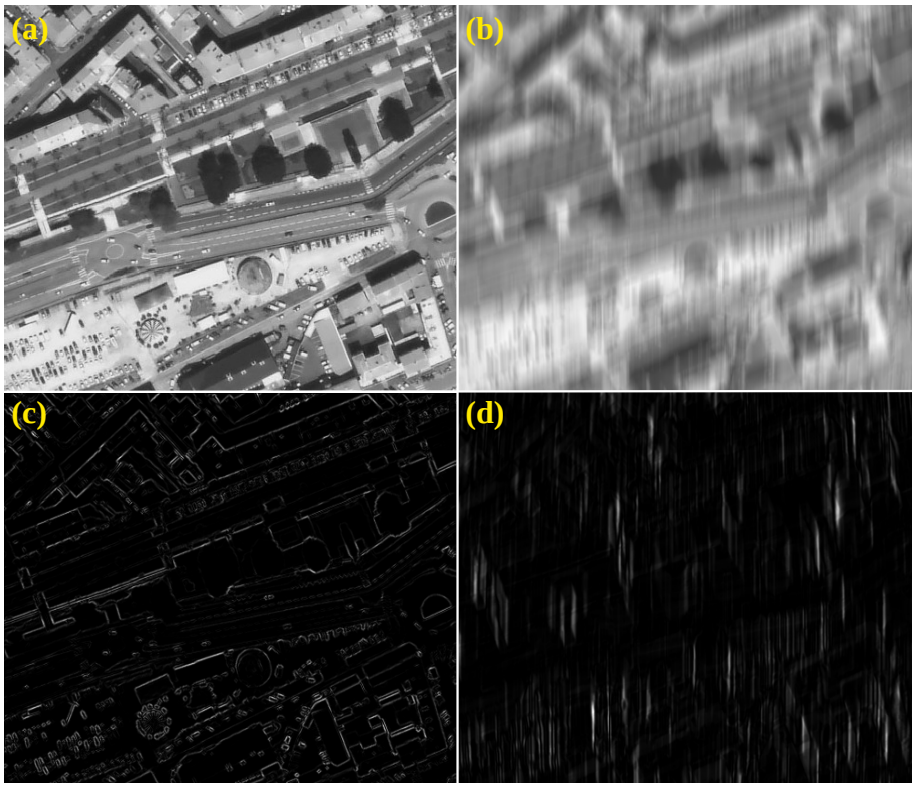


Figure 4.6 – Norm of the image gradient applied to a satellite image of Brest (a) Sharp (b) Vertical blur (c)(d) Resulting norm of the gradient. Visually, it represents the perceived edges of an image.

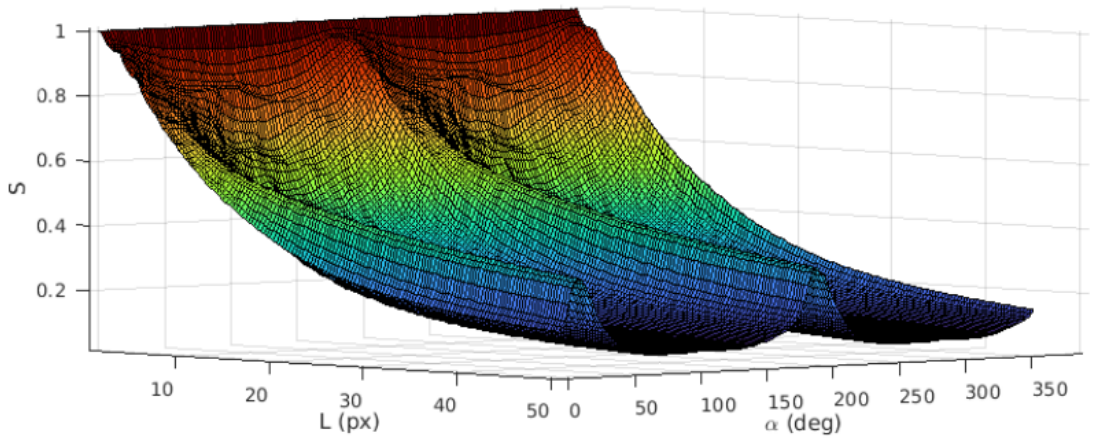


Figure 4.7 – Norm of the image gradient with respect to the motion amplitude L and to the direction of motion α. We can observe a predominance of the influence of the motion amplitude L, but still local maxima due to the orientation α.

thesis are mechanical constraints, i.e., velocity or acceleration ones, which are in general simple enough to be treated with ad hoc techniques as we did in Chapter 3. In visual servoing, more difficult constraints, such as avoiding joint limits or ensuring the visibility of the target, have to be considered with more elaborate methods such as adding features [Kermorgant & Chaumette, 2013] or using the task-priority framework in case the system has some redundancy [Kanoun et al., 2011]. On the other hand, optimal control, such as Quadratic Programming (QP) and Model-Predictive Control (MPC), is also an appealing general approach effective when multiple constraints have to be ensured simultaneously. In this chapter, we propose to see the resolution of the visual servoing scheme as an optimization problem, considering that our primary task is to focus on an object of interest, as usual, but ensuring that the constraints of our system are respected, including the motion blur-free constraint. To do so, we choose to deal with a Quadratic Programming formulation of our visual servoing control scheme. A QP can be presented in the simplest way by the following system:

$$\begin{aligned} \mathbf{v}^* = \arg\min_{\mathbf{v}} \quad & \| \mathbf{Q}\mathbf{v} - \mathbf{r} \|^2 \\ \text{subject to} \quad & \mathbf{A}\mathbf{v} = \mathbf{b} \\ & \mathbf{C}\mathbf{v} \leq \mathbf{d} \end{aligned} \tag{4.5}$$

where

— $\mathbf{v} \in \mathbb{R}^{n_d}$ is the control variable with n_d its dimension. The solution of the QP system is the optimized control variable $\mathbf{v}^*$.

— $\| \mathbf{Q}\mathbf{v} - \mathbf{r} \|^2$ is the cost function, which must be quadratic by definition. It is the quantity that we want to regulate to 0, with $\mathbf{Q} \in \mathbb{R}^{n_s \times n_f}$ and $\mathbf{r} \in \mathbb{R}^{n_s}$ matrix and vector used to define a system of n_s equations, each of them being an element of the cost function to be regulated to 0 (i.e., a cost function by itself).

— $\mathbf{A}\mathbf{v} = \mathbf{b}$ is the equality constraint system, it represents a set of n_e equality equations with $\mathbf{A} \in \mathbb{R}^{n_e \times n_f}$ the equality constraint matrix and $\mathbf{b} \in \mathbb{R}^{n_e}$ the equality constraint vector. This system must be ensured while optimizing for $\mathbf{v}$.

— $\mathbf{C}\mathbf{v} \leq \mathbf{d}$ is the inequality constraint system, it represents a set of n_c inequality equations with $\mathbf{C} \in \mathbb{R}^{n_c \times n_f}$ the inequality constraint matrix and $\mathbf{d} \in \mathbb{R}^{n_c}$ the inequality constraint vector that is represented as an upper bound in the system but can also be a lower bound if we multiply by -1 the associated line of $\mathbf{C}$ and element of $\mathbf{d}$. Once again, this system must be ensured while optimizing for $\mathbf{v}$.

This system is a standard formulation of a QP problem. We can note that the cost function can

also be written:

$$\| \mathbf{Qv} - \mathbf{r} \|^2 = \frac{1}{2}\mathbf{v}^\top \mathbf{H}\mathbf{v} - \mathbf{c}^\top \mathbf{v} \tag{4.6}$$

with $\mathbf{H} = \mathbf{Q}^\top \mathbf{Q}$ and $\mathbf{c} = \mathbf{Q}^\top \mathbf{r}$. In this formulation, $\mathbf{H}$ is the Hessian matrix of the cost function, i.e., taking $f(\mathbf{v}) = \frac{1}{2}\mathbf{v}^\top \mathbf{H}\mathbf{v} - \mathbf{c}^\top \mathbf{v}$, the second derivative of f is directly $f''(\mathbf{v}) = \mathbf{H}$. We will not go into more detail, but having $\mathbf{H}$ positive semidefinite induces a *convex* QP and if $\mathbf{H}$ is positive definite, it is a *strictly convex* QP [Nocedal & Wright, 2006]. Having a strictly convex QP guarantees the existence and unity of the solution (but not that the solution can be reached, this depends on the constraints, we speak about *feasibility*). Moreover, convex QPs are easier to solve.

One asset of quadratic programming is that there exists a huge number of algorithms to resolve the system and provide a solution in very low computational time. They are often separated into different families that basically handle constraint management with different approaches. As examples, qpOASES [Ferreau et al., 2014] (active set resolution); qpSWIFT [Pandala et al., 2019] (proximate interior point method); OSQP [Stellato et al., 2020] (Douglas-Rachford algorithm); proxQP [Bambade et al., 2022] (augmented Lagrangian), etc. In our case, the method used to solve the QP does not make a difference, as our system has low dimensions. We choose the fastest one according to benchmarks, that is, proxQP.

For our use, QP is mainly a powerful tool to represent and solve a constrained system reactively, but there is much theoretical work available to prove the feasibility of the system and the optimality of the solution [Nocedal & Wright, 2006][Nocedal & Wright, n.d.].

QP has already been applied to visual servoing [Agravante et al., 2016], but also MPC [Allibert et al., 2010][Hajiloo et al., 2015][Roque et al., 2020]. Model predictive control has, in fact, an especially high interest if the constraint is raised in a future iteration because of the evolution of the system during a horizon. The MPC is basically simulating the system thanks to an accurate model, and thus, it would optimize the whole trajectory and not only the next control step, as a simple QP would. In [Falanga et al., 2018] and [Tordesillas & How, 2022], an MPC has been designed to limit motion blur while optimizing the trajectory of a quadrotor with respect to perception features. In both papers, the control tackles motion blur by considering the reduction of the projected velocity of a landmark in the cost function. The main differences with our approach are that we explicitly consider motion blur through a sharpness metric measured in the current image, and motion blur limitation is expressed as a constraint instead of a supplementary term in the cost function, which is impossible in our case due to the lack of redundancy in pan-tilt control. According to the classic formulation of a QP, the motion blur-free constraint will need to be mapped as a constraint of our optimized variable, which is the camera velocity.

Thus, our objective is to establish a relationship between the norm of the image gradient and the angular velocity, and to constrain it.

4.3 Quadratic programming-based visual servoing

In this section, we strongly based our work on the control law designed in Section 3.2 on its pan-tilt configuration expressed in equation (3.36). Indeed, we choose here to only consider the centering task, as it is surely the main source of blur when using our control law. Thus, the design of the visual features and interaction matrices will be identical, i.e., we will still propose an image-based visual servoing. The main difference lies in the formulation of the velocity controller and the introduction of error models, based on our previous work. Additionally, we consider the low-level controller to be an integrator.

As the system has now only 2 degrees of freedom, the visual feature chosen is simply an image point $\mathbf{x} = (x, y)$ belonging to the target, typically its centroid and its desired position at the image center is denoted $\mathbf{x}^*$. Thus, the visual servoing control law aims to regulate to 0 a visual error $\mathbf{e}$ defined by:

$$\mathbf{e} = \mathbf{x} - \mathbf{x}^* \tag{4.7}$$

We recall the dynamic equation that links the variation of the visual error with the camera pan-tilt velocity $\boldsymbol{\omega}_c = (\omega_x, \omega_y)^T$:

$$\dot{\mathbf{e}} = \mathbf{L}_{\boldsymbol{\omega}} \boldsymbol{\omega}_c + \frac{\partial \mathbf{e}}{\partial t} \tag{4.8}$$

where the pan-tilt part of the interaction matrix related to $\mathbf{x}$ is now written $\mathbf{L}_{\boldsymbol{\omega}}$ for sake of clarity, and is given by equation (3.12) from Chapter 3:

$$\mathbf{L}_{\boldsymbol{\omega}} = \begin{pmatrix} xy & -(1+x^2) \\ 1+y^2 & -xy \end{pmatrix} \tag{4.9}$$

4.3.1 QP-formulation

In classical IBVS, the control scheme is directly designed through Moore-Penrose pseudo-inverse (and by a simple inverse matrix in the pan-tilt case considering only axes x and y). In QP-based approach, it is expressed as a quadratic cost function to be minimized with respect to the control variables, here $\boldsymbol{\omega}_c$. Additionally, all constraints must be expressed linearly with respect to these variables. From equation (4.8) and considering velocity, acceleration and motion

blur-free constraints, we obtain the following optimization system [Kanoun et al., 2011]:

$$\begin{aligned} \boldsymbol{\omega}_c^* = \arg\min_{\boldsymbol{\omega}_c} \quad & \| \mathbf{L}_{\boldsymbol{\omega}} \boldsymbol{\omega}_c - \dot{\mathbf{e}}^* \|^2 \\ \text{s.t.} \quad & -\boldsymbol{\omega}_{max} \leq \boldsymbol{\omega}_c \leq \boldsymbol{\omega}_{max} \\ & -\boldsymbol{\gamma}_{max} \leq \boldsymbol{\gamma}_c \leq \boldsymbol{\gamma}_{max} \\ & \mathbf{c}_S^T \, \boldsymbol{\omega}_c \geq c_s \end{aligned} \tag{4.10}$$

where

- $\dot{\mathbf{e}}^*$ is the desired evolution chosen for the visual error $\mathbf{e}$, also called the visual error model. It is described in the next section.
- $\boldsymbol{\gamma}_c$ denotes the camera acceleration while $\boldsymbol{\omega}_{max}$ and $\boldsymbol{\gamma}_{max}$ are respectively the velocity and the acceleration bounds. In order to express linearly the acceleration constraint with respect to $\boldsymbol{\omega}_c$, it is implemented by discretizing $\boldsymbol{\gamma}_c = (\boldsymbol{\omega}_c(k+1) - \boldsymbol{\omega}_c(k))/\Delta t$, $\boldsymbol{\omega}_c(k)$ being the result obtained at the previous step.
- Finally, $\mathbf{c}_S$ and c_s are respectively the sharpness constraint vector and a scalar involved for ensuring the motion blur-free constraint. They are determined in Section 4.5.

