HAL open science

Contributions to active visual estimation and control of robotic systems

Riccardo Spica

To cite this version:

Riccardo Spica. Contributions to active visual estimation and control of robotic systems. Signal and Image Processing. Université de Rennes, 2015. English. NNT : 2015REN1S080 . tel-01254754v2

HAL Id: tel-01254754

https://theses.hal.science/tel-01254754v2

Submitted on 10 Mar 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

ANNÉE 2015

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Traitement du Signal et Télécommunications

École doctorale Matisse

présentée par

Riccardo Spica

préparée à l'unité de recherche IRISA – UMR6074
Institut de Recherche en Informatique et Systeme Aléatoires

Contributions to Active Visual Estimation and Control of Robotic Systems

Thèse soutenue à Rennes le 11/12/2015

devant le jury composé de :

Christine Chevallereau / *Président*
Directeur de Recherche CNRS, IRCCyN, Nantes

Seth Hutchinson / *Rapporteur*
Professeur, University of Illinois at Urbana-Champaign

Pascal Morin / *Rapporteur*
Professeur, Université Pierre et Marie Curie / ISIR, Paris

Giuseppe Oriolo / *Examinateur*
Professeur, Sapienza Università di Roma

François Chaumette / *Directeur de thèse*
Directeur de Recherche Inria, Irisa, Rennes

Paolo Robuffo Giordano / *Encadrant*
Chargé de Recherche CNRS, Irisa, Rennes

The purpose of an experiment
is to make Nature speak intelligently:
all that then remains is to listen

Bill Diamond (1978)

Résumé

Depuis ses origines, dans les années 1980, la robotique a été définie comme la science qui étudie la « connexion intelligente entre la perception et l'action » [SK08]. Dans cette définition, il est clair qu'il n'y a pas de poids particulier sur les composantes de perception et d'action d'un système robotique. Ce qui rend un robot (ainsi qu'un être vivant) vraiment intelligent n'est pas, en fait, l'excellence dans une de ces composantes (ou même dans les deux), mais, plutôt, le correct équilibre entre les deux. Si, d'une part, il est immédiatement intuitif que la précision de l'estimation de l'état du robot par rapport à l'environnement qui l'entoure a un fort impact sur la précision de ses actions, il ne faut pas oublier, d'autre part, qu'il a été démontré à plusieurs reprises que la perception est un processus actif aussi bien pour les êtres vivants [Gib62] que pour les systèmes robotiques [Baj88].

Comme pour de nombreux travaux en robotique, nous nous sentons partiellement obligés de « justifier » notre étude en partant d'un exemple représentatif du monde biologique. Il a été montré dans [TTAE00, Tuc00] que les rapaces approchent leur proie en suivant une trajectoire en spirale (voir Fig. 1(b)) plutôt qu'une (plus courte) ligne droite. L'explication reconnue de ce comportement, apparemment contre-productif, est que la ligne de visée de la fovéa, la région rétinienne spécialisée dans la vision aiguë, vise à environ 45 degrés à droite ou à gauche de l'axe frontal de l'oiseau, voir Fig. 1(a). En suivant une trajectoire en spirale autour de la proie, le rapace peut voir la cible dans le champ de vue de la fovéa tout en conservant la tête alignée avec le reste de son corps. Pour voler en ligne droite, le faucon serait forcé de tourner sa tête sur le côté, augmentant ainsi considérablement la résistance de l'air durant le vol ce qui réduirait sa vitesse. Cette trajectoire en spirale est donc le résultat (naturel) d'une maximisation conjointe des performances à la fois de la perception et de l'action.

Dans de nombreuses applications robotisées basées sur des capteurs extéroceptifs, l'état d'un robot par rapport à son environnement peut seulement être par-

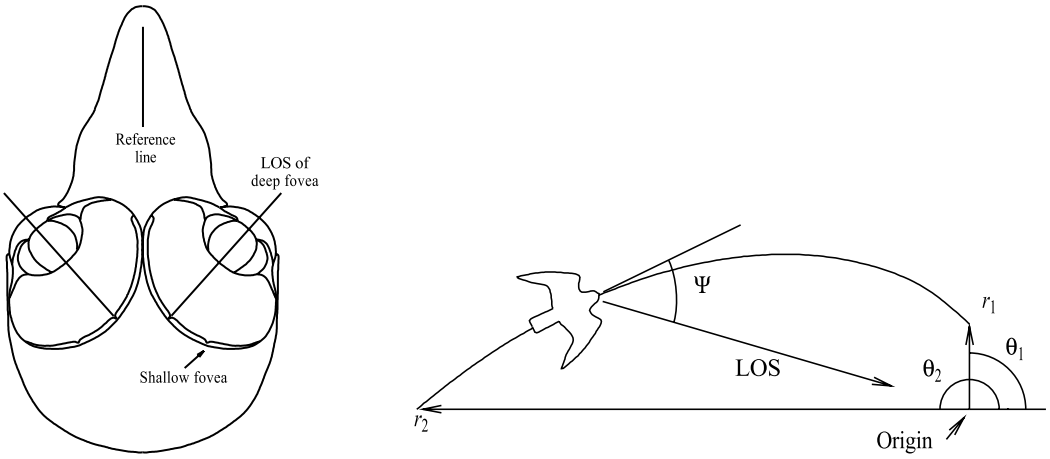


FIGURE 1 – **Optimisation de la trajectoire chez les faucons pèlerins**. Fig. (a) : structure anatomique de l'œil du faucon avec la fovéa (la zone spécialisée dans la vision aiguë) visant à environ 45 degrés par rapport à l'axe frontal, image prise de [Tuc00]. Fig. (b) : trajectoire en spirale suivie par le faucon afin d'optimiser la perception visuelle et l'aérodynamique pendant le vol vers sa proie, image prise de [TTAE00].

tiellement récupéré par ses capteurs embarqués. Dans ces situations, des schémas d'estimation d'état peuvent être exploités pour récupérer en ligne les « informations manquantes » et les fournir à n'importe quel planificateur/contrôleur de mouvement, à la place des états actuels non mesurables. Quand on considère des cas non-triviaux, cependant, l'estimation de l'état doit souvent faire face aux relations non linéaires entre l'environnement et l'espace des capteurs (la projection perspective effectuée par les caméras étant un exemple classique dans ce contexte [MSKS03]). En raison de ces non-linéarités, la convergence et la précision de l'estimation peuvent être fortement affectées par la trajectoire particulière suivie par le robot/capteur qui, au sens large, doit garantir un niveau suffisant d'excitation pendant le mouvement [CM10, AWCS13]. Par exemple, dans le contexte de la reconstruction 3-D à partir de la vision ("Structure from Motion – SfM"), un mauvais choix des entrées du système (par exemple, la vitesse de translation de la caméra) peut rendre la structure 3-D de la scène non observable quelle que soit la stratégie d'estimation utilisée [Mar12, EMMH13, GBR13], ce qui conduit, dans la pratique, à une estimation inexacte de l'état pour des trajectoires avec un faible contenu d'information. Ceci, à son tour, peut dégrader les performances de tout planificateur/contrôleur qui doit générer des actions en fonction de l'état reconstruit, conduisant éventuellement à des échecs ou des instabilités en cas de trop grandes erreurs d'estimation [DOR08, MMR10]. La dépendance entre la performance d'estimation de la trajectoire du robot et la performance du contrôle de la précision d'estimation, crée clairement une relation étroite entre perception et action : la perception doit être optimisée pour améliorer la performance de l'action, et les actions choisies doivent

permettre la maximisation des informations recueillies pendant le mouvement pour faciliter la tâche d'estimation [VTS12].

Sujet de la thèse

Avec ces considérations en tête, nous adoptons dans cette thèse la définition classique de la robotique et nous analysons la relation entre perception et action dans les systèmes robotiques. Étant donné l'ampleur du sujet, nous nous concentrons sur le contexte de l'estimation et de la commande basées sur la vision et, en particulier, sur la classe des shémas de asservissement visuel basé-image (IBVS) [CH06]. En effet, en plus d'être une technique basée sur des capteurs très répandus, voir par exemple [TC05, GH07, MS12], l'IBVS est aussi un bon exemple de tous les problèmes mentionnés précédemment : d'une part, quel que soit l'ensemble choisi des primitives visuelles (par exemple, des points, des droites, des plans et ainsi de suite), la matrice d'interaction associée dépend toujours d'informations 3-D supplémentaires qui ne sont pas directement mesurables à partir d'une image (par exemple, la profondeur d'un point caractéristique ou le rayon d'une sphère). Cette information 3-D doit alors être approximée ou estimée en ligne par l'intermédiaire d'un algorithme de Structure from Motion (SFM), et une connaissance imprécise (en raison, par exemple, de mauvaises approximations ou d'une mauvaise performance du SfM) peut dégrader l'asservissement et même conduire à des instabilités ou à l'échec du suivi de la primitive dans l'image [MMR10]. Par contre, la performance du SfM est directement affectée par la trajectoire particulière suivie par la caméra lors de l'asservissement [Mar12] : le contrôleur IBVS doit alors être en mesure de réaliser la tâche visuelle principale et, en même temps, d'assurer un niveau suffisant de gain d'information pour permettre une estimation précise de l'état.

À cet égard, nous proposons une stratégie d'optimisation de trajectoire en ligne qui permet de maximiser le taux de convergence d'un estimateur SfM, compte tenu de certaines limites sur la norme maximum acceptable de la vitesse de translation de la caméra. Nous montrons aussi comment cette technique peut être associée avec l'exécution simultanée d'une tâche IBVS en utilisant des techniques de résolution de redondance appropriées. Tous les résultats théoriques présentés dans cette thèse sont validés par une vaste campagne expérimentale qui, à notre avis, supporte pleinement nos affirmations.

Structure de la thèse

Le corps de cette thèse est divisé en trois parties principales. La première présente un examen et un état de l'art des principales techniques utilisées en vision par or-

dinateur, commande de robot, estimation d'état et perception active. Les deuxième et troisième parties présentent, par contre, les contributions originales de ce travail dans le contexte de l'estimation 3-D active et de son couplage avec la réalisation d'une tache d'asservissement. Dans ce qui suit, nous proposons un bref résumé du contenu de chaque partie.

Aperçu de la Partie I

Dans la Partie I, nous introduisons quelques préliminaires liés à la vision par ordinateur et à la robotique et nous proposons un examen approfondi de la littérature existante. A cet égard, nous essayons de focaliser l'attention du lecteur sur l'importance d'un couplage correct entre la perception et la commande en soulignant les problèmes éventuels qui peuvent survenir lorsqu'un tel couplage n'est pas correctement assuré.

Au Chapitre 2 nous commençons par résumer le modèle de base de la formation d'une image et les relations géométriques qui existent entre deux vues de la même scène prises à partir de différents points de vue de la caméra. Le chapitre se poursuit par la présentation des techniques standard utilisées pour la modélisation de la cinématique d'un robot et pour commander son mouvement. Nous nous concentrons, en particulier, sur la classe des techniques de commande de mouvement basé sur la vision, avec un accent sur les schémas d'asservissement visuel basé image (IBVS).

Au Chapitre 3 nous passons en revue les techniques classiques utilisées dans la littérature pour estimer l'état d'un robot et/ou la structure de l'environnement à partir de l'information visuelle. Nous proposons une comparaison entre les techniques déterministes et probabilistes et, en particulier, entre les indicateurs les plus couramment utilisés dans les deux approches pour mesurer le « conditionnement » de l'estimation. Ceci nous conduit naturellement à l'introduction de la notion de perception active comme le paradigme correct à adopter pour assurer un couplage efficace entre la perception et l'action.

Aperçu de la Partie II

La Partie II de la thèse présente les contributions de ce travail dans le contexte de l'estimation 3-D active par une caméra en mouvement. Les résultats contenus dans cette partie sont les sujets des contributions originales de l'auteur publiées dans [1, 2, 3, 4, 5, 6].

Au Chapitre 4 nous proposons une caractérisation de la dynamique de l'erreur d'estimation dans la SfM. Malgré la non-linéarité du problème, on montre que la

dynamique d'erreur peut être rendue approximativement équivalente à celle d'un système linéaire du second ordre avec des pôles donnés si les gains d'estimation et la vitesse de translation de la caméra sont correctement sélectionnés en ligne en fonction des mesures visuelles courantes. Par la suite, le chapitre propose l'application de ce schéma général de perception active à un ensemble de primitives géométriques de base : points, plans et primitives cylindriques et sphériques.

Au Chapitre 5 nous présentons une vaste validation expérimentale de l'ensemble des thèses soutenues au Chapitre 4 avec un robot manipulateur équipé d'une caméra embarquée. En particulier, nous montrons que pour l'ensemble des primitives géométriques décrites dans le Chapitre 4, notre stratégie permet de contrôler correctement et de maximiser (autant que possible) le taux de convergence de l'estimateur sous certaines contraintes sur la norme maximale de la vitesse de la caméra.

Aperçu de la Partie III

La Partie III de ce travail aborde le problème plus difficle d'associer la perception active avec l'exécution d'une tâche d'asservissement visuel (IBVS). Une partie de ce travail a été présenté dans les publications de l'auteur [7, 8] et [9], qui, au moment de la rédaction, est sous évaluation en vue de publication.

Au Chapitre 6 nous proposons une stratégie intelligente pour augmenter le taux de convergence de l'estimation de la structure 3-D d'une manière qui est compatible avec l'exécution d'une tâche principale d'asservissement visuel. Nous proposons d'utiliser un opérateur de projection « large » qui permet de maximiser la redondance du robot de manière à donner plus de « liberté » à la maximisation de l'estimation active, tout en réalisant la tâche IBVS. Ensuite, nous étendons notre solution par l'introduction d'une stratégie adaptative qui permet de régler l'effet de l'estimation active et même de déclencher son activation/désactivation en fonction de l'état actuel de l'erreur d'estimation.

Au Chapitre 7 nous présentons, comme déjà fait au Chapitre 5, une vérification expérimentale et approfondie des résultats du Chapitre 6. Nous démontrons que l'optimisation du mouvement de la caméra pendant le transitoire d'un contrôleur IBVS maximise le taux de convergence d'un estimateur SfM. Vu que les quantités estimées sont utilisées pour calculer la matrice d'interaction pour la tâche IBVS considéré, cela se traduit, à son tour, par une amélioration substancielle de la performance de la commande. Enfin, nous montrons qu'en utilisant une stratégie adaptative fondée sur l'état actuel de l'estimateur, on peut également réduire l'effet négatif de déformation que la perception active a sur la trajectoire de la caméra.

Conclusions et annexes

Le Chapitre 7 conclut la description des principales contributions de ce travail.

En plus du contenu décrit jusqu'ici, la thèse contient également un chapitre conclusif et trois annexes.

Au Chapitre 8 nous fournissons un examen final d'ensemble des principaux résultats de la thèse en soulignant aussi quelques questions ouvertes qui restent encore à résoudre. Le Chapitre 8 propose aussi un certain nombre d'extensions possibles à ce travail qui mériteraient d'être étudiées. Certains d'entre elles sont en effet les sujets de l'activité de recherche actuelle de l'auteur.

Dans l'Annexe A nous incluons quelques détails techniques supplémentaires pour la dérivation de certains résultats présentés dans la thèse. Ce contenu n'est pas indispensable pour comprendre le reste de ce travail, mais il est néanmoins inclus ici pour les lecteurs intéressés.

Dans l'Annexe B nous présentons des résultats préliminaires dans le contexte de l'estimation 3-D dense en utilisant directement des informations photométriques. Dans ce cas, l'image de la caméra (perçue comme une nappe dense de niveau de luminosité) est utilisée directement pour calculer le terme d'innovation de l'observateur SfM, supprimant la nécessité de tout traitement préliminaire de l'image (par exemple l'extraction, la recherche de correspondances et le suivi des primitives).

Dans l'Annexe C nous donnons une introduction très courte et plutôt informelle au vaste sujet des systèmes port-Hamiltoniens car ils fournissent une interprétation intéressante, intuitive et physique des shémas d'estimation et d'optimisation utilisés dans cette thèse. Nous introduisons également, très brièvement, la représentation à graphe de liaisons de tels systèmes.

Abstract

As every scientist and engineer knows very well, running an experiment is a process that requires a careful and thorough planning phase. The goal of such a phase is to ensure that the experiment will actually give the scientist as much information as possible about the process that she/he is observing so as to minimize the experimental effort (in terms of, e.g., number of trials, duration of each experiment and so on) needed to reach a trustworthy conclusion.

In a similar way perception, both in a natural and in an artificial settings, is an active process in which the perceiving agent (be it a human, an animal or a robot) tries its best to maximize the amount of information acquired about the environment using its limited sensor capabilities and resources.

In many sensor-based robot applications, the state of a robot w.r.t. the environment can only be partially retrieved from his on-board sensors. In these situations, state estimation schemes can be exploited for recovering online the 'missing information' then fed to any planner/motion controller in place of the actual unmeasurable states. When considering non-trivial cases, however, state estimation must often cope with the nonlinear sensor mappings from the observed environment to the sensor space that make the estimation convergence and accuracy strongly affected by the particular trajectory followed by the robot/sensor.

In this thesis we restrict our attention to the problem of vision based robot control. In fact vision is probably the most important sensor in biological systems and endowing robots with the possibility of controlling their motion using visual information is considered to be one of the hardest, but also most promising, challenges in modern robotics. When relying on vision based control techniques, such as Image Based Visual Servoing (IBVS), some knowledge about the 3-D structure of the scene is needed for a correct execution of the task. However, this 3-D information cannot, in general, be extracted from a single camera image without additional assumptions on the scene. In these cases, one can exploit a Structure from Motion (SfM) estima-

tion process for reconstructing this missing 3-D information. However performance of any SfM estimator is known to be highly affected by the trajectory followed by the camera during the estimation process, thus creating a tight coupling between camera motion (needed to, e.g., realize a visual task) and performance/accuracy of the estimated 3-D structure.

In this context, a main contribution of this thesis is the development of an online trajectory optimization strategy that allows maximization of the converge rate of a SfM estimator by (actively) affecting the camera motion. The optimization is based on the classical Persistence of Excitation (PE) condition used in the adaptive control literature to characterize the well-posedness of an estimation problem. This metric, however, is also strongly related to the Fisher Information Matrix (FIM) employed in probabilistic estimation frameworks for similar purposes. The optimization strategy that we propose can be run online because it is computationally efficient and it is only based on available information (visual measurements and camera velocity).

We also show how this technique can be coupled with the concurrent execution of a IBVS task using appropriate redundancy resolution techniques. In particular we employ a large projection operator that allows to maximize the robot redundancy by controlling the visual task error norm instead of the full task.

All of the theoretical results presented in this thesis are validated by an extensive experimental campaign run using a real robotic manipulator equipped with a camera in-hand. In our opinion, the experiments fully support our claims showing how the observability of a SfM estimation problem can be conveniently increased, even while concurrently executing a IBVS task, and that this results in improved performance for *both* the estimation and the control processes.

Keywords: active perception, control of robotic systems, visual servoing, structure from motion.

Acknowledgements

Here I am, close to finishing my dissertation, and finally finding some time to dedicate to acknowledge, as is only right and proper, all the people that, in one way or another, contributed reaching this new turning point.

To start with, I would like to thank Dr. Paolo Robuffo Giordano. I can sincerely say that I could not have wished for a better advisor. He never denied me his full support and encouragement, even in the long nights before conference deadlines. More than this, I am very grateful for the peer and mutual respect climate that he established since the very beginning of our relationship that, I believe, goes beyond a purely academic collaboration.

My very great appreciation goes to my supervisor Dr. François Chaumette for always guiding my research with his expert advise. He also honoured me with his trust and gave me a number of opportunities that significantly contributed enriching my Doctorate adventure.

One of such opportunities was the possibility to visit the Australian National University in Canberra. For this, I would like to also thank Prof. Robert Mahony for accepting my visit and supporting it. In the short six months that I spent in Canberra we could immediately establish a fruitful collaboration that, I hope, will continue in the future.

During my Doctorate, I also had the chance to visit the Max Planck Institute for Biological Cybernetics in Tübingen. For this I am also thankful to Dr. Antonio Franchi.

I wish to acknowledge Prof. Seth Hutchinson and Prof. Pascal Morin who kindly accepted to review this manuscript. An acknowledgement also goes to the other members of my jury, Dr. Christine Chevallereau and Prof. Giuseppe Oriolo, for their availability to attend my defence.

I cannot refrain to thank my past and current colleagues of the Lagadic team for

always making my Doctorate a pleasant experience. In particular, I wish to thank Giovanni Claudio and Fabrizio Schiano with whom I ended up bonding more. I also wish to thank my colleagues at the Australian National University for warmly welcoming me in their team from the very first day. The same goes with my colleagues at the Max Planck Institute for Biological Cybernetics.

Hearty thanks also go to Francesco Scattone, who shared this work and life experience with me from the very beginning, for always reminding me to not to take things too seriously.

I am very thankful to my long date friends too. Those in Rome and those now spreading all around the world. This quite long experience abroad was not always a bed of roses and it was relieving to know that you would always "be there".

Finally, I want to dedicate these last lines of acknowledgment to my parents and my sister. For just so many reasons that I do not need to explain, thank you!

Contents

Notation **xv**

Chapter 1 Introduction **1**
1.1 Topic of the thesis 3
1.2 Structure of the thesis 3

Part I Preliminaries and state of the art **7**

Chapter 2 Computer vision and robotics fundamentals **9**
2.1 Computer and robot vision 9
2.1.1 A brief history of electronic imaging 10
2.1.2 The pinhole camera model 12
2.1.3 Geometry of multiple views 15
2.1.3.1 The epipolar constraint 16
2.1.3.2 The homography constraint 18
2.1.3.3 The interaction matrix 20
2.2 Robot modeling and control 21
2.2.1 Robot kinematics 21
2.2.2 Robot differential kinematics 22
2.2.3 Kinematic singularities 24
2.2.4 Robot motion control 25
2.2.4.1 Control in the configuration space 26
2.2.4.2 Control in the task space – the non-redundant case 26
2.2.4.3 Control in the task space – the redundant case . . . 28
2.2.4.4 Control at the acceleration level 30
2.3 Visual Servoing 31
2.3.1 General classification of Visual Servoing approaches 32

2.3.2 Image Based Visual Servoing 33

Chapter 3 State estimation **37**
3.1 Estimation from vision 38
3.2 Deterministic frameworks 42
3.2.1 The Luenberger observer 42
3.2.2 Nonlinear state observation 44
3.2.3 A nonlinear observer for SfM 47
3.3 Probabilistic frameworks 49
3.3.1 The Maximum Likelihood Estimator 49
3.3.2 The Fisher Information Matrix and the Cramer-Rao bound . 53
3.3.3 The Kalman-Bucy filter 56
3.4 Active perception 61

Part II Active structure from motion **69**

Chapter 4 A framework for active Structure from Motion **71**
4.1 Interesting properties of the nonlinear Structure from Motion estimator 72
4.2 Characterization of the system transient behavior 73
4.3 Shaping the damping factor 76
4.4 Tuning the stiffness matrix 77
4.5 Application to a class of geometric primitives 80
4.5.1 Active Structure from Motion for a point 80
4.5.1.1 Planar projection model 81
4.5.1.2 Spherical projection model 83
4.5.1.3 Comparison between planar and spherical projection models 85
4.5.2 Active Structure from Motion for a plane 85
4.5.2.1 Plane reconstruction from 3-D points 86
4.5.2.2 Plane reconstruction from discrete image moments . 88
4.5.2.3 Optimizing online the selection of moments 90
4.5.2.4 Using dense image moments 96
4.5.3 Active Structure from Motion for a sphere 98
4.5.4 Active Structure from Motion for a cylinder 100
4.6 Conclusions 104

Chapter 5 Experiments and simulations of active structure from motion **107**
5.1 Active structure estimation for a point 109
5.1.1 Comparison of planar and spherical projection models 109

5.1.2 Depth estimation for a point feature 111
5.1.3 Comparison between the nonlinear observer and the EKF . . 114
5.2 Active structure estimation for a plane 115
5.2.1 Plane estimation from 3-D points (method B) 116
5.2.2 Plane estimation from discrete image moments (method C) . 117
5.2.3 Comparison of the three methods A, B and C 119
5.2.4 Simulation results for the use of adaptive moments 123
5.2.4.1 Unconstrained polynomial basis 124
5.2.4.2 Constrained polynomial basis 127
5.2.5 Simulation results of plane estimation from dense image moments 130
5.3 Active structure estimation for a sphere 131
5.4 Active structure estimation for a cylinder 134
5.5 Conclusions 136

Part III Information aware Visual Servoing 139

Chapter 6 Coupling active SfM and IBVS 141
6.1 Problem description 143
6.2 Plugging active sensing in IBVS 145
6.2.1 Second-order VS using a Large Projection Operator 146
6.2.2 Optimization of the 3-D Reconstruction 147
6.2.3 Second-order Switching Strategy 149
6.3 Adaptive switching 151
6.4 Conclusions 155

Chapter 7 Experimental results of coupling active SfM and IBVS 157
7.1 Using a basic switching strategy 158
7.1.1 First set of experiments 158
7.1.2 Second set of experiments 162
7.1.3 Third set of experiments 165
7.2 Using an adaptive switching strategy 166
7.3 Using a standard Kanade Lucas Tomasi feature tracker 168
7.4 Conclusions 171

Chapter 8 Conclusions and future work 173
8.1 Summary and contributions 173
8.2 Open issues and future perspectives 175

Appendix A Technical details 179

A.1 Derivation of the optimal Kalman-Bucy filter 179
A.1.1 Propagation equation for the error covariance matrix 179
A.1.2 Derivation of the optimal Kalman Filter gain 180
A.2 Dynamics of the weighted image moments 182
A.3 Time-derivative of the limb surface parameters for a spherical target 182
A.4 Estimation of the limb surface parameter for a cylindrical target . . 183
A.5 Derivation of equation (4.84) 184
A.5.1 Proof of Prop. 6.2 184
A.5.2 Properties of $E(t)$ 186

Appendix B Dense photometric structure estimation from motion 189
B.1 System dynamics with planar projection 191
B.2 System dynamics with spherical projection 193
B.3 A nonlinear observer for photometric Structure from Motion 193
B.4 Surface regularization and smoothing 198
B.5 Propagation of depth discontinuities 204
B.5.1 Rarefaction waves 207
B.5.2 Shock waves 209
B.6 Notes on the numerical implementation 214
B.7 Conclusions 217

Appendix C A primer on port-Hamiltonian systems 219
C.1 Introduction to port-Hamiltonian systems 219
C.2 Bond-graphs 224

References 227
Thesis related publications 227
Bibliography 228

Notation

General notation conventions

Throughout this thesis, the following notation conventions will be used:

- Scalar quantities are represented by lowercase symbols such as a, b, and so on.
- Elements of $\mathbb{R}^n$ and similar sets are interpreted as column vectors and represented by bold lowercase symbols such as $\boldsymbol{a}, \boldsymbol{b}$, and so on.
- We use the notation (a, b, c) to indicate a vertical concatenation of elements (scalars, vectors or matrices) and $[a\, b\, c]$ for horizontal concatenations.
- $\boldsymbol{I}_n$ is used to represent the identity matrix of dimension $n \times n$.
- $\boldsymbol{\varnothing}_{n\times m}$ is used to represent the $n \times m$ matrix with all elements equal to zero. If $m = 1$ we also use $\boldsymbol{\emptyset}_n$.
- We use the notation $\boldsymbol{A} \succ 0$ to indicate a positive definite matrix, i.e. such that $\boldsymbol{a}^T \boldsymbol{A} \boldsymbol{a} > 0, \forall \boldsymbol{a}$. In a similar way we define $\boldsymbol{A} \succeq 0$ as a semi-positive matrix. We also write $\boldsymbol{A} \prec 0$ if $-\boldsymbol{A} \succ 0$ and $\boldsymbol{A} \succ \boldsymbol{B}$ if $\boldsymbol{A} - \boldsymbol{B} \succ 0$.
- We use the symbol ${}^c_a\boldsymbol{t}_b$ to indicate the vector that goes from the origin of reference frame $\mathcal{F}_a$ to that of reference frame $\mathcal{F}_b$ and is expressed in reference frame $\mathcal{F}_c$. We also use the notation ${}^a\boldsymbol{p} = {}^a_a\boldsymbol{t}_{\boldsymbol{p}}$ for the vector that goes from the origin of reference frame $\mathcal{F}_a$ to a point $\boldsymbol{p}$ and is expressed in the reference frame $\mathcal{F}_a$. We use a similar notation to indicate velocities, e.g. ${}^c_a\boldsymbol{v}_b$ is the velocity of frame $\mathcal{F}_b$ w.r.t. frame $\mathcal{F}_a$ expressed in frame $\mathcal{F}_c$. However we use the compact notation $\boldsymbol{v}$ and $\boldsymbol{\omega}$ to indicate the linear and angular velocities of the camera frame w.r.t. the world expressed in the camera frame.

- ${}^{b}\boldsymbol{R}_a$ indicates the rotation matrix that transforms vectors from frame $\mathcal{F}_a$ to frame $\mathcal{F}_b$. We can then write ${}^{b}\boldsymbol{p} = {}^{b}\boldsymbol{R}_a\,{}^{a}\boldsymbol{p} + {}^{b}_{b}\boldsymbol{t}_a$. We also use the notation ${}^{b}\boldsymbol{M}_a$ to indicate such transformation in a compact way using the homogeneous representation of 3-D vectors:

$$\begin{bmatrix} {}^{b}\boldsymbol{p} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{b}\boldsymbol{R}_a & {}^{b}_{b}\boldsymbol{t}_a \\ \boldsymbol{\emptyset}_3^T & 1 \end{bmatrix} \begin{bmatrix} {}^{a}\boldsymbol{p} \\ 1 \end{bmatrix} = {}^{b}\boldsymbol{M}_a \begin{bmatrix} {}^{a}\boldsymbol{p} \\ 1 \end{bmatrix}.$$

- $[\boldsymbol{a}]_\times$ the is skew-symmetric matrix built with the components of the 3-D vector $\boldsymbol{a}$ and representing the cross product operator for 3-D vectors, i.e. $[\boldsymbol{a}]_\times \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$.

- $\widehat{\boldsymbol{a}}$ indicates an estimation or approximation of $\boldsymbol{a}$.

- $\widetilde{\boldsymbol{a}} = \widehat{\boldsymbol{a}} - \boldsymbol{a}$ is the error between the approximation of $\boldsymbol{a}$ and its actual value.

- $\check{\boldsymbol{a}}$ is used to indicate the expression of $\boldsymbol{a}$ in another set of coordinates.

- $\boldsymbol{A}^\dagger$ indicates a generalized inverse or the pseudoinverse of $\boldsymbol{A}$.

Nabla and friends

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$, $\boldsymbol{p} \mapsto f(\boldsymbol{p})$ be a generic scalar function of a vector argument. We indicate with

$$\nabla_{\boldsymbol{p}} f(\boldsymbol{p}) = \begin{bmatrix} \left.\frac{\partial f(\boldsymbol{p})}{\partial p_1}\right|_{\boldsymbol{p}} \\ \left.\frac{\partial f(\boldsymbol{p})}{\partial p_2}\right|_{\boldsymbol{p}} \\ \vdots \\ \left.\frac{\partial f(\boldsymbol{p})}{\partial p_n}\right|_{\boldsymbol{p}} \end{bmatrix}$$

the column vector built by stacking the partial derivatives of f w.r.t the elements of $\boldsymbol{p}$. This vector is also called the gradient of f w.r.t $\boldsymbol{p}$ and it can be thought of as the product of f by the nabla (column) vector:

$$\nabla_{\boldsymbol{p}} = \begin{bmatrix} \frac{\partial}{\partial p_1} \\ \frac{\partial}{\partial p_2} \\ \vdots \\ \frac{\partial}{\partial p_n} \end{bmatrix}.$$

This notation is convenient because it allows us to easily define the Hessian of a function, i.e. the matrix of all second order partial derivatives of f w.r.t. $\boldsymbol{p}$ as

$$\nabla\nabla_{\boldsymbol{p}}^T f(\boldsymbol{p}) = (\nabla_{\boldsymbol{p}}\nabla_{\boldsymbol{p}}^T) f(\boldsymbol{p}) = \begin{bmatrix} \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_1^2}\right|_{\boldsymbol{p}} & \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_1 \partial p_2}\right|_{\boldsymbol{p}} & \cdots & \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_1 \partial p_n}\right|_{\boldsymbol{p}} \\ \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_2 \partial p_1}\right|_{\boldsymbol{p}} & \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_2^2}\right|_{\boldsymbol{p}} & \cdots & \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_2 \partial p_n}\right|_{\boldsymbol{p}} \\ \vdots & \vdots & \ddots & \vdots \\ \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_n \partial p_1}\right|_{\boldsymbol{p}} & \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_n \partial p_2}\right|_{\boldsymbol{p}} & \cdots & \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_n^2}\right|_{\boldsymbol{p}} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Also the Laplacian is easily defined as

$$\nabla_{\boldsymbol{p}}^2 f(\boldsymbol{p}) = (\nabla_{\boldsymbol{p}}^T \nabla_{\boldsymbol{p}}) f(\boldsymbol{p}) = \operatorname{tr}\left(\nabla\nabla_{\boldsymbol{p}}^T f(\boldsymbol{p})\right) = \sum_{i=1}^{n} \left.\frac{\partial^2 f(\boldsymbol{p})}{\partial p_i^2}\right|_{\boldsymbol{p}} \in \mathbb{R}.$$

Now let $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$, $\boldsymbol{p} \mapsto \boldsymbol{f}(\boldsymbol{p})$ be a generic (column) vector function of a vector argument. We can extend the gradient operation and define the Jacobian of $\boldsymbol{f}$ w.r.t. $\boldsymbol{p}$ as the matrix:

$$\nabla_{\boldsymbol{p}} \boldsymbol{f}(\boldsymbol{p})^T = \begin{bmatrix} \nabla_{\boldsymbol{p}} f_1(\boldsymbol{p})^T \\ \nabla_{\boldsymbol{p}} f_2(\boldsymbol{p})^T \\ \vdots \\ \nabla_{\boldsymbol{p}} f_m(\boldsymbol{p})^T \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Note that this notation is somehow inappropriate since, if both $\nabla_{\boldsymbol{p}}$ and $\boldsymbol{f}$ are represented as column vectors, one should write the Jacobian as $\boldsymbol{f}\nabla_{\boldsymbol{p}}^T = \left(\nabla_{\boldsymbol{p}} \boldsymbol{f}^T\right)^T$. However we accept this notation as it is common in the literature.

For all of the differential operators introduced above, we will sometimes omit the subscript representing the variable w.r.t. which the differentiation is taken whenever this does not lead to confusion, i.e. whenever the function that is being differentiated depends on a single (vector) variable, e.g., $\nabla_{\boldsymbol{p}} f(\boldsymbol{p}) = \nabla f(\boldsymbol{p})$.

Acronyms and abbreviations

APS	Active Pixel Sensor.
CCD	Charge-Coupled Device.
CMOS	Complementary Metal-Oxide-Semiconductor.
DOF	Degree of Freedom.

EIF	Extended Information Filter.
EKF	Extended Kalman Filter.
FIM	Fisher Information Matrix.
FOV	Field Of View.
IBVS	Image Based Visual Servoing.
IF	Information Filter.
iff	if and only if.
IMU	Inertial Measurement Unit.
KF	Kalman Filter.
KLT	Kanade Lucas Tomasi feature tracker.
LO	Luenberger Observer.
LOS	Line Of Sight.
MLE	Maximum Likelihood Estimator.
MOS	Metal-Oxide-Semiconductor.
MPC	Model Predictive Control.
MPE	Maximum a Posteriori Estimator.
MTF	Modulated Transformer.
NBV	Next Best View.
NLS	Nonlinear Least Squares.
ODE	Ordinary Differential Equation.
OG	Observability Gramian.
PBVS	Position Based Visual Servoing.
PDE	Partial Differential Equation.
PDF	Probability Density Function.
PE	Persistence of Excitation.
pH	port-Hamiltonian.
POMDP	Partially Observable Markov Decision Process.
RANSAC	RANdom SAmple Consensus.
ROS	Robot Operating System.

SAM	Smoothing And Mapping.
SfM	Structure from Motion.
SLAM	Simultaneous Localization and Mapping.
SPLAM	Simultaneous Planning Localization and Mapping.
SSD	Sum of Squared Difference.
SVD	Singular Value Decomposition.
UKF	Unscented Kalman Filter.
V-SLAM	Vision based SLAM.
VO	Visual Odometry.
VS	Visual Servoing.
w.l.o.g.	without loss of generality.

Chapter 1

Introduction

Since its very origins, in the years 1980s, robotics was defined as the science that studies the *"intelligent connection between perception and action"* [SK08]. Clearly in this definition there is no particular stress on either the perception or the action components of a robotic system. What makes a robot (as well as a natural being) really intelligent is not, in fact, the excellence in either one (or even both) of them, but, instead, the correct interplay between the two. On one hand, it is immediately intuitive that the accuracy in the estimation of the robot state w.r.t. the environment that surrounds it has a strong impact on the accuracy of its actions. On the other hand, however, one should not forget that perception has been demonstrated many times to be an *active* process both for natural beings [Gib62] and for robotic systems [Baj88].