4.3.2 Visual error model

4.3.2.1 Translationless pan-tilt

In the nominal case of a static pan-tilt unit observing a motionless object, we consider the decrease of $\mathbf{e}$ such as:

$$\dot{\mathbf{e}}^* = -\lambda \mathbf{e} \tag{4.11}$$

with $\lambda > 0$ an adaptive gain defined in equation (3.23) of Chapter 3. This case will be studied as a proof-of-concept, as it is widely used in the visual servoing community.

4.3.2.2 Satellite in pan-tilt mode

In case the camera is subject to known external motions, which is our use-case since the satellite has its own orbital speed, the visual error model is refined for compensating these external motions. More precisely, for a satellite observing a static terrestrial target, we set:

$$\dot{\mathbf{e}}^* = -\lambda \mathbf{e} - \mathbf{L}_{\boldsymbol{v}} ({}^c\boldsymbol{v}_s - {}^c\boldsymbol{v}_{o_E}) \tag{4.12}$$

where ${}^c\boldsymbol{v}_s$ and ${}^c\boldsymbol{v}_{o_E}$ are respectively the translational velocity of the satellite and the translational velocity induced by Earth's rotation, both expressed in the camera frame, and defined in Chapter 3. We recall the translational part $\mathbf{L}_v$ of the interaction matrix of $\mathbf{x}$, expressed in equation (3.17):

$$\mathbf{L}_v = \begin{pmatrix} -1/Z & 0 & x/Z \\ 0 & -1/Z & y/Z \end{pmatrix} \tag{4.13}$$

with Z the depth of the target in the camera frame.

4.3.2.3 Target tracking

Finally, when the satellite observes a mobile target without any knowledge about its velocity, it is possible to estimate the effect in the image $\frac{\partial \mathbf{e}_t}{\partial t}$ due to this unknown motion and to compensate for it, leading to:

$$\dot{\mathbf{e}}^* = -\lambda \mathbf{e} - \mathbf{L}_v({}^c\boldsymbol{v}_s - {}^c\,\boldsymbol{v}_{o_E}) - \widehat{\frac{\partial \mathbf{e}_t}{\partial t}} \tag{4.14}$$

with $\widehat{\frac{\partial \mathbf{e}_t}{\partial t}}$ the estimation of $\frac{\partial \mathbf{e}_t}{\partial t}$.

Unlike in Chapter 3, we will not use an adaptive visual integrator to compensate for that motion. Indeed, we will see that precise knowledge of each motion involved in the system will be needed to design our sharpness constraint. Thus, we are interested in approaches that estimate the target motion directly in parallel of the control, contrary to the adaptive integrator that was working as a secondary task and eventually eliminating the tracking error induced by this target motion. Thus, we prefer to consider a Kalman filter to perform this task [Welch & Bishop, 1995]. To do so, we have to provide the filter with a quantity to measure, predict, and filter. From equation (1.36) of Chapter 1, we want to determine $\widehat{\frac{\partial \mathbf{e}_t}{\partial t}}$ expressed by:

$$\widehat{\frac{\partial \mathbf{e}_t}{\partial t}} = \hat{\dot{\mathbf{e}}} - \mathbf{L}_{\boldsymbol{\omega}}(k)\boldsymbol{\omega}_c(k) - \mathbf{L}_v(k)({}^c\boldsymbol{v}_s(k) - {}^c\,\boldsymbol{v}_{o_E}(k)) \tag{4.15}$$

and discretizes the quantity $\hat{\dot{\mathbf{e}}} = [\mathbf{e}(k) - \mathbf{e}(k-1)]/\Delta t$ with $\Delta t = 1/F$, F being the frequency of the control loop, so that each term can be obtained for each iteration of the control. $\widehat{\frac{\partial \mathbf{e}_t}{\partial t}}$ is then fed to the Kalman filter, and we consider a constant velocity model.

Kalman filter with constant velocity model

The two main equations of the filter are:

$$\begin{aligned} \mathbf{S}(k+1) &= \mathbf{M}\mathbf{S}(k) + \mathbf{W}(k) \quad \text{State evolution} \\ \mathbf{X}(k) &= \mathbf{T}\mathbf{S}(k) + \mathbf{N}(k) \quad \text{Measure} \end{aligned} \tag{4.16}$$

where:

- $\mathbf{S}(k) = (\mathbf{s}(k), \dot{\mathbf{s}}(k))$ is the state of the system, the vector that we want to estimate accurately, in our case $\mathbf{s}(k) = \mathbf{e}_t(k)$ and $\dot{\mathbf{s}}(k) = \frac{\partial \mathbf{e}_t}{\partial t}(k)$, and we are interested in $\dot{\mathbf{s}}(k)$. $\mathbf{M}$ is the model of the system, for a constant velocity model we have directly $\mathbf{M} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$ and $\mathbf{W}(k)$ is a zero-mean white noise of variance $\mathbf{V}_W$.
- $\mathbf{X}(k)$ is the measurement, $\mathbf{T}$ is the type of measurement, for instance if we only measure velocity, we directly have $\mathbf{T} = (0, 1)^\top$ and $\mathbf{N}(k)$ is a zero-mean white noise of variance $\mathbf{V}_N$.

The Kalman filter proceeds in two steps, a *prediction* of the next state based on the model, and a *filtering* of the prediction based on the measurement.

Prediction step: We predict the next state $\widehat{\mathbf{S}}(k|k-1)$:

$$\begin{aligned} \widehat{\mathbf{S}}(k|k-1) &= \mathbf{M}(k-1)\widehat{\mathbf{S}}(k-1|k-1) \\ \mathbf{\Gamma}_{\tilde{\mathbf{S}}(k|k-1)} &= \mathbf{M}(k-1)\mathbf{\Gamma}_{\tilde{\mathbf{S}}(k-1|k-1)}\mathbf{M}^\top(k-1) + \mathbf{V}_W \end{aligned} \tag{4.17}$$

with $\widehat{\mathbf{S}}(k-1|k-1)$ the estimation at the last step, $\mathbf{\Gamma}_{\tilde{\mathbf{S}}(k|k-1)}$ the covariance of the prediction error and $\mathbf{\Gamma}_{\tilde{\mathbf{S}}(k-1|k-1)}$ the covariance matrix of the estimation error.

Filtering step: The prediction is filtered based on the measurement, we obtain the output of the Kalman filter $\widehat{\mathbf{S}}(k|k)$:

$$\begin{aligned} \widehat{\mathbf{S}}(k|k) &= \widehat{\mathbf{S}}(k|k-1) + \mathbf{G}(k)[\mathbf{X}(k) - \mathbf{T}\widehat{\mathbf{S}}(k|k-1)] \\ \mathbf{G}(k) &= \mathbf{\Gamma}_{\tilde{\mathbf{S}}(k|k-1)}\mathbf{T}^\top[\mathbf{T}\mathbf{\Gamma}_{\tilde{\mathbf{S}}(k|k-1)}\mathbf{T}^\top + \mathbf{V}_N] \\ \mathbf{\Gamma}_{\tilde{\mathbf{S}}(k|k)} &= (\mathbf{I} - \mathbf{G}(k)\mathbf{T}(k))\mathbf{\Gamma}_{\tilde{\mathbf{S}}(k|k-1)} \end{aligned} \tag{4.18}$$

with $\mathbf{G}(k)$ the gain of the filter, which is computed thanks to the covariance matrix, and then the covariance matrix of the estimation error $\Gamma_{\tilde{\mathbf{S}}(k|k)}$ is updated.

With this last visual error model, we have determined all the configurations of the QP-based IBVS that will be experienced in this chapter. We now focus on the determination of the motion blur-free constraint, or sharpness constraint.

4.4 Motion blur metric

In this section, we once again present the metric that we have selected to measure sharpness. We recall the expression of the norm of the image gradient that we will implement and its Jacobian that will be used in the QP-controller is derived. In our application, since the satellite is able to compensate for the translational velocity of its orbit and the velocity involved by Earth's rotation, motion blur occurs in two configurations: as soon as the centering from $\mathbf{x}$ towards $\mathbf{x}^*$ starts, in which case the initial acquired image is sharp, and when the camera tracks a mobile target whose own motion is unknown, in which case the part of the image corresponding to the target is initially blurred.

4.4.1 Norm of the image gradient

As already stated in the preliminaries, the norm of the image gradient is a good indicator of the image sharpness. The norm we consider is expressed as:

$$S = \gamma \sum_{\mathbf{u} \in \text{ROI}} \left[\nabla I_u^2(u,v) + \nabla I_v^2(u,v) \right] \tag{4.19}$$

with $\gamma = \frac{1}{N}$ a normalization factor with N the number of pixels $\mathbf{u} = (u,v)$ in a region of interest (ROI) centered around $\mathbf{x}$. We recall that motion blur can be modeled as a convolution of a sharp image with a particular kernel h. More precisely, if $I^*(u,v)$ denotes the intensity of the sharp image at pixel (u,v) the intensity of the blurred image is given by:

$$I(u,v) = I^*(u,v) \circledast h \tag{4.20}$$

$$= \sum_m \sum_n I^*(u-m, v-n) h(m,n) \tag{4.21}$$

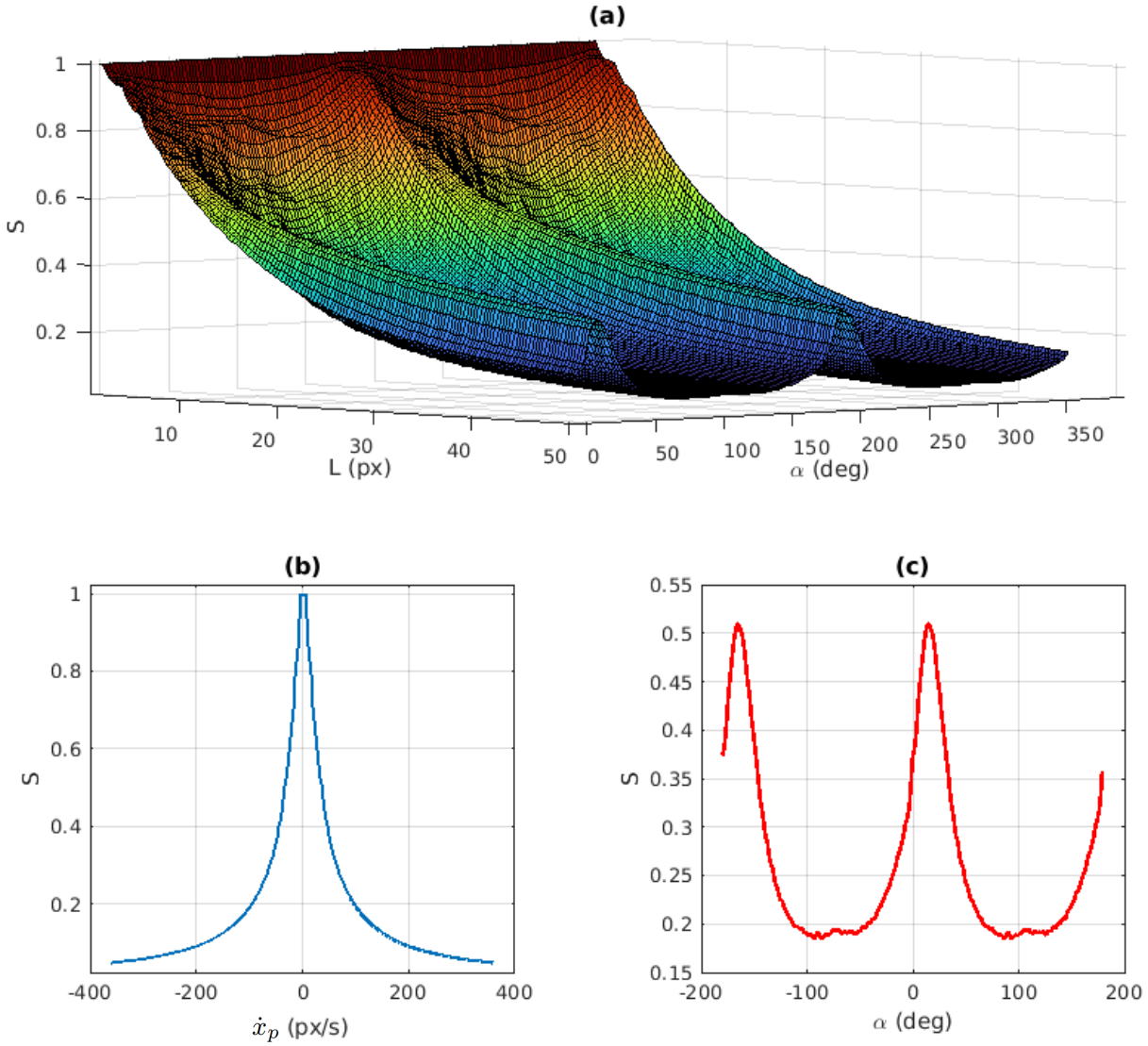


Figure 4.8 – Norm S of the image gradient: (a) S with respect to L and α, (b) S with respect to $\dot{\mathbf{x}}_p = (\dot{x}_p, 0)$px/s, (c) S with respect to α with $\dot{\mathbf{x}}_p = (100, 0)$px/s

Linear motion blur kernel

It is recalled from [Tiwari et al., 2013]:

$$h(a,b) = \begin{cases} 1/L \text{ if} \sqrt{a^2+b^2} \leq \frac{L}{2} \text{ and } \frac{b}{a} = -\tan\alpha \\ 0 \quad \text{else} \end{cases} \tag{4.22}$$

We can precise the parameters of the kernel. The motion amplitude $L = t_k||\dot{\mathbf{x}}_p||$ is the product of the integration time t_k of the camera with the norm of the projected velocity $\dot{\mathbf{x}}_p = (\dot{x}_p, \dot{y}_p)$ in the image, expressed in pixel/s, and $\alpha = \arctan(-\dot{y}_p/\dot{x}_p)$ is the direction of motion. Thanks to this kernel, the sharpness function S has been evaluated on simulated images with different camera velocities (see Figure 4.6). As can be seen on Figure 4.8b, S is maximum when $\dot{\mathbf{x}}_p \approx 0$, which means the image is indeed sharp. We also see local maxima depending on the direction of motion (Figure 4.8c), which are stronger when considering large values of motion amplitude. Still, the influence of the motion amplitude on the norm of the image gradient is clearly predominant.

As S varies according to L and α, it means that it is possible to find a correlation between the variation of S and the variation of the PSF parameters. This correlation is found by computing the Jacobian of the norm of the image gradient.

4.4.2 Jacobian of the sharpness function

We previously mentioned that it is necessary to relate the sharpness function S to the camera pan-tilt velocities to use it in our QP controller. As a first step for that, we determine in this section the Jacobian of S with respect to the parameters L and α involved in kernel h, as well as the variation of these parameters with respect to the acceleration $\ddot{\mathbf{x}}_p = (\ddot{x}_p, \ddot{y}_p)$ in the image. We have:

$$\dot{S} = \frac{\partial S}{\partial L}\dot{L} + \frac{\partial S}{\partial \alpha}\dot{\alpha} \tag{4.23}$$

and we determine each derivative in the following.

4.4.2.1 Motion amplitude

From equation (4.19), the variation of S related to L, $J_L = \frac{\partial S}{\partial L}$, is expressed by:

$$J_L = \gamma \sum_{\mathbf{u}\in \text{ROI}} 2(\nabla I_u \frac{\partial \nabla I_u}{\partial L} + \nabla I_v \frac{\partial \nabla I_v}{\partial L}) \tag{4.24}$$

This leads to computing the derivative of the image gradients with respect to L. They can be expressed from equation (4.21):

$$\begin{aligned}\frac{\partial \nabla I_u(u,v)}{\partial L} &= \nabla_u(I^*(u,v) \circledast \frac{\partial h}{\partial L}) \\ \frac{\partial \nabla I_v(u,v)}{\partial L} &= \nabla_v(I^*(u,v) \circledast \frac{\partial h}{\partial L})\end{aligned} \tag{4.25}$$

Finally, from equation (4.22), the derivative of the motion kernel with respect to L is expressed by:

$$\frac{\partial h}{\partial L} = \begin{cases} -1/L^2 & \text{if} \sqrt{a^2+b^2} \leq \frac{L}{2} \text{ and } \frac{b}{a} = -\tan\alpha \\ 0 & \text{else} \end{cases} \tag{4.26}$$

which can, in fact, be rewritten under the simple form:

$$\frac{\partial h}{\partial L} = -\frac{1}{L}h \tag{4.27}$$

Retracing the development in equation (4.25) then in equation (4.24) gives us:

$$J_L = -\frac{2\gamma}{L} \sum_{\mathbf{u}\in \text{ROI}} \left[\nabla I_u^2(u,v) + \nabla I_v^2(u,v)\right] = -2S/L \tag{4.28}$$

which is an interesting result, as we can directly compute the Jacobian of the motion amplitude with a measure of S and L.

For the variation of L, we have by definition $L = t_k||\dot{\mathbf{x}}_p||$. Since $||.||$ represents the Euclidean norm, the time-derivative of L is given by:

$$\dot{L} = t_k \frac{d||\dot{\mathbf{x}}_p||}{dt} = t_k \frac{\dot{x}_p\ddot{x}_p + \dot{y}_p\ddot{y}_p}{||\dot{\mathbf{x}}_p||} \tag{4.29}$$

This highlights acceleration terms, showing that the variation of motion blur depends on the acceleration of the projected speed, which is not a surprising result.

4.4.2.2 Direction of motion

Similarly, the variation of S related to α, $J_\alpha = \frac{\partial S}{\partial \alpha}$, has the same form as (4.24):

$$J_\alpha = \gamma \sum_{\mathbf{u}\in \text{ROI}} 2(\nabla I_u \frac{\partial \nabla I_u}{\partial \alpha} + \nabla I_v \frac{\partial \nabla I_v}{\partial \alpha}) \tag{4.30}$$

Now we need to determine the derivative of the image gradients with respect to α they are obtained from equation (4.21):

$$\begin{aligned}\frac{\partial \nabla I_u(u,v)}{\partial \alpha} &= \nabla_u(I^*(u,v) \circledast \frac{\partial h}{\partial \alpha}) \\ \frac{\partial \nabla I_v(u,v)}{\partial \alpha} &= \nabla_v(I^*(u,v) \circledast \frac{\partial h}{\partial \alpha})\end{aligned} \tag{4.31}$$

Unfortunately, the α-derivative of h can not be expressed directly since h is a piecewise function and α appears in an edge condition of the kernel. By definition, h is a two-dimensional gate function, one of which defines the length of the gate (the amplitude L) and the other its width (the orientation α). Moreover, we saw during the preliminaries that a linear motion blur PSF is represented by a multiple of point of h that represents an oriented line with L the size of the line and α its direction, it means that the width of the door is really thin, and can be seen as a sequence of Dirac impulsion along its length. Thus, we propose to approximate h by a Gaussian door function that is continuous with respect to α by considering the approximation of its width by a Gaussian approximating a Dirac impulsion. Let us call this function h_α defined by:

$$h_\alpha(a,b) = \begin{cases} \frac{1}{L} e^{-\pi(a\sin\alpha + b\cos\alpha)^2} & \text{if} \sqrt{a^2+b^2} \leq \frac{L}{2} \\ 0 & \text{else} \end{cases} \tag{4.32}$$

We can now express the derivative of the Kernel with respect to α, given by:

$$\frac{\partial h_\alpha}{\partial \alpha} = -2\pi(a\cos\alpha - b\sin\alpha)(a\sin\alpha + b\cos\alpha)h_\alpha \tag{4.33}$$

which is a more complex result, meaning that the variation of the kernel h with respect to α depends on the repartition of the points in the kernel. As for the variation of α, we obtain from its definition:

$$\dot{\alpha} = \frac{\dot{y}_p\ddot{x}_p - \dot{x}_p\ddot{y}_p}{||\dot{\mathbf{x}}_p||^2} \tag{4.34}$$

whose time-derivative contained once again acceleration terms.

4.4.2.3 Complete Jacobian

By combining the previous developments, we obtain:

$$\dot{S} = t_k J_L \frac{\dot{x}_p\ddot{x}_p + \dot{y}_p\ddot{y}_p}{||\dot{\mathbf{x}}_p||} + J_\alpha \frac{\dot{y}_p\ddot{x}_p - \dot{x}_p\ddot{y}_p}{||\dot{\mathbf{x}}_p||^2} \tag{4.35}$$

which, since $L = t_k||\dot{\mathbf{x}}_p||$, can be rewritten as:

$$\dot{S} = t_k \mathbf{J}_S \ddot{\mathbf{x}}_p \tag{4.36}$$

where the Jacobian of S wrt. $\dot{\mathbf{x}}_p$ is given by:

$$\mathbf{J}_S = \frac{\dot{\mathbf{x}}_p^T}{||\dot{\mathbf{x}}_p||} \begin{pmatrix} J_L & -J_\alpha/L \\ J_\alpha/L & J_L \end{pmatrix} \tag{4.37}$$

In (4.23), we considered the influence of both parameters L and α in the variation of S, leading to the expression of the complete Jacobian $\mathbf{J}_S$ in (4.37). As we aim to design a sharpness constraint in section 4.5, we question the impact of such a constraint on $\dot{S}$, and by transitivity, on $\dot{L}$ and $\dot{\alpha}$.

A constraint on $\dot{L}$ would induce a slowdown in the image trajectory linearly to J_L which is calculated directly from the equation (4.28). Imposing a constraint on $\dot{\alpha}$ would change the direction of motion and consequently alter the nature of the image trajectory. This variation is linear with respect to J_α, which, according to equation (4.31), requires that the sharp image I^* is properly computed. Moreover, J_α depends on the pixel distribution in the image, thus, the behavior of the control cannot be reliably predicted. In the presence of complex images with multiple local maxima of S with respect to α, oscillations in the trajectory may occur.

In our case, the centering task is modeled with an exponential decay of the visual error vector that induces a straight-line image trajectory for a static target. As satellite images are most of the time highly detailed and intricate, J_α might be noisy and induce a slight deviation from the original trajectory. Additionally, Figure 4.8 shows that the impact of motion amplitude is predominant for satellite images, and that α has only noticeable influences for large values of L that would not happen in our situation.

That is why we consider only the parameter L in the following, which is easily implemented by setting $J_\alpha = 0$ in equation (4.37).

4.5 Sharpness constraint

In this part, we aim to design a constraint expressed linearly with respect to the camera pan-tilt velocities for limiting as much as possible the motion blur.

A basic idea to achieve this goal would be to state that S should not decrease, leading to the inequality constraint $\dot{S} \geq 0$. Ensuring this condition would force the control to improve the

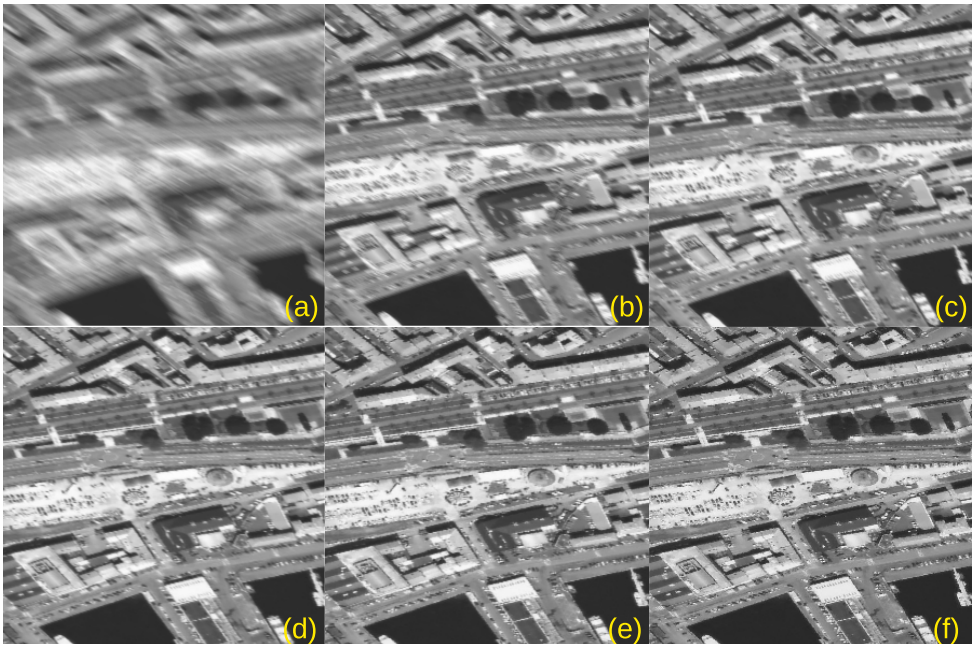


Figure 4.9 – Normalized sharpness S/S^* relatively to a sharp image: (a) 0.2, (b) 0.6, (c) 0.7, (d) 0.8, (e) 0.9, (f) 1.

sharpness, or at least to maintain it steady. However, if the image is initially sharp, camera pan-tilt motions to center the target will require decreasing the sharpness, which is not compatible with the above constraint. A better strategy is to impose S to be higher than an acceptable value for maintaining a certain level of sharpness in the image. This can be expressed through the following condition:

$$S \geq \tau \, S_{max} \tag{4.38}$$

where $\tau \in [0, 1[$ is the level of desired sharpness (see Figure 4.9 for some examples) and S_{max} is the norm of the image gradients for the sharpest image available I_{max}. In case the camera observes a static target, the initial ROI is sharp, from which $I_{max} = I^*$ is directly obtained, and so is S_{max}. When tracking a mobile target, I_{max} and S_{max} are again first selected and computed from the initial ROI, but since the initial ROI is blurred and the control is aimed to decrease this blur, it is possible to update both I_{max} and S_{max} such that:

$$\text{if } S(t) \geq S_{max} \text{ then } I_{max} = I(t) \text{ and } S_{max} = S(t) \tag{4.39}$$

In order to express this constraint linearly wrt. to camera pan-tilt velocities, we propose to use

a control barrier function [Ames et al., 2019] from equation (4.38), defined by:

$$\dot{S} \geq \sigma \text{ with } \sigma = \kappa(\tau S_{max} - S) \tag{4.40}$$

where $\kappa > 0$ is a scalar gain. Ensuring this constraint induces the following behaviors on the system:

— if $S \geq \tau S_{max}$, $\sigma \leq 0$: the control is weakly constrained and allows for decreasing S up to τS_{max};

— if $S < \tau S_{max}$, $\sigma > 0$: the control is fully constrained and forced to increase S over τS_{max}.