As done in many robotic works, we feel partially obliged to "justify" our study by bringing up a representative example from the biological world. It has been shown in [TTAE00, Tuc00] that raptors approach their preys by following a spiral trajectory (see Fig. 1.1(b)) rather than a (shorter) straight path. The accepted explanation for this, at first, counter-intuitive behavior is that the Line Of Sight (LOS) of the deep fovea, the retina region specialized in acute vision, points at approximately 45 degrees to the right or left of the frontal axis of the bird, see Fig. 1.1(a). By following a spiral trajectory around the pray, the raptor can then keep the target within the deep fovea field of view while still maintaining its head aligned with the rest its body. Flying on a straight line, instead, would force the falcon to turn its head to the side thus considerably increasing air drag and significantly reducing the flight speed. This trajectory is then the (natural) result of a joint performance maximization of both perception and action.

In many sensor-based robot applications, the state of a robot w.r.t. the environment can only be partially retrieved from its on-board sensors. In these situations, state estimation schemes can be exploited for recovering online the 'missing informa-

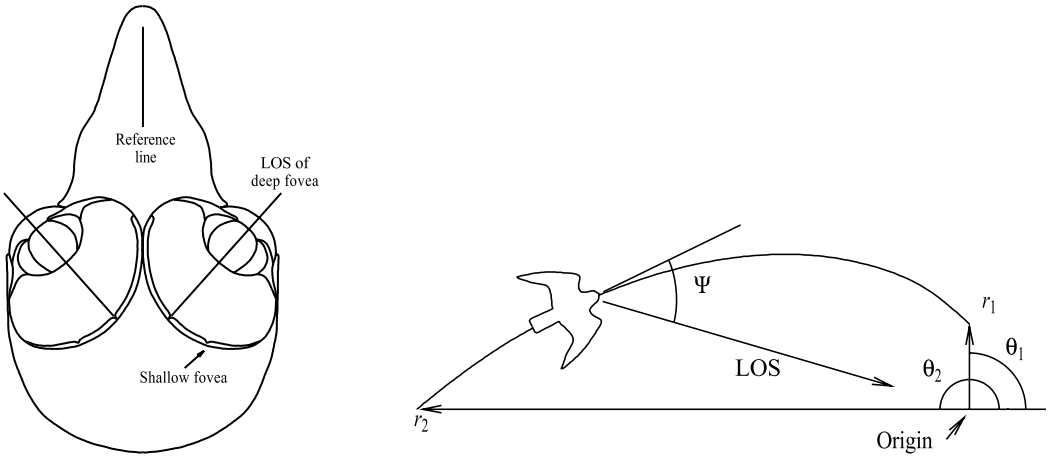


Figure 1.1 – **Trajectory optimization in peregrine falcons**. Fig. (a): anatomical structure of the falcon eye with the deep fovea (the area specialized in acute vision) pointing approximately 45 degrees to the side of the frontal line, image from [Tuc00]. Fig. (b): spiral trajectory followed by the falcon to optimize visual perception and air drag while flying toward its prey, image from [TTAE00].

tion' then fed to any planner/motion controller in place of the actual unmeasurable states. When considering non-trivial cases, however, state estimation must often cope with the nonlinear sensor mappings from the observed environment to the sensor space (the perspective projection performed by cameras being a classical example in this context [MSKS03]). Because of these nonlinearities, the estimation convergence and accuracy can be strongly affected by the particular trajectory followed by the robot/sensor which, loosely speaking, must guarantee a sufficient level of *excitation* during motion [CM10, AWCS13].

For example, in the context of Structure from Motion (SfM), a poor choice of the system inputs (e.g., the camera linear velocity) can make the 3-D scene structure non-observable *whatever* the employed estimation strategy [Mar12, EMMH13, GBR13], resulting, in practice, in inaccurate state estimation for trajectories with low information content. This, in turn, can degrade the performance of any planner/controller that needs to generate actions as a function of the reconstructed states, possibly leading to failures/instabilities in case of too large estimation errors [DOR08, MMR10].

The dependence of the estimation performance on the robot trajectory, and of the control performance on the estimation accuracy, clearly creates a tight coupling between perception and action: perception should be optimized for the sake of improving the action performance, and the chosen actions should allow maximization of the information gathered during motion for facilitating the estimation task [VTS12].

1.1 Topic of the thesis

With these considerations in mind, in this thesis we embrace the classical definition of robotics and we analyze the relationship between perception and action in robotic systems. Given the extent of the topic, we concentrate on the context of robot *visual* estimation/control and, in particular, on the class of Image Based Visual Servoing (IBVS) [CH06] control schemes. Indeed, besides being a widespread sensor-based technique, see, e.g., [TC05, GH07, MS12], IBVS is also a good example of *all* the aforementioned issues: on the one hand, whatever the chosen set of visual features (e.g., points, lines, planar patches and so on), the associated *interaction matrix* always depends on some additional 3-D information not directly measurable from the visual input (e.g., the depth of a feature point or the radius of a sphere). This 3-D information must then be approximated or estimated online via a SfM algorithm, and an inaccurate knowledge (because of, e.g., wrong approximations or poor SfM performance) can degrade the servoing execution and also lead to instabilities or loss of feature tracking [MMR10]. On the other hand, the SfM performance is directly affected by the particular trajectory followed by the camera during the servoing [Mar12]: the IBVS controller should then be able to realize the main visual task while, *at the same time*, ensuring a sufficient level of information gain for allowing an accurate state estimation.

In this context, we propose an online trajectory optimization strategy that allows to maximize the converge rate of a SfM estimator, given some limitations on the maximum acceptable norm of the camera linear velocity. We also show how this technique can be coupled with the concurrent execution of a IBVS task using appropriate redundancy resolution techniques. All of the theoretical results presented in this thesis are further validated by an extensive experimental campaign that, in our opinion, fully support our claims.

1.2 Structure of the thesis

The core of this thesis is divided into three main parts. The first one presents a review and a state of the art of the main techniques used in computer vision, robot control, state estimation and active perception. The second and third parts present, instead the original contributions of this work in the context of active structure estimation from motion and its coupling with the realization of a servoing task. In the following we propose a brief summary of the content of each part.

Outline of Part I

In Part I we introduce some preliminaries related to computer vision and robotics and we propose an extensive review of the existing literature in the field. In this regard, we try to keep the focus of the reader on the importance of a correct coupling between perception and control by highlighting the possible issues that can arise when such a coupling is not correctly ensured.

In Chapt. 2 we start by summarizing, the basic model of image formation and the geometrical relationships between two images of the same scene taken from different camera points of view. The chapter continues by presenting the standard techniques used for modeling the kinematics of a robot and for controlling its motion. We concentrate, in particular, on the class of vision based motion control techniques with an emphasis on IBVS schemes.

In Chapt. 3 we review the standard techniques used in the literature for estimating the state of a robot and/or the structure of the environment from visual information. We propose a comparison between deterministic frameworks and probabilistic ones and, in particular, between the most common metrics used in either approach for measuring the "conditioning" of the estimation. This naturally leads us to the introduction of the concept of active perception as the correct paradigm to deal with the coupling between perception and action.

Outline of Part II

The second part of the thesis presents the contributions of this work in the context of active structure estimation from motion. The results contained in this part were the subjects of the author's original publications [1, 2, 3, 4, 5, 6].

In Chapt. 4 we propose a characterization of the dynamics of the estimation error in SfM. We show that, despite the nonlinearity of the problem, the error dynamics can be made approximately equivalent to that of a second order linear system with assigned poles if the estimation gains and the camera linear velocity are correctly selected online as a function of the current visual measurements. Subsequently, the chapter proposes the application of this general active perception scheme to a set of basic geometric primitives: point features, planes, and spherical and cylindrical targets.

In Chapt. 5 we present an extensive experimental validation of all of the theoretical claims of Chapt. 4 on a real robotic manipulator equipped with a camera in-hand. In particular, we show that, for all of the geometric primitives described

in Chapt. 4, our strategy allows to correctly control and maximize (whenever possible) the convergence rate of the estimator given some constraints of the maximum allowed camera velocity norm.

Outline of Part III

The third part of this work addresses the more challenging problem of coupling active perception with the execution of a visual servoing (IBVS) task. Part of this material was presented in the author's publications [7, 8] and in [9] which, at the time of writing, is under consideration for publication.

In Chapt. 6 we suggest a sensible control strategy for increasing the convergence rate of a SfM estimator in a way that is *compatible* with the execution of a main visual servoing task. We propose to use a large projection operator that allows to maximize the redundancy of the robot so as to give more "freedom" to the active estimation maximization, while still realizing the IBVS task. Next, we extend our solution by introducing an adaptive strategy that allows to tune the effect of the active estimation and even trigger its activation/deactivation as a function of the current state of the estimation error.

In Chapt. 7 we report, as done in Chapt. 5, a thorough experimental verification of the results of Chapt. 6. We demonstrate that the optimization of the camera motion during the transient of a IBVS scheme maximizes the convergence rate of a SfM estimator. Since the estimated quantities are used to calculate the interaction matrix for the considered IBVS task, this results, in turn, in a substantial improvement of the control performance. Finally we show that, by using an adaptive strategy based on the status of the estimator, one can also reduce the negative deformation effects of the active SfM optimization on the camera trajectory.

Conclusions and appendices

Chapter 7 concludes the description of the main contributions of this work. In addition to the content outlined so far, the thesis also contains an additional conclusive chapter and three appendices.

In Chapt. 8 we provide a final overall review of the main results of the thesis by also highlighting some open issues that still remain to be solved. Chapter 8 also proposes a certain number of possible extensions to this work that could be worth investigating. Some of them, are, indeed the subjects of the author's current research activity.

In Appendix A we include some additional technical details for the derivation of some of the results contained in the thesis. This content is not essential to understand the rest of this work, but it is nevertheless included here for the interested reader.

In Appendix B we report some preliminary original results in the context of dense structure estimation from motion using photometric information directly. In this case the camera image (intended as a dense luminance level map) is directly used to compute the innovation term of the SfM observer removing the need for any preliminary image processing (e.g. feature extraction, matching and tracking) step.

In Appendix C we give a very short and rather informal introduction to the vast topic of port-Hamiltonian (pH) systems since they provide an interesting intuitive and physical interpretation of the estimation and optimization schemes used in this thesis. We also introduce very briefly the bond-graph graphical representation of such systems.

Part I

Preliminaries and state of the art

Chapter 2

Computer vision and robotics fundamentals

This chapter provides an overview of the main topics this thesis is concerned with and the related state of the art. The material and results included here are well established and can be found in many computer vision and robotics textbooks. By no means, we can provide a complete picture of such a vast literature, but we will try, at least, to introduce the most essential concepts.

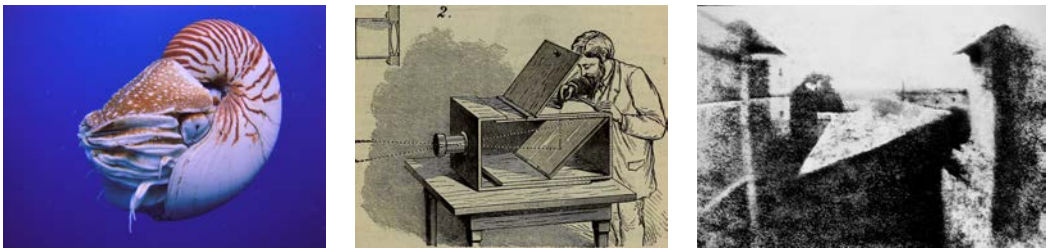
We begin the chapter with an introduction to computer vision in Sect. 2.1. A short historical prospective, in Sect. 2.1.1, foreruns a description of the basic geometrical aspects of the camera projection process (Sect. 2.1.2) and of the geometric relationships between two images taken from different points of view in Sect. 2.1.3.

After this, in Sect. 2.2, we move our focus to robotics. We start by introducing the classical modeling of robot kinematics (Sect. 2.2.1) and differential kinematics (Sect. 2.2.2) w.r.t. a certain task. This gives us all the necessary background information to tackle the problem of robot motion control in Sect. 2.2.4.

We conclude the chapter by putting together robotics and computer vision in the so-called Visual Servoing (VS) framework in Sect. 2.3. In particular, we concentrate our attention on Image Based Visual Servoing (IBVS) which is thoroughly described in Sect. 2.3.2.

2.1 Computer and robot vision

Computer vision is the research field that studies how a computer algorithm can exploit an image or, more in general, a sequence of images, to take decisions such as, for example, identify a possible danger, recognize a person or devise appropriate control inputs for a robotic system. This analysis is obviously many-sided as it can



(a) (b) (c)

Figure 2.1 – **History of electronic imaging**. Fig. (a): picture of a cephalopod Nautilus. The pinhole camera eye is visible in the center of the image. "Nautilus Palau" by Manuae - Own work. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org. Fig. (b): artist using a camera obscura as an aid for drawing, from [Bea72]. Fig. (c): first surviving picture ever taken dating circa 1826. "View from the Window at Le Gras" by Joseph Nicéphore Niépce - Rebecca A. Moss, Coordinator of Visual Resources and Digital Content Library. College of Liberal Arts Office of Information Technology, University of Minnesota. Licensed under Public Domain via Commons - https://commons.wikimedia.org.

be expected given the richness and complexity of information contained in even a single image. In this section, however, we limit ourselves to reviewing the basic geometrical formalism of computer vision.

2.1.1 A brief history of electronic imaging

The history of computer vision starts in the 5-th century BC, in ancient China, when the Mohist philosopher Mozi documented, for the first time, the use of a pinhole to project, on a wall, the inverted image of a person [Nee62]. Western civilizations made the same discovery about a century later, when the Greek philosopher Aristotle described the effect in his book "Problems". This natural phenomenon, can sometimes be observed when sunlight filters through dense foliage and it is the working principle of some primitive animal eyes [Lan05] such as those of the cephalopod Nautilus in Fig. 2.1(a).

In the centuries that followed, many scientists, philosophers and artists experimented with the pinhole and contributed to the development of the so called *camera obscura* (see Fig. 2.1(b)), an elementary optical device that allows to project a neat image of the environment on a planar surface. In 1584, in his work "Magiae naturalis", the Italian scientist and philosopher Giambattista della Porta proposed, for the first time, the addition of concave and convex lenses to improve the focus and brightness of the images projected by the camera obscura. It was only in the 19-th century, however, that, thanks to the contributions of Thomas Wedgwood and Nicéphore Niépce, the device would be used, in conjunction with light-sensitive materials, to create the first permanent pictures (Fig. 2.1(c)).

These primitive cameras required an exposure time of several minutes and were, thus, not suited for real-time video capturing. The invention of the celluloid film by Kodak in 1889 finally made it possible to take instantaneous snapshots and videos at a reasonable cost. On February 12, 1892 Léon Bouly first patented the *cinématographe*, a single device that would allow the capturing and projection of motion movies. Unable to afford the patent fees and to further develop the idea, Bouly would sell the rights to the French engineers Auguste and Louis Lumière who would realize the first movie "Sortie de l'usine Lumière de Lyon" in 1895.

Cameras would have been of little use for robot control without the invention of electric image sensors. The first attempts to transduce light information into electric signals date back to the beginning of the 20-th century when Alan Archibald Campbell-Swinton discussed the use of cathode ray tubes as both capture and display devices to realize a "Distant Electric Vision" [Cam08]. Such a device would actually come to light only thirty years later, when H. Miller and J. W. Strange at EMI, and H. Iams and A. Rose at RCA independently succeeded in transmitting the first images.

Equally fundamental for the development of computer vision, was the discovery of solid-state electronics. The unilateral conducting properties of certain crystals had been discovered by the German scientist Karl Ferdinand Braun in 1874, but it was not until the end of the 1930s that the American physicist R Shoemaker Ohl, at Bell Labs, would obtain a sufficiently pure silicon crystal to realize an efficient rectifying effect. Ohl also discovered and described the photo sensitivity of semiconductor junctions. As the story goes [RH97], Ohl was examining the conducting properties of a cracked silicon crystal when he realized, by accident, that the voltage across the rod would suddenly increase when the crystal was flashed by a desk lamp. This discovery would later lead to the development of the solid-state photo-diode.

In the late 1960s attempts were made to use multiple photo-diodes, arranged in a matrix, to realize image sensors. Particularly promising was the Active Pixel Sensor (APS) technology in which each pixel photo-diode was coupled with its own integrated Complementary Metal-Oxide-Semiconductor (CMOS) transistor amplifier to increase noise rejection and reduce readout time. The beginning of electronic imaging was, however, marked by the invention of the Charge-Coupled Device (CCD) by Willard Boyle and George E. Smith at Bell Labs in 1969 [BS70]. Michael Tompsett, also at Bell Labs, later demonstrated the potential of CCD devices for electronic imaging in 1971 [ABT71, TAB+71]. The high variability in Metal-Oxide-Semiconductor (MOS) production processes and the instability, over time, of MOS transistors characteristics, eclipsed APS-MOS technology in favor of CCDs for image sensors. In the late 1980s and early 1990s, however, advances in

the production process determined a resurgence in the use of CMOS sensors that nowadays dominate the market.

2.1.2 The pinhole camera model

Modern vision sensors are far more sophisticated than a simple pinhole camera. A complex optical system, composed of multiple lenses, is used to focus light on the actual image sensor. Each of these lenses is characterized by its own optical properties including its shape, refraction, absorption and reflectance indices, chromatic aberration, presence of impurities and so on. The projection process can, consequently, be extremely complex to describe accurately. In this work, however, as it is often the case for computer vision and robotic applications, we make the assumption that the camera optics is "good enough" to reduce all distorting effects so that the projection model can be reasonably approximated by that of an ideal pinhole camera with an infinitely small aperture. Under this assumption, only the light rays passing through the aperture position $\boldsymbol{o}_{\mathcal{C}}$ can enter the pinhole camera and hit the sensing surface on the back wall (see Fig. 2.1). We call this point the *camera optical center*. The axis passing through this point and perpendicular to the image plane is, instead, called the *camera optical axis*. The distance f between the image plane and the optical center (measured on the optical axis) is called the *focal length*.

It is convenient to define a reference frame $\mathcal{F}_{\mathcal{C}}$ centered in $\boldsymbol{o}_{\mathcal{C}}$ and with the $\boldsymbol{z}$ axis parallel to the camera optical axis. This is also referred to as a *canonical retinal frame*. Consider a generic 3-D point $\boldsymbol{p} = [X, Y, Z]^T$ in frame $\mathcal{F}_{\mathcal{C}}$. Using elementary geometry and triangle similarities, one can conclude that the light ray originating from $\boldsymbol{p}$ and passing through $\boldsymbol{o}_{\mathcal{C}}$ intersects the image plane in the point $\boldsymbol{\rho} = \left[-f\frac{X}{Z}, -f\frac{Y}{Z}, -f\right]^T$ in $\mathcal{F}_{\mathcal{C}}$, also called the *perspective projection* of $\boldsymbol{p}$, see [MSKS03]. Note that the same is true for any point along the projection line from $\boldsymbol{o}_{\mathcal{C}}$ to $\boldsymbol{\rho}$, i.e. for any point defined as $\boldsymbol{p}' = \alpha\boldsymbol{p}$. A camera is, therefore, a *scale invariant* sensor: the projection of the 3-D world on the 2-D sensing surface causes a "loss" of information about the scale and the distance of the environment. Small objects close to the camera will look the same as larger objects further away from the sensor. This depth ambiguity is the core of the astonishing portraits by the Dutch artist Maurits Cornelis Escher an example of which can be seen in Fig. 2.2. Vector $\boldsymbol{\rho}$ should then be considered as representing a 2-D *direction* rather than a 3-D point. More precisely one can say that $\boldsymbol{\rho}$ belongs to the 2-D *projective space*, i.e. the set of one dimensional subspaces (lines through the origin) of $\mathbb{R}^3$. Due to this scale ambiguity, it is perfectly licit to "re-scale" $\boldsymbol{\rho}$ in any way that is found convenient from a mathematical or numerical point of view. Moreover one can also

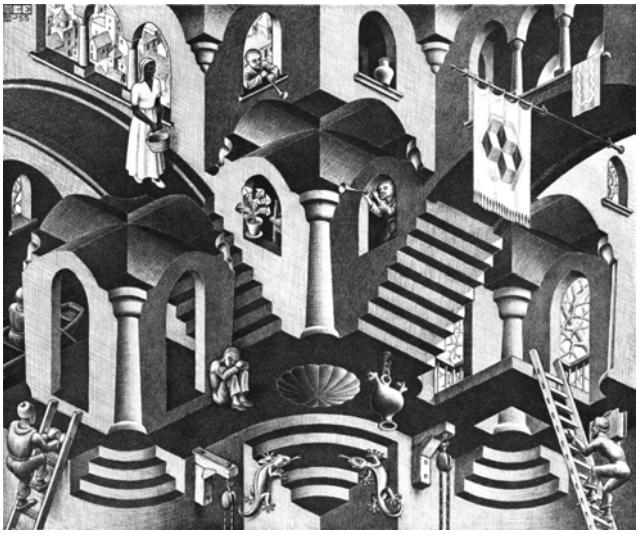


Figure 2.2 – **M. C. Escher (Leeuwarden 1898 – Laren 1972), *"Convex and Concave"*.** Lithograph print, first printed in March 1955.

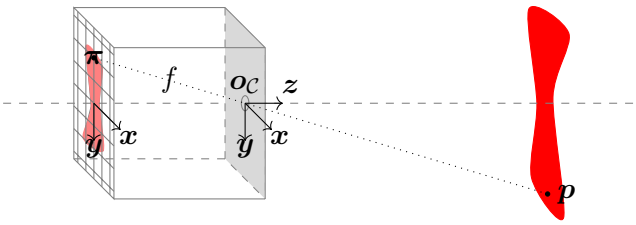


Figure 2.3 – **Projection model of a pinhole camera**.

introduce the handy notation

$$\boldsymbol{\rho} \cong \boldsymbol{p}$$

to indicate equivalence, up to a scale factor, between two quantities.

By dividing $\boldsymbol{\rho}$ by its last component $-f$, e.g., one obtains the vector

$$\boldsymbol{\pi} = \frac{\boldsymbol{\rho}}{-f} = \frac{1}{Z}\boldsymbol{p} \in \mathbb{P}^2 \tag{2.1}$$

where $\mathbb{P}^2$ is the space of 3-D homogeneous vectors defined as

$$\mathbb{P}^2 = \{\boldsymbol{a} : \boldsymbol{a} \in \mathbb{R}^3,\, \boldsymbol{e}_3^T \boldsymbol{a} = 1\}. \tag{2.2}$$

Note that, the notation $\mathbb{P}^2$ is often used in the literature to represent the projective space of dimension 2. For us, instead, it represents a particular representation of such space (the one based on homogeneous vectors). In this representation, $\boldsymbol{\pi}$ also corresponds to the projection of $\boldsymbol{p}$ on a virtual image plane, parallel to the vision sensor and placed at a unitary distance from $\boldsymbol{o}_{\mathcal{C}}$ *in front* of the optical center (see Fig. 2.4). Another convenient representation is obtained by dividing $\boldsymbol{\pi}$ by its norm.

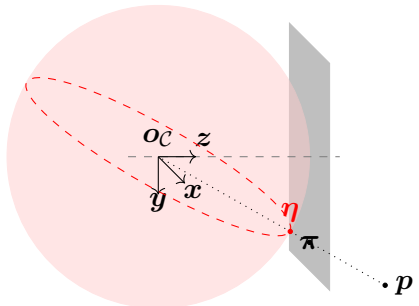


Figure 2.4 – **Planar and spherical perspective projection models**.

In this way, one obtains the unit-norm vector

$$\boldsymbol{\eta} = \frac{\boldsymbol{\pi}}{\|\boldsymbol{\pi}\|} = \frac{1}{\|\boldsymbol{p}\|}\boldsymbol{p} \in \mathbb{S}^2$$

where $\mathbb{S}^2$ is the *unit sphere*, i.e. the space of 3-D unit-norm vectors defined as

$$\mathbb{S}^2 = \{\boldsymbol{a} : \boldsymbol{a} \in \mathbb{R}^3,\ \|\boldsymbol{a}\| = 1\}. \tag{2.3}$$

In this representation, also called the *spherical perspective projection*, $\boldsymbol{\eta}$ corresponds to the intersection of the projection ray of $\boldsymbol{p}$ with a virtual spherical imaging surface with unit radius and centered in $\boldsymbol{o}_{\mathcal{C}}$ (see again Fig. 2.4). The spherical model is particularly convenient when dealing with cameras with a large Field Of View (FOV), such as catadioptric cameras, or when fusing images from different cameras pointing in different directions (generalized cameras) [FC09]. In fact one has, obviously, $\boldsymbol{\eta} = \boldsymbol{\pi}/\|\boldsymbol{\pi}\|$ and, since $\|\boldsymbol{\pi}\| \geq \pi_3 = 1$, $\boldsymbol{\eta}$ is always well defined where as $\boldsymbol{\pi} = \boldsymbol{\eta}/\eta_3$ which is not defined for $\eta_3 = 0$ (points on the plane orthogonal to $\boldsymbol{e}_3$ and passing through $\boldsymbol{o}_{\mathcal{C}}$).

Before concluding this section, we want to point the attention to the fact that, in an actual camera sensor, measurements are not obtained directly w.r.t. the canonical frame introduced above, but, instead, in terms of non-negative integer pixel indices $(j,\, k) \in \mathbb{N}^2$ w.r.t. an image origin which, typically, corresponds to the upper left corner of the image. Assuming that all the pixels are rectangular and have the same size, the transformation from pixel indices to homogeneous coordinates $\boldsymbol{\pi}$ is affine and can then be written in a matrix form as

$$\begin{bmatrix} j \\ k \\ 1 \end{bmatrix} = \begin{bmatrix} fd_x & 0 & j_{\boldsymbol{o}_{\mathcal{C}}} \\ 0 & fd_y & k_{\boldsymbol{o}_{\mathcal{C}}} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \boldsymbol{K}\boldsymbol{\pi} \tag{2.4}$$

where d_x and d_y are the pixel dimensions in metric units and $(j_{\boldsymbol{o}_{\mathcal{C}}},\, k_{\boldsymbol{o}_{\mathcal{C}}})$ are the coordinates, in pixels, of the camera *principal point*: the point in which the optical axis intersects the image plane. The matrix $\boldsymbol{K}$ contains a set of parameters that

are peculiar to a specific camera and therefore is also called the *intrinsic parameter matrix*. Identifying $\boldsymbol{K}$ requires, in general, a *calibration process* and this motivates the fact that $\boldsymbol{K}$ is also called the camera *calibration matrix*. Camera calibration can be a complex problem and, as such, it has generated a vast literature (see, for example [Dua71, Tsa87, Zhe00]). More complex, possibly nonlinear, transformations can also be considered, instead of (2.4), to model distortion effects. This topic is, however, beyond the scope of this thesis and an incredible amount of libraries and toolboxes are available to perform the calibration task. Among these one can mention ViSP [MSC05] and OpenCV [BK12] just to give some examples. From this point on, therefore, we will always assume that the camera in use has been preliminarily *perfectly* calibrated and thus the intrinsic parameters are known so that it is always possible to recover the metric measurement $\boldsymbol{\pi}$ (or equivalently $\boldsymbol{\eta}$) from the raw measurement (j, k). This allows us to carry on all further numerical developments in terms of the metric quantities $\boldsymbol{\pi}$ and $\boldsymbol{\eta}$ with the implicit assumption that (2.4) (or a more complex model) will be used to transform the sensor measurements in the actual implementation of the algorithms. Calibration errors will be ignored or treated as exogenous noise.

Finally one should mention that, for some computer vision applications, such as, for example, 3-D reconstruction from online videos [PNF+08, Har94], it is not desirable or even possible to perform any camera calibration procedure. A vast literature exists concerning the use of uncalibrated cameras in computer vision and robot control. This is, again, beyond the scope of this thesis.

2.1.3 Geometry of multiple views

In Sect. 2.1.2 we explained how the light, emanating from a particular point in the environment, hits the image sensor in a specific position. This section is, instead, devoted to the relationship between the projection of the same point $\boldsymbol{p}$ on two different image planes belonging to different cameras or, equivalently, corresponding to a different position of a single moving camera.

Note that, in general, *identifying* the same point in two different images is all but a trivial task. The problem goes generally under the name of *correspondence resolution* or *feature matching and tracking* or, more in general, *image registration* and has been addressed in countless works (see, e.g., [MS05, LK81]). This thesis, however, is not concerned with the correspondence problem resolution and we assume that a robust tracking algorithm, such as those implemented in the ViSP or OpenCV libraries, allows to correctly identify the same point (or other geometric primitives) in different images.

2.1.3.1 The epipolar constraint

Consider a generic point $\boldsymbol{p}$, fix two different camera poses corresponding to the reference frames $\mathcal{F}_a$, $\mathcal{F}_b$ and let ${}^a\boldsymbol{R}_b$ represent the orientation of $\mathcal{F}_b$ expressed in $\mathcal{F}_a$ and ${}^a_a\boldsymbol{t}_b$ be the coordinates of the origin of $\mathcal{F}_b$ expressed in $\mathcal{F}_a$. The coordinates of $\boldsymbol{p}$ in the two frames are related by the affine transformation

$$ {}^a\boldsymbol{p} = {}^a\boldsymbol{R}_b\, {}^b\boldsymbol{p} + {}^a_a\boldsymbol{t}_b. \tag{2.5} $$

From (2.1) one can write

$$ {}^aZ\, {}^a\boldsymbol{\pi} = {}^a\boldsymbol{R}_b({}^bZ\, {}^b\boldsymbol{\pi}) + {}^a_a\boldsymbol{t}_b. $$

To eliminate the (unknown) depths from the equation one can multiply both sides by $[{}^a_a\boldsymbol{t}_b]_\times$ obtaining

$$ {}^aZ\, [{}^a_a\boldsymbol{t}_b]_\times\, {}^a\boldsymbol{\pi} = {}^bZ\, [{}^a_a\boldsymbol{t}_b]_\times\, {}^a\boldsymbol{R}_b\, {}^b\boldsymbol{\pi}. $$

Scalarly multiplying both sides by ${}^a\boldsymbol{\pi}$ yields

$$ 0 = {}^bZ\, {}^a\boldsymbol{\pi}^T\, [{}^a_a\boldsymbol{t}_b]_\times\, {}^a\boldsymbol{R}_b\, {}^b\boldsymbol{\pi}. $$

Finally, assuming ${}^bZ \neq 0$, one can divide by ${}^bZ$ and obtain the *epipolar constraint*

$$ {}^a\boldsymbol{\pi}^{T\,a}\boldsymbol{E}_b\, {}^b\boldsymbol{\pi} = 0, \tag{2.6} $$

where matrix ${}^a\boldsymbol{E}_b = [{}^a_a\boldsymbol{t}_b]_\times\, {}^a\boldsymbol{R}_b$ is the *essential matrix* encoding the relative pose between $\mathcal{F}_a$ and $\mathcal{F}_b$.

The epipolar constraint (2.6) can also be easily derived from geometric considerations. By looking at Fig. 2.5 one can immediately notice that the origins of the two frames $\boldsymbol{o}_a$ and $\boldsymbol{o}_b$ and the point $\boldsymbol{p}$ form a triangle and therefore lie on the same plane. The triple product between the three vectors ${}^a_a\boldsymbol{t}_b$, ${}^a\boldsymbol{\pi}$, and ${}^b\boldsymbol{\pi}$ (all expressed in the same frame) must, therefore, be equal to zero. From Fig. 2.5 one can also immediately realize that such triangle collapses onto a line if $\boldsymbol{o}_a$, $\boldsymbol{o}_b$ and $\boldsymbol{p}$ are aligned, i.e. if ${}^a_a\boldsymbol{t}_b = \boldsymbol{\emptyset}_3$ (pure camera rotation) or if ${}^a_a\boldsymbol{t}_b$ is parallel to ${}^a\boldsymbol{\pi}$ (translation along the projection ray of $\boldsymbol{p}$). In this case the epipolar constraint (2.6) degenerates.

Matrix ${}^a\boldsymbol{E}_b$ belongs to the *essential space*, the subset of 3×3 real matrices obtained as the product of a skew-symmetric matrix and a rotation matrix:

$$ E = \left\{ \boldsymbol{E} \in \mathbb{R}^{3\times 3} | \boldsymbol{E} = [\boldsymbol{a}]_\times\, \boldsymbol{R},\, \boldsymbol{R} \in SO(3),\, \boldsymbol{a} \in \mathbb{R}^3 \right\}, $$

where $SO(3) = \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} | \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_3, \det(\boldsymbol{R}) = 1\}$ is the *special orthonormal group* of dimension three. Since the epipolar constraint (2.6) is linear in the elements of the essential matrix, it is possible to estimate ${}^a\boldsymbol{E}_b$, up to a single scalar factor (due to the homogeneous form of (2.6)), from a sufficient number of

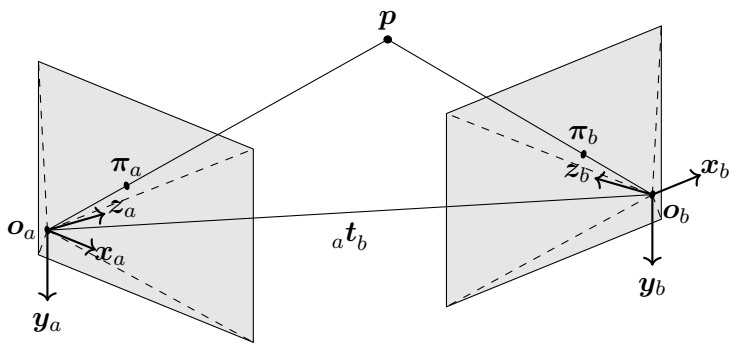


Figure 2.5 – **Geometrical representation of the epipolar constraint**. The points ${}^a\boldsymbol{\pi}$ and ${}^b\boldsymbol{\pi}$ represent the projections of the same point $\boldsymbol{p}$ from two camera vantage points related by the rotation matrix ${}^a\boldsymbol{R}_b$ and the translation vector ${}^a_a\boldsymbol{t}_b$.

point correspondences. The classical work [Lon81] introduced an efficient algorithm based on the use of eight point correspondences to reconstruct ${}^a\boldsymbol{E}_b$. This method is probably the most widely known and it is still adopted nowadays. Nevertheless, more recent algorithms have significantly reduced the number of necessary points. As a matter of fact one can notice that, despite having 9 elements, matrix ${}^a\boldsymbol{E}_b$ only possess 5 Degrees of Freedom (DOFs): 3 for the rotation and 2 for the translation up to a scalar factor. Kruppa [Kru13, Nis04] demonstrated, indeed, that it is possible to reduce the number of necessary (non degenerate) point correspondences to 5, although the solution cannot be obtained in closed form. An efficient algorithm for solving the 5 points problem was proposed in [Nis04], and it is significantly more involved than the 8 points one.

Once ${}^a\boldsymbol{E}_b$ has been reconstructed, one can also decompose it into four solutions [Lon81]

$$ {}^a\boldsymbol{R}_b = \boldsymbol{U}\boldsymbol{R}_{\boldsymbol{z}}^T\left(\pm\tfrac{\pi}{2}\right)\boldsymbol{V}^T, \quad \left[{}^a_a\boldsymbol{t}_b\right]_\times = \boldsymbol{U}\boldsymbol{R}_{\boldsymbol{z}}\left(\pm\tfrac{\pi}{2}\right)\boldsymbol{S}\boldsymbol{U}^T $$

with all possible combinations of $\pm$ and with $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{S}$ being the matrices of the Singular Value Decomposition (SVD) of ${}^a\boldsymbol{E}_b = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, and $\boldsymbol{R}_{\boldsymbol{z}}(\theta)$ being the matrix corresponding to a rotation of an angle θ about the $\boldsymbol{z}$-axis. By imposing that ${}^aZ$ and ${}^bZ$ are both positive (otherwise $\boldsymbol{p}$ would be behind the image plane and hence not visible by the camera) one can actually exclude three of the above solutions and finally select the correct one. The resulting solution is still defined up to a single scalar factor due to the scale ambiguity of the projection process. If additional metric information is available, such as the norm of ${}^a_a\boldsymbol{t}_b$, one can then recover the scale and fully reconstruct the environment structure ($\boldsymbol{p}$) and the transformation between the two camera poses (${}^a_a\boldsymbol{t}_b$, ${}^a\boldsymbol{R}_b$). In human beings, for example, learned knowledge of the distance between the eyes and of familiar objects size are used as priors for environment structure reconstruction and accurate navigation.

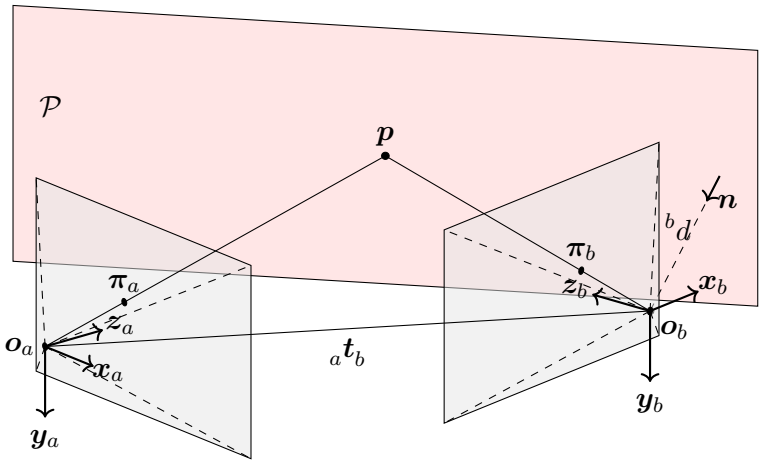


Figure 2.6 – **Geometrical representation of the homography constraint**. The points ${}^a\boldsymbol{\pi}$ and ${}^b\boldsymbol{\pi}$ represent the projections of the same point $\boldsymbol{p}$ from two camera vantage points related by the rigid transformation described by the rotation matrix ${}^a\boldsymbol{R}_b$ and the translation vector ${}^a_a\boldsymbol{t}_b$.

2.1.3.2 The homography constraint

It can be shown that, if the points used for the reconstruction of the epipolar matrix form some particular degenerate configurations, called critical surfaces [Adi85, LH88], the standard 8-points algorithm fails. Most of these critical surfaces rarely occur in practice, however one of them is extremely common in all artificial environments: the plane. If the selected points lie on the same planar surface, one additional constraint can be exploited to obtain a more sensible algorithm for reconstructing the structure of the scene.

Let ${}^b\boldsymbol{n} \in \mathbb{S}^2$ be the unit vector normal to the plane $\mathcal{P}$ expressed in frame $\mathcal{F}_b$. Let also ${}^b d$ indicate the distance between $\mathcal{P}$ and the origin of frame $\mathcal{F}_b$. Any point ${}^b\boldsymbol{p}$ on the plane surface must then satisfy the *planarity constraint*:

$$ {}^b\boldsymbol{n}^T\, {}^b\boldsymbol{p} + {}^b d = 0, \quad \forall\, {}^b\boldsymbol{p} \in \mathcal{P}. $$

Assuming ${}^b d \neq 0$, one can then write $-\frac{{}^b\boldsymbol{n}^T}{{}^b d}\, {}^b\boldsymbol{p} = 1$ and plug this in (2.5) obtaining

$$ {}^a\boldsymbol{p} = \left({}^a\boldsymbol{R}_b - \frac{{}^a_a\boldsymbol{t}_b}{{}^b d}\, {}^b\boldsymbol{n}^T \right) {}^b\boldsymbol{p} = {}^a\boldsymbol{H}_b\, {}^b\boldsymbol{p} $$

where ${}^a\boldsymbol{H}_b$ is the *homography matrix* encoding both the roto-translation between $\mathcal{F}_a$ and $\mathcal{F}_b$ and the geometric parameters of plane $\mathcal{P}$. Using again (2.1), multiplying both sides by matrix $[{}^a\boldsymbol{\pi}]_\times$ and assuming, again, ${}^b Z \neq 0$, one finally obtains the *homography constraint*

$$ [{}^a\boldsymbol{\pi}]_\times\, {}^a\boldsymbol{H}_b\, {}^b\boldsymbol{\pi} = \emptyset_3. \tag{2.7} $$

We can now immediately understand why the epipolar constraint (2.6) cannot be used to reconstruct the essential matrix ${}^{b}\boldsymbol{E}_{a}$ in a planar case. In fact for any 3-D vector $\boldsymbol{a}$ one has:

$$ {}^{a}\boldsymbol{\pi}^{T}\left([\boldsymbol{a}]_{\times}\,{}^{a}\boldsymbol{H}_{b}\right){}^{b}\boldsymbol{\pi} = -\boldsymbol{a}^{T}\left[{}^{a}\boldsymbol{\pi}\right]_{\times}{}^{a}\boldsymbol{H}_{b}\,{}^{b}\boldsymbol{\pi} = 0, $$

i.e. the constraint ${}^{a}\boldsymbol{\pi}^{T}\boldsymbol{E}\,{}^{b}\boldsymbol{\pi}$ is satisfied for a set of matrices $\boldsymbol{E} = [\boldsymbol{a}]_{\times}\,{}^{a}\boldsymbol{H}_{b}$ that are different from ${}^{a}\boldsymbol{E}_{b}$ and do not belong to the essential space.

As for the epipolar constraint, the homography constraint (2.7) is linear in the elements of the homography matrix. Differently from the epipolar case, however, only 4 point correspondences $({}^{a}\boldsymbol{\pi}_{i}, {}^{b}\boldsymbol{\pi}_{i})$ are necessary in this case, provided that no three of them are collinear. In fact there is a total of 7 DOFs (3 for the rotation, 2 for the translation up to a scalar factor and 2 for the unit-norm plane normal) but each point contributes with 2 constraints using (2.7).

Following, e.g., [MSKS03], the homography constraint (2.7) can be rearranged as $\boldsymbol{b}_{i}^{T}\mathbf{x} = 0$ where $\boldsymbol{b}_{i} = {}^{b}\boldsymbol{\pi}_{i} \otimes [{}^{a}\boldsymbol{\pi}_{i}]_{\times} \in \mathbb{R}^{9\times 3}$ (with $\otimes$ indicating the Kronecker product), and $\mathbf{x} = [{}^{a}\boldsymbol{H}_{b11}, {}^{a}\boldsymbol{H}_{b21}, \ldots, {}^{a}\boldsymbol{H}_{b33}]^{T} \in \mathbb{R}^{9}$ is the vector obtained by stacking the columns of ${}^{a}\boldsymbol{H}_{b}$. By now letting $\boldsymbol{B} = (\boldsymbol{b}_{1}, \ldots, \boldsymbol{b}_{N}) \in \mathbb{R}^{3N\times 9}$ be the collection of all the N $\boldsymbol{b}_{i}$, one can compactly rewrite equation (2.7) for all measured pairs as

$$ \boldsymbol{B}\mathbf{x} = \boldsymbol{\emptyset}_{9}. \tag{2.8} $$

Equation (2.7) has a unique (non zero) solution, up to a scalar factor, if and only if (iff) $\operatorname{rank}(\boldsymbol{B}) = 8$. A (least-square) solution $\mathbf{x}$ of (2.8) can then be found by exploiting the SVD of $\boldsymbol{B} = \boldsymbol{U}_{B}\boldsymbol{S}_{B}\boldsymbol{V}_{B}^{T}$ and by taking the column of $\boldsymbol{V}_{B}$ associated to the smallest singular value $\sigma_{1,B}^{2}$.

Since the homography matrix is defined up to a scale factor and the left-multiplication by ${}^{a}\boldsymbol{H}_{b}$ is a left group action, one can also think of ${}^{a}\boldsymbol{H}_{b}$ as an element of the *special linear group* of dimension 3 $SL(3) = \{\boldsymbol{A} \in \mathbb{R}^{3} | \det(\boldsymbol{A}) = 1\}$. This fact was exploited in [HMT+11] to derive a nonlinear observer for the homography matrix.

Using standard algorithms [MSKS03], it is finally possible to decompose the associated recovered homography ${}^{a}\boldsymbol{H}_{b}$ into 4 solutions for $\boldsymbol{R}$, $\frac{\boldsymbol{t}}{d}$ and $\boldsymbol{n}$ (expressed in either frame). The positive depth constraint can be used, as done for the essential matrix, to exclude only *two* of them. The ambiguity among the remaining physically admissible solutions can only be resolved by exploiting prior knowledge of the scene (e.g., approximated known direction of $\boldsymbol{n}$ in one of the two frames, or comparison against the homography estimated from a third frame).

Furthermore, as with the epipolar constraint, the structure of the scene and the camera translation remain defined up to a scalar factor (only the ratio $\frac{\boldsymbol{t}}{d}$ can be

recovered). Additional knowledge about either the camera motion or the scene 3-D structure is then still needed for solving this ambiguity. Moreover, similarly to the epipolar case, one can easily verify that if ${}^a_a\boldsymbol{t}_b = \boldsymbol{\emptyset}_3$ (pure camera rotation) or ${}^a_a\boldsymbol{t}_b$ is parallel to ${}^a\boldsymbol{\pi}$ (translation along the projection ray of $\boldsymbol{p}$), constraint (2.8) is satisfied regardless of the value of $\frac{\boldsymbol{t}}{d}$ and $\boldsymbol{n}$ and therefore point $\boldsymbol{p}$ cannot be used to identify these parameters.

2.1.3.3 The interaction matrix

For the development of the Visual Servoing control laws described in Sect. 2.3 it will be necessary to derive a description of the relationship between the camera motion and the motion of the projection of the environment on the image. To determine such a relationship we start by differentiating w.r.t. time the projection equation (2.1) obtaining

$$\dot{\boldsymbol{\pi}} = \frac{1}{Z}\dot{\boldsymbol{p}} - \frac{1}{Z^2}\boldsymbol{p}\dot{Z} = \frac{1}{Z}\left(\boldsymbol{I}_3 - \boldsymbol{\pi}\boldsymbol{e}_3^T\right)\dot{\boldsymbol{p}}.$$

Assuming that the point $\boldsymbol{p}$ is static in the environment one can write

$$\dot{\boldsymbol{p}} = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{p}$$

where $\boldsymbol{v}$ and $\boldsymbol{\omega}$ are the camera linear and angular velocity expressed in the camera frame. Wrapping everything up one concludes that

$$\dot{\boldsymbol{\pi}} = \begin{bmatrix} -\zeta\left(\boldsymbol{I}_3 - \boldsymbol{\pi}\boldsymbol{e}_3^T\right) & [\boldsymbol{\pi}]_\times \end{bmatrix} \boldsymbol{u} = \boldsymbol{L}_{\boldsymbol{\pi}}\boldsymbol{u}. \tag{2.9}$$

with $\zeta = \frac{1}{Z}$ and $\boldsymbol{u} = [\boldsymbol{v}^T, \boldsymbol{\omega}^T]^T$. Matrix $\boldsymbol{L}_{\boldsymbol{\pi}} \in \mathbb{R}^{3\times 6}$ is called the *Interaction Matrix* of the point feature. Note that the last row of $\boldsymbol{L}_{\boldsymbol{\pi}} \in \mathbb{R}^{3\times 6}$ contains only zeros. For this reason, in general, only the first two rows of (2.9) are considered:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -\left(1+x^2\right) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \boldsymbol{u}. \tag{2.10}$$

By repeating the same calculations one can easily show that [HM02]

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} -\delta\left(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T\right) & [\boldsymbol{\eta}]_\times \end{bmatrix} \boldsymbol{u} = \boldsymbol{L}_{\boldsymbol{\eta}}\boldsymbol{u} \tag{2.11}$$

with $\delta = \frac{1}{\|\boldsymbol{p}\|}$. Note that $\boldsymbol{L}_{\boldsymbol{\eta}} \in \mathbb{R}^{3\times 6}$, however $\operatorname{rank}(\boldsymbol{L}_{\boldsymbol{\eta}}) = 2$ because of the constraint $\|\boldsymbol{\eta}\| = 1$.

2.2 Robot modeling and control

From a mechanical point of view, a manipulator is a set of rigid bodies, called *links*, connected by *joints* possibly actuated by *motors*. In particular, we will concentrate our attention to fixed-base, open-chain, fully-actuated robotic manipulators. In such robots, the first link, called *base*, is rigidly attached to the ground so that it does not move during normal operation. A single sequence (chain) of $n-1$ links, coupled by n joints, connects the base to the *end-effector* of the manipulator, where a tool, specific to the desired task (e.g. a camera), is attached. Moreover, each one of the joints is actuated by a motor.

2.2.1 Robot kinematics

As well known, the configuration space of a rigid body, i.e. its *pose*, comprises both its position and orientation w.r.t. a fixed inertial frame $\mathcal{F}_W$ and can hence be thought of as an element of the *special Euclidean group* of dimension three: $SE(3) = \mathbb{R}^3 \times SO(3)$. The configuration space of $n+1$ *free* rigid bodies of a manipulator is then given by $SE(3) \times SE(3) \times SE(3) \cdots = SE(3)^{n+1}$. Nevertheless we made the assumption that the base of the manipulator is fixed to the ground and therefore its position and orientation are constant during normal operation. The links are also connected to each other by joints that constraint the space of possible motions by applying mechanical reaction forces to the links. Assuming infinitely stiff joints, this introduces kinematic constraints to the space of possible link velocities. Such constraints are of the *holonomic* type: they can be integrated into geometrical constraints on the manipulator configuration variables. This restricts the actual robot configuration space to a n-dimensional smooth sub-manifold $\mathcal{Q}$ of $SE(3)^{n+1}$, locally diffeomorphic to $\mathbb{R}^n$, that contains all possible configurations of the robot links. This subspace can be parametrized by a vector $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T \in \mathcal{Q}$ of n *generalized coordinates*. The vector $\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n]^T \in T_{\boldsymbol{q}}\mathcal{Q}$ represents the *generalized velocity* of the manipulator and it is an element of $T_{\boldsymbol{q}}\mathcal{Q}$, the tangent space of $\mathcal{Q}$ at $\boldsymbol{q}$.

If the manipulator kinematic parameters (relative position and orientation of the joint axes) are known, the position and orientation of each link, and in particular of the robot end-effector, w.r.t. the fixed frame $\mathcal{F}_{\mathcal{W}}$ can be computed as a function of the robot joint configuration $\boldsymbol{q}$ only. This mapping is also called the *robot direct kinematics*. Standard conventions, such as the Denavit-Hartenberg parametrization, exist to evaluate the direct kinematics [SSVO09]. Software toolboxes are also available to simplify the task [Cor11]. Identifying the kinematic parameters of a robot is a process called *kinematic calibration* that usually requires moving the robot in different joint configurations while measuring the pose of its end-effector [SSVO09].

The process can be considerably involved but many solutions have been proposed in the literature and specific software tools have been developed for the purpose [BW04]. Often the robot manufacturer provides the geometric parameters of the robot links and the only thing that remains to identify is the hand-eye transformation between the end-effector frame and the reference frame of the tool/sensor (e.g. the camera) w.r.t. which the task is assigned [MSC05]. In the rest of the thesis, we will then assume that a preliminary calibration process has been performed for the robot in use so that the manipulator kinematic parameters are know with high accuracy.

Depending on the specific task that needs to be accomplished, one can then specify a m-dimensional *task vector* $\boldsymbol{r} \in \mathcal{R}$ (e.g., the position and/or orientation of the end-effector, the distance between two points on two different links, the distance between the end-effector and a point in the world, and so on) and use the direct kinematics to calculate a *task function* or *task-oriented direct kinematics*

$$\boldsymbol{r} : \mathcal{Q} \mapsto \mathcal{R}, \quad \boldsymbol{q} \mapsto \boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q}) \tag{2.12}$$

that maps each joint configuration to the value of the task vector. The task space $\mathcal{R}$ is in general a smooth manifold, locally diffeomorphic to $\mathbb{R}^m$. In some cases, due to joint limits, only a subset $\mathcal{Q}_a$ of the configuration space $\mathcal{Q}$ is actually admissible. The image of $\mathcal{Q}_a$ through $\boldsymbol{r}$ is also called the *workspace* of the robot

$$\mathcal{WS} = \{\boldsymbol{r} \in \mathcal{R} \,|\, \boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q}) \text{ for some } \boldsymbol{q} \in \mathcal{Q}_a\}$$

Note that the task function is, in general, not injective: a robot can, in fact, be *redundant* w.r.t. the task $\boldsymbol{r} \in \mathcal{WS}$ and an infinite set of joint configurations $\boldsymbol{q} \in \mathcal{Q}_a$ might result in the same $\boldsymbol{r}$. Moreover, in most cases, $\boldsymbol{r}$ is not surjective either: if $\boldsymbol{r} \notin \mathcal{WS}$, there exists no configuration $\boldsymbol{q}$ in $\mathcal{Q}_a$ (and possibly even in $\mathcal{Q}$) such that $\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{q})$. The identification of the inverse mapping of (2.12), the *inverse kinematics problem*, is therefore a complex task and many resolution strategies have been proposed [SSVO09].

2.2.2 Robot differential kinematics

Equation (2.12) relates the space of joint configurations $\mathcal{Q}$ to that of the task $\mathcal{R}$. The *task-oriented differential kinematics* describes, instead, the relationship between the joint generalized velocities $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\mathcal{Q}$ and the task velocities $\dot{\boldsymbol{r}} \in T_{\boldsymbol{r}}\mathcal{R}$. This can easily be obtained by differentiating the task function (2.12) w.r.t. the joint generalized coordinates:

$$\dot{\boldsymbol{r}} = \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.13}$$

where $\boldsymbol{J}_{\boldsymbol{r}} = \nabla_{\boldsymbol{q}} \boldsymbol{r}^T \in \mathbb{R}^{m \times n}$ is the (analytic) *task Jacobian*. When the task vector $\boldsymbol{r}$ contains the pose of the end-effector w.r.t. a fixed world frame $\mathcal{F}_{\mathcal{W}}$, one also calls (2.13) the *robot analytic differential kinematics*.

In many situations, the task is naturally expressed in terms of the end-effector pose. This is typically the case, for example, in robotic vision applications where the task is expressed in terms of a desired camera pose or a desired position of some objects in the camera image and the camera itself is rigidly attached to the robot end-effector. In these cases it is often convenient to think of $\boldsymbol{r}$ as composed of two parts:(i) a robot direct kinematic function that, given the configuration $\boldsymbol{q}$ returns the pose of the robot end-effector w.r.t. the inertial frame; (ii) an additional function that describes the task itself in terms of the end-effector pose. For these situations, as it will be clear in Sect. 2.3 it can be useful to calculate the robot *geometric Jacobian* that relates the joint generalized velocities $\dot{\boldsymbol{q}}$ to the end-effector linear and angular velocities, typically expressed in the end-effector frame itself

$$\begin{bmatrix} {}^{\mathcal{E}}_{\mathcal{W}}\boldsymbol{v}_{\mathcal{E}} \\ {}^{\mathcal{E}}_{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{E}} \end{bmatrix} = \boldsymbol{u}_{\mathcal{E}} = \boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.14}$$

with $\boldsymbol{J}_{\mathcal{E}} \in \mathbb{R}^{6 \times n}$. Note that the geometric Jacobian does not contain partial derivatives as (2.13). In fact, if $\boldsymbol{r}$ contains the end-effector position $\boldsymbol{t}_{\mathcal{E}} \in \mathbb{R}^3$ and a minimal parametrization $\boldsymbol{\phi}_{\mathcal{E}} \in \mathbb{R}^3$ of the end-effector orientation (e.g. Euler angles, Tait-Bryan angles, and so on), then one has:

$$\begin{bmatrix} {}^{\mathcal{E}}_{\mathcal{W}}\boldsymbol{v}_{\mathcal{E}} \\ {}^{\mathcal{E}}_{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{E}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\varnothing}_{3\times3} \\ \boldsymbol{\varnothing}_{3\times3} & \boldsymbol{T}(\boldsymbol{\phi}_{\mathcal{E}}) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{t}}_{\mathcal{E}} \\ \dot{\boldsymbol{\phi}}_{\mathcal{E}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\varnothing}_{3\times3} \\ \boldsymbol{\varnothing}_{3\times3} & \boldsymbol{T}(\boldsymbol{\phi}_{\mathcal{E}}) \end{bmatrix} \nabla_{\boldsymbol{q}} \boldsymbol{r}^T \dot{\boldsymbol{q}}$$

where $\boldsymbol{T}(\boldsymbol{\phi}_{\mathcal{E}}) \in \mathbb{R}^{3\times3}$ describes the relationship between the parametrization derivative and the angular velocity. One can then conclude that

$$\boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\varnothing}_{3\times3} \\ \boldsymbol{\varnothing}_{3\times3} & \boldsymbol{T}(\boldsymbol{\phi}_{\mathcal{E}}) \end{bmatrix} \nabla_{\boldsymbol{q}} \boldsymbol{r}^T. \tag{2.15}$$

Note, however, that, in general, $\boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q})$ can be computed directly, and more conveniently, by exploiting the robot kinematics rather than using the expression (2.15) as explained, e.g., in [SSVO09].

Another important role of the geometric Jacobian can be highlighted by considering (2.14) as representing the constitutive equations of a power preserving Modulated Transformer (MTF) in the jargon of bond graphs and port-Hamiltonian (pH) systems (see Appendix C). The joint and end-effector velocities $\dot{\boldsymbol{q}}$ and $\boldsymbol{u}_{\mathcal{E}} = ({}^{\mathcal{E}}_{\mathcal{W}}\boldsymbol{v}_{\mathcal{E}}, {}^{\mathcal{E}}_{\mathcal{W}}\boldsymbol{\omega}_{\mathcal{E}})$ clearly represent the flow variables in the transformer. One can then introduce the corresponding dual effort variables, $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$, that represent the generalized forces applied to the joints and those applied to the end-effector respectively

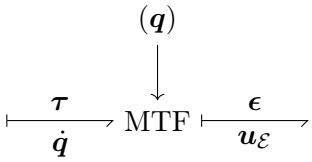


Figure 2.7 – **Bond graph representation of the robot geometric Jacobian.** The Jacobian (2.14) can be represented as a Modulated Transformer (MTF), function of the robot joint configuration (top arrow). The generalized effort and flows variables are indicated on the top and bottom of the MTF ports.

(see Fig. 2.7). Imposing that the total power is preserved by the MTF, one can write

$$\dot{\boldsymbol{q}}^T \boldsymbol{\tau} = \boldsymbol{u}_{\mathcal{E}}^T \boldsymbol{\epsilon} = \dot{\boldsymbol{q}}^T \boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q})^T \boldsymbol{\epsilon} \Leftrightarrow \boldsymbol{\tau} = \boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q})^T \boldsymbol{\epsilon},$$

i.e. the transpose of the geometric Jacobian describes the relationship between the generalized forces applied to the end-effector and the ones applied to the joints. This relationship, that goes under the name of *kineto-static duality*, can also be extended to the analytic Jacobian (2.13) as well as to the velocity transformation described by the feature interaction matrix (2.28). This latter possibility, for example, was exploited in [MS12] to extend the classical VS framework to the control of robot dynamics.

Finally the second-order differential kinematics can also be easily obtained by differentiation w.r.t. time of the above first-order relationships:

$$\ddot{\boldsymbol{r}} = \boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\boldsymbol{r}}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{2.16}$$

and

$$\begin{bmatrix} {}^{\mathcal{E}}_{\mathcal{W}}\dot{\boldsymbol{v}}_{\mathcal{E}} \\ {}^{\mathcal{E}}_{\mathcal{W}}\dot{\boldsymbol{\omega}}_{\mathcal{E}} \end{bmatrix} = \boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\mathcal{E}}(\boldsymbol{q})\dot{\boldsymbol{q}}.$$

2.2.3 Kinematic singularities

For both the task analytic Jacobian in (2.13) and the geometric one in (2.14) there can exist particular configurations $\boldsymbol{q}$ for which the Jacobian looses rank. In correspondence to such configurations, called *kinematic singularities*, one can, e.g., find an infinite set of task velocities $\dot{\boldsymbol{r}}$ that cannot be realized by any choice of $\dot{\boldsymbol{q}}$. Moreover, close to singular configurations, large joint velocities $\dot{\boldsymbol{q}}$ can result in very small task velocities $\dot{\boldsymbol{r}}$. Exploiting the duality relationship between generalized velocities and forces, one can immediately realize that in correspondence to kinematic singularities there exist infinite forces in the task space that correspond to a zero torque for at least one of the joints. It is then intuitive that such configurations will arise, e.g., when two ore more joint axes are aligned as shown in Fig. 2.8. In this configurations, in fact, the applied force $\boldsymbol{\epsilon}$ can be totally absorbed using torques

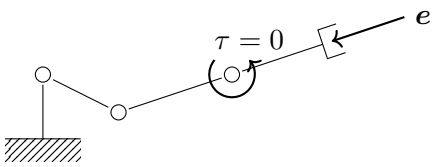


Figure 2.8 – **Kinematic singularity.** The compensation of the the applied force $\boldsymbol{\epsilon}$ does not require any action from the third joint since the force will be absorbed by first two joints and by the structure of the last two aligned links.

applied to the joints before and after the aligned links and by the robot structure of the aligned links.

2.2.4 Robot motion control

Once the robot kinematics has been identified, the control problem can be stated as follows: with the robot in an initial configuration $\boldsymbol{q}_0 \in \mathcal{Q}_a$, identify a sequence of motor commands that allows to reach (one of) the robot configuration(s) $\boldsymbol{q}_\mathrm{d} \in \mathcal{Q}_a$ corresponding to a desired value of the task vector $\boldsymbol{r}_\mathrm{d} \in \mathcal{WS}$, i.e. such that $\boldsymbol{r}(\boldsymbol{q}_\mathrm{d}) = \boldsymbol{r}_\mathrm{d}$. Obviously the problem can only be solved if $\boldsymbol{q}_\mathrm{d}$ exists and $\boldsymbol{q}_\mathrm{d} \in \mathcal{Q}_a$ and $\boldsymbol{q}_0 \in \mathcal{Q}_a$ lie in the same connected component of $\mathcal{Q}_a$. From now on, we will always assume this to be true.

In a physical robotic manipulator, the joints are actuated by motors that apply forces/torques on the robot links. The resulting motion of the links is then a result of the *robot dynamics* which depend, in addition to the robot geometric properties (the same involved in the kinematic model described above), of the joint/links dynamic parameters such as mass, inertia tensor, elasticity, viscosity and so on. The dynamic model of the robot can be obtained, using either Lagrange or Newton methods as described, e.g. in [SSVO09]. The dynamic parameters can be identified via an additional *dynamic calibration* procedure. Once this is done, dynamic control techniques can be used to compute the necessary motor commands and regulate the task to the desired value (see [ACGM94, DLMO00, MS12], and again [SSVO09] for some examples of such dynamic control techniques). In most cases, however, knowledge of the dynamic model is not strictly necessary to obtain satisfactory performance. A kinematic command (typically a velocity) can, in fact, often be considered as system input in lieu of the motor forces and torques. This is made possible by the presence of lower level feedback control loops that are capable of executing any desired velocity reference, provided that this is physically admissible (i.e. sufficiently smooth) and does not exceed the robot actuators capabilities. The control architecture can thus be split into a velocity-level controller that generates velocity references, and a dynamics controller that ensures their realization. Commercial robots are typically equipped with such low-level feedback loops within

"closed" architectures that, most of the time, do not even allow the user to have direct control over the torques and forces applied by the motors.

In this thesis we do not address the problem of controlling the robot dynamics. Instead, we assume that, thanks to the presence of a low-level dynamic control feedback, the robot can be modeled as an ideal integrator that takes as input a velocity reference $\dot{\boldsymbol{q}}$ and produces as output a configuration $\boldsymbol{q}(t) = \int_{t_0}^{t} \dot{\boldsymbol{q}}\, \mathrm{d}\tau$. For our purposes, we will also devise some control laws that generate an acceleration level command $\ddot{\boldsymbol{q}}$. This command will then be numerically integrated over time before sending it (as a velocity level command) to the robot low-level controller.

2.2.4.1 Control in the configuration space

A first possibility for controlling the robot motion is to first solve the inverse kinematics problem and calculate (one of) the joint configurations $\boldsymbol{q}_\mathrm{d}$ that satisfy $\boldsymbol{r}(\boldsymbol{q}_\mathrm{d}) = \boldsymbol{r}_\mathrm{d}$. If a solution exists, then the control law

$$\dot{\boldsymbol{q}} = -k\left(\boldsymbol{q} - \boldsymbol{q}_\mathrm{d}\right), \quad k > 0 \tag{2.17}$$

results in the exponential convergence of $\boldsymbol{q}$ to $\boldsymbol{q}_\mathrm{d}$

$$\boldsymbol{q}(t) = \boldsymbol{q}_0 e^{-\lambda t} + \boldsymbol{q}_\mathrm{d}\left(1 - e^{-\lambda t}\right).$$

With this solution, the joint configuration will evolve from $\boldsymbol{q}_0$ to $\boldsymbol{q}_\mathrm{d}$ in a straight line. Since, in general, $\boldsymbol{r}(\boldsymbol{q})$ is not a linear function, this results in unpredictable trajectories in the task space. In addition to this, if one wished to specify a time varying task trajectory $\boldsymbol{r}_\mathrm{d}(t)$, this strategy would require the resolution of the inverse kinematic problem online.

2.2.4.2 Control in the task space – the non-redundant case

An alternative solution is to devise a control law that regulates *directly* the task variable $\boldsymbol{r}$ avoiding the resolution of the inverse kinematics problem. This can be done by inverting the robot kinematics at a differential level, i.e. by solving the *inverse differential kinematics* problem. Assume that a task velocity $\dot{\boldsymbol{r}}$ is assigned. Also assume, for the moment, that $n = m$ and that the task Jacobian is not singular, i.e. $\det\left(\boldsymbol{J}_{\boldsymbol{r}}\right) \neq 0$. Under these assumptions, a joint velocity that realizes the desired task velocity can be easily obtained by inverting (2.13)

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{r}}^{-1}\dot{\boldsymbol{r}}. \tag{2.18}$$

At this stage, the problem is simplified to that of choosing a sensible $\dot{\boldsymbol{r}}$. If $\boldsymbol{r}_\mathrm{d} = const$, using

$$\dot{\boldsymbol{r}} = -k\left(\boldsymbol{r} - \boldsymbol{r}_\mathrm{d}\right) = -k\boldsymbol{e}, \quad k > 0 \tag{2.19}$$

with $\boldsymbol{e} = \boldsymbol{r} - \boldsymbol{r}_{\mathrm{d}}$, will result in the task error dynamics

$$\dot{\boldsymbol{e}} = -k\boldsymbol{e}$$

which yields the error behavior

$$\boldsymbol{e}(t) = \boldsymbol{e}(t_0)e^{-\lambda(t-t_0)} \qquad \forall t \geq t_0, \tag{2.20}$$

and thus ensures the exponential convergence of $\boldsymbol{r}$ to $\boldsymbol{r}_{\mathrm{d}}$. In this case, the task vector $\boldsymbol{r}$ will evolve in a straight line from its initial value $\boldsymbol{r}_0 = \boldsymbol{r}(\boldsymbol{q}_0)$ to $\boldsymbol{r}_{\mathrm{d}}$[1]. Since, in general, $\boldsymbol{r}(\boldsymbol{q})$ is not linear, the joint vector will instead follow a generic, non straight, trajectory. If one wished to solve a *tracking problem* in which the reference $\dot{\boldsymbol{r}}_{\mathrm{d}}$ is not constant, then

$$\dot{\boldsymbol{r}} = \dot{\boldsymbol{r}}_{\mathrm{d}} - k\left(\boldsymbol{r} - \boldsymbol{r}_{\mathrm{d}}\right) = \dot{\boldsymbol{r}}_{\mathrm{d}} - k\boldsymbol{e}, \quad k > 0 \tag{2.21}$$

with $\dot{\boldsymbol{r}}_{\mathrm{d}} = \frac{\mathrm{d}\boldsymbol{r}_{\mathrm{d}}}{\mathrm{d}t}$ should be used instead of (2.19).

If the robot is in a *singular configuration* then $\det\left(\boldsymbol{J}_{\boldsymbol{r}}\right) = 0$ and the inverse Jacobian cannot be computed. More in general if $\rho = \operatorname{rank}\left(\boldsymbol{J}_{\boldsymbol{r}}\right) < m$ the linear system (2.13) does not have a solution and the robot kinematics are *over constrained*. Leveraging classical linear least-squares results, we can reformulate the inverse kinematics problem as an optimization one: we seek to find a solution $\dot{\boldsymbol{q}}$ that results in a task velocity that is as close as possible, according to a specified metric $\boldsymbol{W} \in \mathbb{R}^{m\times m} \succ 0$, to the desired one

$$\min_{\dot{\boldsymbol{q}}} \frac{1}{2}(\boldsymbol{J}_{\boldsymbol{r}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{r}})^T \boldsymbol{W}\left(\boldsymbol{J}_{\boldsymbol{r}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{r}}\right). \tag{2.22}$$

As well known, thanks to the convexity of the problem, a unique solution always exists and can be computed by using the (weighted) right Moore-Penrose pseudoinverse

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{r}}^{\dagger}\boldsymbol{r}. \tag{2.23}$$

In particular, if $\operatorname{rank}\left(\boldsymbol{J}_{\boldsymbol{r}}\right) = n$, one has

$$\boldsymbol{J}_{\boldsymbol{r}}^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{J}_{\boldsymbol{r}}(\boldsymbol{J}_{\boldsymbol{r}}\boldsymbol{W}^{-1}\boldsymbol{J}_{\boldsymbol{r}}^T)^{-1}.$$

If $\operatorname{rank}\left(\boldsymbol{J}_{\boldsymbol{r}}\right) < n$, the pseudoinverse can still be computed by resorting on a SVD of $\boldsymbol{J}_{\boldsymbol{r}}$ [MK89]. Note however that, while (2.23) is certainly useful for symbolic manipulations, the numerical calculation of the pseudoinverse plus its multiplication by the task velocity $\dot{\boldsymbol{r}}$ is less numerically accurate and much more computationally expensive than the direct resolution of the linear optimization problem (2.22) and,

[1] Note that a straight line trajectory in the task space $\mathcal{R}$ does not result, in general, in a straight line trajectory of the robot end-effector in the 3-D Eclidean space.

therefore, it should be avoided if possible [GMW81]. The full control law can finally be calculated by substituting $\dot{\boldsymbol{r}}$ with, e.g., the expressions in (2.19) and results in the error dynamics

$$\dot{\boldsymbol{e}} = -k\boldsymbol{J}_{\boldsymbol{r}}\boldsymbol{J}_{\boldsymbol{r}}^{\dagger}\boldsymbol{e}.$$

Differently from before, the error evolution is not exponential in general and one can only prove local convergence of the task error to zero[2]. Matrix $\boldsymbol{J}_{\boldsymbol{r}}^{\dagger}$ has a null space of dimension $m-\rho$ and there exist values of the task error $\boldsymbol{e} \in \ker(\boldsymbol{J}_{\boldsymbol{r}}^{\dagger})$ that correspond to local minima in which the controller can get stuck [CH06].

2.2.4.3 Control in the task space – the redundant case

A dual situation is the one in which $n > m$ and $\operatorname{rank}(\boldsymbol{J}_{\boldsymbol{r}}) = m$. In this case the kinematics of the robot are *redundant* w.r.t. the assigned task $\boldsymbol{r}$ and infinite solutions $\dot{\boldsymbol{q}}$ can be found that result in the same value of $\dot{\boldsymbol{r}}$. One simple solution in this case is to introduce an additional $(n-m)$-dimensional task function defined as:

$$\begin{cases} \boldsymbol{r}_2 = \boldsymbol{r}_2(\boldsymbol{q}) \\ \dot{\boldsymbol{r}}_2 = \boldsymbol{J}_{\boldsymbol{r}_2}(\boldsymbol{q})\dot{\boldsymbol{q}} \end{cases}, \quad \boldsymbol{r}_2 \in \mathbb{R}^{(n-m)},\ \boldsymbol{J}_{\boldsymbol{r}_2}(\boldsymbol{q}) \in \mathbb{R}^{(n-m)\times n}.$$

One can then apply the same strategy as in the square case to the *extended* system:

$$\begin{bmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{r}}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_{\boldsymbol{r}} \\ \boldsymbol{J}_{\boldsymbol{r}_2} \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}_{ex}\dot{\boldsymbol{q}}, \quad \boldsymbol{J}_{ex} \in \mathbb{R}^{n\times n}.$$

This resolution framework, introduced for the first time in [Ser89], takes the name of *task augmentation* or *extended Jacobian method*. Its main disadvantage is the introduction of *algorithmic singularities* that arise when $\operatorname{rank}(\boldsymbol{J}_{ex}) < n$ in spite of the fact that $\operatorname{rank}(\boldsymbol{J}_{\boldsymbol{r}}) = m$ and $\operatorname{rank}(\boldsymbol{J}_{\boldsymbol{r}_2}) = n-m$.

An alternative approach is to describe the problem, as done for the overconstrained case, with the jargon of linear least-squares optimization as, e.g.

$$\begin{aligned} &\min_{\dot{\boldsymbol{q}}} \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{W}\dot{\boldsymbol{q}} \\ &\text{s.t. } \boldsymbol{J}_{\boldsymbol{r}}\dot{\boldsymbol{q}} - \dot{\boldsymbol{r}} = \emptyset_m \end{aligned}$$

where $\boldsymbol{W} \in \mathbb{R}^{n\times n} \succ 0$. A unique solution always exists and can be computed by using the (weighted) left Moore-Penrose pseudoinverse

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{r}}^{\dagger}\dot{\boldsymbol{r}}. \tag{2.24}$$

[2] In this overconstrained situation, the ideal behavior (2.20) could *still* be obtained if the m-dimensional error vector $\boldsymbol{e}$ happens to be spanned by the ρ-dimensional range space of $\boldsymbol{J}_{\boldsymbol{r}}$ ($\rho < m$) during the whole task execution. However, in most cases this special condition cannot be expected to hold.

In particular, if $\operatorname{rank}(\boldsymbol{J_r}) = m$, one has

$$\boldsymbol{J_r}^{\dagger} = \left(\boldsymbol{J_r}^T \boldsymbol{W} \boldsymbol{J_r}\right)^{-1} \boldsymbol{J_r}^T \boldsymbol{W}. \tag{2.25}$$

The same considerations as above, concerning the disadvantages of computing the pseudoinverse instead of directly solving the optimization problem, remain valid for this case. Moreover, as shown in [CK88], a direct symbolic computation of the joint velocity, without explicit calculation of the whole Jacobian pseudoinverse, is also possible thanks to appropriate block decomposition of $\boldsymbol{J_r}$.