We now have to express equation (4.40) under the form given in equation (4.10), i.e.:

$$\mathbf{c}_S^T \boldsymbol{\omega}_c \geq c_s \tag{4.41}$$

For that, by considering a forward Euler discretization of equation (4.36), taking arbitrarily $\Delta t = t_k$, we obtain:

$$\dot{S} = \mathbf{J}_S(k)(\dot{\mathbf{x}}_p(k+1) - \dot{\mathbf{x}}_p(k)) \tag{4.42}$$

The final step is to relate equation (4.42) with the pan-tilt velocities of the camera. This is done by expressing the projected velocity $\dot{\mathbf{x}}_p$ with respect to $\boldsymbol{\omega}_c$, which is now described.

4.5.1 Fixed pan-tilt observing a motionless target

In the nominal configuration, the projected speed in the image is given from (4.8):

$$\dot{\mathbf{x}}_p = p\, \mathbf{L}_\omega\, \boldsymbol{\omega}_c \tag{4.43}$$

where p is the ratio between the focal length and the size of a pixel for converting meters into pixels. We can now substitute $\dot{\mathbf{x}}_p(k+1)$ in equation (4.42) to obtain:

$$\dot{S} = \mathbf{J}_S(k)\ (p\mathbf{L}_\omega(k)\ \boldsymbol{\omega}_c(k+1) - \dot{\mathbf{x}}_p(k)) \tag{4.44}$$

The sharpness constraint expressed in equation (4.40) can thus be written as equation (4.41) with:

$$\begin{aligned} \mathbf{c}_S^T &= p\ \mathbf{J}_S(k)\ \mathbf{L}_\omega(k) \\ c_s &= \sigma + \mathbf{J}_S(k)\dot{\mathbf{x}}_p(k) \end{aligned} \tag{4.45}$$

This inequality constraint is linear through $\boldsymbol{\omega}_c(k+1)$ and can be handled by our QP scheme.

4.5.2 Predictive sharpness constraint

The main issue with equation (4.45) is that the constraint is based on current measurements. It means that if the image is sharp at instant k, the constraint is not activated for computing $\boldsymbol{\omega}_c(k+1)$, so if blur is created at instant $k+1$ due to $\boldsymbol{\omega}_c(k+1)$, the QP will consider it only at the next step. Therefore, we propose to consider a one-step predictive constraint designed as follows:

— Predict the next velocity based on inverse kinematics through the dynamic model $\widehat{\boldsymbol{\omega}_c}(k+1) = \mathbf{L}_{\boldsymbol{\omega}}^{-1}\dot{\mathbf{e}}^*$ (which will be really near for the control law obtained in equation (3.36) of Chapter 3),

— Blur the ROI I_{max} obtained in (4.39) using (4.21) according to $\widehat{\boldsymbol{\omega}_c}(k+1)$, and compute $\widehat{\mathbf{J}_S}(k+1)$,

— Fill the constraint for the QP at instant k by using:

$$\begin{aligned} \mathbf{c}_{\hat{S}}^T &= p\ \widehat{\mathbf{J}_S}(k+1)\ \mathbf{L}_{\boldsymbol{\omega}}(k) \\ c_{\hat{s}} &= \sigma + \widehat{\mathbf{J}_S}(k+1)\dot{\mathbf{x}}_p(k) \end{aligned} \tag{4.46}$$

This predictive constraint will be tested together with non-predictive ones.

4.5.3 Satellite in pan-tilt mode

In this use case, we introduce the compensation of known external motions $\boldsymbol{\upsilon}_{ext}$ (such as orbital motions). Consequently, the projected velocity in the image becomes:

$$\dot{\mathbf{x}}_p = p\left(\mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c + \mathbf{L}_{\boldsymbol{\upsilon}}\boldsymbol{\upsilon}_{ext}\right) \tag{4.47}$$

From equation (4.42) the time-derivative of the sharpness function transforms into:

$$\dot{S} = \mathbf{J}_S(k)\Big(p\mathbf{L}_{\boldsymbol{\omega}}(k)\boldsymbol{\omega}_c(k+1) + p\mathbf{L}_{\boldsymbol{\upsilon}}(k)\boldsymbol{\upsilon}_{ext}(k+1) - \dot{\mathbf{x}}_p(k)\Big) \tag{4.48}$$

The associate sharpness constraint becomes:

$$\begin{aligned} \mathbf{c}_S^T &= p\,\mathbf{J}_S(k)\mathbf{L}_{\boldsymbol{\omega}}(k) \\ c_s &= \sigma + \mathbf{J}_S(k)(\dot{\mathbf{x}}_p(k) - p\mathbf{L}_{\boldsymbol{\upsilon}}(k)\boldsymbol{\upsilon}_{ext}(k+1)) \end{aligned} \tag{4.49}$$

Finally, we can also consider a predictive sharpness constraint for the same reasons as explained before:

$$\begin{aligned} \mathbf{c}_{\hat{S}}^T &= p\,\widehat{\mathbf{J}_S}(k+1)\mathbf{L}_{\boldsymbol{\omega}}(k) \\ c_{\hat{s}} &= \sigma + \widehat{\mathbf{J}_S}(k+1)(\dot{\mathbf{x}}_p(k) - p\mathbf{L}_{\boldsymbol{v}}(k)\boldsymbol{v}_{ext}(k+1)) \end{aligned} \tag{4.50}$$

4.5.4 Satellite in pan-tilt mode and target tracking

The last use case is to consider a mobile target and to inject the estimated effect of its motion in the projected image speed:

$$\dot{\mathbf{x}}_p = p\,(\mathbf{L}_{\boldsymbol{\omega}}\boldsymbol{\omega}_c + \mathbf{L}_{\boldsymbol{v}}\boldsymbol{v}_{ext} + \widehat{\frac{\partial \mathbf{e}_t}{\partial t}}) \tag{4.51}$$

from which non-predictive and predictive constraints can be directly obtained, similarly as for the other previous use-cases:

$$\dot{S} = \mathbf{J}_S(k)\Big(p\mathbf{L}_{\boldsymbol{\omega}}(k)\boldsymbol{\omega}_c(k+1) + \mathbf{L}_{\boldsymbol{v}}(k)\boldsymbol{v}_{ext}(k+1) + \frac{\partial \mathbf{e}_t}{\partial t}(k+1) - \dot{\mathbf{x}}_p(k)\Big) \tag{4.52}$$

The associate sharpness constraint is refined including the estimation of $\frac{\partial \mathbf{e}_t}{\partial t}$:

$$\begin{aligned} \mathbf{c}_S^T &= \mathbf{J}_S(k)p\mathbf{L}_{\boldsymbol{\omega}}(k) \\ c_s &= \sigma + \mathbf{J}_S(k)(\dot{\mathbf{x}}_p(k) - p\mathbf{L}_{\boldsymbol{v}}(k)\boldsymbol{v}_{ext}(k+1) - p\widehat{\frac{\partial \mathbf{e}_t}{\partial t}}(k+1)) \end{aligned} \tag{4.53}$$

Finally, we also consider the predictive sharpness in the same idea as equation (4.46) and equation (4.50):

$$\begin{aligned} \mathbf{c}_{\hat{S}}^T &= \widehat{\mathbf{J}_S}(k+1)p\mathbf{L}_{\boldsymbol{\omega}}(k) \\ c_{\hat{s}} &= \sigma + \widehat{\mathbf{J}_S}(k+1)(\dot{\mathbf{x}}_p(k) - p\mathbf{L}_{\boldsymbol{v}}(k)\boldsymbol{v}_{ext}(k+1) - p\widehat{\frac{\partial \mathbf{e}_t}{\partial t}}(k+1)) \end{aligned} \tag{4.54}$$

Now that we have determined all the possible constraints according to the configuration adopted by the QP, we will validate our approach with test scenarios.

4.6 Simulation results

In this section, we evaluate the QP-based IBVS law with the sharpness constraint for several configurations of the system: fixed pan-tilt tracking a static object (nominal case for pan-tilt servoing), and satellite on pan-tilt mode tracking a terrestrial target, motionless or not.

The blur simulation is managed by the Matlab function *fspecial* with a C++ implementation to communicate easier with our Visual servoing framework [E. Marchand et al., 2005], and the Kalman filter with a constant velocity model is also taken from ViSP (vpKalmanFilter). The satellite path simulator is implemented as in Chapter 3, a camera is fixed to a satellite that moves in a sun-synchronous circular orbit of 500 km and acquires generated images with a 50-cm resolution (GSD) of Brest, Barcelona, and an airplane above water moving at 800 km/h, set in their real location on the Earth's surface. We also use an SSD inverse compositional template tracker [E. Marchand et al., 2005][Baker & Matthews, 2004] to track the ROI, the target point $\mathbf{x}$ is again the centroid of the ROI. For all experiments, the camera frequency F has been set to 5 Hz, which corresponds to the acquisition frequency specified by Airbus D&S , and we have selected $t_k = 1/F$ for the integration time of an image. This high value usually generates a large motion blur. Finally, the gain λ is set exactly in the same proportion as in Section 3.5 of Chapter 3.

In the following, we highlight the influences on the control of the sharpness constraint defined in (4.40) tuned with different desired level of sharpness using $\tau \in [0, 1[$ while κ is set to the highest value that does not induce any oscillatory behavior, typically $\kappa = 1$. We compare the constraint outcomes on visual error, angular velocity, sharpness metric, and image trajectory. For sake of visibility, the sharpness metric (4.19) is shown normalized on every figure, i.e., divided by the highest value S^* of S recorded during the servo to get $S/S^* \in [0, 1]$. A video illustrating the results is available here.

4.6.1 Translation-less pan-tilt tracking a static target

In the first configuration, we consider a nominal case where the pan-tilt unit is fixed (i.e., no translational motions) and has to center a static object. This classic case for a standard pan-tilt camera, yet not realistic with the control of a satellite, allows us to analyze clearly the impact of the non-predictive and the predictive sharpness constraints described in Section 4.5.1 and 4.5.2, and to conclude on the best one to use.

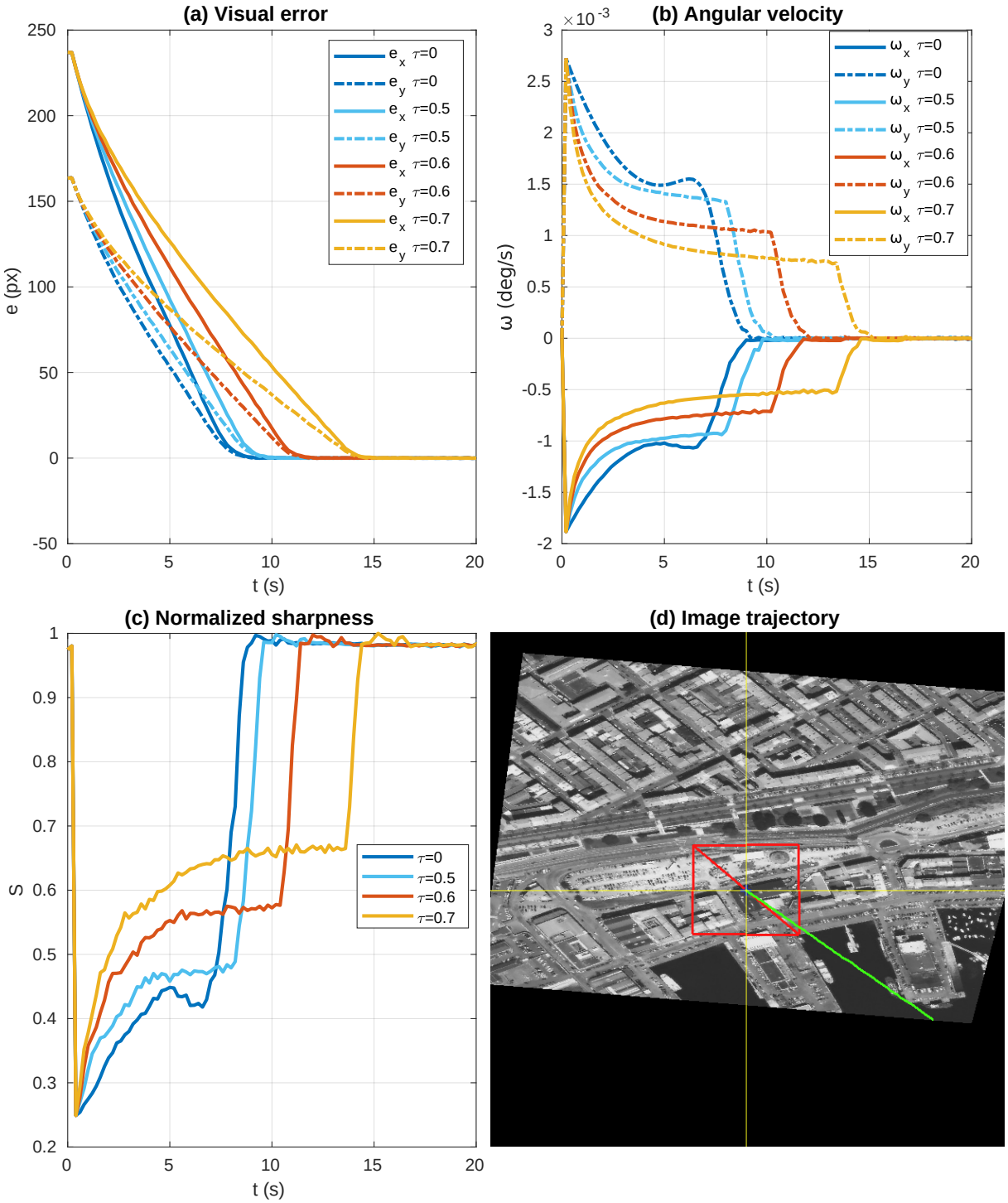


Figure 4.10 – Non-predictive constraint (4.45) applied to the fixed pan-tilt configuration tracking a location in the port of Brest, wrt. τ. $S^* = 393.2$. The image trajectory is all the time a perfect straight line.