As it is well known from linear algebra, if $\boldsymbol{J_r} \in \mathbb{R}^{m \times n}$ and $\operatorname{rank}(\boldsymbol{J_r}) = \rho < n$, then the linear mapping (2.13) has a $(n-\rho)$-dimensional *null space* or *kernel* defined as

$$\ker(\boldsymbol{J_r}) = \{\dot{\boldsymbol{q}} \in \mathbb{R}^n | \boldsymbol{J_r}\dot{\boldsymbol{q}} = \boldsymbol{\emptyset}_m\}.$$

Let us denote as $\boldsymbol{P_r} \in \mathbb{R}^{n \times n}$ the *projection matrix* whose image space corresponds to the null space of $\boldsymbol{J_r}$, i.e. $\boldsymbol{J_r}\boldsymbol{P_r} = \boldsymbol{\varnothing}_{m \times n}$. If $\dot{\boldsymbol{q}}_{\boldsymbol{r}}$ is a solution to (2.13), then also $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}_{\boldsymbol{r}} + \boldsymbol{P_r}\dot{\boldsymbol{q}}_w$, $\forall \dot{\boldsymbol{q}}_w \in \mathbb{R}^n$ is a solution to (2.13). Using Lagrange's multipliers one can, in fact, easily demonstrate [SSVO09] that the solution to the optimization problem

$$\begin{aligned} &\min_{\dot{\boldsymbol{q}}} \frac{1}{2}(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_w)^T \boldsymbol{W} (\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_w) \\ &\text{s.t. } \boldsymbol{J_r}\dot{\boldsymbol{q}} - \dot{\boldsymbol{r}} = \boldsymbol{\emptyset}_m \end{aligned}$$

is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J_r}^{\dagger}\dot{\boldsymbol{r}} + \left(\boldsymbol{I}_n - \boldsymbol{J_r}^{\dagger}\boldsymbol{J_r}\right)\dot{\boldsymbol{q}}_w. \tag{2.26}$$

with $\boldsymbol{J_r}^{\dagger}$ as in (2.25). The matrix $\left(\boldsymbol{I}_n - \boldsymbol{J_r}^{\dagger}\boldsymbol{J_r}\right)$ is then *one* of the possible matrices $\boldsymbol{P_r}$ that allow to project vector $\dot{\boldsymbol{q}}_w$ into the null space of $\boldsymbol{J_r}$ thus ensuring that (2.13) remains satisfied. Usually the joint velocity $\dot{\boldsymbol{q}}_w$ is chosen in the direction of the gradient of a *secondary* (scalar) *objective function* that one wishes to maximize

$$\dot{\boldsymbol{q}}_w = k_{\boldsymbol{r}_2} \nabla_{\boldsymbol{q}} w(\boldsymbol{q}), \quad k_{\boldsymbol{r}_2} > 0.$$

For this reason (2.26) is also called the *projected gradient* redundancy resolution technique [Lie77]. Using (2.26), with $\dot{\boldsymbol{r}}$ as in (2.19) or (2.21), results again in the globally exponentially stable task error dynamics $\dot{\boldsymbol{e}} = -k\boldsymbol{e}$ since the effect of $\dot{\boldsymbol{q}}_w$ is not visible in the space $\mathcal{R}$. Moreover the dynamics of the secondary task will be described by:

$$\dot{w} = (\nabla_{\boldsymbol{q}} w)^T \dot{\boldsymbol{q}} = (\nabla_{\boldsymbol{q}} w)^T \left[\boldsymbol{J_r}^{\dagger}\boldsymbol{r} + k_w \left(\boldsymbol{I}_n - \boldsymbol{J_r}^{\dagger}\boldsymbol{J_r}\right) \nabla_{\boldsymbol{q}} w\right].$$

If the primary and secondary task are *compatible*, i.e. $(\nabla_{\boldsymbol{q}} w)^T \boldsymbol{J_r}^{\dagger} (\boldsymbol{r} - \boldsymbol{J_r}\nabla_{\boldsymbol{q}} w) \geq 0$, then

$$\dot{w} \geq k_w \|\nabla_{\boldsymbol{q}} w\|^2 > 0$$

and the secondary cost function will be optimized.

The projected gradient method has proven very effective for the resolution of the inverse differential kinematics problem for redundant manipulators, nevertheless its implementation requires a considerable computational effort when many DOFs are involved. In this regard, alternative and more efficient solutions have been proposed such as the *reduced gradient* method [LO91]. When the application involves many different tasks that inherently have different priority levels, one can also consider the use of a *task priority* resolution framework [NHY87] that explicitly models the effect of the higher priority tasks on the lower priority ones. Along the same lines, the *task sequencing* redundancy resolution method [CCSS91] regulates each task, one at a time, from the one with the highest priority to the one with lowest one. This ensures that an "artificial" degree of redundancy is maintained during the resolution of all but the least priority task. Finally an incredible number of variations and possibly combinations of the above mentioned methods exists in the literature. For a survey on the matter, we suggest to refer to [SK08]. For the purpose of this paper, however, given the simplicity of the involved task and the relatively small number of DOFs considered, we will limit our attention to the use of a simple projected gradient resolution strategy with the understanding that an extension to different redundancy resolution frameworks would also be possible.

Finally note that the problem of controlling the robot motion in the configuration space, described in Sect. 2.2.4.1, can be regarded as a special case of the more general task space control problem in which one has $\boldsymbol{r} = \dot{\boldsymbol{q}}$ and the task function and the task Jacobian correspond to the identity function. In this case, obviously, the task Jacobian is always square and full rank and (2.18), with (2.19), trivially reduces to (2.17).

2.2.4.4 Control at the acceleration level

We conclude this section by considering the extension of the redundancy resolution frameworks introduced so far to the case of an acceleration-level kinematic control. From (2.16) one immediately has

$$\ddot{\boldsymbol{r}} - \dot{\boldsymbol{J}}_{\boldsymbol{r}}\dot{\boldsymbol{q}} = \boldsymbol{J}_{\boldsymbol{r}}\ddot{\boldsymbol{q}},$$

therefore the second order differential kinematics inversion problem is formally equivalent to the first order one, and all of the control schemes introduced above can be easily extended to the acceleration level by adopting the following substitutions in the formulas

$$\dot{\boldsymbol{q}} \to \ddot{\boldsymbol{q}}, \quad \dot{\boldsymbol{r}} \to \ddot{\boldsymbol{r}} - \dot{\boldsymbol{J}}_{\boldsymbol{r}}\dot{\boldsymbol{q}}, \quad \dot{\boldsymbol{q}}_w \to \ddot{\boldsymbol{q}}_w.$$

Differently from before, however, at the acceleration level one also needs to make sure that a sufficient level of *damping* is introduced in the system to stabilize the

second-order dynamics of the task error. If one designs

$$\ddot{\boldsymbol{r}} = \ddot{\boldsymbol{r}}_{\mathrm{d}} - k_d \left(\dot{\boldsymbol{r}} - \dot{\boldsymbol{r}}_{\mathrm{d}}\right) - k_p \left(\boldsymbol{r} - \boldsymbol{r}_{\mathrm{d}}\right), \quad k_d > 0,\, k_p > 0$$

with, possibly, $\ddot{\boldsymbol{r}}_{\mathrm{d}} = \dot{\boldsymbol{r}}_{\mathrm{d}} = \boldsymbol{\emptyset}_m$ for regulation problems, then, in the redundant and square cases, the dynamics of the task error becomes

$$\ddot{\boldsymbol{e}} = -k_d \dot{\boldsymbol{e}} - k_p \boldsymbol{e}$$

and it is always globally exponentially stable.

A damping in the null space of the task must also be introduced to dissipate internal motions. This is obtained by using, e.g.

$$\ddot{\boldsymbol{q}}_w = \nabla_{\dot{\boldsymbol{q}}} w(\dot{\boldsymbol{q}}) - k_{d,2} \dot{\boldsymbol{q}}.$$

If the internal low-level robot controller only accepts velocity-level commands (as it is the case for the robot used in the experiments of Chapts. 5 and 7) the acceleration reference, generated using the above strategy, must be numerically integrated over time before actually feeding it to the robot control unit.

2.3 Visual Servoing

For the control laws described in Sect. 2.2.4 to be implemented in practice, one needs to recover the current value of the task variable $\boldsymbol{r}$ (and possibly $\dot{\boldsymbol{r}}$) needed to calculate the feedback terms in (2.19–2.21). In other words, one needs to recover an estimation of the current robot state in relation to the assigned task. This can be done by exploiting the presence of *sensors*, such as joint encoders, lasers, sonars, and so on. According to the definition given in [CH06], the term Visual Servoing (VS), or *visual servo control*, refers, in particular, to the use of computer vision data (extracted from camera images) to control the robot motion. The first attempts of using a camera for robot control can be dated back to the early 70s [BP73, SI73]. These works where based on an approach to the problem that is fundamentally different from modern VS techniques: the camera would be used to take snapshots of the scene, understand the current situation and devise from that a new reference trajectory that would be fed to a motion control architecture; this latter would then move the robot "blindly", without directly exploiting any additional visual information, for a certain amount of time after which a new snapshot would be taken. This approach was mainly imposed by the very limited performance of the computing architectures available at that time and was abandoned with the progressive improvements in computational power. The term Visual Servoing was apparently introduced by the authors of [HP79] to distinguish their work from this pioneering

use of *visual feedback*: the novelty of their work was the use of simple but fast image processing algorithms to allow the exploitation of visual information in real-time reactive behaviors. In essence, this is still the accepted definition of VS [Cor93]. As such, VS is more of a general *paradigm* rather than a specific collection of techniques. In fact, a variety of different implementations are possible tailored to the specific applications. Different types of cameras (perspective, catadioptric [BMH03], or generalized cameras [CMS11]) both monocular and stereoscopic [HCM94] can be used. Cameras can be mounted either on the robot end-effector (*eye-in-hand* [WSN85]) or on an external fixed base (*eye-to-hand* [WWR93, RK98, HDE98]) or in a combination of both (as it is typical in humanoid robotics [APC13, TK01]). Both fixed-base and mobile manipulators [MOP07, CAK99] can be considered. VS control formalism can also be applied to other sensors than traditional cameras such as RGB-D sensors [TM12], camera/laser-stripe sensors [KMM+96] and ultrasound probes [MKC08]. The development of a fully functioning VS system requires deep mastering of a wide range of techniques spanning the fields of image processing, computer vision, and control and estimation theory. Since this thesis is more concerned with the latter aspects, we refer the reader to specialized textbooks such as, e.g. [Rus11, Sze10], and we concentrate our attention to the control and estimation aspects involved.

2.3.1 General classification of Visual Servoing approaches

A large variety of different visual control schemes have been proposed in the literature. The reader can refer to the classical works [Cor93, HHC+96] or the more recent two-parts review paper [CH06, CH07] for a complete overview. This multiplicity of solutions, however, are conceptually equivalent from a control point and can be treated with a unique formulation that is based on the interpretation of the visual information as a task variable in the sense explained in Sect. 2.2.4. Depending on the type of information that enters in the definition of the task variable, which in the context of VS is usually called the *visual features vector*, one can distinguish two main approaches:

- in Position Based Visual Servoing (PBVS) [WHB96, TMCG02] the visual information is used to estimate a set of 3-D parameters such as, e.g., the pose of the camera (or the robot end-effector) w.r.t. some reference coordinate frame. In the computer vision literature, this is referred to as the 3-D *localization problem*. Once the localization is solved, the control problem is equivalent to a classical geometric control of the robot end-effector pose.

- in Image Based Visual Servoing (IBVS) the information extracted from the camera (e.g. the position of some key points or the contour of an object in

the image) enters *directly* in the definition of the task vector and the robot pose is never reconstructed. The goal configuration is described in terms of the value that the features assume when camera is in the desired pose.

Both strategies present their advantages and disadvantages and the choice between the two can sometimes be determined by the specific application requirements. Since, for our purposes, both solutions are valid, we refer again the reader to [Cor93, HHC+96, CH06, CH07] for a deeper discussion about the two approaches, and we dedicate the next session to provide additional details about the IBVS control framework.

2.3.2 Image Based Visual Servoing

Consider a camera, with an attached reference frame $\mathcal{F}_\mathcal{C}$, that measures a set of *visual features* $\boldsymbol{s} \in \mathbb{R}^m$ (e.g., the x and y coordinates of a point, the parameters of some lines, and so on), possibly by resorting on some image processing algorithm. We make the assumption [ECR92] that the measurements $\boldsymbol{s}$ only depend on the *shape* of the part $\mathcal{O}$ of environment observed by the camera (e.g., the radius of a sphere/cylinder, the contour of a planar patch, and so on) and on the *pose* of the part w.r.t. the camera (e.g., the position $\boldsymbol{p}$ of a point, the orientation and distance of a plane, and so on). Let the shape be identified by a set of constant parameters $\boldsymbol{\theta}$. Let also $\mathcal{F}_\mathcal{O}$ be a reference frame attached to the object. Its pose w.r.t. the camera frame $\mathcal{F}_\mathcal{C}$ is then represented by the homogeneous transformation matrix

$$
{}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{O}} = \begin{bmatrix} {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{O}} & {}^{\mathcal{C}}_{\mathcal{C}}\boldsymbol{t}_{\mathcal{O}} \\ \boldsymbol{\emptyset}_3^T & 1 \end{bmatrix}.
$$

If a fixed reference frame $\mathcal{F}_\mathcal{W}$ is also defined, we can express ${}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{O}}$ as

$$
{}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{O}} = {}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{W}}\,{}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{O}}
$$

where ${}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{O}}$ and ${}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{C}}$ represent the pose of the object and the camera w.r.t. the world fixed frame.

In a eye-in-hand configuration, the object pose is fixed w.r.t. the world and the camera is rigidly attached to the robot end-effector with a constant transformation ${}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{E}}$ so that its pose can be expressed as a function of the robot joint configuration $\boldsymbol{q}$ exploiting the direct kinematics as described in Sect. 2.2.1. One can then write

$$
\boldsymbol{s} = \boldsymbol{s}({}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{O}}, \boldsymbol{\theta}) = \boldsymbol{s}({}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{E}}\,{}^{\mathcal{E}}\boldsymbol{M}_{\mathcal{W}}(\boldsymbol{q})\,{}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{O}}, \boldsymbol{\theta}) = \boldsymbol{s}(\boldsymbol{q}, {}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{E}}, {}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{O}}, \boldsymbol{\theta}).
$$

In a eye-to-hand configuration, instead, the camera pose w.r.t. the world frame is fixed and the object of interest (e.g., a tool) is mounted on the robot end-effector

with a constant transformation ${}^{\mathcal{E}}\boldsymbol{M}_{\mathcal{O}}$ so that its pose can be expressed, again, as a function of the robot joint configuration, hence

$$\boldsymbol{s} = \boldsymbol{s}({}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{O}}, \boldsymbol{\theta}) = \boldsymbol{s}({}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{W}}\,{}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{E}}(\boldsymbol{q})\,{}^{\mathcal{E}}\boldsymbol{M}_{\mathcal{O}}, \boldsymbol{\theta}) = \boldsymbol{s}(\boldsymbol{q}, {}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{W}}, {}^{\mathcal{E}}\boldsymbol{M}_{\mathcal{O}}, \boldsymbol{\theta}).$$

A hybrid between the two situations is also possible. All of these configurations, despite their differences, are all formally equivalent in the sense that, by considering the ${}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{E}}$ and ${}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{O}}$, or ${}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{W}}$ and ${}^{\mathcal{E}}\boldsymbol{M}_{\mathcal{O}}$, as part of $\boldsymbol{\theta}$, one can always write

$$\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{q}, \boldsymbol{\theta}).$$

Due to this formal equivalence, from now on we will limit our attention to the eye-in-hand configuration case.

Note that, in some applications such as target tracking [CRE91], one might have that neither ${}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{O}}$ nor ${}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{W}}$ can be considered constant because some external non-controlled agent is causing them to change with time. In these cases, which are not considered in this thesis, one might have $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{q}, \boldsymbol{\theta}, t)$.

The basic idea behind IBVS is to define the task vector in Sect. 2.2.1 as a function of the sole image features:

$$\boldsymbol{r} = \boldsymbol{s}(\boldsymbol{q}, \boldsymbol{\theta}). \tag{2.27}$$

The first step to define the VS control problem is to define a desired value of the task. Consider, for simplicity, the case of a regulation task in which a desired constant value $\boldsymbol{s}_{\mathrm{d}}$ is specified. As for the generic task oriented control framework described in Sects. 2.2.4.2 and 2.2.4.3, $\boldsymbol{s}_{\mathrm{d}}$ is defined as the value of $\boldsymbol{s}$ when the robot (or the camera) is in the desired configuration. This value can be either computed using (2.27) or, if possible, experimentally measured by manually moving the robot to the desired configuration. In this latter case, the goal of VS is to ensure that the robot can then reach this learned configuration regardless of its initial configuration.

In order to use one of the control schemes introduced in Sects. 2.2.4.2 and 2.2.4.3, one also needs to calculate the task Jacobian for the image features. In VS, rather than symbolically differentiating (2.27) w.r.t. $\boldsymbol{q}$, it is in general convenient to exploit the fact that $\boldsymbol{s}$ is a function of the camera pose ${}^{\mathcal{C}}\boldsymbol{M}_{\mathcal{W}}$ and, hence, its derivative can be expressed as a function of the camera linear and angular velocity. In particular, as shown in Sect. 2.1.3.3, the following relationship holds [CH06]

$$\dot{\boldsymbol{s}} = \boldsymbol{L}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{\chi})\boldsymbol{u} \tag{2.28}$$

where $\boldsymbol{L}_{\boldsymbol{s}} \in \mathbb{R}^{m\times 6}$ is the *interaction matrix* of the considered visual features, $\boldsymbol{\chi} \in \mathbb{R}^{p}$ is a vector of unmeasurable 3-D quantities associated to $\boldsymbol{s}$ (e.g., the depth Z for

a point feature or the radius R for a sphere), and $\boldsymbol{u} = (\boldsymbol{v}, \boldsymbol{\omega}) \in \mathbb{R}^6$ is the camera linear/angular velocity expressed in the camera frame. Furthermore, one has

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{E}} \boldsymbol{u}_{\mathcal{E}} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{E}} \boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q}) \dot{\boldsymbol{q}}$$

with $\boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q})$ being the *robot geometric Jacobian*, introduced in (2.14), relating end-effector and robot joint velocities and

$${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{E}} = \begin{bmatrix} {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{E}} & [{}^{\mathcal{C}}\boldsymbol{t}_{\mathcal{E}}]_{\times} {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{E}} \\ \boldsymbol{\varnothing}_{3\times 3} & {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{E}} \end{bmatrix}$$

being the *twist matrix* that transforms linear and angular velocities from the end-effector frame to the camera frame. Note that matrix ${}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{E}}$ contains the (constant) roto-translation between the end-effector frame and the camera frame. These quantities must be identified by a, so called, *hand-to-eye calibration* process, see e.g. the classical work [TL89]. One can then introduce a *camera geometric Jacobian*, defined as

$$\boldsymbol{J}_{\mathcal{C}}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}_{\boldsymbol{v}}(\boldsymbol{q}) \\ \boldsymbol{J}_{\boldsymbol{\omega}}(\boldsymbol{q}) \end{bmatrix} = {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{E}} \boldsymbol{J}_{\mathcal{E}}(\boldsymbol{q}), \qquad \boldsymbol{J}_{\mathcal{C}}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}, \tag{2.29}$$

and conclude that

$$\boldsymbol{J}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{\chi}, \boldsymbol{q}) = \boldsymbol{L}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{\chi}) \boldsymbol{J}_{\mathcal{C}}(\boldsymbol{q}) \in \mathbb{R}^{m\times n} \tag{2.30}$$

is the *visual task Jacobian* associated to (2.27). To simplify the notation, whenever this does not lead to confusion, we will neglect the subscript $\boldsymbol{s}$ in $\boldsymbol{J}_{\boldsymbol{s}}$. It is worth noting that, because of the structure in (2.30),

$$\rho = \operatorname{rank}(\boldsymbol{J}) \leq \min\{\operatorname{rank}(\boldsymbol{L}_{\boldsymbol{s}}), \operatorname{rank}(\boldsymbol{J}_{\mathcal{C}})\} \leq \min\{m, n, 6\}$$

for any m and n. By then defining $\boldsymbol{e} = \boldsymbol{s} - \boldsymbol{s}_{\mathrm{d}}$ as the visual error vector, one has

$$\dot{\boldsymbol{e}} = \boldsymbol{J}\dot{\boldsymbol{q}}. \tag{2.31}$$

As explained in Sect. 2.2.4.3, in a *redundant* case w.r.t. the visual task ($\rho < n$), the standard choice for regulating $\boldsymbol{e}(t) \to \boldsymbol{\emptyset}_m$ is to apply the control law

$$\dot{\boldsymbol{q}} = -\lambda \boldsymbol{J}^{\dagger}\boldsymbol{e} + (\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}\boldsymbol{J})\dot{\boldsymbol{q}}_w = -\lambda \boldsymbol{J}^{\dagger}\boldsymbol{e} + \boldsymbol{P}\dot{\boldsymbol{q}}_w, \qquad \lambda > 0, \tag{2.32}$$

where $\boldsymbol{J}^{\dagger}$ denotes the Moore-Penrose pseudoinverse of matrix $\boldsymbol{J}$ and $\dot{\boldsymbol{q}}_w \in \mathbb{R}^n$ is an arbitrary vector projected on the null-space of the main visual task through the action of the projector $\boldsymbol{P} = (\boldsymbol{I}_n - \boldsymbol{J}^{\dagger}\boldsymbol{J}) \in \mathbb{R}^{n\times n}$. Vector $\dot{\boldsymbol{q}}_w$ is typically exploited for additional optimization purposes during the servoing, such as maximization/minimization of some suitable scalar cost function $w(\boldsymbol{q})$. In *non-redundant* cases ($\rho = n$), $\boldsymbol{P} = \boldsymbol{\varnothing}_{n\times n}$ and the control action (2.32) reduces to

$$\dot{\boldsymbol{q}} = -\lambda \boldsymbol{J}^{\dagger}\boldsymbol{e}, \qquad \lambda > 0, \tag{2.33}$$

with all the system DOFs constrained by the realization of the visual task. If, instead, $\rho = m$ (possible only if $m \leq \min\{n, 6\}$), i.e., if the servoing task is *feasible* for the given camera/robot system, both control actions (2.32–2.33) will result in a perfectly decoupled and exponential convergence for the visual error $\boldsymbol{e}(t)$ as in (2.20). Finally, if $\rho < m$ (e.g., when $m > n$ and/or $m > 6$), the visual task is *overconstrained* w.r.t. the camera/robot system and the ideal exponential behavior (2.20) will, in general, only be approximated during motion.

Remark 2.1. *Note that, in general, the feature interaction matrix* (2.28)*, and thus the IBVS task Jacobian* (2.30)*, depend on some unmeasurable quantity* $\boldsymbol{\chi}$*, related to the scene 3-D structure, that cannot be directly measured using a camera sensor only. In particular, these geometric quantities only appear in the first* 3 *columns of the feature interaction matrix* (2.28)*, i.e. those related to the camera linear velocity, see, e.g.* $1/Z$ *in* (2.10)*. In practice, then, whenever the specified IBVS tasks involves the camera translational DOFs, none of the IBVS control laws introduced in this section, can be implemented* exactly*, but it will be necessary to substitute the actual* $\boldsymbol{J}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{\chi}, \boldsymbol{q})$ *with an approximation* $\widehat{\boldsymbol{J}}_{\boldsymbol{s}} = \boldsymbol{J}_{\boldsymbol{s}}(\boldsymbol{s}, \widehat{\boldsymbol{\chi}}, \boldsymbol{q})$ *with* $\widehat{\boldsymbol{J}}_{\boldsymbol{s}} \to \boldsymbol{J}_{\boldsymbol{s}}$ *for* $\widehat{\boldsymbol{\chi}} \to \boldsymbol{\chi}$*. Regardless of the degree of redundancy of the considered robot/visual task combination, the error dynamics will be governed by the eigenvalues of matrix* $\boldsymbol{J}_{\boldsymbol{s}}\widehat{\boldsymbol{J}}_{\boldsymbol{s}}^{\dagger}$*. It is already clear, then, that ensuring a high estimation accuracy for* $\widehat{\boldsymbol{\chi}}$ *will be crucial to guarantee that* $\boldsymbol{J}_{\boldsymbol{s}}\widehat{\boldsymbol{J}}_{\boldsymbol{s}}^{\dagger} \succ 0$ *(so that* $\boldsymbol{e} \to \boldsymbol{\emptyset}_m$*) and, more in general, to improve performance of any IBVS control law.*

Chapter 3

State estimation

As discussed in Sect. 2.3, the first attempts of using a camera to control robot motion used the vision sensor only to reconstruct the current state of the robot at some distant time instants. More modern PBVS approaches rely on visual information to recover an estimation of the current camera pose. Finally, even in IBVS frameworks, despite the fact that the camera pose is not directly entering in the definition of the task, the features interaction matrix in (2.28) and, as a consequence, the task Jacobian in (2.30), still depend on some 3-D geometric parameters that cannot be *directly* extracted from a camera image but, in general, must be estimated using a sequence of images together with some additional metric information.

This need for recovering (online) an approximation of the current state of the system is not peculiar to VS applications, but characterizes pretty much all control problems to the point that it has led to the emergence of an entire dedicated branch of research that can take different names depending on the particular application and assumptions.

In this chapter we introduce and analyze the problem of *state estimation* in general, but with a focus to visual applications in particular. We start, in Sect. 3.1, by introducing the problem of state/parameter estimation from vision and by reviewing the main "branches" in which this has evolved in the literature. We dedicate the final part of Sect. 3.1 to position our work w.r.t. to these different approaches. In Sects. 3.2 and 3.3 we introduce the basic deterministic and probabilistic frameworks that have been proposed to solve the estimation problem. We also explain and compare the standard metrics that are commonly used to quantify the "well-posedness" of the estimation problem and the amount of information available about the quantities to be estimated. This naturally leads us to the introduction, in Sect. 3.4, of the active perception problem which is the main topic and motivation of this thesis.

3.1 Estimation from vision

Almost all animal species have, at least, a very simple eye-spot and about 96% of all known species have a proper eye, capable of recognizing patterns and control locomotion [LF92]. During the course about 600 million years, animal eyes have evolved in a multiplicity of different forms [LF92, Lan05]. The sense of vision has then brilliantly succeeded the test of natural selection and it is probably the most essential in our daily life experience. The main reason for this is to be found in the tremendous amount of information that vision can provide about the geometry, appearance (texture and color) and even the semantic content of the environment. In a similar way, a camera is probably one of the most informative sensors that a robotic system can be equipped with. Camera sensors also have the advantage of being very small and lightweight and usually, due to their intrinsically passive nature, more power efficient than other sensors (e.g. laser range finders, ultrasound and sonar sensors need to actively generate a signal to be able to acquire the information).

The power of vision, however, also comes at a considerable cost. It is estimated that about 40% of human brain pathways are connected to the retina and up to 50% of the neural tissue might be directly or indirectly devoted to the processing of visual information, more than all other senses combined. Visual information processing also takes two thirds of the electrical activity of our brain when we open our eyes [Fix57, Bow12]. In a similar way, processing information from a vision sensor has proven to be an incredible challenge for engineering and science. Computer vision algorithms are often extremely eager of computational power to the point of being one of the main factors pushing forward the development of highly efficient and specialized computer architectures [FM05, DBSM07].

In this thesis, we are only concerned with the geometric aspects of computer vision. The first work that was directly dedicated to the mathematical modeling of the problem of reconstructing the geometric structure of a scene from two (projected) views is attributed to Kruppa [Kru13, MSKS03]. Kruppa was, in particular, interested in the problem of estimating both the structure of the scene and the relative pose between the two view points. This problem is known in computer vision as *Structure from Motion (SfM)*. As explained in Sect. 2.1.3.1, Kruppa proved that 5 point correspondences from two different points of view, are sufficient to solve the problem up to a finite number of solutions (and to a scale factor). With the increase of computers computational power, research became interested in using a large number of images (instead of just two), both calibrated and not calibrated, to accurately estimate both the structure of complex environments (represented by thousands of points) and the camera poses in each of the view points. In this context

one can mainly distinguish two different approaches.

Typical SfM algorithms, stemming from the computer vision community, process *altogether* a large number of images, usually from uncalibrated cameras, in an offline optimization process. This optimization is also known as *bundle adjustment* [TMHF00] and was initially conceived for photogrammetry applications. It can lead to very impressive results in the reconstruction of complex scenes as demonstrated in, e.g., [SSS06].

A different approach, more typical of robotic applications, is to process the set of images *sequentially*, as new frames are captured by the camera, in real time. In these approaches one is typically mainly interested in the relative pose ${}^{\mathcal{C}_{k-1}}\boldsymbol{M}_{\mathcal{C}_k}$ of the camera between frame $k-1$ and frame k (although sometimes more than two frames might be processed to improve accuracy). If necessary, a pose w.r.t. the initial frame (usually chosen as the fixed reference) can be computed by integrating the sequence of estimated transformations, i.e. ${}^{\mathcal{W}}\boldsymbol{M}_{\mathcal{C}_k} = {}^{\mathcal{C}_0}\boldsymbol{M}_{\mathcal{C}_1}\,{}^{\mathcal{C}_1}\boldsymbol{M}_{\mathcal{C}_2}\ldots{}^{\mathcal{C}_{k-1}}\boldsymbol{M}_{\mathcal{C}_k}$. Because of this integration process, the problem is typically known as *Visual Odometry (VO)* [SF11, FS12] for its similarity with the more classical wheel odometry. Even if VO is not directly interested in the reconstruction of the scene geometry, bundle adjustment can sometimes be applied to a limited selection of the most recent frames (sliding window bundle adjustment) to improve accuracy.

Somewhat in between these two approaches is the, so called, *Vision based SLAM (V-SLAM)* problem [PPTN08, Dav03]. This is an extension of the more general Simultaneous Localization and Mapping (SLAM) problem [DWB06, BDW06] to the case of visual measurements. In V-SLAM, similarly to SfM, the focus is on reconstructing both the camera pose (localization) and the structure of the environment (mapping) w.r.t. a *global* reference frame (typically coincident with the first image). Differently from SfM, however, in V-SLAM there is a temporal causality relationship between the frames. The images (and the corresponding camera poses) are normally processed in a sequence, and the camera motion is integrated, similarly to what happens in VO, until the moving camera revisits a certain part of the scene. When this *loop closure* arises, a global optimization is performed that involves both the camera pose and the section of the map correlated (in a statistical sense) to the loop path. While in SfM the estimation problem is usually solved using batch optimization techniques (such as bundle adjustment) both V-SLAM and VO mostly adopt recursive filtering strategies, such as (Extended/Unscented) Kalman Filters, Particle Filters and Recursive Least Squares/Maximum Likelihood estimators.

Note that the boundary line between V-SLAM and SfM is often very blurry. The use of the two terms seems sometimes mostly related to the "traditions" of different scientific communities with SfM being more common in the computer vision

community and V-SLAM being more widely used in the robotics one. Sometimes, in fact, the term *real-time SfM* is also used to refer to V-SLAM [CFJS02, DRMS07, SMD10].

Finally, both VO and V-SLAM algorithms, usually assume that the camera intrinsic parameters are known from some preliminary calibration process although some exceptions can be found in the literature [KWHT10, TEC12].

Regardless of the approach taken, the problem of estimating the 3-D geometric structure of a scene and/or the camera pose from a sequence of 2-D images acquired by a camera belongs to the class of the so-called *inverse problems*: starting from the *result* of a physical process (the images obtained by projection) one has to infer the *cause* (the 3-D structure and camera pose) that originated it. This type of problems is usually difficult to solve and mathematically ill-conditioned because the information contained in the measurements is typically not sufficient to entirely reconstruct the process, unless additional priors (or measurements) can be introduced. In the case of vision, the projection process described in Sect. 5.1.1 reduces the 3-D world into a 2-D image. The third dimension is "lost" in the process and this causes the scale ambiguity that affects image measurements (see Sect. 5.1.1). By no means this scale can be reconstructed from the 2-D images *only*: additional metric information (such as the known dimension of an object in the scene or the metric distance between the camera view points in the different images) must also be included. Priors on the position/shape of some observed objects or on the initial camera motion have been assumed, e.g., in [ZKA+09, KM07]. Another possibility, often exploited in SfM [XS12], VO [NNB06] and V-SLAM [SBO+10] is to use a stereo vision sensor with a known fixed distance between the left and right cameras. This strategy is also adopted by many animals, including humans, that have (at least) two front facing eyes with overlapping fields of view. An alternative, is to use a single monocular moving camera and exploit additional sensors to retrieve a metric measurement of the distance between multiple views. In some cases a monocular camera can be mounted on a slider mechanism as in [Mor80]. This is conceptually equivalent to stereo-vision and it is a strategy also adopted in nature: to increase their FOV and effectively detect predators, some animals (like pigeons) have sideways eyes with scarcely overlapping fields of view; by bobbing their heads back and forth they can however recover depth perception. Another possibility is to use additional metric sensors, such as joint encoders [CBBJ96], Inertial Measurement Units (IMUs) [GBSR15, Mar12], sonar [HMTP13] and so on, to recover some metric information such as the motion of the camera or the distance from the scene and thus reconstruct the scale of the environment. Finally, note that the monocular estimation problem is also interesting when dealing with a stereo sensor. As a matter of fact, when the distance to the environment is much larger than the distance

between the left and right cameras, stereo-vision degenerates to the monocular case. This is the reason why, e.g., panorama pictures look in general more realistic than close range ones.

With respect to these previous works, in this thesis we address the problem of 3-D structure reconstruction from monocular image measurements and *known* and *controlled* camera motion [CBBJ96]. This is, in fact, the typical situation when considering robotic fixed-base manipulators, where the camera is attached to the robot end-effector and its velocity can be accurately measured and controlled using the robot differential kinematics (2.14) and joint encoders and motors. Even when dealing with mobile or flying robots, one can sometimes assume that at least a rough estimate of the camera velocity is known from wheel odometry or aerodynamic drag [AABM14]. However, presence of non negligible dynamics, non-holonomic constraints [FMS06, SSVO09] and underactuation [AOM02, AOB14] complicates the control problem and thus the study of this kind of platforms was not considered in this thesis and is left to future extensions.

As in SfM, we are mostly interested in retrieving the scene 3-D geometry. However, w.r.t. SfM works, we do not consider arbitrarily complex scenes, but instead we limit our attention to a class of basic geometric primitives (points, spheres, cylinders, planes) that can be treated in close form and efficiently estimated. Similarly to VO and V-SLAM, we use a real-time recursive estimation framework. Our approach is particularly similar to the "sensor-based" or "ego-centric" filtering strategy presented in [GBSO13]. In both cases, the robot/camera builds a 3-D model of the environment in its own body/sensor frame via a filtering technique: a Kalman Filter (KF) in [GBSO13] and similar works, and a deterministic nonlinear filter in our case. In fact, differently from most V-SLAM papers, we do not explicitly model noise in our estimation process but, instead, we adopt a deterministic observer, originally introduced in [DOR07], that is capable (thanks to its stability properties) of rejecting the effect of noise as an exogenous disturbance. The main advantage in doing so, is that this will allow us to recover the dynamics of the error in closed form and thus to predict the performance of the estimation and also to actively optimize it using classical control techniques. The main focus of this thesis, in fact, is not the estimation problem *per se*, but rather the *control* of the camera motion to guarantee that the estimation is well defined. Indeed, as already anticipated in Sect. 2.1.3, certain camera motions may result in a degeneration of the multiple view geometry. As it will be experimentally demonstrated in Chapt. 5, the control policy that we propose results in improved estimation performance *regardless* of the adopted estimation algorithm.

3.2 Deterministic frameworks

In deterministic estimation frameworks the presence of noise in the system dynamics or in the measurement process is ignored. The design of the estimation algorithm is based on the nominal system or, equivalently, on the expected value of a stochastic system. Care is taken to always ensure that the resulting estimation error dynamics is (possibly globally and exponentially) asymptotically stable. In essence this means that the estimation will always "tend to" match with the real system state and will reject the disturbing effect of unmodeled parts of the system dynamics such as noise.

3.2.1 The Luenberger observer

The first mathematical theory behind recursive estimation for linear systems is due to David Luenberger [Lue64, Lue66]. Consider the state space representation of a linear time-invariant dynamic system

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \boldsymbol{A}\mathbf{x} + \boldsymbol{B}\mathbf{u} \\ \boldsymbol{s} = \boldsymbol{C}\mathbf{x} + \boldsymbol{D}\mathbf{u} \end{cases} \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^q$ is the system dynamic *state* that is initially equal to $\mathbf{x}_0$ (unknown), $\mathbf{u} \in \mathbb{R}^v$ is the *input* vector, $\boldsymbol{s} \in \mathbb{R}^m$ is the measurable *output*, $\boldsymbol{A} \in \mathbb{R}^{q\times q}$ is the *system matrix*, $\boldsymbol{B} \in \mathbb{R}^{q\times v}$ is the *input matrix*, $\boldsymbol{C} \in \mathbb{R}^{m\times q}$ is the *output matrix* and finally $\boldsymbol{D} \in \mathbb{R}^{m\times q}$. The idea behind the Luenberger observer is to construct an artificial dynamic system that "imitates" the real one. The reasoning behind this is that if, given the same input $\mathbf{u}$, the outputs $\boldsymbol{s}$ and $\widehat{\boldsymbol{s}}$ generated by the real system and by the observer are *identical* (up to the differential level q), then the two systems must be in the same state and one can use the state $\widehat{\mathbf{x}}$ of the observer as an estimation of the state $\mathbf{x}$ of the real system. If the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are known, a sensible way of constructing an observer for (3.1) is the Luenberger Observer (LO)

$$\begin{cases} \dot{\widehat{\mathbf{x}}} = \boldsymbol{A}\widehat{\mathbf{x}} + \boldsymbol{B}\mathbf{u} - \boldsymbol{K}\left(\widehat{\boldsymbol{s}} - \boldsymbol{s}\right) \\ \widehat{\boldsymbol{s}} = \boldsymbol{C}\widehat{\mathbf{x}} + \boldsymbol{D}\mathbf{u} \end{cases} \tag{3.2}$$

where $\boldsymbol{K} \in \mathbb{R}^{q\times m} \succ 0$ is a constant gain. With this choice it is easy to verify that the dynamics of the error $\widetilde{\mathbf{x}} = \widehat{\mathbf{x}} - \mathbf{x}$ is given by

$$\dot{\widetilde{\mathbf{x}}} = \left(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}\right)\widetilde{\mathbf{x}}$$

and its solution

$$\widetilde{\mathbf{x}} = e^{(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})(t-t_0)}\widetilde{\mathbf{x}}_0$$

converges to zero, regardless of the value of $\widetilde{\mathbf{x}}_0$, if $(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}) \prec 0$. Note that the input $\mathbf{u}$ does not appear in the error dynamics and thus, it has no effect on the

convergence of the estimation. This property only characterizes, in general, linear systems.

The conditions for the existence of a gain $\boldsymbol{K}$ s.t. $(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}) \prec 0$ were studied by Kalman who first demonstrated that the gain $\boldsymbol{K}$ always exists iff the system is *observable* [Kal60], i.e. if and only if

$$\operatorname{rank} \boldsymbol{\mathcal{O}} = q, \text{ with } \boldsymbol{\mathcal{O}} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{q-1} \end{bmatrix}. \tag{3.3}$$

or, equivalently,

$$\boldsymbol{\mathcal{G}} = \int_{t_0}^{t} e^{\boldsymbol{A}^T(\tau - t_0)} \boldsymbol{C}^T \boldsymbol{C} e^{\boldsymbol{A}(\tau - t_0)} \, \mathrm{d}\tau \succ 0, \quad \text{for some } t > t_0.$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{q \times q}$ is also called the Observability Gramian (OG) of the system. Note, again, that for linear time-invariant systems, the observability is an intrinsic property of the system and only depends on the structure of matrix $\boldsymbol{A}$ and $\boldsymbol{C}$ and not on the inputs applied to the system. Interesting for the following considerations is the fact that a system is observable iff one cannot find a state transformation such that:

$$\begin{cases} \dot{\check{\mathbf{x}}} = \begin{bmatrix} \check{\boldsymbol{A}}_{11} & \boldsymbol{\varnothing}_{m \times (q-m)} \\ \check{\boldsymbol{A}}_{21} & \check{\boldsymbol{A}}_{22} \end{bmatrix} \check{\mathbf{x}} + \check{\boldsymbol{B}}\mathbf{u} \\ \check{\boldsymbol{s}} = \begin{bmatrix} \check{\boldsymbol{C}}_1 & \boldsymbol{\varnothing}_{m \times (q-m)} \end{bmatrix} \check{\mathbf{x}} + \check{\boldsymbol{D}}\mathbf{u} \end{cases}$$

i.e. a part of the state (the bottom one in this case) does not have any effect on either the rest of the state or the measurements.

The observability results also extend to linear time-varying systems of the form:

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \boldsymbol{A}(t)\mathbf{x} + \boldsymbol{B}(t)\mathbf{u} \\ \boldsymbol{s} = \boldsymbol{C}(t)\mathbf{x} + \boldsymbol{D}(t)\mathbf{u} \end{cases}. \tag{3.4}$$

In this case, the OG is given by

$$\boldsymbol{\mathcal{G}} = \int_{t_0}^{t} \boldsymbol{\Phi}(\tau, t_0)^T \boldsymbol{C}(\tau)^T \boldsymbol{C}(\tau) \boldsymbol{\Phi}(\tau, t_0) \, \mathrm{d}\tau \succ 0, \quad \text{for some } t > 0. \tag{3.5}$$

where $\boldsymbol{\Phi}(t, t_0)$ is the state transition matrix corresponding to $\boldsymbol{A}(t)$ that satisfies

$$\mathbf{x}(t) = \boldsymbol{\Phi}(t, t_0)\mathbf{x}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t, \tau)\boldsymbol{B}(\tau)\mathbf{u}(\tau) \, \mathrm{d}\tau.$$

3.2.2 Nonlinear state observation

The case of nonlinear systems is generally more complex than that of linear ones. Consider a generic nonlinear system $\mathcal{P}$ described by

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}) \\ \boldsymbol{s} = \boldsymbol{g}(\mathbf{x}, \mathbf{u}) \end{cases}.$$

As with the linear case one can build a state observer whose dynamics is a composition of a prediction term, which tries to make the observer evolve as the real system, and an innovation term, which uses the discrepancy between the outputs $\widehat{\boldsymbol{s}}$ and $\boldsymbol{s}$ generated by the two systems to compensate for the fact that the observer was initialized from an initial state $\overline{\mathbf{x}}_0$ that is different, in general, from the initial state $\mathbf{x}_0$ of the real system. Differently from linear systems, however, the design of these terms must be done, in general, case by case without relying on any general design rule apart from some specific classes of nonlinear systems. An extension of the concept of observability to nonlinear systems was proposed in [HK77]. There, two states $\mathbf{x}_0$ and $\mathbf{x}_1$ are defined as *indistinguishable* if, starting from $\mathbf{x}_0$ and $\mathbf{x}_1$ and for every admissible input $\mathbf{u}(t)$, $t \in [t_0, t_f]$, the system $\mathcal{P}$ produces identical outputs $\boldsymbol{s}(t)$, $t \in [t_0, t_f]$. A system is then *observable* if for every state $\mathbf{x}$, the set of states indistinguishable from $\mathbf{x}$ is equal to $\mathbf{x}$ itself. If this is true, then it is possible to reconstruct the state from which the system has started (and consequently the current state) by looking at the input-output map over a certain period of time. Note that the fact that a (nonlinear) system is observable does not imply that *every* admissible input $\mathbf{u}(t)$, $t \in [t_0, t_f]$ allows to distinguish two states by looking at the output $\boldsymbol{s}$, but only that there exists *at least one* input $\mathbf{u}(t)$, $t \in [t_0, t_f]$ that allows such a distinction. Moreover, in general, a system may not be *instantaneously observable*: it might be necessary to travel for a long distance/time to be able to distinguish a state $\mathbf{x}$ from other states. Because of this, the authors of [HK77] introduce the concept of *local weak observability*. Let U be a subset of the state space, $\mathbf{x}_0$ is U-*indistinguishable* from $\mathbf{x}_1$ if none of the inputs $\mathbf{u}(t)$, $t \in [t_0, t_f]$ that, starting both $\mathbf{x}_0$ and $\mathbf{x}_1$, result in a trajectory that lies in U, i.e such that $\mathbf{x}(t) \in U, \forall t \in [t_0, t_f]$, allows to distinguish $\mathbf{x}_0$ from $\mathbf{x}_1$. The system $\mathcal{P}$ is then said locally weakly observable if, for each state $\mathbf{x} \in \mathbb{R}^q$, there exists an open neighborhood U such that, for every open neighborhood $V \subseteq U$ of $\mathbf{x}$, there is no state in V (other than $\mathbf{x}$) that is V-indistinguishable from $\mathbf{x}$. Intuitively this means that there exist some input that allows to instantaneously distinguish a state from its neighbours, see [HK77]. If a system is locally weakly observable, then the system state can be estimated from the measured outputs, the known control inputs and a certain number (depending on the particular system) of their derivatives. We want to stress again the fact that not all admissible inputs might be appropriate for the estimation to be effective,

even if the system is observable. [HK77] also introduces a simple algebraic test for local weak observability. To simplify the algebra we limit our attention to driftless control-affine systems that have the form:

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \boldsymbol{L}(\mathbf{x})\mathbf{u} = \sum_{i=1}^{v} \boldsymbol{l}_i(\mathbf{x})\mathrm{u}_i \\ \boldsymbol{s} = \boldsymbol{g}(\mathbf{x}) \end{cases}.$$

with $\boldsymbol{L}(\mathbf{x}) \in \mathbb{R}^{q\times v}$ and $\boldsymbol{l}_i(\mathbf{x}) \in \mathbb{R}^q$. Let us denote as $\mathrm{L}_{\boldsymbol{l}}\boldsymbol{g}$ the *Lie derivative* of $\boldsymbol{g}$ w.r.t. $\boldsymbol{l}$ defined as:

$$\mathrm{L}_{\boldsymbol{l}}\boldsymbol{g}(\mathbf{x}) = \nabla_{\mathbf{x}}\boldsymbol{g}(\mathbf{x})^T\boldsymbol{l}(\mathbf{x}) \in \mathbb{R}^q.$$

Lie differentiation is a recursive operation and we can define

$$\mathrm{L}_{\boldsymbol{l}_2}\mathrm{L}_{\boldsymbol{l}_1}\boldsymbol{g}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathrm{L}_{\boldsymbol{l}_1}\boldsymbol{g}(\mathbf{x})^T\boldsymbol{l}_2(\mathbf{x}),\ \mathrm{L}_{\boldsymbol{l}}^k\boldsymbol{g}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathrm{L}_{\boldsymbol{l}}^{k-1}\boldsymbol{g}(\mathbf{x})^T\boldsymbol{l}(\mathbf{x}),\ \text{with } \mathrm{L}^0\boldsymbol{g}(\mathbf{x}) = \boldsymbol{g}(\mathbf{x}).$$

We can now build a matrix $\boldsymbol{\mathcal{O}}$ analogous to the one in (3.3) by piling up the single matrices obtained by taking the Jacobians of the Lie derivatives of $\boldsymbol{g}(\mathbf{x})$ w.r.t. $\boldsymbol{l}_i$. If this matrix has rank q, then the system is locally weakly observable and an estimator can be designed that reconstructs $\mathbf{x}$ from measurements of $\boldsymbol{s}$ and known (and appropriate) inputs $\mathbf{u}$.

To give an example let us consider the dynamics of a single point feature. Let us define $\mathbf{x} = (\boldsymbol{\eta}, \delta) \in \mathbb{R}^4$ and $\mathbf{u} = (\boldsymbol{v}, \boldsymbol{\omega})$ as in Sect. 2.1.3.3. The derivative of δ can be easily calculated as:

$$\dot{\delta} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\|\boldsymbol{p}\|}\right) = \frac{\boldsymbol{p}^T}{\|\boldsymbol{p}\|^3}\dot{\boldsymbol{p}} = \frac{\boldsymbol{p}^T}{\|\boldsymbol{p}\|^3}(-\boldsymbol{v} - [\boldsymbol{p}]_\times\boldsymbol{\omega}) = -\delta^2\boldsymbol{\eta}^T\boldsymbol{v}.$$

Therefore, using also (2.11), we have

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} -\delta(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T) & [\boldsymbol{\eta}]_\times \\ \delta^2\boldsymbol{\eta}^T & \boldsymbol{\emptyset}_3^T \end{bmatrix}\mathbf{u} \\ \boldsymbol{s} = \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\emptyset}_3 \end{bmatrix}\mathbf{x} \end{cases}. \tag{3.6}$$

We begin by defining the zero-th order Lie derivative and its Jacobian:

$$\mathrm{L}^0\boldsymbol{g} = \boldsymbol{\eta}, \quad \nabla_{\mathbf{x}}\mathrm{L}^0\boldsymbol{g}^T = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\emptyset}_3 \end{bmatrix} \tag{3.7}$$

The first order Lie derivatives w.r.t. $\boldsymbol{l}_i$, $i = 1, \ldots 6$ are given by the columns of

$$\mathrm{L}_{\boldsymbol{l}_i}^1\boldsymbol{g} = \text{column } i \text{ of } \begin{bmatrix} -\delta(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T) & [\boldsymbol{\eta}]_\times \end{bmatrix}.$$

Piling up their gradients and (3.7), we conclude

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\emptyset}_3 \\ \delta(\boldsymbol{\eta}^T \boldsymbol{e}_1 \boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{e}_1^T) & -(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T)\boldsymbol{e}_1 \\ \delta(\boldsymbol{\eta}^T \boldsymbol{e}_2 \boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{e}_2^T) & -(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T)\boldsymbol{e}_2 \\ \delta(\boldsymbol{\eta}^T \boldsymbol{e}_3 \boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{e}_3^T) & -(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T)\boldsymbol{e}_3 \\ -[\boldsymbol{e}_1]_\times & \boldsymbol{\emptyset}_3 \\ -[\boldsymbol{e}_2]_\times & \boldsymbol{\emptyset}_3 \\ -[\boldsymbol{e}_3]_\times & \boldsymbol{\emptyset}_3 \end{bmatrix} \begin{array}{l} \\ \leftarrow v_{\boldsymbol{x}} \\ \leftarrow v_{\boldsymbol{y}} \\ \leftarrow v_{\boldsymbol{z}} \\ \leftarrow \omega_{\boldsymbol{x}} \\ \leftarrow \omega_{\boldsymbol{y}} \\ \leftarrow \omega_{\boldsymbol{z}} \end{array}$$

where we highlighted the rows corresponding to each input. It is easy to verify that $\boldsymbol{\mathcal{O}}$ has rank 4 by taking the rows number 1, 2, 3, 4, 8, 12

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 2\delta\boldsymbol{\eta}_{\boldsymbol{x}} & 0 & 0 & \boldsymbol{\eta}_{\boldsymbol{x}}^2 - 1 \\ 0 & 2\delta\boldsymbol{\eta}_{\boldsymbol{y}} & 0 & \boldsymbol{\eta}_{\boldsymbol{y}}^2 - 1 \\ 0 & 0 & 2\delta\boldsymbol{\eta}_{\boldsymbol{z}} & \boldsymbol{\eta}_{\boldsymbol{z}}^2 - 1 \end{bmatrix}$$

and considering that, since $\|\boldsymbol{\eta}\| = 1$, the last column cannot contain only zeros. The Structure from Known Motion problem is, therefore, observable as already shown, e.g. in [SP94]. However, this does not mean that *any* input $\mathbf{u}$ allows to distinguish between different states. In fact if $\boldsymbol{v} = \alpha\boldsymbol{\eta}$ the system reduces to:

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \begin{bmatrix} \boldsymbol{\emptyset}_3 & [\boldsymbol{\eta}]_\times \\ \delta^2 & \boldsymbol{\emptyset}_3^T \end{bmatrix} \mathbf{u} \\ \boldsymbol{s} = \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\emptyset}_3 \end{bmatrix} \mathbf{x} \end{cases} .$$

with now $\mathbf{u} = \boldsymbol{u} = (\alpha, \boldsymbol{\omega})$. If we repeat the calculation of $\boldsymbol{\mathcal{O}}$ we obtain

$$\boldsymbol{\mathcal{O}} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\emptyset}_3 \\ \boldsymbol{\emptyset}_{3\times 3} & \boldsymbol{\emptyset}_3 \\ -[\boldsymbol{e}_1]_\times & \boldsymbol{\emptyset}_3 \\ -[\boldsymbol{e}_2]_\times & \boldsymbol{\emptyset}_3 \\ -[\boldsymbol{e}_3]_\times & \boldsymbol{\emptyset}_3 \end{bmatrix} \begin{array}{l} \\ \leftarrow \alpha \\ \leftarrow \omega_{\boldsymbol{x}} \\ \leftarrow \omega_{\boldsymbol{y}} \\ \leftarrow \omega_{\boldsymbol{z}} \end{array}$$

that can never have rank 4. This is clearly due to the fact that, with this choice of inputs, a part of the system state (δ) does not affect neither the output nor the dynamics of the other state components. In general, in the case of nonlinear systems, together with the above observability conditions, one will have to impose a *persistent excitation* condition on the system inputs. In other words, apart from trivial cases in which the state can be directly measured by inverting $\boldsymbol{s}$, one will

have to guarantee that the inputs are selected in such a way that every component of the state $\mathbf{x}$ has an effect (possibly only for a limited amount of time) on the history of the output $\boldsymbol{s}$.

3.2.3 A nonlinear observer for SfM

In this section we focus our attention to a particular class of nonlinear systems which are typical of SfM problems. For the reader's convenience, we start recalling here a classical formulation of the Persistence of Excitation (PE) Lemma in the context of adaptive control as stated in, e.g., [MT95].

Lemma 3.1 (Persistence of Excitation)**.** *Consider the system*[1]*:*

$$\begin{cases} \dot{\widetilde{\boldsymbol{s}}} = -\boldsymbol{H}\widetilde{\boldsymbol{s}} + \boldsymbol{\Omega}^T(t)\,\widetilde{\boldsymbol{\chi}} \\ \dot{\widetilde{\boldsymbol{\chi}}} = -\alpha\boldsymbol{\Omega}(t)\,\widetilde{\boldsymbol{s}} \end{cases}. \tag{3.8}$$

where $\widetilde{\boldsymbol{s}} \in \mathbb{R}^m$, $\widetilde{\boldsymbol{\chi}} \in \mathbb{R}^p$, $\boldsymbol{H} \succ 0$, *and* $\alpha > 0$. *If* $\|\boldsymbol{\Omega}(t)\|$ *and* $\left\|\dot{\boldsymbol{\Omega}}(t)\right\|$ *are uniformly bounded and the PE condition is satisfied, that is, there exists a time interval* $T > 0$ *and a scalar* $\gamma > 0$ *such that*

$$\int_t^{t+T} \boldsymbol{\Omega}(\tau)\,\boldsymbol{\Omega}^T(\tau)\,\mathrm{d}\tau \succeq \gamma\boldsymbol{I}_p \succ 0, \quad \forall t \geq t_0, \tag{3.9}$$

then $(\widetilde{\boldsymbol{s}}, \widetilde{\boldsymbol{\chi}}) = (\boldsymbol{\emptyset}_m, \boldsymbol{\emptyset}_p)$ *is a globally exponentially stable equilibrium point.*

Let now $\mathbf{x} = [\boldsymbol{s}^T \;\; \boldsymbol{\chi}^T]^T \in \mathbb{R}^{m+p}$ be the state of a dynamical system, where $\boldsymbol{s} \in \mathbb{R}^m$ represents a *measurable* component of $\mathbf{x}$ and $\boldsymbol{\chi} \in \mathbb{R}^p$ an *unmeasurable* one. Assume further that the following holds

$$\begin{cases} \dot{\boldsymbol{s}} = \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{s}, \mathbf{u}) + \boldsymbol{\Omega}^T(\boldsymbol{s}, \mathbf{u})\boldsymbol{\chi} \\ \dot{\boldsymbol{\chi}} = \boldsymbol{f}_{\boldsymbol{\chi}}(\boldsymbol{s}, \boldsymbol{\chi}, \mathbf{u}) \end{cases} \tag{3.10}$$

with $\mathbf{u} \in \mathbb{R}^v$ being an input vector, and $\boldsymbol{\Omega}(t) \in \mathbb{R}^{p\times m}$ a generic but *known* time-varying quantity. Note that in formulation (3.10) vector $\boldsymbol{\chi}$ is required to appear *linearly* in the dynamics of $\boldsymbol{s}$ (first equation). Furthermore, matrix $\boldsymbol{\Omega}(\cdot) \in \mathbb{R}^{p\times m}$ and vectors $\boldsymbol{f}_{\boldsymbol{s}}(\cdot) \in \mathbb{R}^m$ and $\boldsymbol{f}_{\boldsymbol{\chi}}(\cdot) \in \mathbb{R}^p$ are assumed to be generic but *known* and sufficiently smooth functions w.r.t. their arguments which are all available apart from the unknown value of $\boldsymbol{\chi}$ in $\boldsymbol{f}_{\boldsymbol{\chi}}(\cdot)$. For the case of structure from motion for point feature, considering, again, (3.6), one has for example $\boldsymbol{s} = \boldsymbol{\eta}$, $\boldsymbol{\chi} = \delta$, and $\mathbf{u} = \boldsymbol{u}$

$$\begin{cases} \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{s}, \mathbf{u}) = [\boldsymbol{s}]_\times\,\boldsymbol{\omega} \\ \boldsymbol{\Omega}(\boldsymbol{s}, \mathbf{u}) = -(\boldsymbol{I}_3 - \boldsymbol{s}\boldsymbol{s}^T) \\ \boldsymbol{f}_{\boldsymbol{\chi}}(\boldsymbol{s}, \boldsymbol{\chi}, \mathbf{u}) = \boldsymbol{\chi}^2\boldsymbol{s}^T\boldsymbol{v} \end{cases}$$

[1] A more general version of this lemma was discussed in [1]. Such a generalization, however, is not necessary for the developments of the rest of this thesis and therefore will be omitted.

Exploiting (3.8–3.9), a sensible estimation scheme for retrieving the (unmeasurable) value of $\boldsymbol{\chi}$ can be devised, as suggested by [DOR07, RDO08, DOR08], in the following way: let $\widehat{\mathbf{x}} = [\widehat{\boldsymbol{s}},\, \widehat{\boldsymbol{\chi}}]^T \in \mathbb{R}^{n+p}$ be the estimated state, $\widetilde{\boldsymbol{s}} = \widehat{\boldsymbol{s}} - \boldsymbol{s}$, $\widetilde{\boldsymbol{\chi}} = \widehat{\boldsymbol{\chi}} - \boldsymbol{\chi}$, $\widetilde{\mathbf{x}} = [\widetilde{\boldsymbol{s}}^T\ \widetilde{\boldsymbol{\chi}}^T]^T$, and design the following update rule

$$\begin{cases} \dot{\widehat{\boldsymbol{s}}} = \boldsymbol{f}_s(\boldsymbol{s},\, \mathbf{u}) + \boldsymbol{\Omega}^T(\boldsymbol{s},\, \mathbf{u})\widehat{\boldsymbol{\chi}} - \boldsymbol{H}\widetilde{\boldsymbol{s}} \\ \dot{\widehat{\boldsymbol{\chi}}} = \boldsymbol{f}_\chi(\boldsymbol{s},\, \widehat{\boldsymbol{\chi}},\, \mathbf{u}) - \alpha\boldsymbol{\Omega}(\boldsymbol{s},\, \mathbf{u})\widetilde{\boldsymbol{s}} \end{cases}. \tag{3.11}$$

The error dynamics then takes the form

$$\begin{cases} \dot{\widetilde{\boldsymbol{s}}} = -\,\boldsymbol{H}\widetilde{\boldsymbol{s}} + \boldsymbol{\Omega}^T(\boldsymbol{s},\, \mathbf{u})\widetilde{\boldsymbol{\chi}} \\ \dot{\widetilde{\boldsymbol{\chi}}} = -\,\alpha\boldsymbol{\Omega}(\boldsymbol{s},\, \mathbf{u})\widetilde{\boldsymbol{s}} + [\boldsymbol{f}_\chi(\boldsymbol{s},\, \widehat{\boldsymbol{\chi}},\, \mathbf{u}) - \boldsymbol{f}_\chi(\boldsymbol{s},\, \boldsymbol{\chi},\, \mathbf{u})] \\ \quad = -\,\alpha\boldsymbol{\Omega}(\boldsymbol{s},\, \mathbf{u})\widetilde{\boldsymbol{s}} + \boldsymbol{d}(\widetilde{\mathbf{x}}, t) \end{cases}. \tag{3.12}$$

System (3.12) matches almost perfectly the formulation (3.8) apart from the 'spurious' term $\boldsymbol{d}(\widetilde{\mathbf{x}}, t)$. This can be considered as a *vanishing* disturbance for the nominal system (3.8), i.e., such that $\boldsymbol{d}(\boldsymbol{\emptyset}_p, t) = \boldsymbol{\emptyset}_p$. Therefore, it is typically possible to still prove *local* exponential convergence of the origin of (3.12) by resorting to Lyapunov arguments and by imposing suitable bounds on the initial condition $\widetilde{\mathbf{x}}(t_0)$ and/or on $\|\boldsymbol{d}(\widetilde{\mathbf{x}}, t)\|$, see [DOR08, RDO08] for some examples in this sense.

The PE condition (3.9) plays the role of an observability criterion: convergence of the estimation error $\widetilde{\mathbf{x}}(t) \to \boldsymbol{\emptyset}_q$ is possible iff the square matrix $\boldsymbol{\Omega}\,(t)\,\boldsymbol{\Omega}^T\,(t) \in \mathbb{R}^{p\times p}$ keeps being full rank in the integral sense of (3.9). We note that if $m \geq p$, that is, if the number of independent available measurements $\boldsymbol{s}$ is larger or equal to the number of independent estimated quantities $\boldsymbol{\chi}$, then it is in principle possible to *instantaneously* satisfy (3.9) by enforcing

$$\boldsymbol{\Omega}\,(t)\,\boldsymbol{\Omega}^T\,(t) \succeq \frac{\gamma}{T}\boldsymbol{I}_p, \quad \forall t \geq t_0. \tag{3.13}$$

On the other hand, if $m < p$ then $\det(\boldsymbol{\Omega}\,(t)\,\boldsymbol{\Omega}^T\,(t)) \equiv 0$ by construction. Nevertheless, in this case, it could still be possible to satisfy (3.9) in an integral sense if the $l-$dimensional range space of $\boldsymbol{\Omega}\,(t)\,\boldsymbol{\Omega}^T\,(t)$ $(l \leq m)$ can span $\mathbb{R}^p$ during the period T. In this work, however, we will only consider the first situation $m \geq p$ and thus aim at fulfilling the (more restrictive) condition (3.13).

Remark 3.1. *Note that the* local *stability properties of the error dynamics* (3.12) *are due to the perturbation term* $\boldsymbol{d}(\widetilde{\mathbf{x}},\, t)$ *which affects an otherwise globally exponentially stable error system. Indeed, in the special case* $\dot{\boldsymbol{\chi}} = \boldsymbol{\emptyset}_p$ *(unknown but* constant *parameters), one has* $\boldsymbol{d}(\widetilde{\mathbf{x}},\, t) \equiv \boldsymbol{\emptyset}_p$ *and* global *exponential convergence for the error system* (3.12)*. This is, for instance, the case of the structure estimation problems for spherical and cylindrical objects considered in Sects. 4.5.3 and 4.5.4. We stress,*

however, that the estimation scheme (3.11) *is not restricted to this particular situation but can be applied (with, in this case, only local convergence guarantees) to the more general case of* state observation *problems in which the unknown* $\boldsymbol{\chi}$ *is subject to a non-negligible dynamics as in* (3.10)*. The structure estimation for a point feature and a planar surface discussed in Sects. 4.5.1 and 4.5.2 falls in this second class.*

Before concluding this section we want to mention that many other deterministic observers have been proposed in the literature for solving the SfM problem. Some references and a recent comparison of some of these solutions can be found, e.g., in [GBC+15].

3.3 Probabilistic frameworks

Differently from deterministic frameworks, probabilistic ones explicitly take into consideration, in the modeling and design phase, the presence of stochastic terms in the system dynamics. Both the system state $\mathbf{x}$ and the measurements $\boldsymbol{s}$ are treated as random processes and the estimation in general consists in finding a (deterministic) $\widehat{\mathbf{x}}$ that minimizes some statistical property of the estimation error which is also considered as a random process. In this section we briefly present two very common probabilistic frameworks: the Maximum Likelihood Estimator (MLE) is only treated here because it allows to simply introduce the Cramer-Rao bound and the Fisher Information Matrix; the Kalman Filter (KF) will instead be used in Chapt. 5 for a basic comparison between deterministic and probabilistic estimation frameworks.

3.3.1 The Maximum Likelihood Estimator

Assume that one wants to estimate the value of a random vector $\boldsymbol{\chi} \in \mathbb{R}^p$. We use the notation $\boldsymbol{\chi} \sim p_{\boldsymbol{\chi}}(\boldsymbol{\chi})$ to indicate the Probability Density Function (PDF) of $\boldsymbol{\chi}$. The quantity $p_{\boldsymbol{\chi}}(\boldsymbol{\chi})$ is also called the *a priori* PDF of $\boldsymbol{\chi}$ as it represents the confidence that one has about the different values that $\boldsymbol{\chi}$ may have *before* taking any additional measurement. Assume now that a measurement $\boldsymbol{s} \in \mathbb{R}^m$ is taken and that this measurement (also a random vector) is *correlated* with $\boldsymbol{\chi}$, meaning that

$$p_{\boldsymbol{s},\boldsymbol{\chi}}(\boldsymbol{s},\boldsymbol{\chi}) \neq p_{\boldsymbol{s}}(\boldsymbol{s})p_{\boldsymbol{\chi}}(\boldsymbol{\chi})$$

where $p_{\boldsymbol{s},\boldsymbol{\chi}}(\boldsymbol{s},\boldsymbol{\chi})$ is the joint probability density function of the two random vectors or, equivalently,

$$p_{\boldsymbol{\chi}|\boldsymbol{s}}(\boldsymbol{\chi}|\boldsymbol{s}) \neq p_{\boldsymbol{\chi}}(\boldsymbol{\chi})$$

i.e. the observation of $\boldsymbol{s}$ "tells something" about $\boldsymbol{\chi}$ that makes one change his confidence (the PDF) of $\boldsymbol{\chi}$. The quantity $p_{\boldsymbol{\chi}|\boldsymbol{s}}(\boldsymbol{\chi}|\boldsymbol{s})$ is also called the *a posteriori* PDF of $\boldsymbol{\chi}$ as it represents the confidence that one has about the different values that $\boldsymbol{\chi}$ may have *after* taking the measurement $\boldsymbol{s}$. The Maximum Likelihood Estimator (MLE) technique consists in choosing the estimation $\widehat{\boldsymbol{\chi}}$ as the value that maximizes the probability of obtaining the measurement $\boldsymbol{s}$ [BSLK04]

$$\widehat{\boldsymbol{\chi}} = \arg\max_{\boldsymbol{\chi}} p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi}).$$

Assuming that $p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi})$ is differentiable, one can find this value by equating to zero its derivative w.r.t. $\boldsymbol{\chi}$ which gives rise to the *likelihood equation*

$$\nabla_{\boldsymbol{\chi}} p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi})\Big|_{\widehat{\boldsymbol{\chi}}} = \boldsymbol{\emptyset}_p.$$

In practice it is often more convenient to work with the logarithm of the likelihood equation, called the *log-likelihood equation*:

$$\nabla_{\boldsymbol{\chi}} \log p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi})\Big|_{\widehat{\boldsymbol{\chi}}} = \boldsymbol{\emptyset}_p. \tag{3.14}$$

To make an example, assume, that the a priori PDF of $\boldsymbol{\chi}$ is a Gaussian distribution with mean $\mathrm{E}\{\boldsymbol{\chi}\} = \overline{\boldsymbol{\chi}}_0$ and covariance $\mathrm{E}\left\{(\boldsymbol{\chi} - \overline{\boldsymbol{\chi}}_0)(\boldsymbol{\chi} - \overline{\boldsymbol{\chi}}_0)^T\right\} = \boldsymbol{\Sigma}_0 \in \mathbb{R}^{p\times p}$ so that

$$p_{\boldsymbol{\chi}}(\boldsymbol{\chi}) = \frac{1}{\sqrt{(2\pi)^p \det(\boldsymbol{\Sigma}_0)}} e^{-\frac{1}{2}(\boldsymbol{\chi}-\overline{\boldsymbol{\chi}}_0)^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\chi}-\overline{\boldsymbol{\chi}}_0)}. \tag{3.15}$$

We also write $\boldsymbol{\chi} \sim \mathcal{N}(\overline{\boldsymbol{\chi}}_0, \boldsymbol{\Sigma}_0)$. Assume now that the measurement model is linear with additive noise

$$\boldsymbol{s} = \boldsymbol{\Omega}^T \boldsymbol{\chi} + \boldsymbol{v} \tag{3.16}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{m\times p}$ is *deterministic* and $\boldsymbol{v} \in \mathbb{R}^m$ represents the measurement noise, which is assumed to be Gaussian distributed with zero mean and covariance matrix $\mathrm{E}\left\{\boldsymbol{v}\boldsymbol{v}^T\right\} = \boldsymbol{R} \in \mathbb{R}^{m\times m}$. The PDF of $\boldsymbol{s}$ given $\boldsymbol{\chi}$ is easily computed as a multivariate Gaussian distribution

$$p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi}) = \frac{1}{\sqrt{(2\pi)^m \det(\boldsymbol{R})}} e^{-\frac{1}{2}\left(\boldsymbol{s}-\boldsymbol{\Omega}^T\boldsymbol{\chi}\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{s}-\boldsymbol{\Omega}^T\boldsymbol{\chi}\right)}. \tag{3.17}$$

Using (3.14) one obtains the maximum likelihood estimation of $\boldsymbol{\chi}$

$$\widehat{\boldsymbol{\chi}} = \arg\min_{\boldsymbol{\chi}} \left\{\frac{1}{2}\left(\boldsymbol{s} - \boldsymbol{\Omega}^T\boldsymbol{\chi}\right)^T \boldsymbol{R}^{-1}\left(\boldsymbol{s} - \boldsymbol{\Omega}^T\boldsymbol{\chi}\right)\right\} = \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T\right)^{-1}\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{s}. \tag{3.18}$$

Since we have only considered one measurement, in fact, the best we can do is to use a weighted pseudoinverse based on the noise content of each component of the measurement. Note that the estimation is only possible if $\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T \succ 0$. The

information provided by the prior (3.15) was not considered for the calculation of (3.18). The prior can be treated as an additional virtual measurement affected by a Gaussian uncertainty

$$\overline{\boldsymbol{\chi}}_0 = \boldsymbol{\chi} + \boldsymbol{v}_0,\ \boldsymbol{v}_0 \sim \mathcal{N}(\boldsymbol{\emptyset}_p, \boldsymbol{\Sigma}_0) \Rightarrow p_{\overline{\boldsymbol{\chi}}_0|\boldsymbol{\chi}}(\overline{\boldsymbol{\chi}}_0|\boldsymbol{\chi}) = \frac{1}{\sqrt{(2\pi)^q \det(\boldsymbol{\Sigma}_0)}} e^{-\frac{1}{2}(\overline{\boldsymbol{\chi}}_0-\boldsymbol{\chi})^T \boldsymbol{\Sigma}_0^{-1}(\overline{\boldsymbol{\chi}}_0-\boldsymbol{\chi})}.$$

The joint probability distribution of $(\overline{\boldsymbol{\chi}}_0, \boldsymbol{s})$, under the assumption that $\boldsymbol{v}_0$ and $\boldsymbol{v}$ are not correlated, is given by

$$\begin{aligned} p_{\overline{\boldsymbol{\chi}}_0,\boldsymbol{s}|\boldsymbol{\chi}}(\overline{\boldsymbol{\chi}}_0, \boldsymbol{s}|\boldsymbol{\chi}) &= p_{\overline{\boldsymbol{\chi}}_0|\boldsymbol{\chi}}(\overline{\boldsymbol{\chi}}_0|\boldsymbol{\chi}) p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi}) \\ &= c\, e^{-\frac{1}{2}\left[(\boldsymbol{s}-\boldsymbol{\Omega}^T\boldsymbol{\chi})^T \boldsymbol{R}^{-1}(\boldsymbol{s}-\boldsymbol{\Omega}^T\boldsymbol{\chi}) + (\overline{\boldsymbol{\chi}}_0-\boldsymbol{\chi})^T \boldsymbol{\Sigma}_0^{-1}(\overline{\boldsymbol{\chi}}_0-\boldsymbol{\chi})\right]} \end{aligned}$$

where c is a normalizing constant that does not depend of $\boldsymbol{\chi}$ and hence can be ignored. Using again (3.14) one obtains

$$\widehat{\boldsymbol{\chi}} = \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T + \boldsymbol{\Sigma}_0^{-1}\right)^{-1} \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{s} + \boldsymbol{\Sigma}_0^{-1}\overline{\boldsymbol{\chi}}_0\right). \tag{3.19}$$

It can be shown [BSLK04] that this result would also be obtained by considering an alternative estimation technique: the Maximum a Posteriori Estimator (MPE). This latter defines the optimal estimation $\widehat{\boldsymbol{\chi}}$ as the one that maximizes the a posteriori distribution of $\boldsymbol{\chi}$, i.e.

$$\widehat{\boldsymbol{\chi}} = \arg\max_{\boldsymbol{\chi}} p_{\boldsymbol{\chi}|\boldsymbol{s},\overline{\boldsymbol{\chi}}_0}(\boldsymbol{\chi}|\boldsymbol{s}, \overline{\boldsymbol{\chi}}_0).$$

Taking the expected value of the estimation error $\widetilde{\boldsymbol{\chi}} = \widehat{\boldsymbol{\chi}} - \boldsymbol{\chi}$ with $\widehat{\boldsymbol{\chi}}$ in (3.19), one has

$$\begin{aligned} \mathrm{E}\{\widehat{\boldsymbol{\chi}} - \boldsymbol{\chi}\} = \mathrm{E}\{\widehat{\boldsymbol{\chi}}\} - \mathrm{E}\{\boldsymbol{\chi}\} &= \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T + \boldsymbol{\Sigma}_0^{-1}\right)^{-1} \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\,\mathrm{E}\{\boldsymbol{s}\} + \boldsymbol{\Sigma}_0^{-1}\overline{\boldsymbol{\chi}}_0\right) - \overline{\boldsymbol{\chi}}_0 \\ &= \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T + \boldsymbol{\Sigma}_0^{-1}\right)^{-1} \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T\,\mathrm{E}\{\boldsymbol{\chi}\} + \boldsymbol{\Sigma}_0^{-1}\overline{\boldsymbol{\chi}}_0\right) - \overline{\boldsymbol{\chi}}_0 = \boldsymbol{\emptyset}_p \end{aligned}$$

and the estimation $\widehat{\boldsymbol{\chi}}$ is said to be *unbiased*.