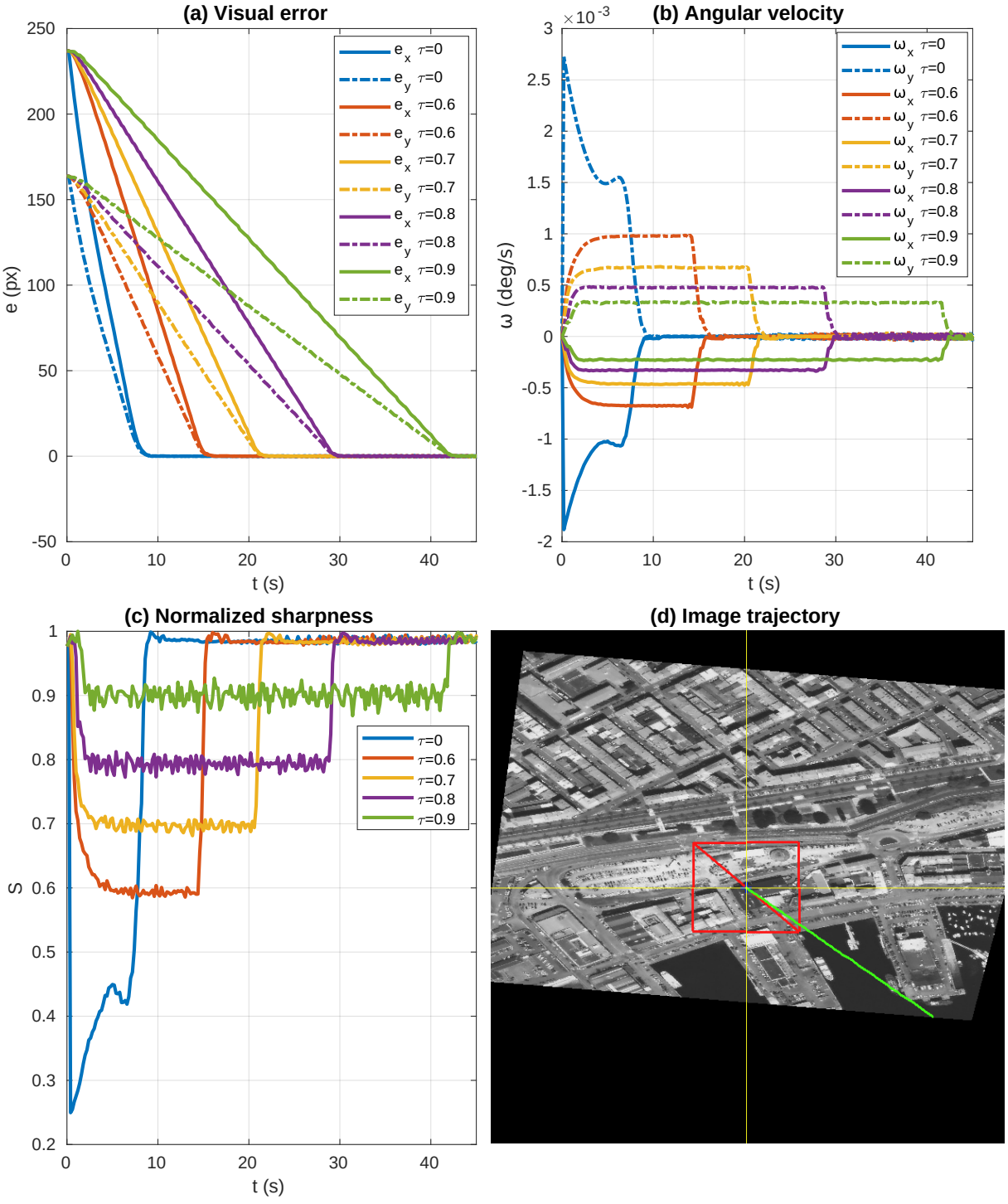


Figure 4.11 – Predictive constraint (4.46) applied to the fixed pan-tilt configuration, wrt. τ. The target is identical to Figure 4.10 to compare the impact of the prediction on the control. The trajectory is still unchanged. $S^* = 392.5$.

4.6.1.1 Non-predictive constraint

The results of using the non-predictive constraint are illustrated in Figure 4.10. Whatever the value of τ, the sharpness drops as soon as the centering task is activated (Figure 4.10c). Indeed, since the image is initially sharp, the non-predictive constraint is satisfied so the centering task is initially not constrained and generates motion blur. Just after this drop, for $\tau = 0$ that corresponds to not considering the sharpness constraint in the QP, we can note that the sharpness naturally increases due to the decrease of the angular velocity. For $\tau \neq 0$, we are in the case $S < \tau S_{max}$ and the constraint is involved for increasing as fast as possible the sharpness over τS_{max}. The velocity is slow downed at a stronger pace wrt. τ (Figure 4.10b) and the visual error converges more slowly (Figure 4.10a). Finally, the trajectory of the target in the image is a pure straight line in all cases (Figure 4.10d), and the centering task is successfully completed.

4.6.1.2 Predictive constraint

The case considering the predictive constraint is reported in Figure 4.11. Unlike the previous case, since this constraint is based on a one-step prediction, the centering task is constrained as soon as it is activated, which allows avoiding any initial large drop (apart of course when $\tau = 0$). We are now in the case $S \geq \tau S_{max}$ and the control decreases the sharpness up to τS_{max} (Figure 4.11c), which preserves much more the quality of the image, especially for $\tau = 0.9$. The angular velocity is now saturated at a value that is lower with the increase of τ (Figure 4.11b), which induces a slower convergence of the visual error (Figure 4.11a). The image trajectory of the target is still a pure straight line (Figure 4.11d), with a perfect centering of the target.

To conclude on this nominal case, both sharpness constraints improve the quality of the image compared to the unconstrained case while the target is perfectly centered. Still, the comparison between the non-predictive and the predictive constraints shows that the predictive constraint brings a far better behavior of the system. That is why only the predictive constraint will be considered in the following.

4.6.2 Satellite in pan-tilt mode

In the second configuration, the satellite is in pan-tilt mode aiming to center a selected terrestrial target, static or not. As in Chapter 3, the satellite is initially compensating for all the orbital motions thanks to the satellite inner loop control, i.e., the pan-tilt focuses on a particular terrestrial location such that the target is visible in the image before the centering task is activated.

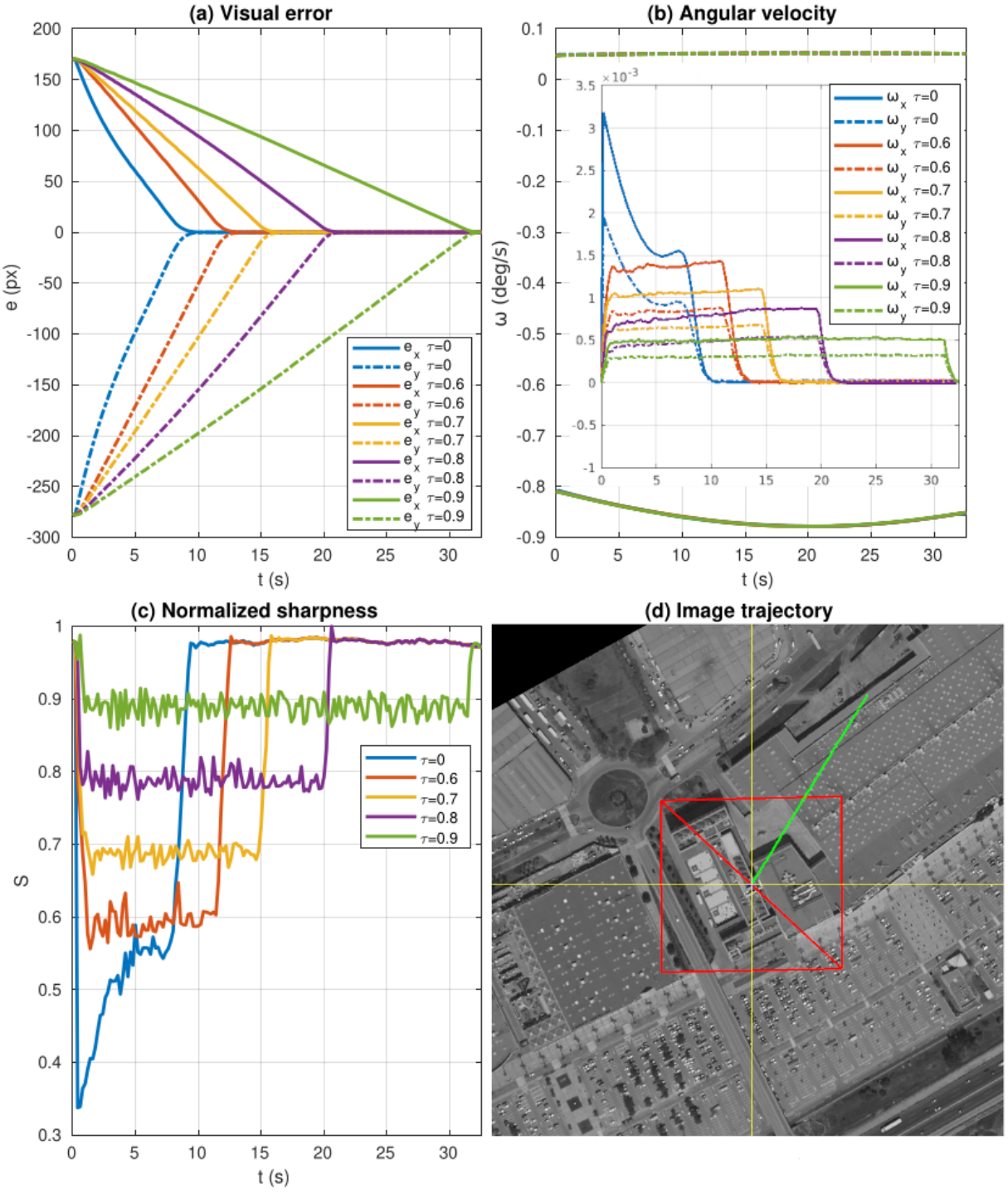


Figure 4.12 – Predictive constraint (4.50) applied to the pan-tilt satellite configuration tracking a static terrestrial object located in a commercial center of Barcelona, wrt. τ. $S^* = 131.7$. The inner figure in (b) shows the velocity without the compensation of the orbital motions. Note that the case $\tau = 0$ corresponds to the pan-tilt mode presented in equation (3.36) of Chapter 3, and similar results can be observed.

4.6.2.1 Static target

When the target is motionless, the initial ROI is sharp as in the previous case, and similar results are obtained (see Figure 4.12). Indeed, when $\tau \neq 0$, the sharpness is preserved over τS_{max} (Figure 4.12c) while the target is successfully centered with a convergence time increasing with τ (Figure 4.12a). We see in Figure 4.12b that the velocity compensating for the orbital motions is preponderant. By considering only the velocity component due to the centering task (inner figure), saturations are again observed and they are stronger wrt. τ, as in the previous case. Finally, the image trajectory of the target is still not impacted (Figure 4.12d).

4.6.2.2 Target with an unknown motion

In this last situation, the target has a large own motion (800 km/h) that is blurring the ROI at the beginning as seen in Figure 4.13d. The estimation of the corresponding motion $\widehat{\frac{\partial \mathbf{e}_t}{\partial t}}$ in the image is handled by a Kalman filter with a constant velocity model. As a sharp image is initially not available, S_{max} is updated during the control according to (4.39). As a result, the sharpness threshold τS_{max} has initially a low value that will be improved by the natural increase of the sharpness due to the compensation of the target's own motion and the reduction of the visual error. In this case, we set $\tau = 0.9$, the best level of sharpness obtained so far. The results are depicted in Figure 4.13. First, the predictive constraint allows tracking and centering the airplane, while the non-consideration of the sharpness constraint leads to an immediate failure of the template tracking due to blur accumulation. Indeed, the predictive constraint favors an increase in sharpness (Figure 4.13c) induced by a larger influence of the compensation of the airplane's motion (Figure 4.13b) compared to the centering task at the beginning (Figure 4.13b inner window). This effect is also observed in Figure 4.13d, the centering tasks starts to bend the image trajectory only from $2s$ and its impact becomes preponderant at $4s$ when the aircraft's motion is fully compensated. Lately, the centering is perfectly accomplished (Figure 4.13a) and the maximum sharpness is reached accordingly (Figure 4.13c).

Although the tracking and centering are perfectly executed, we do not obtain as good sharpness as in previous cases. It is explained because the threshold τS_max is very low and, in this case, it would be a better strategy to specify a threshold by hand. Yet, we do not have this knowledge so far. Another solution would be to study the influence of the parameter κ of the control barrier function defining the sharpness constraint, as it is the only parameter that we considered fixed. In this case, we still set $\tau = 0.9$, the best level of sharpness obtained so far, and we study the impact of the parameter κ on the control. The results are depicted in Figure 4.14. First,

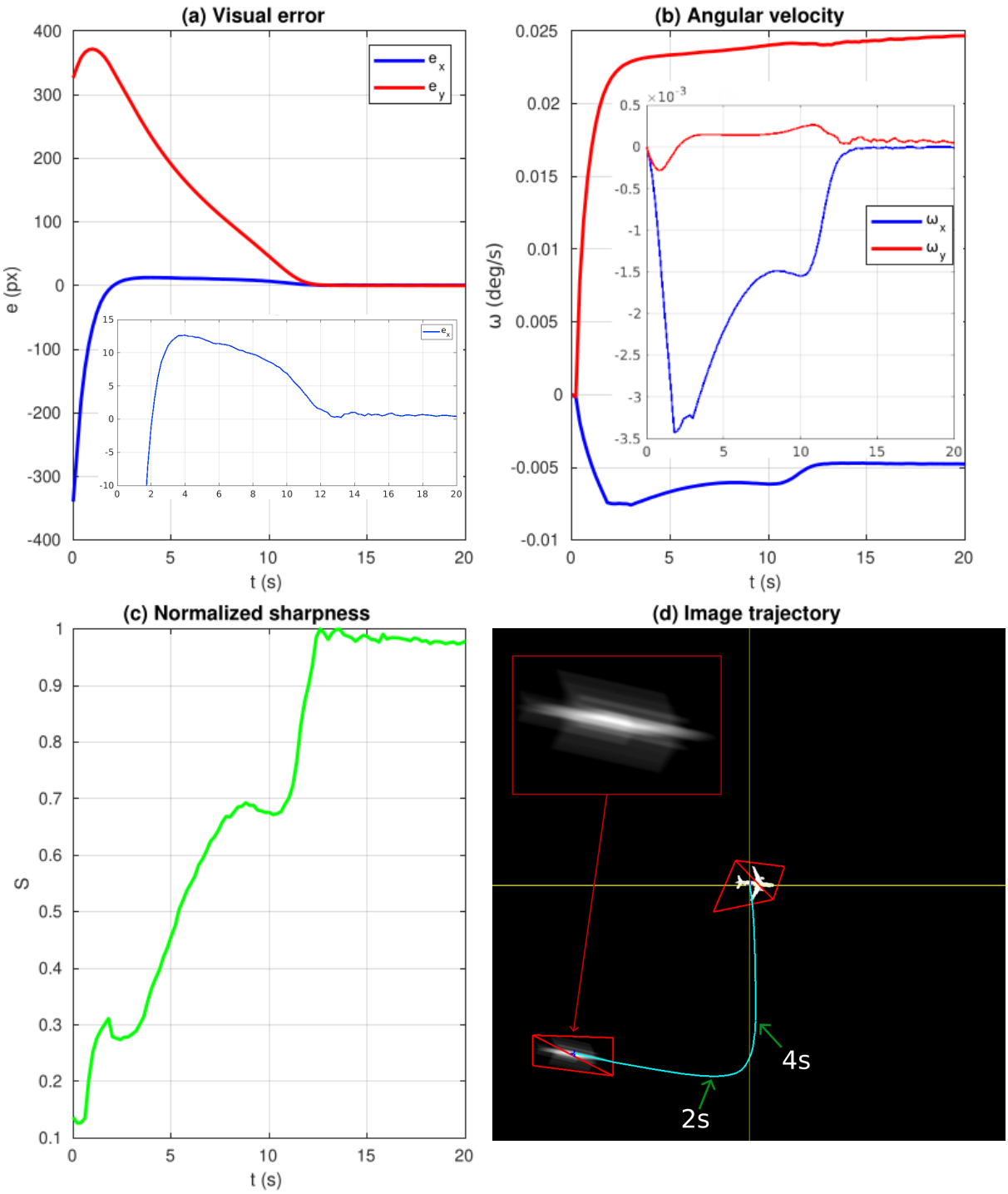


Figure 4.13 – Predictive constraint (4.54) applied to the pan-tilt satellite configuration tracking an airplane flying at 800 km/h above water. $S^* = 523.2$. Velocity in (b) are subtracted from orbital motion compensations, the inner figure also subtracts the compensation for the aiplane motion.

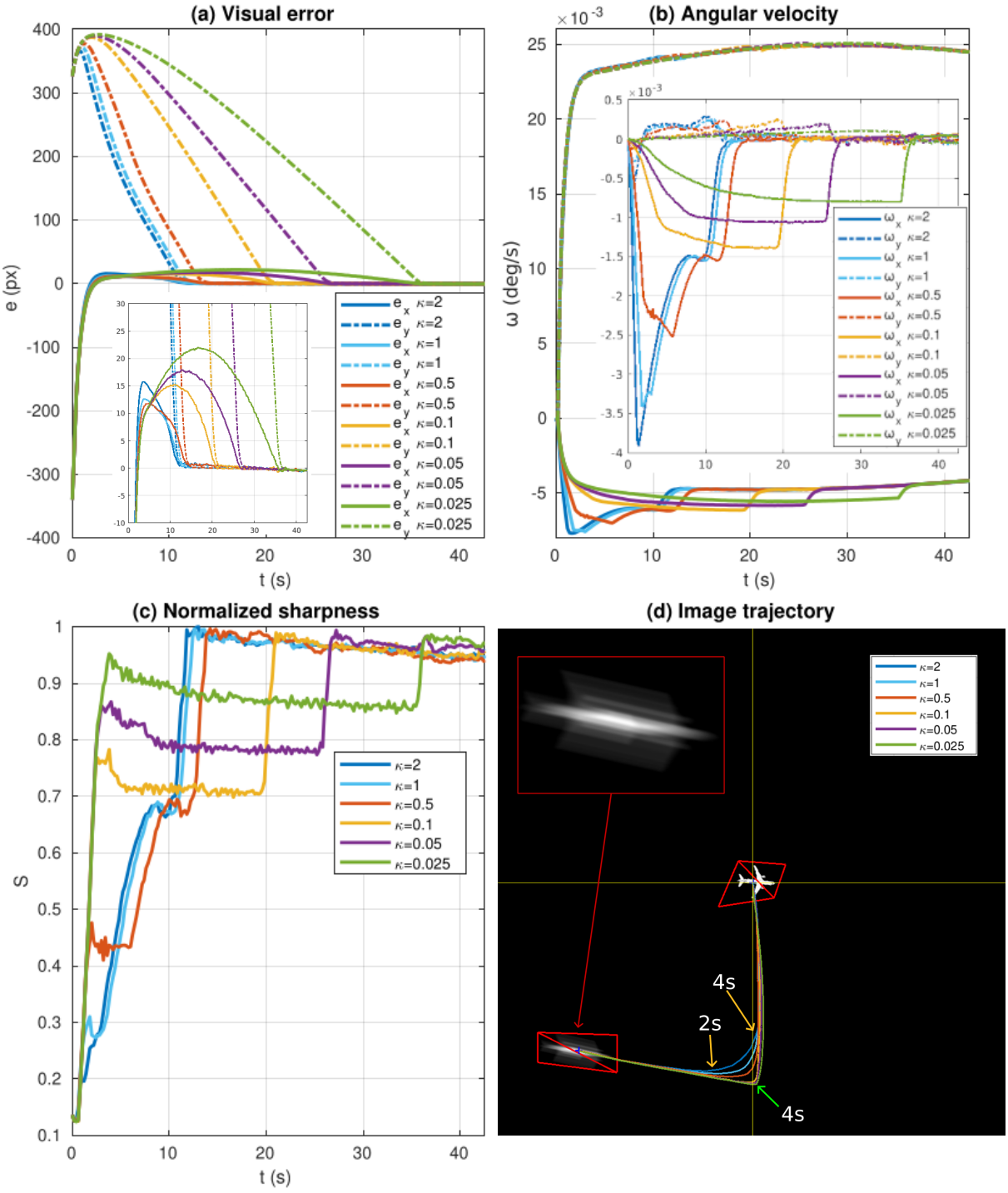


Figure 4.14 – Predictive constraint (4.54) applied to the pan-tilt satellite configuration tracking an airplane flying at 800 km/h above water wrt. τ. $S^* = 523.2$. Velocity in (b) are subtracted from orbital motion compensations, the inner figure also subtracts the compensation for the aiplane motion. We can observe the influence of the parameter κ on the trajectory, leading to a decoupling of the compensation of the aircraft motion and of its centering.

once again, regardless the value of κ, the predictive constraint allows tracking and centering the airplane (Figure 4.14d). Then, in detail, the lower the gain κ, the more the predictive constraint favors the increase in sharpness due to the compensation of the airplane's motion which is translated by a stronger slowdown of the centering task, it is observed in Figure 4.14b inner window with a lower peak of velocity at the beginning for lower κ. This effect is also observed in the image trajectory (Figure 4.14d) where the centering task bends the trajectory later with weaker κ. Indeed, the influence of the centering task becomes preponderant at these inflection points that occur at approximately 2 s for $\kappa \geq 0.5$ and 4 s $\kappa \leq 0.1$. At this very moment, if the sharpness S has not reached a higher threshold than what would naturally be achieved through the reduction of the visual error (Fig. 4.13a and c), then the system is weakly constrained ($\kappa \geq 0.5$) when performing the task; otherwise, it is constrained to maintain sharpness above this threshold ($\kappa \leq 0.1$) which is thus higher with lower κ, and similar results as previous configurations can be obtained. Consequently, the cases $\kappa \leq 0.1$ provide a better behaviour in terms of sharpness ($S/S^* > 0.7$) compared to $\kappa \geq 0.5$ which are converging faster but with images of worse quality. Note that, when considering $\kappa \leq 0.1$ the behavior of the control is the exact opposite as what we obtained in Chapter 3 for target tracking: the primary task becomes the compensation of the target motion as it is what improves sharpness the most, and the secondary task is the centering task that is smoothly operated after the aircraft is fixed in the image.

This use case shows that the predictive constraint allows for the tracking and centering of a static or high speed moving target on the Earth while maintaining a certain level of sharpness. With this last result, we can validate the effectiveness of our novel sharpness constraint,which has be proven to perform remarkably well in many scenarii including the ones presented previously in the thesis.

4.7 Conclusion

In this chapter, we have developed a new QP-based IBVS controller expressed under a motion blur-free constraint to operate simultaneously a centering and target tracking task in various configurations of a pan-tilt camera satellite.

We first started by investigating the way motion blur is characterized in the computer vision community, which allowed us to introduce the notion of *point spread function* and see how it is implemented, estimated, and how images are finally traditionally restored based on it. Then, we studied sharpness metrics especially those who can sense motion blur as a degradation of image sharpness, and selected the norm of the image gradient as an appropriate metric. We

ended this preliminary part by discussing constrained control and presented a nominal example of quadratic programming.

Then, we went into the core of the chapter by proposing a QP formulation of our previous image-based visual servoing scheme, and we introduced error models to be consistent with this new formulation. We proposed several configurations of this system, mainly focus in our satellite system, but also the nominal case of a fixed pan-tilt camera in order to speak for the widest applications possible and not only for satellites. After that, we determined analytically the Jacobian of the norm of the image gradient based on partial derivatives with respect to the PSF parameters, as we had established before their influences in the variation of the metric. Then, we designed our sharpness constraint through iterative reasoning and finally selected a control barrier function to express the constraint based on metric measurements as a velocity constraint to be injected into the QP. We proposed this constraint with a predictive step based on our work in Chapter 3, which strictly enhanced the behavior of the system. Finally, we validated our constrained controller on different configurations including the nominal pan-tilt camera, and our previous satellite scenarios. For each case, the control successfully completed the centering task with respect to the sharpness constraint, leading to preserve the quality of the image during an acquisition. Specifically, the sharpness constraint made possible target tracking of a high-speed object together with improvements in image quality, which we believe to be a very interesting result, either in robotics or in the astronautic fields. In the end, this approach can be easily transferred to any system using a camera, provided that visual control law can be formulated as a quadratic cost function.

With this last contribution chapter, we have proposed a complementary direction for that thesis. It is now time to conclude our entire study.

CONCLUSION

In this thesis, we presented a novel approach for the rotational control of an Earth observation satellite. We defended the idea that visual servoing is a powerful candidate to operate precise and robust pointing of an LEO satellite subject to extreme orbital motions with respect to its system constraints. We went deeper and showed that difficult or even impossible missions such as the on-board target tracking of a moving object could now be handled. Parallel to this contribution, we proposed through an analytical development a sharpness constraint to increase image quality during visual servo control, which might be applied to any system using a camera and not only video satellites.

Our contribution

In Chapter 3, we first proposed to detail the relationship between the camera velocity and the variation of the visual error vector in the context of a satellite subject to external motions induced by its orbit and the motion of the target. We demonstrated that an angular velocity controller compensating for these motions can solve the visual regulation problem and be stable in the sense of Lyapunov if the interaction matrix involved in the control scheme is well-estimated. We then designed an IBVS controller through the selection of 3 image features, allowing to solve a centering and an orientation task. We then determined the compensation terms based on orbital mechanics, and proposed to solve the tracking problem by decomposing the target motion with known displacement and residual motion, the last one being tackled by a visual integrator. We refined the control law with control skills from the literature and proposed an ad hoc technique to simply handle velocity and acceleration constraints. This control law was validated thanks to the simulation of camera views embedded on a satellite, modeled with the same laws established in Chapter 2, and promising results were shown.