The MLE can easily be extended to the case in which one takes multiple independent measurements $\boldsymbol{s}_k = \boldsymbol{s}(t_k), k = 1, \ldots K$, which are different samples of a random process $\boldsymbol{s}(t)$, and wants to use them, together with the initial prior (3.15), to estimate the value $\boldsymbol{\chi}_k$ of another random process $\boldsymbol{\chi}(t)$ at the same instants t_k. We indicate with $\mathcal{X} = \boldsymbol{\chi}_0, \boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_K$ the set of unknowns and with $\mathcal{S} = \overline{\boldsymbol{\chi}}_0, \boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_K$ the set of measurements. Considering a generic nonlinear measurement model with additive Gaussian noise

$$\boldsymbol{s}(t_k) = \boldsymbol{g}(\boldsymbol{\chi}(t_k), t_k) + \boldsymbol{v}(t_k)$$

with $\boldsymbol{v}_k = \boldsymbol{v}(t_k) \sim \mathcal{N}(\boldsymbol{\emptyset}_m, \boldsymbol{R}_k)$ and assuming that the variables $\boldsymbol{s}_k$, $\boldsymbol{\chi}_k$ and $\boldsymbol{v}_k$ are all independent, we can write the joint PDF of the measurements as

$$p_{\mathcal{S}|\mathcal{X}}(\mathcal{S}|\mathcal{X}) = c\, e^{\left[-\frac{1}{2}\sum_{k=0}^{K}(\boldsymbol{s}_k - \boldsymbol{g}(\boldsymbol{\chi}_k,t_k))^T \boldsymbol{R}_k^{-1}(\boldsymbol{s}_k - \boldsymbol{g}(\boldsymbol{\chi}_k,t_k))\right]}$$

where we used $\boldsymbol{s}_0 = \boldsymbol{\chi}_0$, $\boldsymbol{g}(\boldsymbol{\chi}(t_0), t_0) = \overline{\boldsymbol{\chi}}_0$, $\boldsymbol{R}_0 = \boldsymbol{\Sigma}_0$, and c is a normalizing factor that does not depend on $\mathcal{X}$ and thus can be ignored. Therefore the MLE (3.14) returns

$$\widehat{\mathcal{X}} = \arg\min_{\mathcal{X}} \left[\frac{1}{2} \sum_{k=1}^{K} (\boldsymbol{s}_k - \boldsymbol{g}(\boldsymbol{\chi}_k, t_k))^T \boldsymbol{R}_k^{-1} (\boldsymbol{s}_k - \boldsymbol{g}(\boldsymbol{\chi}_k, t_k)) \right]$$

which is a Nonlinear Least Squares (NLS) problem. This kind of batch resolution strategies, based on MLE and NLS, have been exploited in many SLAM works where $\boldsymbol{\chi}_k$ typically represents a set of robot and landmark locations and $\boldsymbol{s}_k$ is the set of all taken measurements. The problem in this case also takes the name of trajectory Smoothing And Mapping (SAM) [DK06, KRD08] because of its similarity to the signal smoothing problem. In this context, in general, one also exploits the fact that the different robot poses $\boldsymbol{\chi}_k$ are not independent but, instead, they are related by the dynamics of the system:

$$\boldsymbol{\chi}_k = \boldsymbol{f}(\boldsymbol{\chi}_{k-1}, \mathbf{u}_k) + \boldsymbol{w}_k$$

where $\mathbf{u}_k \in \mathbb{R}^v$ is a control input and $\boldsymbol{w}_k \in \mathbb{R}^p$ is a Gaussian distributed process noise $\boldsymbol{w}_k \sim \mathcal{N}(\boldsymbol{\emptyset}_p, \boldsymbol{Q}_k)$. Therefore one can write

$$p_{\boldsymbol{\chi}_k|\boldsymbol{\chi}_{k-1}}(\boldsymbol{\chi}_k|\boldsymbol{\chi}_{k-1}) = e^{-\frac{1}{2}(\boldsymbol{\chi}_k - \boldsymbol{f}(\boldsymbol{\chi}_{k-1}, \mathbf{u}_k))^T \boldsymbol{Q}_k^{-1} (\boldsymbol{\chi}_k - \boldsymbol{f}(\boldsymbol{\chi}_{k-1}, \mathbf{u}_k))}.$$

The joint measurements PDF can be easily obtained, by exploiting the uncorrelation, as a simple product between the single Gaussian PDFs

$$\begin{aligned} p_{\mathcal{S}|\mathcal{X}}(\mathcal{S}|\mathcal{X}) &= c\, p_{\overline{\boldsymbol{\chi}}_0|\boldsymbol{\chi}_0}(\overline{\boldsymbol{\chi}}_0|\boldsymbol{\chi}_0) \prod_{k=1}^{K} p_{\boldsymbol{\chi}_k|\boldsymbol{\chi}_{k-1}}(\boldsymbol{\chi}_k|\boldsymbol{\chi}_{k-1}) p_{\boldsymbol{s}_k|\boldsymbol{\chi}_k}(\boldsymbol{s}_k|\boldsymbol{\chi}_k) \\ &= c\, e^{-\frac{1}{2}\left[(\boldsymbol{\chi}_0 - \overline{\boldsymbol{\chi}}_0)^T \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\chi}_0 - \overline{\boldsymbol{\chi}}_0) + \sum_{k=1}^{K} (\boldsymbol{\chi}_k - \boldsymbol{f}_k)^T \boldsymbol{Q}_k^{-1} (\boldsymbol{\chi}_k - \boldsymbol{f}_k) + (\boldsymbol{s}_k - \boldsymbol{g}_k)^T \boldsymbol{R}_k^{-1} (\boldsymbol{s}_k - \boldsymbol{g}_k)\right]} \end{aligned}$$

where c is a normalization factor that does not depend on $\mathcal{X}$ and, for brevity of notation, we wrote $\boldsymbol{f}_k = \boldsymbol{f}(\boldsymbol{\chi}_{k-1}, \mathbf{u}_k)$ and $\boldsymbol{g}_k = \boldsymbol{g}(\boldsymbol{\chi}_k, t_k)$. The application of (3.14) leads to [KGS+11]

$$\begin{aligned} \widehat{\mathcal{X}} = \arg\min_{\mathcal{X}} \frac{1}{2} \Bigg[& (\boldsymbol{\chi}_0 - \overline{\boldsymbol{\chi}}_0)^T \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\chi}_0 - \overline{\boldsymbol{\chi}}_0) \\ & + \sum_{k=1}^{K} (\boldsymbol{\chi}_k - \boldsymbol{f}_k)^T \boldsymbol{Q}_k^{-1} (\boldsymbol{\chi}_k - \boldsymbol{f}_k) + (\boldsymbol{s}_k - \boldsymbol{g}_k)^T \boldsymbol{R}_k^{-1} (\boldsymbol{s}_k - \boldsymbol{g}_k) \Bigg] \end{aligned}$$

Different techniques can be used to find the numerical solution to the NLS problem [DS96]. Iterative methods, such as Gauss-Newton or Levenberg-Marquardt algorithms are often preferred because of their efficiency [KGS+11]. Incremental NLS solutions have also been proposed in [KJR+11, PSI+13] to reduce computational time by exploiting the sparsity of the SLAM problem and efficient block-matrix operations.

3.3.2 The Fisher Information Matrix and the Cramer-Rao bound

Let us calculate the estimation error covariance for the MLE (3.19). Assuming that $\boldsymbol{\chi}$ and $\boldsymbol{v}$ are not correlated and after some tedious but straightforward calculation one obtains

$$\boldsymbol{\Sigma} = \mathrm{E}\left\{(\widehat{\boldsymbol{\chi}} - \boldsymbol{\chi})^T (\widehat{\boldsymbol{\chi}} - \boldsymbol{\chi})\right\} = \left(\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}. \tag{3.20}$$

It is clear, than, that the quantity $\boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T$ represents the "gain", in terms of reduction of uncertainty, that one has by using $\boldsymbol{s}$ to ameliorate the estimation of $\boldsymbol{\chi}$. This concept is formalized by the so called Fisher Information Matrix (FIM)

$$\boldsymbol{\mathcal{I}}_F = -\mathrm{E}\left\{(\nabla_{\boldsymbol{\chi}} \log p_{\mathcal{S}|\boldsymbol{\chi}}(\mathcal{S}|\boldsymbol{\chi}))(\nabla_{\boldsymbol{\chi}} \log p_{\mathcal{S}|\boldsymbol{\chi}}(\mathcal{S}|\boldsymbol{\chi}))^T\right\}. \tag{3.21}$$

The quantity $\nabla_{\boldsymbol{\chi}} \log p_{\mathcal{S}|\boldsymbol{\chi}}(\mathcal{S}|\boldsymbol{\chi})$, the gradient of the log-likelihood function that appears in (3.14), is also called the *score* of the measurement $\mathcal{S}$ since it represents the amount of information that $\mathcal{S}$ contains about $\mathcal{X}$ or the sensitivity of the measurements w.r.t. the unknown quantities. One can also prove [BSLK04] that the following definition of $\boldsymbol{\mathcal{I}}_F$ is equivalent to (3.21)

$$\boldsymbol{\mathcal{I}}_F = -\mathrm{E}\left\{\nabla\nabla_{\boldsymbol{\chi}}^T \log p_{\mathcal{S}|\boldsymbol{\chi}}(\mathcal{S}|\boldsymbol{\chi})\right\} \tag{3.22}$$

and hence the FIM represents the curvature of the log-likelihood function that appears in (3.14). As such it represents a quantitative measure of the "quality" of the maximum point found with (3.14). The importance of the FIM lies in the fact that it can be proved (see again [BSLK04]) that for any unbiased estimation $\widehat{\boldsymbol{\chi}}$, the covariance of the estimation error satisfies the *Cramer-Rao lower bound*:

$$\mathrm{E}\left\{(\widehat{\boldsymbol{\chi}} - \boldsymbol{\chi})^T (\widehat{\boldsymbol{\chi}} - \boldsymbol{\chi})\right\} \succeq \boldsymbol{\mathcal{I}}_F^{-1}. \tag{3.23}$$

In our first example, we used the prior (3.15) and the measurement (3.17), therefore $\mathcal{S} = \{\overline{\boldsymbol{\chi}}_0, \boldsymbol{s}\}$ and

$$\begin{aligned} -\log\left(p_{\mathcal{S}|\boldsymbol{\chi}}(\mathcal{S}|\boldsymbol{\chi})\right) &= -\log\left(p_{\boldsymbol{\chi}}(\boldsymbol{\chi}) p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi})\right) = -\log p_{\boldsymbol{\chi}}(\boldsymbol{\chi}) - \log\left(p_{\boldsymbol{s}|\boldsymbol{\chi}}(\boldsymbol{s}|\boldsymbol{\chi})\right) \\ &= (\boldsymbol{\chi} - \overline{\boldsymbol{\chi}}_0)^T \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\chi} - \overline{\boldsymbol{\chi}}_0) + \left(\boldsymbol{s} - \boldsymbol{\Omega}^T\boldsymbol{\chi}\right)^T \boldsymbol{R}^{-1} \left(\boldsymbol{s} - \boldsymbol{\Omega}^T\boldsymbol{\chi}\right) \end{aligned}$$

and taking the expected value of the second order partial derivatives w.r.t. $\boldsymbol{\chi}$ we conclude

$$\boldsymbol{\mathcal{I}}_F = \boldsymbol{\Omega}\boldsymbol{R}^{-1}\boldsymbol{\Omega}^T + \boldsymbol{\Sigma}_0^{-1}$$

which is equal to the covariance of the MLE in (3.20). In the case of MLE, then, the Cramer-Rao bound(3.23) is satisfied with the equal sign and therefore the estimator is called *efficient*.

The Fisher information matrix for the multiple measurement case can also be easily computed as

$$\boldsymbol{\mathcal{I}}_F = \boldsymbol{\Sigma}_0^{-1} + \sum_{k=1}^{K} \nabla_{\boldsymbol{\chi}_k} \boldsymbol{g}(\boldsymbol{\chi}_k, t_k)^T \boldsymbol{R}_k^{-1} \nabla_{\boldsymbol{\chi}_k} \boldsymbol{g}(\boldsymbol{\chi}_k, t_k).$$

In a simple case, analogous to (3.16), where one has a linear time varying measurement model $\boldsymbol{s}_k = \boldsymbol{\Omega}_k^T \boldsymbol{\chi}_k + \boldsymbol{v}_k$ this results in

$$\boldsymbol{\mathcal{I}}_F = \boldsymbol{\Sigma}_0^{-1} + \sum_{k=1}^{K} \boldsymbol{\Omega}_k \boldsymbol{R}_k^{-1} \boldsymbol{\Omega}_k^T. \tag{3.24}$$

Note the resemblance between the expression (3.24) and the quantity involved in the PE condition (3.9). As a matter of fact (3.24) is the expression of the FIM that one would obtain from system (3.10) by assuming as measurement the quantity $\dot{\boldsymbol{s}} - \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{s}, \mathbf{u})$. In SfM, this corresponds to the de-rotated optical flow field, sometimes used for 3-D reconstruction [GBSR15]. The similarity is more evident if we consider the expression of (3.24) for $\boldsymbol{R}_k = r_{\boldsymbol{s}}^2 \boldsymbol{I}_m$ and in the limit for $\Delta_t = t_k - t_{k-1} \to 0$. With this assumption the summation can be approximated by a continuous time integral in a similar way to what done in [WSM14]

$$\boldsymbol{\mathcal{I}}_F = \boldsymbol{\Sigma}_0^{-1} + \frac{1}{r_{\boldsymbol{s}}^2} \sum_{k=1}^{K} \boldsymbol{\Omega}_k \boldsymbol{\Omega}_k^T \approx \boldsymbol{\mathcal{I}}_0 + i_{\boldsymbol{s}} \int_t^{t+T} \boldsymbol{\Omega}(\tau) \boldsymbol{\Omega}(\tau)^T \, \mathrm{d}\tau.$$

where $\boldsymbol{\mathcal{I}}_0 = \boldsymbol{\Sigma}_0^{-1}$ and $i_{\boldsymbol{s}} = \frac{1}{r_{\boldsymbol{s}}^2}$ represent the amount of information contained in the initial prior and in each measurement. We can then conclude that:

$$\int_t^{t+T} \boldsymbol{\Omega}(\tau) \boldsymbol{\Omega}(\tau)^T \, \mathrm{d}\tau = \frac{\boldsymbol{\mathcal{I}}_F - \boldsymbol{\mathcal{I}}_0}{i_{\boldsymbol{s}}}$$

i.e. the integral in (3.9) can be interpreted as an index of the *efficiency* of the experiment: how much information was acquired during the experiment ($\boldsymbol{\mathcal{I}}_F - \boldsymbol{\mathcal{I}}_0$) w.r.t. the information contained in the measurements ($i_{\boldsymbol{s}}$). Note that the FIM (3.22) and the PE matrix in (3.9) are characteristic properties of the system and of the experiment *only* and do not depend in any way of the kind of estimator that is used to recover $\widehat{\boldsymbol{\chi}}$. The Cramer-Rao bound (3.23), on the contrary, qualifies the particular estimation scheme that is being used: only an efficient estimator will make correctly use of all the available information and produce an estimation with the least possible uncertainty.

We conclude this section by reporting the expression of the FIM for nonlinear continuous-process discrete-measurement systems of the form:

$$\begin{cases} \dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}, t) \\ \boldsymbol{s}_k = \boldsymbol{g}(\mathbf{x}_k, t_k) + \boldsymbol{v}_k \end{cases}.$$

Our derivation is very similar to [Tay79], however, instead of computing the FIM relative to the most recent system state $\mathbf{x}_K$ we are interested in knowing the FIM w.r.t. the estimation of the initial state $\mathbf{x}_0$: this will respond the question of how much information the measurements $\boldsymbol{s}_k, k = 1, \ldots, K$ give about the initial state $\mathbf{x}_0$ of the system. This is a more similar question to the one addressed in observability analysis. Following [Tay79] we can consider $\mathcal{S} = \{\overline{\mathbf{x}}_0, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_K\}$, $\mathcal{X} = \{\mathbf{x}_0, \ldots, \mathbf{x}_K\}$ and write

$$-\log\left(p_{\mathcal{S}|\mathcal{X}}(\mathcal{S}|\mathcal{X})\right) = c + \left(\mathbf{x}_0 - \overline{\mathbf{x}}_0\right)^T \boldsymbol{\Sigma}_0^{-1} \left(\mathbf{x}_0 - \widehat{\mathbf{x}}_0\right) + \sum_{k=1}^{K} \left(\boldsymbol{s}_k - \boldsymbol{g}_k\right)^T \boldsymbol{R}_k^{-1} \left(\boldsymbol{s}_k - \boldsymbol{g}_k\right)$$

and taking the second order differentials w.r.t. $\mathbf{x}_0$ we conclude

$$\boldsymbol{\mathcal{I}}_F = \boldsymbol{\Sigma}_0^{-1} + \sum_{k=1}^{K} \boldsymbol{\Phi}_{k,0}^T \boldsymbol{C}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{C}_k \boldsymbol{\Phi}_{k,0}$$

where

$$\boldsymbol{C}_k = \nabla_{\mathbf{x}_k} \boldsymbol{g}(\mathbf{x}_k, t_k)^T$$

and $\boldsymbol{\Phi}_{k,0} = \boldsymbol{\Phi}(t_k, t_0)$ is the solution to the ordinary differential equation

$$\frac{\partial}{\partial t} \boldsymbol{\Phi}(t, t_0) = \nabla_{\mathbf{x}} \boldsymbol{f}(\mathbf{x}, \mathbf{u}, t)^T \boldsymbol{\Phi}(t, t_0)$$

from $t = t_0$ to $t = t_k$ subject to the initial condition

$$\boldsymbol{\Phi}(t_0, t_0) = \boldsymbol{I}_q.$$

If we consider the linear time varying system (3.4), we have

$$\boldsymbol{C}_k = \boldsymbol{C}(t_k), \quad \frac{\partial}{\partial t} \boldsymbol{\Phi}(t, t_0) = \boldsymbol{A}(t) \boldsymbol{\Phi}(t, t_0)$$

so $\boldsymbol{\Phi}_{k,0}$ is the state transition matrix corresponding to $\boldsymbol{A}(t)$ and we have

$$\boldsymbol{\mathcal{I}}_F = \boldsymbol{\Sigma}_0^{-1} + \sum_{k=1}^{K} \boldsymbol{\Phi}_{k,0}^T \boldsymbol{C}_k^T \boldsymbol{R}_k^{-1} \boldsymbol{C}_k \boldsymbol{\Phi}_{k,0}. \tag{3.25}$$

Note the strong similarity between (3.25) and the OG in (3.5). The similarity is more evident if we assume $\boldsymbol{R}_k^{-1} = i_{\boldsymbol{s}} \boldsymbol{I}_m$, we write $\boldsymbol{\Sigma}_0^{-1} = \boldsymbol{\mathcal{I}}_0$ and we make the sample time $\Delta t = t_k - t_{k-1}$ tend to zero. Then we have

$$\frac{\boldsymbol{\mathcal{I}}_F(t) - \boldsymbol{\mathcal{I}}_0}{i_{\boldsymbol{s}}} = \int_{t_0}^{t} \boldsymbol{\Phi}(\tau, t_0)^T \boldsymbol{C}(\tau) \boldsymbol{C}(\tau) \boldsymbol{\Phi}(\tau, t_0) \, \mathrm{d}\tau = \boldsymbol{\mathcal{G}}(t).$$

We can conclude that the OG (3.5), the PE condition (3.9) and the FIM (3.22) are, in essence, just different forms of the same intuitive concept: regardless of the type of estimation algorithm that one decides to use, the maximum possible performance

(in terms of reduced final uncertainty or converge rate) is determined, all gains being equal, by the amount of information that is contained in the measurements. This latter is an intrinsic characteristic of the system and, in the nonlinear case, of the trajectory followed by the system during the experiment. If one has control on the system evolution, through the input $\mathbf{u}$, one can then engage in an *Experiment Design* problem, i.e. in the study of optimal input profiles $\mathbf{u}(\tau), \tau \in [t_0, t]$ that result in the maximum information gain. This problem and how to conciliate it with the execution of additional tasks, is the main focus of this work.

3.3.3 The Kalman-Bucy filter

Probably the most well known and widely used probabilistic state estimator is the Kalman Filter (KF), sometimes called the Kalman-Bucy Filter when presented in its continuous-time version. This algorithm takes its name from Rudolf E. Kalman, who introduced the first discrete-time version of the filter in [Kal60], and Richard S. Bucy, who contributed extending the filter to continuous time in [KB61]. The filter, originally intended for linear system, was later extended to nonlinear system dynamics by the NASA Ames Research Center where it was incorporated in the Apollo on-board computer for estimating the trajectory of the lunar module [SSM62, MS85]. An interesting historical perspective as well as an overview of past and current applications of Kalman filtering can be found in [GA10].

Consider a generic time-varying linear system $\mathcal{P}$ with state-space dynamic equations:

$$\mathcal{P}: \quad \begin{cases} \dot{\mathbf{x}} = \boldsymbol{A}(t)\mathbf{x} + \boldsymbol{B}(t)\mathbf{u} + \boldsymbol{G}(t)\boldsymbol{w} \\ \boldsymbol{s} = \boldsymbol{C}(t)\mathbf{x} + \boldsymbol{D}(t)\mathbf{u} + \boldsymbol{H}(t)\boldsymbol{w} + \boldsymbol{v} \end{cases} \tag{3.26}$$

where $\mathbf{x} \in \mathbb{R}^q$ is the system dynamic *state*, $\mathbf{u} \in \mathbb{R}^v$ is the *input* vector, $\boldsymbol{s} \in \mathbb{R}^m$ is the measurable *output*, $\boldsymbol{A}(t) \in \mathbb{R}^{q\times q}$ is the *system matrix*, $\boldsymbol{B}(t) \in \mathbb{R}^{q\times v}$ is the *input matrix*, $\boldsymbol{C}(t) \in \mathbb{R}^{m\times q}$ is the *output matrix*, $\boldsymbol{G}(t) \in \mathbb{R}^{q\times w}$, $\boldsymbol{H}(t) \in \mathbb{R}^{m\times w}$ and finally, $\boldsymbol{w} \in \mathbb{R}^w$ and $\boldsymbol{v} \in \mathbb{R}^m$ are two Gaussian-distributed random processes with zero mean and known covariance matrices, i.e.

$$\begin{aligned} \mathrm{E}\left\{\boldsymbol{w}(t)\right\} = \boldsymbol{\emptyset}_w, \quad & \mathrm{E}\left\{\boldsymbol{w}(t)\boldsymbol{w}(\tau)^T\right\} = \delta(t-\tau)\boldsymbol{Q}(t), \\ \mathrm{E}\left\{\boldsymbol{v}(t)\right\} = \boldsymbol{\emptyset}_m, \quad & \mathrm{E}\left\{\boldsymbol{v}(t)\boldsymbol{v}(\tau)^T\right\} = \delta(t-\tau)\boldsymbol{R}(t), \\ & \mathrm{E}\left\{\boldsymbol{w}(t)\boldsymbol{v}(\tau)^T\right\} = \delta(t-\tau)\boldsymbol{M}(t), \end{aligned} \tag{3.27}$$

where δ is the Dirac's delta function. We further assume that $\mathcal{P}$ is observable according to the definition introduced in Sect. 3.2.1 and that the initial state $\mathbf{x}(t_0)$ is a Gaussian random vector with known mean and covariance matrix

$$\mathrm{E}\left\{\mathbf{x}(t_0)\right\} = \overline{\mathbf{x}}_0, \quad \mathrm{E}\left\{\left[\mathbf{x}(t_0) - \overline{\mathbf{x}}_0\right]\left[\mathbf{x}(t_0) - \overline{\mathbf{x}}_0\right]^T\right\} = \boldsymbol{\Sigma}_0$$

and that $\mathbf{x}_0$ is independent on $\boldsymbol{w}$ and $\boldsymbol{v}$.

The structure of the KF is similar to (3.2), but uses a time varying update gain $\boldsymbol{K}(t) \in \mathbb{R}^{q\times m}$:

$$\dot{\widehat{\mathbf{x}}} = \boldsymbol{A}(t)\widehat{\mathbf{x}} + \boldsymbol{B}(t)\mathbf{u} - \boldsymbol{K}(t)\left[\boldsymbol{C}(t)\widehat{\mathbf{x}} + \boldsymbol{D}(t)\mathbf{u} - \boldsymbol{s}\right]. \tag{3.28}$$

Observer (3.28) generates an unbiased estimation of $\mathbf{x}$ as it can be easily verified by calculating the expected value of the estimation error. If the filter is initialized with $\widehat{\mathbf{x}}(t_0) = \overline{\mathbf{x}}_0$, then

$$\mathrm{E}\left\{\widetilde{\mathbf{x}}(t_0)\right\} = \mathrm{E}\left\{\widehat{\mathbf{x}}(t_0) - \mathbf{x}(t_0)\right\} = \mathrm{E}\left\{\overline{\mathbf{x}}_0 - \mathbf{x}(t_0)\right\} = \overline{\mathbf{x}}_0 - \mathrm{E}\left\{\mathbf{x}(t_0)\right\} = \boldsymbol{\emptyset}_q.$$

Moreover, by subtracting (3.26–3.28), one obtains the error dynamics:

$$\dot{\widetilde{\mathbf{x}}} = \left[\boldsymbol{A}(t) - \boldsymbol{K}(t)\boldsymbol{C}(t)\right]\widetilde{\mathbf{x}} - \left[\boldsymbol{G}(t) - \boldsymbol{K}(t)\boldsymbol{H}(t)\right]\boldsymbol{w} + \boldsymbol{K}(t)\boldsymbol{v}. \tag{3.29}$$

Since $\boldsymbol{A}(t)$, $\boldsymbol{K}(t)$, $\boldsymbol{C}(t)$, $\boldsymbol{G}(t)$ and $\boldsymbol{H}(t)$ are deterministic matrices and $\mathrm{E}\left\{\boldsymbol{w}\right\} = \boldsymbol{\emptyset}_w$, $\mathrm{E}\left\{\boldsymbol{v}\right\} = \boldsymbol{\emptyset}_m$ by hypothesis, the expected value of the error derivative in (3.29) is

$$\mathrm{E}\left\{\dot{\widetilde{\mathbf{x}}}\right\} = \frac{\mathrm{d}\,\mathrm{E}\left\{\widetilde{\mathbf{x}}\right\}}{\mathrm{d}t} = \left[\boldsymbol{A}(t) - \boldsymbol{K}(t)\boldsymbol{C}(t)\right]\mathrm{E}\left\{\widetilde{\mathbf{x}}\right\},$$

and remains identically zero if $\mathrm{E}\left\{\widetilde{\mathbf{x}}(t_0)\right\} = \boldsymbol{\emptyset}_q$ as already shown.

Up to this point, the KF does not differ in any way from the deterministic Luenberger Observer (LO) introduced in Sect. 3.2.1. The core difference between the two estimators is, in fact, in the computation of the update gain $\boldsymbol{K}(t)$. As a matter of fact, in the KF, this gain is selected *online* as an optimal trade-off between the process and measurement noise contents. As shown in, e.g., [KB61] and briefly reported in Appendix A.1.1, the dynamics of the error covariance matrix $\boldsymbol{\Sigma}(t) = \mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\widetilde{\mathbf{x}}(t)^T\right\}$ is given by

$$\begin{aligned}\dot{\boldsymbol{\Sigma}}(t) &= \left[\boldsymbol{A}(t) - \boldsymbol{K}(t)\boldsymbol{C}(t)\right]\boldsymbol{\Sigma}(t) + \boldsymbol{\Sigma}(t)\left[\boldsymbol{A}(t) - \boldsymbol{K}(t)\boldsymbol{C}(t)\right]^T \\ &\quad + \left[\boldsymbol{G}(t) - \boldsymbol{K}(t)\boldsymbol{H}(t)\right]\boldsymbol{Q}(t)\left[\boldsymbol{G}(t) - \boldsymbol{K}(t)\boldsymbol{H}(t)\right]^T + \boldsymbol{K}(t)\boldsymbol{R}(t)\boldsymbol{K}(t)^T \\ &\quad - \boldsymbol{K}(t)\boldsymbol{M}(t)\left[\boldsymbol{G}(t) - \boldsymbol{K}(t)\boldsymbol{H}(t)\right]^T - \left[\boldsymbol{G}(t) - \boldsymbol{K}(t)\boldsymbol{H}(t)\right]\boldsymbol{M}(t)^T\boldsymbol{K}(t)^T\end{aligned} \tag{3.30}$$

while the initial error covariance can be easily computed as

$$\begin{aligned}\boldsymbol{\Sigma}(t_0) &= \mathrm{E}\left\{\widetilde{\mathbf{x}}(t_0)\widetilde{\mathbf{x}}(t_0)^T\right\} = \mathrm{E}\left\{\left[\widehat{\mathbf{x}}(t_0) - \mathbf{x}(t_0)\right]\left[\widehat{\mathbf{x}}(t_0) - \mathbf{x}(t_0)\right]^T\right\} \\ &= \mathrm{E}\left\{\left[\overline{\mathbf{x}}_0 - \mathbf{x}(t_0)\right]\left[\overline{\mathbf{x}}_0 - \mathbf{x}(t_0)\right]^T\right\} = \boldsymbol{\Sigma}_0.\end{aligned}$$

The Pontryagin minimum principle can be used to minimize the cost functional $\mathrm{E}\left\{\|\widetilde{\mathbf{x}}\|^2\right\}$, as shown in [AT67] and in Appendix A.1.1, resulting in the optimal update gain

$$\boldsymbol{K}(t) = \left[\boldsymbol{\Sigma}(t)\boldsymbol{C}(t)^T + \check{\boldsymbol{M}}(t)\right]\check{\boldsymbol{R}}(t)^{-1}, \tag{3.31a}$$

with

$$\check{\boldsymbol{M}}(t) = \boldsymbol{G}(t)\left[\boldsymbol{Q}(t)\boldsymbol{H}(t)^T + \boldsymbol{M}(t)^T\right], \tag{3.31b}$$

$$\check{\boldsymbol{R}}(t) = \boldsymbol{H}(t)\boldsymbol{Q}(t)\boldsymbol{H}(t)^T + \boldsymbol{R}(t) + \boldsymbol{H}(t)\boldsymbol{M}(t)^T + \boldsymbol{M}(t)\boldsymbol{H}(t)^T. \tag{3.31c}$$

Injecting (3.31a) in (3.30) and wrapping everything up, one finally obtains the optimal KF dynamics for estimating the state of (3.26):

$$\begin{cases} \dot{\boldsymbol{\Sigma}}(t) = \boldsymbol{A}(t)\boldsymbol{\Sigma}(t) + \boldsymbol{\Sigma}(t)\boldsymbol{A}(t)^T + \boldsymbol{G}(t)\boldsymbol{Q}(t)\boldsymbol{G}(t)^T - \boldsymbol{K}(t)\check{\boldsymbol{R}}(t)\boldsymbol{K}(t)^T & \text{(3.31d)} \\ \dot{\hat{\mathbf{x}}} = \boldsymbol{A}(t)\hat{\mathbf{x}} + \boldsymbol{B}(t)\mathbf{u} - \boldsymbol{K}(t)\left[\boldsymbol{C}(t)\hat{\mathbf{x}} + \boldsymbol{D}(t)\mathbf{u} - \boldsymbol{s}\right] & \text{(3.31e)} \end{cases}.$$

Note that the computation of $\boldsymbol{K}(t)$ in (3.31a) requires knowledge of the current $\boldsymbol{\Sigma}(t)$ which can only be obtained by integrating (3.31d) over time. This represents a considerable additional computational effort w.r.t. the non-optimal LO described in Sect. 3.2.1.

An alternative, and equivalent, form of the KF is the so-called Information Filter (IF) which is based on use of the *canonical parametrization* of the multivariate Gaussian distribution that can be obtained by using the transformation

$$\begin{cases} \boldsymbol{i}(t) = \boldsymbol{\Sigma}(t)^{-1}\mathbf{x}(t) & \text{(3.32a)} \\ \boldsymbol{\mathcal{I}}(t) = \boldsymbol{\Sigma}(t)^{-1} & \text{(3.32b)} \end{cases}.$$

$\boldsymbol{i}(t) \in \mathbb{R}^q$ and $\boldsymbol{\mathcal{I}}(t) \in \mathbb{R}^{q\times q}$ are usually called the *information vector* and *information matrix* respectively. To avoid confusion with the FIM described in Sect. 3.3.2, however, we prefer to refer to $\boldsymbol{\mathcal{I}}(t)$ as the *precision matrix* as suggested in [TBF05].

The dynamics of $\boldsymbol{\mathcal{I}}(t)$ can be easily calculated by considering that $\boldsymbol{\mathcal{I}}(t)\boldsymbol{\Sigma}(t) = \boldsymbol{I}_q$ and hence, taking the time derivative of both sides, one obtains

$$\dot{\boldsymbol{\mathcal{I}}}(t)\boldsymbol{\Sigma}(t) + \boldsymbol{\mathcal{I}}(t)\dot{\boldsymbol{\Sigma}}(t) = \boldsymbol{\varnothing}_{q\times q} \Rightarrow \dot{\boldsymbol{\mathcal{I}}}(t) = -\boldsymbol{\mathcal{I}}(t)\dot{\boldsymbol{\Sigma}}(t)\boldsymbol{\mathcal{I}}(t). \tag{3.33}$$

As for $\boldsymbol{i}(t)$ one obviously has:

$$\dot{\boldsymbol{i}}(t) = \dot{\boldsymbol{\mathcal{I}}}(t)\mathbf{x}(t) + \boldsymbol{\mathcal{I}}(t)\dot{\mathbf{x}}(t). \tag{3.34}$$

Using (3.31, (3.33) and (3.34), one obtains the propagation equations for the IF

$$\begin{cases} \dot{\boldsymbol{\mathcal{I}}}(t) = -\boldsymbol{\mathcal{I}}(t)\boldsymbol{A}(t) - \boldsymbol{A}(t)^T\boldsymbol{\mathcal{I}}(t) - \boldsymbol{\mathcal{I}}(t)\boldsymbol{G}(t)\boldsymbol{Q}(t)\boldsymbol{G}(t)^T\boldsymbol{\mathcal{I}}(t) - \boldsymbol{K}(t)\check{\boldsymbol{R}}(t)\boldsymbol{K}(t)^T & \text{(3.35a)} \\ \dot{\hat{\boldsymbol{i}}} = -\left[\boldsymbol{A}(t)^T + \boldsymbol{\mathcal{I}}(t)\boldsymbol{G}(t)\boldsymbol{Q}(t)\boldsymbol{G}(t)^T\boldsymbol{\mathcal{I}}(t) - \boldsymbol{K}(t)\check{\boldsymbol{M}}(t)^T\right]\hat{\boldsymbol{i}} & \text{(3.35b)} \\ \qquad + \boldsymbol{\mathcal{I}}(t)\boldsymbol{B}(t)\mathbf{u} - \boldsymbol{K}(t)\left[\boldsymbol{D}(t)\mathbf{u} - \boldsymbol{s}\right] \end{cases}$$

with

$$\boldsymbol{K}(t) = \left[\boldsymbol{C}(t)^T + \boldsymbol{\mathcal{I}}(t)\check{\boldsymbol{M}}(t)\right]\check{\boldsymbol{R}}(t)^{-1} \tag{3.35c}$$

$$\check{\boldsymbol{M}}(t) = \boldsymbol{G}(t)\left[\boldsymbol{Q}(t)\boldsymbol{H}(t)^T + \boldsymbol{M}(t)^T\right] \tag{3.35d}$$

$$\check{\boldsymbol{R}}(t) = \boldsymbol{H}(t)\boldsymbol{Q}(t)\boldsymbol{H}(t)^T + \boldsymbol{R}(t) + \boldsymbol{H}(t)\boldsymbol{M}(t)^T + \boldsymbol{M}(t)\boldsymbol{H}(t)^T \tag{3.35e}$$

The extension of KF to nonlinear system dynamics of the form

$$\begin{cases} \dot{\mathbf{x}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{w}) \\ \boldsymbol{s} = \boldsymbol{g}(\mathbf{x}, \mathbf{u}, \boldsymbol{w}) + \boldsymbol{v} \end{cases}$$

can be easily obtained by linearizing the dynamics around a nominal state trajectory [SSM62] or around the current estimated state $\widehat{\mathbf{x}}(t)$ as described in [MS85]. The latter solution, in particular, usually results in a superior estimation accuracy and takes the name of Extended Kalman Filter (EKF). Its dynamics are given by

$$\dot{\widehat{\mathbf{x}}} = \boldsymbol{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{w})\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}} - \boldsymbol{K}(t)\left[\boldsymbol{g}(\mathbf{x}, \mathbf{u}, \boldsymbol{w})\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}} - \boldsymbol{s}\right] \tag{3.36}$$

and (3.31a) to (3.31d) with

$$\begin{aligned} \boldsymbol{A}(t) &= \nabla_{\mathbf{x}}\boldsymbol{f}^T\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}}, \quad \boldsymbol{B}(t) = \nabla_{\mathbf{u}}\boldsymbol{f}^T\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}}, \quad \boldsymbol{G}(t) = \nabla_{\boldsymbol{w}}\boldsymbol{f}^T\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}}, \\ \boldsymbol{C}(t) &= \nabla_{\mathbf{x}}\boldsymbol{g}^T\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}}, \quad \boldsymbol{D}(t) = \nabla_{\mathbf{u}}\boldsymbol{g}^T\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}}, \quad \boldsymbol{H}(t) = \nabla_{\boldsymbol{w}}\boldsymbol{g}^T\Big|_{\substack{\mathbf{x}=\widehat{\mathbf{x}} \\ \boldsymbol{w}=\emptyset_w}}. \end{aligned} \tag{3.37}$$

Contrarily to the KF, in general, only *local* convergence of the EKF can be proved [Kre03] and, in some cases, the filter can even converge to biased estimations.

In a similar way, one can also devise an Extended Information Filter (EIF), whose dynamics are:

$$\begin{aligned} \dot{\widehat{\boldsymbol{i}}} = &-\left[\boldsymbol{A}(t)^T + \boldsymbol{\mathcal{I}}(t)\boldsymbol{G}(t)\boldsymbol{Q}(t)\boldsymbol{G}(t)^T\boldsymbol{\mathcal{I}}(t) - \boldsymbol{K}(t)\check{\boldsymbol{M}}(t)^T\right]\widehat{\boldsymbol{i}} \\ &+ \boldsymbol{\mathcal{I}}(t)\left[\boldsymbol{f}(\mathbf{x}, \mathbf{u}, \boldsymbol{w})\Big|_{\substack{\mathbf{x}=\boldsymbol{\mathcal{I}}^{-1}\widehat{\boldsymbol{i}} \\ \boldsymbol{w}=\emptyset_w}} - \boldsymbol{A}(t)\widehat{\mathbf{x}}\right] - \boldsymbol{K}(t)\left[\boldsymbol{g}(\widehat{\mathbf{x}}, \mathbf{u}, \boldsymbol{w})\Big|_{\substack{\mathbf{x}=\boldsymbol{\mathcal{I}}^{-1}\widehat{\boldsymbol{i}} \\ \boldsymbol{w}=\emptyset_w}} - \boldsymbol{s} - \boldsymbol{C}\widehat{\mathbf{x}}\right] \end{aligned} \tag{3.38}$$

with (3.35a) and (3.35c) to (3.35e) and using the same linearizations introduced in (3.37). Note that the EIF requires to retrieve an estimation of the system state $\widehat{\mathbf{x}} = \boldsymbol{\mathcal{I}}^{-1}\widehat{\boldsymbol{i}}$ for computing the process and output functions. This represents one of the main disadvantages of the EIF w.r.t. the EKF and motivates the wider adoption of the EKF, especially for estimation problems that involve a large state space. In some cases, however, the interaction between state variables is only local and, as

a result, the precision matrix becomes sparse. One can then exploit this sparsity (which does not extend to $\boldsymbol{\Sigma}$) to improve the computational efficiency of the IF filter [TLK+04]. In addition to this, the IF and EIF allow to easily represent global uncertainty as $\boldsymbol{\mathcal{I}} = \boldsymbol{\emptyset}_{q\times q}$, where as the KF and EKF would require the use of an infinite covariance matrix. More in general, the IF tends to be numerically more stable than the KF.

Another alternative extension of the KF to nonlinear system dynamics, also considerably popular in the literature (see, e.g., [LCG+13, HKM08] and references therein), is the so-called Unscented Kalman Filter (UKF) proposed in [JU97]. This filter is based on a different strategy (namely the Unscented Transform [JUD95]) for propagating Gaussian distributions through nonlinear system dynamics: instead of propagating the expected value and covariance through a linearization of the system equations (as done in EKF), the UKF uses the nonlinear system dynamics to propagate an appropriately selected set of samples of the original Gaussian PDF; the propagated samples are then used to find a new "best-fitting" Gaussian distribution. The UKF can, sometimes, be more computationally expensive than the EKF but it is easier to develop (it does not require the symbolic calculation of the linearizations (3.37)) and can outperform the EKF in terms of accuracy, especially in presence of highly nonlinear system dynamics.

For more details about the KF, IF and their extensions to nonlinear system, we suggest to refer to [TBF05, LXP07]. We also want to report an interesting result demonstrated in [Tay79]: for the case of nonlinear systems with negligible process noise ($\boldsymbol{w} \equiv \boldsymbol{\emptyset}_w$), the dynamics of the FIM have the same form as those of the precision matrix in the EKF, the only difference being that the linearizations (3.37) are evaluated along the *true* state trajectory $\mathbf{x}$, instead of the current estimation $\widehat{\mathbf{x}}$. This allows to conclude that, if the EKF estimate $\widehat{\mathbf{x}}$ actually converges to $\mathbf{x}$, then, from that moment on, the EKF will be efficient and make perfect use of all available information.

We conclude this section by showing how the general EKF structure specializes to the dynamics (3.10) which is typical of SfM applications. Since we assumed that $\boldsymbol{s}$ is measurable, we can directly exploit its knowledge for the computation of the prediction (∼feedforward) component of the EKF thus obtaining the following estimation dynamics analogous to (3.11)

$$\begin{cases} \dot{\widehat{\boldsymbol{s}}} = \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{s},\, \mathbf{u}) + \boldsymbol{\Omega}^T(\boldsymbol{s},\, \mathbf{u})\widehat{\boldsymbol{\chi}} - \boldsymbol{K}_{\boldsymbol{s}}\widetilde{\boldsymbol{s}} \\ \dot{\widehat{\boldsymbol{\chi}}} = \boldsymbol{f}_{\boldsymbol{\chi}}(\boldsymbol{s},\, \widehat{\boldsymbol{\chi}},\, \mathbf{u}) - \boldsymbol{K}_{\boldsymbol{\chi}}\widetilde{\boldsymbol{s}}. \end{cases} \tag{3.39}$$

where $\boldsymbol{K} = (\boldsymbol{K}_{\boldsymbol{s}}, \boldsymbol{K}_{\boldsymbol{\chi}})$ is computed using (3.31a) to (3.31d). Note that this is equivalent to treating $\boldsymbol{s}$ as an additional component of the input vector $\mathbf{u}$. A similar strategy was exploited in [GBSO13] to obtain an asymptotically stable EKF filter

for SLAM applications. As done in (3.11), we also still carry on an estimation of $\boldsymbol{s}$ that allows us to compute the innovation term for the estimator as a function of $\widetilde{\boldsymbol{s}} = \widehat{\boldsymbol{s}} - \boldsymbol{s}$. Finally we assume that $\boldsymbol{s}$ and $\mathbf{u}$ are affected by independent Gaussian distributed additive noise with $\mathrm{E}\left\{\boldsymbol{s}\boldsymbol{s}^T\right\} = \boldsymbol{\Sigma}_{\boldsymbol{s}}$ and $\mathrm{E}\left\{\mathbf{u}\mathbf{u}^T\right\} = \boldsymbol{\Sigma}_{\mathbf{u}}$. With these choices we have

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{s}} & \boldsymbol{\varnothing}_{m\times v} \\ \boldsymbol{\varnothing}_{v\times m} & \boldsymbol{\Sigma}_{\mathbf{u}} \end{bmatrix}, \quad \boldsymbol{R} = \boldsymbol{\Sigma}_{\boldsymbol{s}}, \quad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{s}} \\ \boldsymbol{\varnothing}_{v\times m} \end{bmatrix}$$

and the linearization (3.37) becomes[2]

$$\begin{aligned} \boldsymbol{A}(t) &= \begin{bmatrix} \boldsymbol{\varnothing}_{m\times m} & \boldsymbol{\Omega}(\boldsymbol{s},\mathbf{u})^T \\ \boldsymbol{\varnothing}_{p\times m} & \nabla_{\boldsymbol{\chi}}\boldsymbol{f}_{\boldsymbol{\chi}}{}^T(\boldsymbol{s},\widehat{\boldsymbol{\chi}},\mathbf{u}) \end{bmatrix} \\ \boldsymbol{B}(t) = \boldsymbol{G}(t) &= \begin{bmatrix} \nabla_{\boldsymbol{s}}(\boldsymbol{f}_{\boldsymbol{s}} + \boldsymbol{\Omega}^T\boldsymbol{\chi})^T(\boldsymbol{s},\mathbf{u}) & \boldsymbol{L}(\boldsymbol{s},\widehat{\boldsymbol{\chi}}) \\ \nabla_{\boldsymbol{s}}\boldsymbol{f}_{\boldsymbol{\chi}}{}^T(\boldsymbol{s},\widehat{\boldsymbol{\chi}},\mathbf{u}) & \nabla_{\mathbf{u}}\boldsymbol{f}_{\boldsymbol{\chi}}{}^T(\boldsymbol{s},\widehat{\boldsymbol{\chi}},\mathbf{u}) \end{bmatrix}, \\ \boldsymbol{C}(t) &= \begin{bmatrix} \boldsymbol{I}_m & \boldsymbol{\varnothing}_{m\times p} \end{bmatrix}, \qquad \boldsymbol{D}(t) = \boldsymbol{H}(t) = \boldsymbol{\varnothing}_{m\times(m+v)} \end{aligned} \tag{3.40}$$

where $\boldsymbol{L}$ is the interaction matrix of the considered visual measurements $\boldsymbol{s}$.

3.4 Active perception

As we have seen in the previous sections of this chapter, when dealing with nonlinear system dynamics and measurement models, the performance of the estimation process is not only determined by the efficiency of the chosen observer, i.e. its ability to exploit the information in a sensible way, but also by the amount of information that is available to perform the estimation, which is represented and quantified, in a similar way, by the observability Gramian, the persistence of excitation condition and the Fisher information matrix. All these quantities, in general, are strongly dependent on the trajectory followed by the system during the estimation process, also called the *experiment*. If the system is controllable through some input ports $\mathbf{u}$ one can consider the problem of *actively* driving the system so that the maximum amount of information is gathered during the experiment. The problem, however, would not be well defined (the objective function would be unbounded from above), without some additional constraint on, e.g., the duration T of the experiment or the total energy available. This view of perception and sensing as an active process can be found, under different forms, in many research fields.

J. J. Gibson and his wife E. J. Gibson, among the most important psychologists of the 20th century, formulated their theory of the *active observer* in the 1960s. For them *"perceiving is active, a process of obtaining information about the world. We*

[2]Note that we could have equivalently considered $\boldsymbol{R} = \boldsymbol{\varnothing}_{m\times m}$ and $\boldsymbol{H} = [\boldsymbol{I}_m, \boldsymbol{\varnothing}_{m\times v}]$

don't simply see, we look. The visual system is a motor system as well as a sensory one. When we seek information in an optic array, the head turns, the eyes turn to fixate, the lens accommodates to focus" [Gib88, Gib79]. We do not touch, but we feel [Gib62].

Possibly the research field that traditionally has been the most interested in this problem is that of statistics. A statistical analysis is, in fact, meaningless without a corresponding measure of the reliability of the results. Since, in general, one can only perform a limited number of experiments, it is important to select the ones that are most significant. Ronald Fisher is regarded as the founder of the modern methods for experimental design [Yat64]. For the first time, in fact, he realized that the main responsibility of the statistician was not the results of its analysis but rather *"the processes by which the data had come into existence"*. He started formally investigating the data gathering phase that had previously been conducted using empirical considerations. Fisher's work was initially applied to agricultural engineering and crop selection but, since then, statistical techniques for optimal experimental design have been used in medical diagnostics [AEK+04, SGL+13], biology [CSKW03, KT09], chemistry [LH06, Lea09], public opinion polling and many other fields.

As pointed out by R. Bajcsy [Baj88], who contributed extending the concept to the robotics and computer vision fields, the term *active perception* is not to be confused with the *active sensing* strategies utilized in many robotics and computer vision works. In that case, in fact, the word active refers to the fact that the sensor, in order to be able to *extract* some information, needs to preliminary *inject* information into the system. E.g. an echographic probe or a laser path finder need to actively generate a signal to be able to perceive the environment. While active sensing can be considered as a particular instance of active perception, this latter concept is more general and refers to the way a sensor is used rather than to its intrinsic functioning mechanisms. Furthermore, it is perfectly possible to use a passive sensor (such as a camera) to do some active perception [Baj88].

In the system identification and adaptive control literature, the problem is usually referred to as *input design* or *optimal experiment design* [Meh74]. The first results in this field are attributed to [Lev60] who first studied the problem of identifying the parameters of linear SISO dynamic systems with the minimum possible uncertainty, i.e., by minimizing the estimation covariance matrix. In this case, one aims at designing the set of inputs that allow to reconstruct, looking at the corresponding system outputs for a certain amount of time, the dynamic parameters of the system, e.g., its poles and zeros. The problem of persistence of excitation of the input signal arises[3]. The analysis is usually done in the spectral domain

[3]Note that we are now considering the problem of estimating the *parameters* of a linear system

and, as well know [BS83], one needs an input signal with at least n spectral lines to identify a system with n parameters with white noise being, consequently, the most exciting signal [Lev60]. In adaptive control a term that is often used is that of *non-uniformly observable systems* [BH96] with reference to the fact that, for generic nonlinear systems, the convergence of estimation schemes might depend on the input. Input optimization solutions based on the minimization of the covariance matrix [BRG13] or on the maximization of the observability Gramian [RBG13] have been proposed. Mutual information between the measured outputs and the unknown system parameters, was also exploited as a metric for the quality of an experiment in [AK71].

A similar problem arises in experimental robot calibration. In this case, one needs to find the robot joint trajectory that results in a sufficient excitation of the robot dynamics/kinematics such that the identification process can be successful and robust w.r.t. noise and other disturbances. In most cases, the calibration procedure is done by resolving a system of (possibly nonlinear) equations using (weighted) least squares techniques. The condition number of such system is often used as a quantitative measure and optimization criterion for the selection of robot trajectories [GK92, RVA+06]. In a similar way, B. Armstrong suggests to maximize the condition number of the persistent excitation matrix over the robot trajectory [Arm89]. In some cases the non deterministic properties of the measurements are explicitly taken into account and the problem is solved using maximum likelihood techniques [SGT+97, WSM14]. In this case the optimization criterion is some metric of the (expected) parameters estimate covariance matrix or of the FIM. The search space for the optimization is usually constructed via a time-discretization of the system inputs [WSM14] or using a set of basis functions [Arm89, Par06, RLH12].

Another similar issue is that of *optimal sensor placement* or *dynamic sensor placement* for sensor networks [KM85, Zha92, UC05]. In this case the different measurements are distributed in space rather than in time: one needs to observe a distributed phenomenon (e.g. for air/water quality monitoring, fault detection, weather forecasting, and so on) through the use of a (limited) set of sensors displaced in different positions of the environments. The problem is then that of identifying the most significant quantities to measure and the optimal positions (possibly time-varying) of the sensors. The optimization function is usually a metric of the FIM [UC05] or of the estimation covariance. The constraint is usually on the total number of sensors and possibly on their dynamics (if they can move). In this context, a term that is often used is that of *optimal coverage* [CMKB02] with reference to the *maximum coverage problem* [CK08] in statistics and computational geometry.

and not its state. The observability of the state of a linear system, in fact, does not depend on the inputs as shown in Sect. 3.2.1.

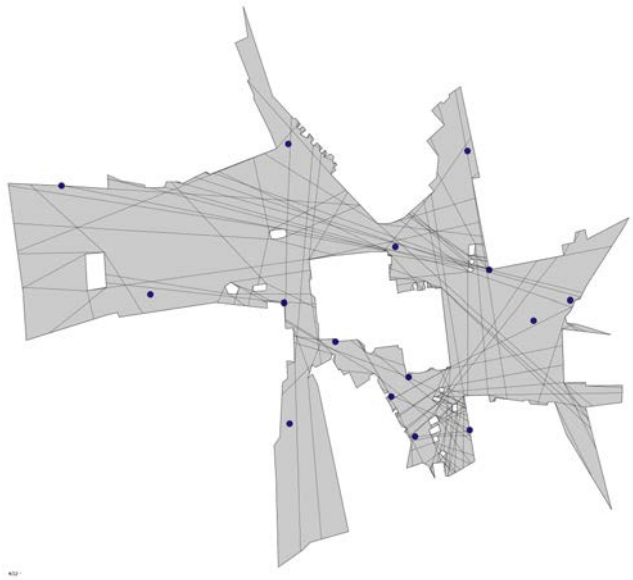


Figure 3.1 – **Resolution of a real-life art gallery problem for the city center of Bremen**. From [BDRDS+13].

This is usually posed in this way: given a set of sets (e.g. the fields of view covered by some sensors when placed in certain discretized positions), select k sets, among them, in such a way that they contain the maximum number of elements (for the sensor placement problem we can think of the total observed area as the quantity to maximize). A related topic, in computational geometry, is the *art gallery problem* [O'r87] in which one tries to find the minimum number of guards, and their distribution, such that every section of a museum lies in the FOV of at least one guard. This problem was demonstrated to be NP-hard in [LL86] but some sub-optimal approximated solutions have been proposed (see, e.g. [BDRDS+13]). Given the complexity of the problem, most of these works use off-line solutions.

The maximization of information is also at the core of *Active SLAM* or Simultaneous Planning Localization and Mapping (SPLAM). In this framework, one addresses the problem of designing a robot trajectory that optimally explores the environment while the map is built. The objective to optimize is, in general, constructed in terms of entropy or information gain [SR05, SB03, BMW+02] or estimate covariance matrix [LHD06]. In this context one often distinguishes two different phases:(i) in the *exploration* phase the robot enlarges the map by visiting new sections of the environment; (ii) in the *localization* phase the robot revisits know sections of the map (loop closure) to reduce the uncertainty related to its position. The optimal policy is usually found by selecting the best control action (in terms of expected information gain) among a limited set of candidates. Since the actual information gain depends on the measurements, which are not known in advance,

the process must be repeated at each iteration based on the current estimate of the environment.

Other interesting strategies are those based on modeling robot motion and observations as a Partially Observable Markov Decision Process (POMDP). This framework is probably the most general one as the uncertainty about the system state is explicitly modeled in the calculation of the control action. The resolution algorithms are, however, very complex and, in general, not suitable for a real-time implementation. Examples of the use of POMDP for active sensing can be found, e.g., in [VM07, CH10, SVL10].

Finally, more specific to vision, we can mention the problem of *Active vision* or *Next Best View* calculation or *View Path Planning* for object reconstruction [Pit99, BWDA00, SRR03] and recognition [HK+89, RCB04]. As with the active SLAM, also in this case the decision about the next viewpoint position is in general made by computing the expected information gain (in terms of entropy or coverage) on a limited set of candidate action and selecting the one that is expected to be the most convenient. The seminal work [AWB88] showed that some complex and ill-posed problems such as structure from motion, become much easier if the observer is active and can control the geometric parameters (e.g. the position or orientation) of the sensor.

SLAM and active vision can also be used jointly to reconstruct a map of the environment and localize a robot within it with the best possible accuracy with the aim of performing some task. In [DM02], which is considered as a pioneering work in this field, the robot trajectory is assigned and determined by a certain task. While the robot is following it, the goal is to change the gaze of a stereo head so that it points toward the feature, in the current map segment, with the largest uncertainty. [AWCS13] runs a probabilistic planner on a known map to find a path from the starting configuration to the assigned goal that minimizes uncertainty measured as a function of the state covariance matrix. The authors of [FPS14] suggest a planning strategy that takes into account both structure and texture since, as well known, vision algorithms perform worse in presence of uniform patterns. The authors also show that local greedy control strategies perform worse than finite horizon planning techniques.

With respect to all these works, in this thesis we address the problem of active vision from a classical control perspective. As it will be shown in Chapt. 4, the nonlinear deterministic observer that we use, described in Sect. 3.2.3, enforces an evolution of the estimation error that is equivalent to that of a second order linear system. Moreover, a complete online control over the eigenvalues of such evolution is possible by acting both on the estimation gains and on the camera linear veloc-

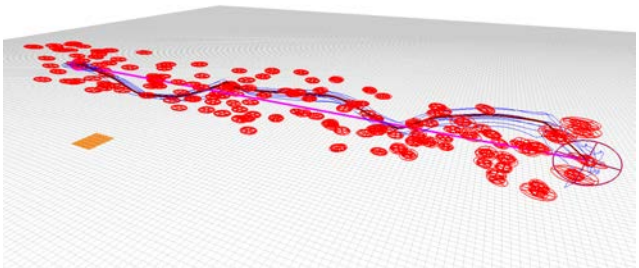


Figure 3.2 – **Trajectory planning to minimize uncertainty for micro aerial vehicles**. The objective is to reach the final pose (to the left) while minimizing the localization uncertainty. The resulting trajectory (black line) is more informative than the direct magenta path. From [AWCS13].

ity. In particular, as it can already be intuitively expected from the developments in Sect. 3.2.3, the "natural frequencies" of the error dynamics are directly proportional to the instantaneous persistence of excitation condition introduced in (3.13). Thanks to the structure of the SfM problem, the PE condition in (3.9) and, in particular, its instantaneous value (3.13) can not just be calculated online using only known information (the visual measurements and the camera linear velocity), but it can also be controlled by acting (online) on the camera linear velocity. One can then use standard control techniques, such as those described in Sect. 2.2.4, to regulate some metric of (3.13) (e.g. its smallest eigenvalue) to obtain the fastest possible convergence rate (and similarly the minimum estimation uncertainty) given some additional constraints on the control inputs. In the next part of the thesis, we will assume that the SfM is the primary task that must be accomplished. The active estimation strategy, thus, has complete freedom on choosing the value of the camera velocity and only a constraint on its norm (that otherwise would grow indefinitely) is imposed. In the last part of the thesis, instead, constraints on the camera velocity will be introduced by the introduction of a main IBVS task that must be executed. The maximization of the observability will then have to be realized only in the null space of this primary task, using some of the redundancy resolution techniques described in Sect. 2.2.4.

Contrarily to some probabilistic frameworks described in this section, we do not take into account the presence of noise neither in the measurements nor in the system dynamics. Probabilistic techniques, in fact, usually do not allow for a complete characterization of the estimation error dynamics. The robustness of the proposed approach w.r.t. noise is however demonstrated by the reporting many real experiments in Chapts. 5 and 7. Moreover, since the excitation of a system (and similarly the acquired Fisher information) does not depend on the choice of the observer, the optimized camera trajectory that our method produces will result

in improved performance also for probabilistic estimation techniques as it will be briefly shown in Chapt. 5.

Finally we use a greedy optimization technique and do not plan on a finite horizon as some of the discussed techniques do. The reason why we do this is that the evaluation of the PE condition (and similarly of the OG and FIM) in the future, here as in all other works, requires a prediction of the future measurements which in turn depends on the estimated quantity. Only locally optimal solutions can then be found, in general, even if a finite planning horizon is considered. Nevertheless, as it will be discussed in the conclusions, the use of a longer planning horizon, coupled with a online re-planning, could result in better results especially once the estimation is close to convergence.

Part II

Active structure from motion

Chapter 4

A framework for active Structure from Motion

The previous chapters briefly introduced the fundamental theoretical concepts that are necessary for the understanding of the rest of this thesis. From this chapter on, we will instead explain the contributions of this work in the context of active structure from motion. As already explained in depth, SfM is a nonlinear estimation problem. As such it results in non uniformly observable system dynamics: singular inputs $\mathbf{u}$ (i.e. camera velocities) exist such that the system evolution starting from two different initial states and under the action of $\mathbf{u}$ results in identical measurements $\boldsymbol{s}(t)$. If such an input is used, then obviously the system input-output mapping $(\mathbf{u}, \boldsymbol{s})$ does not allow to reconstruct the full state of the system. For SfM this results in the impossibility of reconstructing the environment geometry (e.g. the depth of a point feature or the radius of sphere). The strategy that we propose allows to calculate online, based only on measurable quantities, the best motion, i.e. the one that results in the maximum excitation of the system dynamics (measured by the PE condition in (3.13)) and in the maximum amount of acquired information (in the sense of the FIM introduced in Sect. 3.3.2).

We start the chapter by highlighting, in Sect. 4.1, some interesting properties of the nonlinear SfM estimator described in Sect. 3.2.3 that will provide intuitive interpretations for some of the results of this thesis. We continue by characterizing the dynamics of the SfM estimation error as resulting from the use of the nonlinear observer (3.10), and showing its dependence on the observer gains and on system inputs in Sect. 4.2. We then describe how to actively tune these latter online, as a function of the current measurements, to fix the desired damping factor (Sect. 4.3) and natural frequencies (Sect. 4.4). We conclude the chapter by detailing, in Sect. 4.5, how this general policy specializes to specific geometric primitives: points (Sect. 4.5.1), planes (Sect. 4.5.2), spheres (Sect. 4.5.3), and cylinders

(Sect. 4.5.4).

The results contained in this chapter have been presented in different international venues [1, 2, 3, 4, 5] as well as in a journal publication [6].

4.1 Interesting properties of the nonlinear Structure from Motion estimator

We now perform some manipulations of system (3.12) in order to slightly simplify its structure and highlight some important features exploited in the following.

Being $\alpha > 0$, we can consider the following invertible change of coordinates

$$\begin{cases} \check{\tilde{\boldsymbol{s}}} = \tilde{\boldsymbol{s}} \\ \check{\tilde{\boldsymbol{\chi}}} = \dfrac{\tilde{\boldsymbol{\chi}}}{\sqrt{\alpha}} \end{cases}. \tag{4.1}$$

In the new coordinates, system (3.12) takes the form

$$\begin{bmatrix} \dot{\check{\tilde{\boldsymbol{s}}}} \\ \dot{\check{\tilde{\boldsymbol{\chi}}}} \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{\varnothing}_{m\times m} & \check{\boldsymbol{\Omega}}^T(t) \\ -\check{\boldsymbol{\Omega}}(t) & \boldsymbol{\varnothing}_{p\times p} \end{bmatrix} - \begin{bmatrix} \check{\boldsymbol{H}} & \boldsymbol{\varnothing}_{m\times p} \\ \boldsymbol{\varnothing}_{p\times m} & \boldsymbol{\varnothing}_{p\times p} \end{bmatrix} \right) \begin{bmatrix} \check{\tilde{\boldsymbol{s}}} \\ \check{\tilde{\boldsymbol{\chi}}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\emptyset}_m \\ \check{\boldsymbol{d}}(\check{\tilde{\mathbf{x}}}, t) \end{bmatrix}, \tag{4.2}$$

with $\check{\boldsymbol{H}} = \boldsymbol{H}$, $\check{\boldsymbol{\Omega}}(t) = \sqrt{\alpha}\boldsymbol{\Omega}(t)$, and $\check{\boldsymbol{d}} = \frac{1}{\sqrt{\alpha}}\boldsymbol{d}$. We can then note the following facts:

1. In the new coordinates, system (4.2) has an evident port-Hamiltonian (pH) structure (see Appendix C) which is perfectly recovered in the unperturbed case ($\check{\boldsymbol{d}} \equiv \boldsymbol{\emptyset}_p$). The Hamiltonian (storage function) for (4.2) is the lower-bounded scalar function

$$\mathcal{H}(\check{\tilde{\boldsymbol{s}}}, \check{\tilde{\boldsymbol{\chi}}}) = \frac{1}{2}\check{\tilde{\boldsymbol{s}}}^T\check{\tilde{\boldsymbol{s}}} + \frac{1}{2}\check{\tilde{\boldsymbol{\chi}}}^T\check{\tilde{\boldsymbol{\chi}}} = \frac{1}{2}\tilde{\boldsymbol{s}}^T\tilde{\boldsymbol{s}} + \frac{1}{2\alpha}\tilde{\boldsymbol{\chi}}^T\tilde{\boldsymbol{\chi}} \geq 0. \tag{4.3}$$

   Following the pH interpretation, the symmetric component of $\check{\boldsymbol{H}}$, $\check{\boldsymbol{H}}_s = \frac{1}{2}(\check{\boldsymbol{H}} + \check{\boldsymbol{H}}^T)$, represents the dissipative action in the system, while matrix $\check{\boldsymbol{\Omega}}(t)$ defines the internal power-preserving interconnection among the $\check{\tilde{\boldsymbol{s}}}$ and $\check{\tilde{\boldsymbol{\chi}}}$ components of the state. The PE condition (3.13) can then be interpreted as a requirement of a *persistent energy exchange* among the two parts of the system which permits a complete depletion of the total stored energy thanks to the dissipative action provided by $\check{\boldsymbol{H}}_s$.

2. The gain α is a free design parameter and can be suitably exploited to fulfill two objectives. First, since

$$\dot{\mathcal{H}} = -\check{\tilde{\boldsymbol{s}}}^T\check{\boldsymbol{H}}_s\check{\tilde{\boldsymbol{s}}} + \check{\tilde{\boldsymbol{\chi}}}^T\check{\boldsymbol{d}} = -\tilde{\boldsymbol{s}}^T\boldsymbol{H}_s\tilde{\boldsymbol{s}} + \frac{1}{\alpha}\tilde{\boldsymbol{\chi}}^T\boldsymbol{d},$$

one can conclude that, for a bounded disturbance $\|\boldsymbol{d}\| \leq M$, it is always possible to attenuate at will its (possibly destabilizing) contribution by letting the gain $\alpha \to \infty$, basically obtaining a 'semi-global' vs. local stability condition. Furthermore, being

$$\check{\boldsymbol{\Omega}}(t)\check{\boldsymbol{\Omega}}^T(t) = \alpha\boldsymbol{\Omega}(t)\boldsymbol{\Omega}^T(t), \tag{4.4}$$

it is also possible to directly affect the norm of $\check{\boldsymbol{\Omega}}\check{\boldsymbol{\Omega}}^T$ by acting on the gain α. Having an explicit control over the norm of $\check{\boldsymbol{\Omega}}\check{\boldsymbol{\Omega}}^T$, or equivalently over its (real) eigenvalues, will be pivotal for the next developments. We note however that increasing α might also result in a higher sensitivity of the observer to measurement noise that was not taken into account in this analysis.

3. Finally, it is worth to informally re-analyze the stability proof for (4.2) in the new coordinates $(\check{\tilde{\boldsymbol{s}}}, \check{\tilde{\boldsymbol{\chi}}})$ for the case $\check{\boldsymbol{d}} = \boldsymbol{\emptyset}_p$. First of all, system (4.2) is clearly still in the form of (3.8) with $\alpha = 1$ and $\check{\boldsymbol{H}} = \boldsymbol{H} \succ 0$. Second, from $\dot{\mathcal{H}} = -\check{\tilde{\boldsymbol{s}}}^T\check{\boldsymbol{H}}_s\check{\tilde{\boldsymbol{s}}} \leq 0$ we can conclude, using Lyapunov's stability theorems, boundedness of the state trajectories $(\check{\tilde{\boldsymbol{s}}}(t), \check{\tilde{\boldsymbol{\chi}}}(t))$. Being $\ddot{\mathcal{H}} = -2\check{\tilde{\boldsymbol{s}}}^T\check{\boldsymbol{H}}_s\left(\check{\boldsymbol{\Omega}}^T(t)\check{\tilde{\boldsymbol{\chi}}} - \check{\boldsymbol{H}}\check{\tilde{\boldsymbol{s}}}\right)$ and exploiting the assumption of a bounded $\|\boldsymbol{\Omega}(t)\|$ from Lemma 3.1, allows to further conclude boundedness of $\ddot{\mathcal{H}}$ which, invoking Barbalat's Lemma, grants $\dot{\mathcal{H}} \to 0$ and $\check{\tilde{\boldsymbol{s}}}(t) \to \boldsymbol{\emptyset}_m$. Finally, by restricting the system dynamics to the set $\check{\tilde{\boldsymbol{s}}}(t) \equiv \dot{\check{\tilde{\boldsymbol{s}}}}(t) \equiv \boldsymbol{\emptyset}_m$, the first row of (4.2) reduces to $\boldsymbol{\emptyset}_m = \check{\boldsymbol{\Omega}}^T(t)\check{\tilde{\boldsymbol{\chi}}}$. Assuming, as previously stated, $m \geq p$ and full (instantaneous) rank of $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ as in (3.13) also implies full rankness of the (high-rectangular) matrix $\check{\boldsymbol{\Omega}}^T \in \mathbb{R}^{m\times p}$ and, consequently, that $\check{\tilde{\boldsymbol{\chi}}} \equiv \boldsymbol{\emptyset}_p$ as well (thus concluding the proof).

   It is now worth noting that the same proof still holds for a sufficiently smooth time-varying dissipation matrix $h_1\boldsymbol{I}_m \preceq \check{\boldsymbol{H}}(t) \preceq h_2\boldsymbol{I}_m$, $0 < h_1 \leq h_2 < \infty$, with bounded $\left\|\dot{\check{\boldsymbol{H}}}(t)\right\|$: this important feature opens the possibility of suitably shaping the dissipation matrix $\check{\boldsymbol{H}}$ over time in order to fulfill additional objectives of interest, as it will be the case in the next developments.

4.2 Characterization of the system transient behavior

In this section, we will give a characterization of the transient behavior of system (4.2). We will assume that $\boldsymbol{\Omega}(t) = \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{\varsigma}, \mathbf{u})$ where $\boldsymbol{\varsigma} \in \mathbb{R}^r$ indicates a set of time-varying measurable quantities that are not included neither in $\boldsymbol{s}$ (i.e. they do not appear in the observer update term) nor in $\mathbf{u}$ (because they cannot be directly controlled). This is a quite natural requirement in many situations, and it is certainly the case for the SfM applications considered in this work, in which, as it will be shown, $\boldsymbol{\Omega}$ always depends on a set of visual measurements and on the

linear velocity of the camera $\boldsymbol{v}$. This structure allows to (actively) exploit the input vector $\mathbf{u}(t)$ in order to affect matrix $\check{\boldsymbol{\Omega}}$ and, as a consequence, the system transient response. Note that the introduction of $\boldsymbol{\varsigma}$ was not necessary for the developments of Sect. 3.2.3 because, from an estimation point of view, the elements of $\boldsymbol{\varsigma}$ can be thought of as being part of vector $\mathbf{u}$ in the sense that they are measurable quantities used to calculate the prediction term of the estimator. In this section, however, we start dealing with the control problem and thus we need to make this distinction since the value of $\boldsymbol{\varsigma}$, differently from $\mathbf{u}$, cannot be arbitrarily assigned.

Finally, for this analysis, we will neglect the disturbance term $\check{\boldsymbol{d}}(t)$, since, as explained before, its distorting effects can be typically made arbitrarily small by a proper choice of the gain α. This claim will also be confirmed by the simulation and experimental results in Chapts. 5 and 7.

Following the pH interpretation of system (4.2), in particular with in mind a standard mechanical system, one can identify vector $\check{\tilde{\boldsymbol{\chi}}}$ as playing the role of a 'position'-like quantity, and vector $\check{\tilde{\boldsymbol{s}}}$ as that of a 'velocity'-like quantity upon which a dissipative action is present. Therefore, analogously to a mechanical system, we focus the analysis on the dynamics of vector $\ddot{\check{\tilde{\boldsymbol{\chi}}}}$.

Being $\dot{\check{\tilde{\boldsymbol{\chi}}}} = -\check{\boldsymbol{\Omega}}\check{\tilde{\boldsymbol{s}}}$, it is

$$\begin{aligned}\ddot{\check{\tilde{\boldsymbol{\chi}}}} &= -\dot{\check{\boldsymbol{\Omega}}}\check{\tilde{\boldsymbol{s}}} - \check{\boldsymbol{\Omega}}\dot{\check{\tilde{\boldsymbol{s}}}} = -\dot{\check{\boldsymbol{\Omega}}}\check{\tilde{\boldsymbol{s}}} - \check{\boldsymbol{\Omega}}(-\check{\boldsymbol{H}}\check{\tilde{\boldsymbol{s}}} + \check{\boldsymbol{\Omega}}^T\check{\tilde{\boldsymbol{\chi}}}) = \\ &= (\check{\boldsymbol{\Omega}}\check{\boldsymbol{H}} - \dot{\check{\boldsymbol{\Omega}}})\check{\tilde{\boldsymbol{s}}} - \check{\boldsymbol{\Omega}}\check{\boldsymbol{\Omega}}^T\check{\tilde{\boldsymbol{\chi}}} = (\dot{\check{\boldsymbol{\Omega}}}\check{\boldsymbol{\Omega}}^\dagger - \check{\boldsymbol{\Omega}}\check{\boldsymbol{H}}\check{\boldsymbol{\Omega}}^\dagger)\dot{\check{\tilde{\boldsymbol{\chi}}}} - \check{\boldsymbol{\Omega}}\check{\boldsymbol{\Omega}}^T\check{\tilde{\boldsymbol{\chi}}}\end{aligned} \tag{4.5}$$

with $\check{\boldsymbol{\Omega}}^\dagger \in \mathbb{R}^{m\times p}$ denoting the pseudo-inverse of matrix $\check{\boldsymbol{\Omega}}$. Let $\check{\boldsymbol{U}}\check{\boldsymbol{S}}\check{\boldsymbol{V}}^T = \check{\boldsymbol{\Omega}}$ be the SVD of matrix $\check{\boldsymbol{\Omega}}$, where $\check{\boldsymbol{S}} = [\check{\boldsymbol{\Sigma}} \;\; \boldsymbol{\mathcal{O}}_{p\times(m-p)}]$, $\check{\boldsymbol{\Sigma}} = \mathrm{diag}(\check{\sigma}_i) \in \mathbb{R}^{p\times p}$, and $0 \le \check{\sigma}_1 \le \cdots \le \check{\sigma}_p$, from which it directly follows $\check{\boldsymbol{\Omega}}\check{\boldsymbol{\Omega}}^T = \check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}}^2\check{\boldsymbol{U}}^T$ and $\check{\boldsymbol{\Omega}}^\dagger = \check{\boldsymbol{V}}^T\check{\boldsymbol{S}}^\dagger\check{\boldsymbol{U}}$ where, as usual, $\check{\boldsymbol{S}}^\dagger = [\check{\boldsymbol{\Sigma}}^{-1} \;\; \boldsymbol{\mathcal{O}}_{p\times(m-p)}]^T$ with $\check{\boldsymbol{\Sigma}}^{-1} = \mathrm{diag}(\check{\sigma}_i^\dagger)$, $\check{\sigma}_i^\dagger = 1/\check{\sigma}_i$ if $\check{\sigma}_i > 0$ and $\check{\sigma}_i^\dagger = 0$ otherwise.