Lately, we improved our system and added the inner dynamics of the satellite, which led us to consider a predictive saturation algorithm to handle mechanical and dynamical constraints of the satellite, without disturbing the reliability of the control. The final simulations were carried out with more difficult scenarios, such as visual tracking and staring of moving targets, and were replicated on our robotic platform. We showed that visual servoing performs remarkably

well in this context and that numerous applications can be envisaged and investigated.

One of these applications was to limit motion blur and ensure image quality while performing an acquisition, i.e., throughout visual control. In Chapter 4, we proposed to investigate the characteristics of motion blur through its point spread function, and we surveyed the various techniques to estimate and simulate such functions. Then, after an overview of the existing sharpness metrics, we selected the norm of the image gradient as an adequate metric because of its sensitivity to motion blur. According to our strategy to constrain control inputs with respect to sharpness, we proposed an IBVS-QP controller together with different visual error models inspired by Chapter 3 and validated its convergence with respect to various velocity constraints.

Then, we sought to express a sharpness condition as an angular velocity constraint. We started from the expression of the norm of the image gradients and obtained its Jacobian related to the projected velocity. The Jacobian was computed analytically by taking the partial derivatives of the metric with respect to the parameters of the point spread function. Then, we designed a sharpness constraint through a control barrier function, adaptable to the visual error model, and proposed with a one-step prediction to improve its behavior. Finally, we applied this novel control law to the nominal case of a pan-tilt camera, with the aim of transposing this approach to any systems using such devices, bus also to our previous satellite scenarios and succeeded in performing challenging acquisitions in potentially highly-blurred situations.

By improving the control of satellites using robotic control approaches, and proposing original contributions in the field of vision-based control starting from an astronautic problem, we believe this research track could inspire other efforts jointing both areas. In fact, further prospects can already be expressed from this topic.

Limits and perspectives

Indeed, along this great journey (which we have started for a while now), many choices were made to achieve our main objectives, and some doors have been closed or simply not opened.

For example, we only considered "classic" visual features to specify the task to be achieved, we believe more complex features could be used, either geometrical ones or also photometric, specifically if we have in advance an image of the target area, which is likely the case for imagery satellite. Related to the selection of visual features, we employed a template tracking algorithm to extract them, which was effective for satellite images and adaptable for selecting different regions of interest. However, concerning mobile target tracking, personalized techniques, such as 3D model-based tracking, would be more adequate if the target to follow is

known in advance (again, most likely the case for vehicle monitoring). Also, object detection was not addressed in this thesis, still, it represents an essential add-on for further studies. In terms of target tracking, we considered either a visual integrator or a linear Kalman filter with a constant-motion model. Of course, more complex motions could be better tracked by extended Kalman filters or other filtering techniques, but this does not change the way the control law is designed. Moreover, our different digital twins and mock-ups were realistic enough to validate our works, but we did not mention that, obviously, having a real satellite to perform these experiments could be the cherry on the cake.

Other improvements could be made in our study on motion blur. The sharpness metric we used was a good choice regarding its sensitivity to motion blur, still, we could not see the influence of the direction of motion for our different cases, as it becomes noticeable for larger motion blur than what we experienced in our scenarios. This parameter could be evaluated more accurately using other metrics, which would require a dedicated study but a similar approach in the computation of the Jacobian. The control barrier function was a promising proposal for the design of the sharpness constraint, but its principal interest lies in having a known desired level of sharpness. Indeed, for an initially blurred object, where this threshold is not necessarily determined, more adequate strategies could be proposed, for instance, aiming to first totally deblur the target before centering it. Lately, the one-step prediction we employed to create the predictive constraint yielded appealing results that support the idea of attempting a deeper predictive approach, such as model predictive control. This would enable us to optimize the reduction of motion blur throughout the entire trajectory, and not only the next control input.

We can use the advances we have made to address certain research prospects.

A complete constellation for orbital monitoring

No need to mention that one of the direct applications of this thesis is to track mobile objects from a satellite, and every approach we used for visual tracking allowed us to measure the motion of the object and eventually predict its short-term trajectory. However, this can be done only in time windows of one to two minutes at most for one satellite. One evident perspective is thus to increase the number of satellites with almost the same orbital parameters and consider a constellation of tracking satellites. This way, through the communication of the trajectory of a moving object from a first satellite, a second one could anticipate the presence of the target and thus increase the duration of tracking. Still, this application can be relevant for terrestrial targets as long as we consider a consequent number of satellites.

On the other hand, this constellation could also be oriented to track and estimate the motion

of space objects. Indeed, many satellites are inactive (more than 5,000 satellites in LEO), and many space programs are now focusing on de-orbiting these space junk. But before operating a rendezvous, these orbital objects need to be precisely localized, since their orbital parameters have changed from the day they were set into orbit due to orbital perturbations. Of course, these dead satellites can be roughly localized from Earth with telemetry measurements to update their orbital parameters, but this is a tricky task. In addition, a constellation of orbital monitoring satellites could orient their cameras in advance by compensating for part of the orbital motion of the target satellite, based on these approximated parameters, and reduce the relative velocity between them. Once the satellite can be observed (if it happens to be!) it can be tracked by visual servoing, and its velocity can be estimated. The operation is repeated for each satellite available in the constellation, and each observation provides a better approximation of the orbital parameters of the space target.

Yet, one of the biggest issues is to perfectly automatically detect and track a satellite from a flow of images with all the space constraints (occlusion, variable lighting..), but we believe ongoing works on the subject will soon propose relevant solutions, and during the meantime adequate hypotheses can be taken such as the ones presented previously in the literature. Looking further ahead, it would be really appealing to consider this issue but with a fleet of satellites, located close to each other, using coordinated control taken from multi-robot theory, and increasing the number of points of view of an orbital object, so that a satellite that lost the target could still have the opportunity to track it later thanks to distributed control from another satellite. Visual servoing is, again, a credible candidate for performing these tasks.

To end this manuscript, we believe to have treated an interesting subject of research and we hope to have convinced other scientists to launch their camera into orbit.

BIBLIOGRAPHY

Abdul Hafez, A., Mithun, P., Anurag, V., Shah, S., & Madhava Krishna, K., (2017), Reactionless visual servoing of a multi-arm space robot combined with other manipulation tasks, *Robotics and Autonomous Systems*, *91*.

Agravante, D., Claudio, G., Spindler, F., & Chaumette, F., (2016), Visual servoing in an optimization framework for the whole-body control of humanoid robots, *IEEE Robotics and Automation Letters*, *2*(2), 608–615.

Aharchi, M., & Ait Kbir, M., (2020), A review on 3d reconstruction techniques from 2d images, *Innovations in Smart Cities Applications Edition 3: The Proceedings of the 4th International Conference on Smart City Applications 4*, 510–522.

Alepuz, J., Emami, M. R., & Pomares, J., (2016), Direct image-based visual servoing of free-floating space manipulators, *Aerospace Science and Technology*, *55*.

Allen, P., Timcenko, A., Yoshimi, B., & Michelman, P., (1993), Automated tracking and grasping of a moving object with a robotic hand-eye system, *IEEE Transactions on Robotics and Automation*, *9*(2), 152–165.

Allibert, G., Courtial, E., & Chaumette, F., (2010), Predictive control for constrained image-based visual servoing, *IEEE Transactions on Robotics*, *26*(5), 933–939.

Ames, A., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P., (2019), Control barrier functions: theory and applications, *2019 18th European control conference (ECC)*, 3420–3431.

Arulampalam, M., Maskell, N., S.and Gordon, & Clapp, T., (2002), A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *IEEE Transactions on signal processing*, *50*(2), 174–188.

Baker, S., & Matthews, I., (2004), Lucas-kanade 20 years on: a unifying framework, *International journal of computer vision*, *56*, 221–255.

Bambade, A., El-Kazdadi, S., Taylor, A., & Carpentier, J., (2022), Prox-QP: yet another quadratic programming solver for robotics and beyond, *RSS 2022-Robotics: Science and Systems*.

Bensalah, F., & Chaumette, F., (1995), Compensation of abrupt motion changes in target tracking by visual servoing, *Proceedings 1995 IEEE/RSJ International Conference on Intel-*

ligent Robots and Systems. Human Robot Interaction and Cooperative Robots, *1*, 181–187.

Bernard, M., Decluseau, D., Gabet, L., & Nonin, P., (2012), 3d capabilities of pleiades satellite, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci*, *39*, B3.

Chaumette, F., (2004), Image moments: a general and useful set of features for visual servoing, *IEEE Transactions on Robotics*, *20*(4), 713–723.

Chaumette, F., & Hutchinson, S., (2008), Visual servoing and visual tracking, *In* B. Siciliano & O. Khatib (Eds.), *Handbook of robotics* (pp. 563–583), Springer.

Chaumette, F., Rives, P., & Espiau, B., (1991), Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing., *ICRA*, 2248–2253.

Chaumette, F., Rives, P., & Espiau, B., (1993), Classification and realization of the different vision-based tasks. *In Visual servoing: real-time control of robot manipulators based on visual sensory feedback* (pp. 199–228), World Scientific.

Chaumette, F., & Santos, A., (1993), Tracking a moving object by visual servoing, *IFAC Proceedings Volumes*, *26*(2), 643–648.

Chen, C., Chen, S., Hu, G., Chen, B., Chen, P., & Su, K., (2021), An auto-landing strategy based on pan-tilt based visual servoing for unmanned aerial vehicle in gnss-denied environments, *Aerospace Science and Technology*, *116*, 106891.

Chen, S., & Williams, L., (2023), View interpolation for image synthesis. *In Seminal graphics papers: pushing the boundaries, volume 2* (pp. 423–432).

Chen, S., (2011), Kalman filter for robot vision: a survey, *IEEE Transactions on industrial electronics*, *59*(11), 4409–4420.

Chen, X., Steyn, W., & Hashida, Y., (2000), Ground-target tracking control of earth-pointing satellites, *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 4547.

Cho, S., & Lee, S., (2009), Fast motion deblurring. *In Acm siggraph asia 2009 papers* (pp. 1–8).

Cho, S., Matsushita, Y., & Lee, S., (2007), Removing non-uniform motion blur from images, *2007 IEEE 11th International Conference on Computer Vision*.

Collewet, C., & Marchand, E., (2011), Photometric visual servoing, *IEEE Transactions on Robotics*, *27*(4), 828–834.

Corke, P., (1993), Visual control of robot manipulators–a review, *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, 1–31.

Crétual, A., & Chaumette, F., (2000), Dynamic stabilization of a pan and tilt camera for submarine image visualization, *Computer Vision and Image Understanding*, *79*(1), 47–65.

Cui, L., (2016), *Robust micro/nano-positioning by visual servoing* [Doctoral dissertation, Université Rennes 1].

Cui, L., Marchand, E., Haliyo, S., & Régnier, S., (2015), Hybrid automatic visual servoing scheme using defocus information for 6-dof micropositioning, *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 6025–6030.

Curtis, H., (2013), *Orbital mechanics for engineering students*, Butterworth-Heinemann.

Dai, S., & Wu, Y., (2008), Motion from blur, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

Dame, A., & Marchand, É., (2011), Mutual information-based visual servoing, *IEEE Transactions on Robotics*, *27*(5), 958–969.

Dauner, J., Elsner, L., Ruf, O., Borrmann, D., Scharnagl, J., & Schilling, K., (2023), Visual servoing for coordinated precise attitude control in the tom small satellite formation, *Acta Astronautica*, *202*, 760–771.

Elbeltagy, A., Youssef, A., Bayoumy, A., & Elhalwagy, Y., (2018), Fixed ground-target tracking control of satellites using a nonlinear model predictive control.

Espiau, B., Chaumette, F., & Rives, P., (1992), A new approach to visual servoing in robotics, *ieee Transactions on Robotics and Automation*, *8*(3), 313–326.

Falanga, D., Foehn, P., Lu, P., & Scaramuzza, D., (2018), PAMPC: perception-aware model predictive control for quadrotors, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1–8.

Ferreau, H., Kirches, C., Potschka, A., Bock, H., & Diehl, M., (2014), qpOASES: a parametric active-set algorithm for quadratic programming, *Mathematical Programming Computation*, *6*, 327–363.

Gleyzes, M. A., Perret, L., & Kubik, P., (2012), Pleiades system architecture and main performances, *Int. Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *39*(1), 537–542.

Hajiloo, A., Keshmiri, M., Xie, W., & Wang, T., (2015), Robust online model predictive control for a constrained image-based visual servoing, *IEEE Transactions on Industrial Electronics*, *63*(4), 2242–2250.

Hartley, R., & Zisserman, A., (2003), *Multiple view geometry in computer vision*, Cambridge university press.

Hashimoto, K., & Kimura, H., (1995), Visual servoing with nonlinear observer, *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, *1*, 484–489.

Inaba, N., Oda, M., & Hayashi, M., (2003), Visual servoing of space robot for autonomous satellite capture, *Transactions of the Japan Society for Aeronautical and Space Sciences*, *46*(153), 173–179.

Ismail, Z., & Varatharajoo, R., (2010), A study of reaction wheel configurations for a 3-axis satellite attitude control, *Advances in Space Research*, *45*(6), 750–759.

Jain, R., Kasturi, R., & Schunck, B., (1995), *Machine vision* (Vol. 5), McGraw-hill New York.

Jasiobedzki, P., Greenspan, M., Roth, G., Ng, H., & Witcomb, N., (2002), Video-based system for satellite proximity operations, *7th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2002), ESTEC, Noordwijk, The Netherlands*.

Ji, H., & Liu, C., (2008), Motion blur identification from image gradients, *2008 IEEE Conference on Computer Vision and Pattern Recognition*.

Kang, S., (1998), Survey of image-based rendering techniques, *Videometrics VI*, *3641*, 2–16.

Kanoun, O., Lamiraux, F., & Wieber, P., (2011), Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task, *IEEE Transactions on Robotics*, *27*(4), 785–792.

Keipour, A., Pereira, G., Bonatti, R., Garg, R., Rastogi, P., Dubey, G., & Scherer, S., (2022), Visual servoing approach to autonomous uav landing on a moving vehicle, *Sensors*, *22*(17), 6549.

Kermorgant, O., & Chaumette, F., (2013), Dealing with constraints in sensor-based robot control, *IEEE Transactions on Robotics*, *30*(1), 244–257.

Klančar, G., Blažič, S., Matko, D., & Mušič, G., (2012), Image-based attitude control of a remote sensing satellite, *Journal of intelligent & robotic systems*, *66*, 343–357.

Kristiansen, R., & Nicklasson, P., (2005), Satellite attitude control by quaternion-based backstepping, *Proceedings of the 2005, American Control Conference, 2005.*, 907–912.

Krotkov, E., (1988), Focusing, *International Journal of Computer Vision*, *1*(3), 223–237.

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., & Matas, J., (2018), Deblurgan: blind motion deblurring using conditional adversarial networks, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8183–8192.

Lappas, V., Steyn, W., & Underwood, C., (2002), Attitude control for small satellites using control moment gyros, *Acta Astronautica*, *51*(1-9), 101–111.

Liebe, C., (1995), Star trackers for attitude determination, *IEEE Aerospace and Electronic Systems Magazine*, *10*(6), 10–16.

Lim, S., Elgammal, A., & Davis, L., (2003), Image-based pan-tilt camera control in a multi-camera surveillance environment, *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings*, *1*, I–645.

Liu, C., Huang, X., & Wang, M., (2012), Target tracking for visual servoing systems based on an adaptive kalman filter, *International Journal of Advanced Robotic Systems*, *9*(4), 149.

Liu, M., & Jasiobedzki, P., (2002), Behaviour based visual servo controller for satellite capture, *Proceeding 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2002),(19–21 November 2002)*.

Liu, X., Zhou, H., Chang, Y., Xiang, X., Zhao, K., & Tang, D., (2020), Visual-based online control of gimbal on the uav for target tracking, *2020 chinese automation congress (cac)*, 5754–5759.

Ma, H., Zou, W., Zhu, Z., Zhang, C., & Kang, Z., (2020), Moving to oop: an active observation approach for a novel composite visual servoing configuration, *IEEE Transactions on Instrumentation and Measurement*, *70*, 1–15.

Malis, E., Chaumette, F., & Boudet, S., (1999), 2 1/2 d visual servoing, *IEEE Transactions on Robotics and Automation*, *15*(2), 238–250.

Marchand, É., (2007), Control camera and light source positions using image gradient information, *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 417–422.

Marchand, E., Chaumette, F., Spindler, F., Perrier, M., & Toulon, I., (2001), Controlling the manipulator of an underwater rov using a coarse calibrated pan/tilt camera, *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, *3*, 2773–2778.

Marchand, E., Spindler, F., & Chaumette, F., (2005), ViSP for visual servoing: a generic software platform with a wide class of robot control skills, *IEEE Robotics and Automation Magazine*, *12*(4), 40–52.

Marchand, É., & Collewet, C., (2010), Using image gradient as a visual feature for visual servoing, *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5687–5692.

Marchand, É., Spindler, F., & Chaumette, F., (2005), Visp for visual servoing: a generic software platform with a wide class of robot control skills, *IEEE Robotics & Automation Magazine*, *12*(4), 40–52.

McGuire, M., Hennessy, P., Bukowski, M., & Osman, B., (2012), A reconstruction filter for plausible motion blur, *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 135–142.

Mithun, P., Pandya, H., Gaud, A., Shah, S., & Krishna, K., (2018), Image based visual servoing for tumbling objects, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2901–2908.

Myers, R., Oleson, S., Curran, F., & Schneider, S., (1994), Small satellite propulsion options, *Joint Propulsion Conference*, (E-9063).

Nah, S., Hyun Kim, T., & Mu Lee, K., (2017), Deep multi-scale convolutional neural network for dynamic scene deblurring, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3883–3891.

Navarro, F. J., F. Serón, & Gutierrez, D., (2011), Motion blur rendering: state of the art, *Computer Graphics Forum*, *30*(1), 3–26.

Nayar, S., & Ben-Ezra, M., (2004), Motion-based motion deblurring, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), 689–698.

Nocedal, J., & Wright, S., (2006), Quadratic programming, *Numerical optimization*, 448–492.

Nocedal, J., & Wright, S. J., (n.d.), *Numerical optimization*, Springer.

Pandala, A. G., Ding, Y., & Park, H., (2019), Qpswift: a real-time sparse quadratic program solver for robotic applications, *IEEE Robotics and Automation Letters*, *4*(4), 3355–3362.

Petit, A., Marchand, É., & Kanani, K., (2011), Vision-based space autonomous rendezvous: a case study, *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 619–624.

Potmesil, M., & Chakravarty, I., (1983), Modeling motion blur in computer-generated images, *ACM SIGGRAPH Computer Graphics*, *17*(3), 389–399.

Ramos, J., Woodbury, T., & Hurtado, J., (2018), Vision-based tracking of non-cooperative space bodies to support active attitude control detection, *2018 AIAA SPACE and Astronautics Forum and Exposition*, 5353.

Remondino, F., & Fraser, C., (2006), Digital camera calibration methods: considerations and comparisons, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *36*(5), 266–272.

Robic, M., Fraisse, R., Lagadec, K., Marchand, É., & Chaumette, F., (2022), Visual servoing of an earth observation satellite of the lion constellation, *73rd International Astronautical Congress (IAC)*.

Robic, M., Fraisse, R., Marchand, É., & Chaumette, F., (2022), Vision-based rotational control of an agile observation satellite, *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2211–2218.

Roque, P., Bin, E., M., P., & Dimarogonas, D., (2020), Fast model predictive image-based visual servoing for quadrotors, *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7566–7572.

Senior, A., Hampapur, A., & Lu, M., (2005), Acquiring multi-scale images by pan-tilt-zoom control and automatic multi-camera calibration, *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, *1*, 433–438.

Shan, Q., Jia, J., & Agarwala, A., (2008), High-quality motion deblurring from a single image, *Acm transactions on graphics (tog)*, *27*(3), 1–10.

Sharp, C. S., Shakernia, O., & Sastry, S., (2001), A vision system for landing an unmanned aerial vehicle, *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, *2*, 1720–1727.

Soubirane, J., (2019), Shaping the future of earth observation with pléiades neo, *2019 9th International Conference on Recent Advances in Space Technologies (RAST)*, 399–401.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., & Boyd, S., (2020), OSQP: an operator splitting solver for quadratic programs, *Mathematical Programming Computation*, *12*(4), 637–672, https://doi.org/10.1007/s12532-020-00179-2

Stockman, G., & Shapiro, L., (2001), *Computer vision*, Prentice Hall PTR.

Stolle, S., & Rysdyk, R., (2003), Flight path following guidance for unmanned air vehicles with pan-tilt camera for target observation, *Digital Avionics Systems Conference, 2003. DASC'03. The 22nd*, *2*, 8–B.

Subbarao, M., & Tyan, J.-K., (1998), Selecting the optimal focus measure for autofocusing and depth-from-focus, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 864–870.

Sun, Y., & Duthaler, B., S.and Nelson, (2005), Autofocusing algorithm selection in computer microscopy, *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 70–76.

Thieuw, A., & Marcille, H., (2007), Pleiades-HR CMGs-based attitude control system design, development status and performances, *IFAC Proceedings Volumes*, *40*(7), 834–839.

Tiwari, S., Shukla, V., Singh, A., & Biradar, S., (2013), Review of motion blur estimation techniques, *Journal of Image and Graphics*, *1*(4), 176–184.

Tordesillas, J., & How, J., (2022), Panther: perception-aware trajectory planner in dynamic environments, *IEEE Access*, *10*, 22662–22677.

Tummala, A., & Dutta, A., (2017), An overview of cube-satellite propulsion technologies and trends, *Aerospace*, *4*(4), 58.

Votel, R., & Sinclair, D., (2012), Comparison of control moment gyros and reaction wheels for small earth-observing satellites.

Wang, H., Guo, D., Xu, H., Chen, W., Liu, T., & Leang, K., (2017), Eye-in-hand tracking control of a free-floating space manipulator, *IEEE Trans. on Aerospace and Electronic Systems*, *53*(4), 1855–1865.

Wang, M., Fan, C., & Song, C., (2022), Image-based visual tracking attitude control research on small video satellites for space targets, *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 174–179.

Wang, X., Li, T., Sun, S., & Corchado, J., (2017), A survey of recent advances in particle filters and remaining challenges for multitarget tracking, *Sensors*, *17*(12), 2707.

Wee, C., & Paramesran, R., (2007), Measure of image sharpness using eigenvalues, *Information Sciences*, *177*(12), 2533–2552, https://www.sciencedirect.com/science/article/pii/S002002550700014X

Welch, G., & Bishop, G., (1995), An introduction to the kalman filter.

Won, C., (1999), Comparative study of various control methods for attitude control of a leo satellite, *Aerospace science and technology*, *3*(5), 323–333.

Wu, Y., Tang, F., & Li, H., (2018), Image-based camera localization: an overview, *Visual Computing for Industry, Biomedicine, and Art*, *1*(1), 1–13.

Xu, L., & Jia, J., (2010), Two-phase kernel estimation for robust motion deblurring, *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, 157–170.

Yaqi, C., You, H., Tiantian, T., & Yu, L., (2022), A new target tracking filter based on deep learning, *Chinese Journal of Aeronautics*, *35*(5), 11–24.

Yitzhaky, Y., & Kopeika, N., (1997), Identification of blur parameters from motion blurred images, *Graphical models and image processing*, *59*(5), 310–320.

Zhao, W., Liu, H., Lewis, F., Valavanis, K., & Wang, X., (2019), Robust visual servoing control for ground target tracking of quadrotors, *IEEE Transactions on Control Systems Technology*, *28*(5), 1980–1987.

Zollhöfer, M., Stotko, P., Görlitz, A., Theobalt, C., Nießner, M., Klein, R., & Kolb, A., (2018), State of the art on 3d reconstruction with rgb-d cameras, *Computer graphics forum*, *37*(2), 625–652.

LIST OF FIGURES

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Asservissement visuel de l'orientation d'un satellite d'observation de la Terre

Mot clés : Asservissement visuel, robotique spatiale, suivi visuel

Résumé : Cette thèse vise à développer des lois de commande basées vision pour un satellite d'observation en basse orbite capable de s'orienter sur ses trois degrés de liberté. En effet, les satellites d'observation de la Terre doivent pointer précisément leur capteur vers la surface de la Terre, et les méthodes de commande traditionnelles font face à des problèmes en raison de la vitesse orbitale du satellite et des perturbations externes devenant fortes en orbite terrestre basse. Un guidage en temps réel de l'attitude d'un satellite à partir des informations d'un capteur de vision peut offrir davantage de fonctionnalités pour améliorer la robustesse, la précision et la flexibilité du pointage, et permettre des missions plus complexes, telles que le suivi d'un objet en mouvement, ce qui restait jusqu'à présent un problème ouvert. L'asservissement visuel est une approche prometteuse dans ce contexte. Elle a en effet déjà prouvé son efficacité pour effectuer des tâches robotiques (déplacement, manipulation, observation,..), avec de récentes applications à la robotique spatiale. En formulant la mission d'un satellite comme une tâche d'asservissement visuel, il devient possible d'appliquer ces techniques pour contrôler précisément son orientation. Cette thèse propose des lois d'asservissement visuel pour résoudre les problèmes liés aux mouvements rapides du satellite, à ses contraintes mécaniques, à la poursuite de cibles mobiles, et à la qualité de l'image, en atténuant spécifiquement le flou de mouvement lors d'une acquisition.

Title: Visual Servoing of the Orientation of an Earth Observation Satellite

Keywords: Visual servoing, Space robotics, Visual tracking

Abstract: This thesis aims to develop vision-based control laws for a low orbit observation satellite capable of moving on its three rotational degrees of freedom. Indeed, Earth observation satellites are used to precisely point their sensor toward the Earth's surface, and traditional control methods face challenges due to orbital motion and external perturbations becoming intense in low Earth orbits. Real-time satellite attitude guidance through vision sensor feedback can offer more features to improve the robustness, accuracy, and flexibility of the pointing, and allows for more complex missions, such as tracking a moving target. Visual servoing, a vision-based closed-loop control approach, is appealing in this context. Visual servoing has indeed already proven its effectiveness in performing various robotic tasks (navigation, manipulation, observation..), with recent applications to space robotics. By defining the mission of an observation satellite as a visual servoing task, it becomes possible to apply these different techniques to precisely control its orientation. This thesis proposes personalized visual control laws to address issues related to fast satellite motion, its mechanical constraints, on-board visual tracking of moving targets, and image quality, specifically mitigating motion blur during an acquisition.