As for $\dot{\check{\boldsymbol{\Omega}}}$ it is $\dot{\check{\boldsymbol{\Omega}}} = \dot{\check{\boldsymbol{U}}}\check{\boldsymbol{S}}\check{\boldsymbol{V}}^T + \check{\boldsymbol{U}}\dot{\check{\boldsymbol{S}}}\check{\boldsymbol{V}}^T + \check{\boldsymbol{U}}\check{\boldsymbol{S}}\dot{\check{\boldsymbol{V}}}^T$. Exploiting the orthonormality of $\check{\boldsymbol{U}}$, we have $\check{\boldsymbol{U}}^T\check{\boldsymbol{U}} = \boldsymbol{I}_p \implies \dot{\check{\boldsymbol{U}}}^T\check{\boldsymbol{U}} + \check{\boldsymbol{U}}^T\dot{\check{\boldsymbol{U}}} = \boldsymbol{\mathcal{O}}_{p\times p}$. Denoting the skew-symmetric matrix $\check{\boldsymbol{U}}^T\dot{\check{\boldsymbol{U}}} = \check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}$, it is $\dot{\check{\boldsymbol{U}}} = \check{\boldsymbol{U}}\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}$ and, following the same arguments, one has $\dot{\check{\boldsymbol{V}}}^T\check{\boldsymbol{V}} = \check{\boldsymbol{\Gamma}}_{\boldsymbol{V}} = -\check{\boldsymbol{\Gamma}}_{\boldsymbol{V}}^T$ and $\dot{\check{\boldsymbol{V}}}^T = \check{\boldsymbol{\Gamma}}_{\boldsymbol{V}}\check{\boldsymbol{V}}^T$. Therefore,

$$\dot{\check{\boldsymbol{\Omega}}} = \check{\boldsymbol{U}}(\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}\check{\boldsymbol{S}} + \dot{\check{\boldsymbol{S}}} + \check{\boldsymbol{S}}\check{\boldsymbol{\Gamma}}_{\boldsymbol{V}})\check{\boldsymbol{V}}^T. \tag{4.6}$$

We highlight that, as shown in [PL00], matrices $\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}$, $\check{\boldsymbol{\Gamma}}_{\boldsymbol{V}}$ and $\dot{\check{\boldsymbol{S}}}$ can be computed in *closed-form* from the knowledge of $\check{\boldsymbol{U}}$, $\check{\boldsymbol{V}}$, $\check{\boldsymbol{S}}$ and of the closed-form expression of $\check{\boldsymbol{\Omega}}$. This is also valid in our context since an explicit expression of $\check{\boldsymbol{\Omega}}$ is assumed available, while matrices $\check{\boldsymbol{U}}$, $\check{\boldsymbol{V}}$ and $\check{\boldsymbol{S}}$ can be numerically retrieved from $\check{\boldsymbol{\Omega}}$ via any

standard SVD routine. Finally, exploiting (4.6) we have

$$\begin{aligned}\dot{\check{\boldsymbol{\Omega}}}\check{\boldsymbol{\Omega}}^{\dagger} =& \check{\boldsymbol{U}}\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}\check{\boldsymbol{S}}\check{\boldsymbol{S}}^{\dagger}\check{\boldsymbol{U}}^{T} + \check{\boldsymbol{U}}\dot{\check{\boldsymbol{S}}}\check{\boldsymbol{S}}^{\dagger}\check{\boldsymbol{U}}^{T} + \check{\boldsymbol{U}}\check{\boldsymbol{S}}\check{\boldsymbol{\Gamma}}_{\boldsymbol{V}}\check{\boldsymbol{S}}^{\dagger}\check{\boldsymbol{U}}^{T} \\ =& \check{\boldsymbol{U}}(\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}} + \dot{\check{\boldsymbol{\Sigma}}}\check{\boldsymbol{\Sigma}}^{-1} + \check{\boldsymbol{\Sigma}}\bar{\boldsymbol{\Gamma}}_{\boldsymbol{V}}\check{\boldsymbol{\Sigma}}^{-1})\check{\boldsymbol{U}}^{T}\end{aligned} \tag{4.7}$$

where $\bar{\boldsymbol{\Gamma}}_V = -\bar{\boldsymbol{\Gamma}}_V^T$ is the $p \times p$ upper-left block of matrix $\check{\boldsymbol{\Gamma}}_{\boldsymbol{V}}$.

At this point, the dissipation matrix is purposely taken as

$$\check{\boldsymbol{H}} = \check{\boldsymbol{V}} \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{\varnothing}_{p\times(m-p)} \\ \boldsymbol{\varnothing}_{(m-p)\times p} & \boldsymbol{D}_2 \end{bmatrix} \check{\boldsymbol{V}}^T \tag{4.8}$$

with $\boldsymbol{D}_1 \in \mathbb{R}^{p\times p} \succ 0$, $\boldsymbol{D}_2 \in \mathbb{R}^{(m-p)\times(m-p)} \succ 0$, and, thus, $\check{\boldsymbol{H}} \succ 0$ as well. This choice in fact yields

$$\check{\boldsymbol{\Omega}}\check{\boldsymbol{H}}\check{\boldsymbol{\Omega}}^{\dagger} = \check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}}\boldsymbol{D}_1\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T. \tag{4.9}$$

By combining (4.5) with (4.7–4.9), and exploiting the diagonal form of matrix $\check{\boldsymbol{\Sigma}}$, we finally obtain

$$\begin{aligned}\ddot{\check{\boldsymbol{\chi}}} =& \check{\boldsymbol{U}}(\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}} + \dot{\check{\boldsymbol{\Sigma}}}\check{\boldsymbol{\Sigma}}^{-1} + \check{\boldsymbol{\Sigma}}\bar{\boldsymbol{\Gamma}}_{\boldsymbol{V}}\check{\boldsymbol{\Sigma}}^{-1} - \check{\boldsymbol{\Sigma}}\boldsymbol{D}_1\check{\boldsymbol{\Sigma}}^{-1})\check{\boldsymbol{U}}^T\dot{\check{\boldsymbol{\chi}}} - \check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}}^2\check{\boldsymbol{U}}^T\check{\boldsymbol{\chi}} \\ =& (\check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}})(\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}\check{\boldsymbol{\Sigma}} + \dot{\check{\boldsymbol{\Sigma}}}\check{\boldsymbol{\Sigma}}^{-1} + \bar{\boldsymbol{\Gamma}}_{\boldsymbol{V}} - \boldsymbol{D}_1)(\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T)\dot{\check{\boldsymbol{\chi}}} - \check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}}^2\check{\boldsymbol{U}}^T\check{\boldsymbol{\chi}} \\ =& (\check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}})(\check{\boldsymbol{\Pi}} - \boldsymbol{D}_1)(\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T)\dot{\check{\boldsymbol{\chi}}} - (\check{\boldsymbol{U}}\check{\boldsymbol{\Sigma}})\check{\boldsymbol{\Sigma}}^2(\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T)\check{\boldsymbol{\chi}}\end{aligned} \tag{4.10}$$

where

$$\check{\boldsymbol{\Pi}} = \check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}\check{\boldsymbol{\Sigma}} + \dot{\check{\boldsymbol{\Sigma}}}\check{\boldsymbol{\Sigma}}^{-1} + \bar{\boldsymbol{\Gamma}}_{\boldsymbol{V}}. \tag{4.11}$$

The expression obtained in (4.10) has a clear and neat structure: it indicates presence of a change of coordinates

$$\boldsymbol{\epsilon} = (\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T)\check{\boldsymbol{\chi}} \tag{4.12}$$

in which, in the approximation $\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T \approx \mathit{const}$, the system exhibits the simple (and almost diagonal) form

$$\ddot{\boldsymbol{\epsilon}} = (\check{\boldsymbol{\Pi}} - \boldsymbol{D}_1)\dot{\boldsymbol{\epsilon}} - \check{\boldsymbol{\Sigma}}^2\boldsymbol{\epsilon}, \tag{4.13}$$

that is, a (unit-)mass-spring-damper system with diagonal stiffness matrix $\check{\boldsymbol{\Sigma}}^2$.

The convergence rate of (4.13) is then related to the slowest mode of the system, i.e., that associated to the element $\check{\sigma}_1^2$ in $\check{\boldsymbol{\Sigma}}^2$. Therefore, in order to impose a given convergence speed and overall transient behavior to (4.13) (and to the estimation error dynamics (4.5)), one can try to '*place the poles*' of (4.13) by:(i) regulating $\check{\sigma}_1^2$ to a desired value $\check{\sigma}_{1,\mathrm{d}}^2$ and, at the same time, (ii) shaping the damping factor $\boldsymbol{D}_1$ in order to prevent the occurrence of oscillatory modes ($\sim$ complex poles). Sections 4.3 and 4.4 explain how to achieve these two objectives.

Remark 4.1. *Note that, in the special situation $p = 1$ (only one quantity to be estimated), if $\sigma_1(t) \equiv const$ then $\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T \equiv const$ in (4.12) and matrix $\check{\boldsymbol{\Pi}}$ has no disturbing effects on (4.13). Therefore, in this case it is always possible to* exactly *enforce the ideal estimation error dynamics (4.15) by just keeping $\|\boldsymbol{\Omega}(t)\|^2 = \sigma_1^2(t) = const$ during the camera motion. This situation will apply to the case studies discussed in Sects. 4.5.1, 4.5.3 and 4.5.4.*

4.3 Shaping the damping factor

A reasonable choice for matrix $\boldsymbol{D}_1$ could be

$$\boldsymbol{D}_1 = \check{\boldsymbol{\Pi}} + \boldsymbol{C} \tag{4.14}$$

with $\boldsymbol{C}$ any positive definite matrix, without loss of generality (w.l.o.g.) a diagonal one $\boldsymbol{C} = \mathrm{diag}(c_i)$, $c_i > 0$, so as to obtain a completely decoupled transient behavior for (4.13)

$$\ddot{\epsilon}_i + c_i\dot{\epsilon}_i + \check{\sigma}_i^2\epsilon_i = 0, \quad i = 1 \dots p. \tag{4.15}$$

For instance, taking $c_i = c_i^* = 2\check{\sigma}_i$ would (conveniently) result in a critically damped state evolution.

Matrix $\boldsymbol{D}_1$, however, is bound to remain positive definite over time, a constraint which, clearly, is not necessarily met by (4.14) for any arbitrary pair $(\boldsymbol{C}, \check{\boldsymbol{\Pi}})$. Let $\check{\boldsymbol{\Pi}}_s$ and $\check{\boldsymbol{\Pi}}_a$ represent the symmetric/skew-symmetric components of $\check{\boldsymbol{\Pi}}$, and similarly for $\boldsymbol{D}_1$ and $\boldsymbol{C}$, with then

$$\begin{cases} \check{\boldsymbol{\Pi}}_s = \dfrac{1}{2}(\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}\check{\boldsymbol{\Sigma}} - \check{\boldsymbol{\Sigma}}\check{\boldsymbol{\Gamma}}_{\boldsymbol{U}}\check{\boldsymbol{\Sigma}}^{-1}) + \dot{\check{\boldsymbol{\Sigma}}}\check{\boldsymbol{\Sigma}}^{-1} \\ \boldsymbol{D}_{1s} = \check{\boldsymbol{\Pi}}_s + \boldsymbol{C}_s \end{cases}. \tag{4.16}$$

It is obviously $\boldsymbol{D}_1 \succ 0 \iff \boldsymbol{C}_s \succ -\check{\boldsymbol{\Pi}}_s$. In the special case of $\check{\boldsymbol{\Sigma}} = \check{\sigma}\boldsymbol{I}_p = const$ (constant and coincident singular values), $\check{\boldsymbol{\Pi}}_s = \boldsymbol{\varnothing}_{p\times p}$ and thus (4.14) can be safely implemented for any choice of $\boldsymbol{C} \succ 0$. This possibility, however, requires a very stringent constraint on matrix $\check{\boldsymbol{\Omega}}\check{\boldsymbol{\Omega}}^T$ which may be hard to enforce in practice. Alternatively, a less stringent condition may be obtained by suitably bounding $\|\check{\boldsymbol{\Pi}}_s\| \preceq q\boldsymbol{I}_p$, $q \geq 0$: in this case, any $\boldsymbol{C}_s \succ q\boldsymbol{I}_p$ would then guarantee $\boldsymbol{D}_1 \succ 0$.

While this latter possibility is certainly viable, we note however the following: in the general case, satisfying the requirement $\boldsymbol{C}_s \succ q\boldsymbol{I}_p$ would guarantee positive definiteness of $\boldsymbol{D}_1$ in (4.14) but at the possible expense of imposing an *over-damped transient behavior* to the system. In fact, in the general case, one could have $\boldsymbol{C}_s \succ q\boldsymbol{I}_p \succ \mathrm{diag}(c_i^*)$. In other words, having aimed at obtaining a completely decoupled behavior for the evolution of $\boldsymbol{\epsilon}(t)$ as in (4.15) could entail a unnecessary degradation of the transient response.

A full characterization of possible bounds on $\check{\mathbf{\Pi}}_s$ is out of the scope of this work and thus will not be addressed here apart from the following qualitative considerations. Boundedness of $\|\mathbf{\Omega}(t)\|$ and $\|\dot{\mathbf{\Omega}}(t)\|$ required by Lemma 3.1 together with (3.13) are *sufficient* conditions for guaranteeing boundedness of $\|\check{\mathbf{\Pi}}_s\|$ as well. In fact, by a rough inspection of $\check{\mathbf{\Pi}}_s$, we can conclude that it can be made arbitrarily small by:(i) limiting $\|\check{\mathbf{\Sigma}}\|$ (implied by limited $\|\mathbf{\Omega}\|$), (ii) limiting $\det(\check{\mathbf{\Sigma}})$ *from below* in order to prevent $\check{\mathbf{\Sigma}}^{-1} \to \infty$ (implied by (3.13)), and (iii) limiting the rate of change of $\check{\mathbf{\Sigma}}$ and $\check{\boldsymbol{U}}$, that is, by bounding $\|\dot{\check{\mathbf{\Sigma}}}\|$ and $\|\check{\mathbf{\Gamma}}_{\boldsymbol{U}}\|$ (implied by limited $\|\dot{\mathbf{\Omega}}\|$).

In any case, in absence of as a deeper analysis, in the following we will *not* aim for a cancellation of matrix $\check{\mathbf{\Pi}}$, but we will rather neglect its effects on the transient by just taking $\boldsymbol{D}_1 = \mathrm{diag}(c_i^*) > 0$. This can of course result in a poorer overall behavior (for not compensating for $\check{\mathbf{\Pi}}$), but avoids the introduction of any unnecessary lower bound on $\boldsymbol{D}_1$. The simulation and experimental results reported in Chapt. 5 will anyway show that not compensating for matrix $\check{\mathbf{\Pi}}$ has a marginal effect.

4.4 Tuning the stiffness matrix

We recall that matrix $\check{\mathbf{\Sigma}}^2 = \mathrm{diag}(\check{\sigma}_i^2)$ contains the p eigenvalues of the square symmetric matrix $\check{\mathbf{\Omega}}\check{\mathbf{\Omega}}^T$ in (4.4). Let then $\mathbf{\Sigma}^2 = \mathrm{diag}(\sigma_i^2)$ represent the eigenvalues of matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ in the original coordinates ($\widetilde{\boldsymbol{s}}$, $\widetilde{\boldsymbol{\chi}}$). From (4.4) it follows that, in order to affect $\check{\mathbf{\Sigma}}^2$, one can either($i$) act on the gain α for a given $\mathbf{\Sigma}^2$, or (ii) actively adjust $\mathbf{\Sigma}^2$ for a given gain α (or, of course, any combination of both actions). The effect of gain α has already been discussed in Sect. 4.1: in short, one can exploit it to freely amplify/attenuate the eigenvalues of $\check{\mathbf{\Sigma}}^2$ as clear from (4.4). However, we note that, regardless of any choice of the gain, one still needs to ensure a minimum threshold $\sigma_1^2(t) \geq \sigma_{min}^2 > 0$ for the estimation to converge, i.e., for fulfilling condition (3.13): this can only be achieved by actively tuning matrix $\mathbf{\Sigma}^2$. The rest of the section is then devoted to this issue.

We start by noting that, from (4.4), one has $\check{\mathbf{\Sigma}}^2 = \alpha\mathbf{\Sigma}^2 \implies \check{\sigma}_i^2 = \alpha\sigma_i^2$. Therefore, seeking a desired value $\check{\sigma}_{i,\mathrm{d}}^2$ is equivalent to imposing $\sigma_i^2 \to \sigma_{i,\mathrm{d}}^2 = \check{\sigma}_{i,\mathrm{d}}^2/(\alpha)$. We can then focus on the regulation of the eigenvalues σ_i^2.

An explicit expression of the time derivative of the eigenvalues σ_i^2 can be obtained as follows: being $\mathbf{\Omega}(t) = \mathbf{\Omega}(\mathbf{y}, \mathbf{u})$, with $\mathbf{y} = (\boldsymbol{s}, \boldsymbol{\varsigma})$, it is $\sigma_i^2(t) = \sigma_i^2(\mathbf{y}, \mathbf{u})$ which, exploiting the results of [PL00, YFG+10], allows to conclude

$$\frac{\mathrm{d}}{\mathrm{d}t}\sigma_i^2 = \sum_{j=1}^{v}\left(\mathbf{v}_i^T\frac{\partial(\mathbf{\Omega}\mathbf{\Omega}^T)}{\partial \mathrm{u}_j}\mathbf{v}_i\dot{\mathrm{u}}_j\right) + \sum_{j=1}^{n}\left(\mathbf{v}_i^T\frac{\partial(\mathbf{\Omega}\mathbf{\Omega}^T)}{\partial s_j}\mathbf{v}_i\dot{s}_j\right) \tag{4.17}$$

where $\mathbf{v}_i \in \mathbb{R}^p$ is the *normalized* eigenvector associated to σ_i^2. Letting

$$\boldsymbol{J}_{\mathbf{u},i} = \left[\mathbf{v}_i^T \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial \mathrm{u}_1} \mathbf{v}_i \ \ldots \ \mathbf{v}_i^T \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial \mathrm{u}_v} \mathbf{v}_i \right] \in \mathbb{R}^{1\times v} \tag{4.18}$$

and

$$\boldsymbol{J}_{\mathbf{y},i} = \left[\mathbf{v}_i^T \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial \mathrm{y}_1} \mathbf{v}_i \ \ldots \ \mathbf{v}_i^T \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial \mathrm{y}_n} \mathbf{v}_i \right] \in \mathbb{R}^{1\times n}, \tag{4.19}$$

eq. (4.17) can be compactly rewritten as

$$\dot{(\sigma_i^2)} = \boldsymbol{J}_{\mathbf{u},i}\dot{\mathbf{u}} + \boldsymbol{J}_{\mathbf{y},i}\dot{\mathbf{y}}. \tag{4.20}$$

Note, again, that the Jacobian matrices $\boldsymbol{J}_{\mathbf{u},i}$ and $\boldsymbol{J}_{\mathbf{y},i}$ in (4.18–4.19) can be computed in *closed-form* from the knowledge of the eigenvectors $\mathbf{v}_i$ and of a closed-form expression for matrix $\boldsymbol{\Omega}$.

At this point, any differential inversion technique can be applied to (4.20) in order to affect the behavior of the i-th eigenvalue $\sigma_i^2(t)$ by acting upon vector $\dot{\mathbf{u}}$: this must then be treated as the 'actual' input vector, with $\mathbf{u}$ regarded, instead, as an internal state. The eigenvalues σ_i^2 can, in fact, be used to construct a task vector $\boldsymbol{r}$ or a cost function $w(\dot{\boldsymbol{q}})$ and the techniques described in Sect. 2.2.4 (in particular the ones in Sect. 2.2.4.4) can be used to regulate/maximize its value. Sect. 4.5 will discuss some examples in this sense. As illustration, the classical choice

$$\dot{\mathbf{u}} = \boldsymbol{J}_{\mathbf{u},i}^{\dagger} \left[-k(\sigma_i^2 - \sigma_{i,\mathrm{d}}^2) - \boldsymbol{J}_{\mathbf{y},i}\dot{\mathbf{y}} \right], \qquad k > 0, \tag{4.21}$$

would result in a perfect exponential convergence of $\sigma_i^2(t) \to \sigma_{i,\mathrm{d}}^2$.

Remark 4.2. *We note, however, that in general it is not possible to compensate for the term $\boldsymbol{J}_{\mathbf{y},i}\dot{\mathbf{y}}$ as simply done in (4.21) (or in any other equivalent law). Indeed, the formulation (3.10) implies a direct dependency of $\dot{\boldsymbol{s}}$ from the unmeasurable $\boldsymbol{\chi}$, and a similar dynamics can, in general, be found for $\boldsymbol{\varsigma}$, so that an exact evaluation of $\dot{\mathbf{y}}$ is not obtainable in practice. A possible workaround is to replace $\dot{\mathbf{y}}$ with an estimation $\widehat{\dot{\mathbf{y}}}$ obtained by plugging $\widehat{\boldsymbol{\chi}}$ in (3.10) and in the dynamics of $\boldsymbol{\varsigma}$, in place of $\boldsymbol{\chi}$. As $\widehat{\boldsymbol{\chi}} \to \boldsymbol{\chi}$ one obviously has $\widehat{\dot{\mathbf{y}}} \to \dot{\mathbf{y}}$, thus allowing for an* asymptotic *compensation of $\boldsymbol{J}_{\mathbf{y},i}\dot{\mathbf{y}}$. Another possibility, when viable, is to enforce $\dot{\mathbf{y}} \equiv \boldsymbol{\emptyset}_{m+r}$ during the system evolution. A combination of both strategies is, of course, also possible. The next sections will present some examples in this sense.*

In all SfM problems, matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ and its eigenvalues σ_i^2 are only function of $\mathbf{y}$ and of the camera linear velocity $\boldsymbol{v}$ (never of the angular one). This is a direct consequence of the fact that, as already pointed out, the unknown 3-D parameters $\boldsymbol{\chi}$ only appear in the columns of the interaction matrix that are related to the camera translational motion. Therefore (4.20) reduces to

$$\dot{(\sigma_i^2)} = \boldsymbol{J}_{\boldsymbol{v},i}\dot{\boldsymbol{v}} + \boldsymbol{J}_{\mathbf{y},i}\dot{\mathbf{y}} \approx \boldsymbol{J}_{\boldsymbol{v},i}\dot{\boldsymbol{v}} \tag{4.22}$$

where the approximation is valid under the assumption that $\dot{\mathbf{y}}$ is kept (approximately) constant. In particular, roughly speaking, $\|\boldsymbol{\Omega}\boldsymbol{\Omega}^T\|$ is monotonically increasing with $\|\boldsymbol{v}\|^2$: the faster the camera motion the faster the SfM convergence regardless of any other optimization action. Use of the control law (4.21) could then result in a sub-optimal velocity direction compensated by a growth of $\|\boldsymbol{v}\|$. To avoid such a situation, we prefer to consider the optimization of the observability as a secondary objective function to be maximized in the null space of a main task that has the only purpose of keeping the linear velocity norm constant:

$$\dot{\boldsymbol{v}} = -k_1 \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2} (\kappa - \kappa_\mathrm{d}) + k_2 \left(\boldsymbol{I}_3 - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2} \right) \nabla_{\boldsymbol{v}} w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) \tag{4.23}$$

with $k_1 > 0$, $k_2 \geq 0$, $\kappa = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{v}$, $\kappa_\mathrm{d} = \frac{1}{2}\boldsymbol{v}_0^T\boldsymbol{v}_0$, and $w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)$ is a function of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ that quantifies the observability. The analysis in Sect. 4.2 shows that the smallest eigenvalue σ_1^2 directly affects the convergence rate of the employed estimator, and thus a reasonable choice is to maximize σ_1^2 over time. This corresponds to the so called *E-optimality* criterion introduced in [Ehr55].

$$w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \Phi_E = \sigma_1^2 \Rightarrow \nabla_{\boldsymbol{v}} w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \boldsymbol{J}_{\boldsymbol{v},1}^T.$$

Another optimality criterion, often used in the context of experimental design, is the *A-optimality* introduced by [Che53]. This aims at maximizing the trace of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$:

$$w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \Phi_A = \mathrm{tr}(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \sum_{i=1}^{p} \sigma_i^2 \Rightarrow \nabla_{\boldsymbol{v}} w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \sum_{i=1}^{p} \boldsymbol{J}_{\boldsymbol{v},i}^T. \tag{4.24}$$

Unfortunately, the evaluation of the derivative/gradient of an eigenvalue as in (4.17) is not well-defined for repeated eigenvalues [Fri96]. In order to avoid this issue, one can also consider the quantity

$$w(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \Phi_D = \det(\boldsymbol{\Omega}\boldsymbol{\Omega}^T) = \prod_{i=1}^{p} \sigma_i^2 \tag{4.25}$$

as a conditioning measure for matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$. Indeed, from classical linear algebra [Ber09] the following relationship holds for a square matrix $\boldsymbol{A}$

$$\frac{\mathrm{d}}{\mathrm{d}t} \det(\boldsymbol{A}) = \mathrm{tr}\left(\mathrm{adj}\,(\boldsymbol{A}) \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}t} \right) \tag{4.26}$$

with $\mathrm{tr}\,(\cdot)$ and $\mathrm{adj}\,(\cdot)$ being the *trace* and *adjugate* operators, respectively. Contrarily to the derivative of an eigenvalue, the relationship (4.26) is always well-defined with, in particular, no possible ill-conditioning due to repeated eigenvalues. Optimization of (4.25) goes under the name of *D-optimality* criterion and was first proposed by [Wal43].

A variety of other optimality criteria have been proposed in the literature. We suggest to refer to [RAS09] for an overview and comparison of the different ones. Some of them turn out to be equivalent under certain assumptions[Kie74]. Finally, if $p = 1$, one obviously has $\Phi_E = \Phi_A = \Phi_D = \sigma_1^2$.

4.5 Application to a class of geometric primitives

In this section we illustrate the application of the proposed active estimation framework to four concrete SfM problems:(i) estimation of the 3-D coordinates of a point feature, (ii) estimation of the distance and orientation of a planar surface, (iii) estimation of the 3-D position and radius of a spherical target, and (iv) estimation of the 3-D position and radius of a cylindrical target.

In the point feature case, the effects of the adopted projection model on the estimation convergence are also explicitly considered by discussing the differences between the two popular choices of *planar* and *spherical* projection models introduced in Sect. 2.1.2.

For the planar target, we consider both the case of a collection of discrete points of interest extracted and tracked form the surface of a textured planar object, and that of a dense planar patch segmented in the image. Discrete and dense image moments can be used (among others) as a measurement to recover the plane 3-D parameters in the two cases. Since the choice of image moments affects the performance of the estimation, we also propose to adaptively select on-line the most informative image moments to use.

Finally for the spherical and cylindrical targets, we propose the use of two *novel* minimal parametrization that allow to express the sphere/cylinder 3-D structures in terms of measured visual features and of a single unknown parameter (the sphere/-cylinder radius). This allows, in both cases, to reduce the SfM task to the estimation of a single unknown quantity (the sphere/cylinder radius), thus satisfying the requirements of Remark 4.1 for *exactly* imposing the ideal dynamics (4.15) to the estimation error.

4.5.1 Active Structure from Motion for a point

The first case study that we will consider is that of the estimation of the depth of a single point feature. We propose two different estimation schemes using either a planar or a spherical projection model. We will show that the two schemes have different and somehow complementary properties.

4.5.1.1 Planar projection model

Let $\boldsymbol{\pi} = (x,\, y,\, 1) = (X/Z,\, Y/Z,\, 1) \in \mathbb{R}^3$ be the perspective projection of a 3-D point $\boldsymbol{p} = (X,\, Y,\, Z)$ onto the image plane of a calibrated pinhole camera. As it is well know (see [CH06] and Sect. 2.1.3.3), the differential relationship between the image motion of a point feature and the camera linear/angular velocity $\boldsymbol{u} = (\boldsymbol{v},\, \boldsymbol{\omega}) \in \mathbb{R}^6$ expressed in camera frame is given by the interaction matrix (2.10) as a function of the *depth* Z of the feature point. The dynamics of Z is

$$\dot{Z} = \begin{bmatrix} 0 & 0 & -1 & -yZ & xZ & 0 \end{bmatrix} \boldsymbol{u}. \tag{4.27}$$

The expression in (2.10) is not linear in Z but it is linear in $\zeta = 1/Z$. Therefore, by defining $\boldsymbol{s} = (x,\, y) \in \mathbb{R}^2$ and $\chi = \zeta \in \mathbb{R}$, with then $m = 2$ and $p = 1$, we obtain for (3.10)

$$\begin{cases} \boldsymbol{f}_{\boldsymbol{s}}\left(\boldsymbol{s},\, \boldsymbol{\omega}\right) = \begin{bmatrix} xy & -\left(1+x^2\right) & y \\ 1+y^2 & -xy & -x \end{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{\Omega}\left(\boldsymbol{s},\, \boldsymbol{v}\right) = \begin{bmatrix} xv_z - v_x & yv_z - v_y \end{bmatrix} \\ f_{\chi}\left(\boldsymbol{s},\, \chi,\, \boldsymbol{u}\right) = v_z\chi^2 + \left(y\omega_x - x\omega_y\right)\chi \end{cases}, \tag{4.28}$$

with the perturbation term $d(\widetilde{\mathbf{x}},\, t)$ in (3.12) taking the expression

$$d(\widetilde{\chi},\, t) = v_z\left(\widehat{\chi}^2 - \chi^2\right) + \left(y\omega_x - x\omega_y\right)\widetilde{\chi}, \tag{4.29}$$

so that $d(0,\, t) = 0$ as expected. Note that, once χ has been estimated, one can obviously retrieve the 3-D position of the point feature as $\boldsymbol{p} = \boldsymbol{\pi}/\zeta = \boldsymbol{s}/\chi$.

In the point feature case matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ reduces to its single eigenvalue which, for a planar projection model, takes the expression

$$\sigma_1^2 = \|\boldsymbol{\Omega}\|^2 = (xv_z - v_x)^2 + (yv_z - v_y)^2. \tag{4.30}$$

Furthermore, as explained in Remark 4.1, in this case if $\sigma_1(t) \equiv const > 0$ then by construction $\check{\boldsymbol{\Sigma}}^{-1}\check{\boldsymbol{U}}^T = const$ in (4.12) and matrix $\check{\boldsymbol{\Pi}}$ has no distorting effect on the behavior of (4.13). Therefore, it is always possible to *exactly* enforce the 'ideal' estimation error dynamics (4.15) by keeping $\|\boldsymbol{\Omega}\|^2 = \sigma_1^2 = const$.

Moreover, using (4.30), the Jacobian $\boldsymbol{J}_{v,1}$ in (4.22) is given by

$$\boldsymbol{J}_{\boldsymbol{v},1} = 2\begin{bmatrix} v_x - xv_z \\ v_y - yv_z \\ \left(xv_z - v_x\right)x + \left(yv_z - v_y\right)y \end{bmatrix}^T. \tag{4.31}$$

Since σ_1^2 does not depend on $\boldsymbol{\omega}$ (this is always true for SfM problems), it is possible to freely exploit the camera angular velocity for fulfilling additional goals of interest

without interfering with the regulation of $\sigma_1^2(t)$ (only affected by $\boldsymbol{v}$). For instance, as in [CBBJ96], one can use $\boldsymbol{\omega}$ for keeping $\boldsymbol{s} \simeq const$ so as to make the effects of $\dot{\boldsymbol{s}}$ negligible when inverting (4.22) w.r.t. $\dot{\boldsymbol{v}}$, see Remark 4.2.

We now note that σ_1^2 in (4.30) depends on both the camera linear velocity $\boldsymbol{v}$ and on the location $\boldsymbol{\pi}$ of the feature point on the image plane. Since the value of σ_1^2 directly affects the convergence speed of the estimation error, it is interesting to study what conditions on $\boldsymbol{s}$ and $\boldsymbol{v}$ result in the largest possible σ_1^2 (i.e., the fastest possible convergence for a given gain α). Letting $\boldsymbol{e}_3 = [0, 0, 1]^T$ being the camera optical axis, it is (by inspection)

$$\begin{bmatrix} \boldsymbol{\Omega}^T \\ 0 \end{bmatrix} = [\boldsymbol{e}_3]_\times [\boldsymbol{\pi}]_\times \boldsymbol{v}$$

where $[\boldsymbol{a}]_\times$ is the skew-symmetric matrix representing the cross product operator for 3-D vectors (i.e., $[\boldsymbol{a}]_\times \boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$). Therefore,

$$\begin{aligned} \sigma_1^2 &= \begin{bmatrix} \boldsymbol{\Omega} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega}^T \\ 0 \end{bmatrix} = \left\| [\boldsymbol{e}_3]_\times [\boldsymbol{\pi}]_\times \boldsymbol{v} \right\|^2 \\ &= \|\boldsymbol{\pi}\|^2 \|\boldsymbol{v}\|^2 \sin^2(\theta_{\boldsymbol{\pi},\boldsymbol{v}}) \sin^2\left(\theta_{\boldsymbol{e}_3,[\boldsymbol{\pi}]_\times \boldsymbol{v}}\right) \end{aligned}$$

where $\theta_{\boldsymbol{\pi},\boldsymbol{v}}$ and $\theta_{\boldsymbol{e}_3,[\boldsymbol{\pi}]_\times \boldsymbol{v}}$ represent the angles between vectors $(\boldsymbol{\pi}, \boldsymbol{v})$ and vectors $(\boldsymbol{e}_3, [\boldsymbol{\pi}]_\times \boldsymbol{v})$, respectively. The maximum attainable value for σ_1^2 is then

$$\sigma_{max}^2 = \max_{\boldsymbol{\pi},\boldsymbol{v}} (\sigma_1^2) = \|\boldsymbol{\pi}\|^2 \|\boldsymbol{v}\|^2 . \tag{4.32}$$

This maximum is obtained when the camera linear velocity $\boldsymbol{v}$ is such that $\boldsymbol{\pi} \perp \boldsymbol{v}$ and $\boldsymbol{e}_3 \perp [\boldsymbol{\pi}]_\times \boldsymbol{v}$, i.e., rearranging in matrix form

$$\begin{bmatrix} \boldsymbol{\pi}^T \\ \boldsymbol{e}_3^T [\boldsymbol{\pi}]_\times \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} x & y & 1 \\ -y & x & 0 \end{bmatrix} \boldsymbol{v} = 0. \tag{4.33}$$

If $\boldsymbol{\pi} \neq \boldsymbol{e}_3$ (point feature *not* at the center of the image plane), system (4.33) has (full) rank 2 and admits the unique solution (up to a scalar factor)

$$\boldsymbol{v} \cong [\boldsymbol{\pi}]_\times^2 \boldsymbol{e}_3.$$

This requires the linear velocity $\boldsymbol{v}$ to be orthogonal to $\boldsymbol{\pi}$ and to lie on the plane defined by vectors $\boldsymbol{\pi}$ and $\boldsymbol{e}_3$ (i.e., $\boldsymbol{v}$ must belong to a straight line as shown in Fig. 4.1(a)).

If $\boldsymbol{\pi} = \boldsymbol{e}_3$ (point feature at the center of the image plane), system (4.33) loses rank and any $\boldsymbol{v} \perp \boldsymbol{e}_3$ is a valid solution, see Fig. 4.1(b).

It is then possible to draw the following conclusions: for a given norm of the linear velocity $\|\boldsymbol{v}\|$ (i.e., the amount of 'control effort'), system (4.33) determines

the direction of $\boldsymbol{v}$ resulting in $\sigma_1^2 = \sigma_{max}^2$ (maximization of σ_1^2). These conditions are summarized in Figs. 4.1(a) and 4.1(b). The value of σ_{max}^2 is, however, also a function of the feature point location $\boldsymbol{\pi}$ which can be arbitrarily positioned on the image plane. In particular, $\sigma_{max}^2 = \|\boldsymbol{v}\|^2$ for $\boldsymbol{\pi} = \boldsymbol{e}_3$ and $\sigma_{max}^2 = \|\boldsymbol{\pi}\|^2\|\boldsymbol{v}\|^2 > \|\boldsymbol{v}\|^2$ $\forall\, \boldsymbol{\pi} \neq \boldsymbol{e}_3$, with $\lim_{\|\boldsymbol{\pi}\|\to\infty} \sigma_{max}^2(\boldsymbol{\pi}) = \infty$. The value of $\|\boldsymbol{\pi}\|$ (distance of the point feature from the projection center) thus acts as an amplification factor for σ_{max}^2. Therefore,

1. the smallest σ_{max}^2 (i.e., the *slowest* 'optimal' convergence for the depth estimation error) is obtained for the smallest value of $\|\boldsymbol{\pi}\|$, i.e., when $\boldsymbol{\pi} = \boldsymbol{e}_3 \implies \|\boldsymbol{\pi}\| = 1$ (feature point at the center of the image plane). It is worth noting that in this case $v_z = 0$ (from the condition $\boldsymbol{v} \perp \boldsymbol{\pi}$) and $\sigma_{max}^2 = \|\boldsymbol{v}\|^2 = v_x^2 + v_y^2$: the camera moves on the surface of a sphere with a constant radius (depth) pointing at the feature point. Also, being in this case $\dot{\chi} = \dot{Z}/Z^2 = 0$, one has $d(\widetilde{\chi}, t) \equiv 0$ and global convergence for the estimation error (see Remark 3.1);

2. the largest σ_{max}^2 (i.e., the *fastest* 'optimal' convergence for the depth estimation error) is obtained for the largest possible value of $\|\boldsymbol{\pi}\|$. In the usual case of a rectangular image plane centered at the origin, this translates into keeping the feature point positioned at one of the four image corners. However, compared with the previous case, this results in a $d(\widetilde{\chi}, t) \neq 0$ and only local convergence for the estimation error. Moreover, as shown in [CBBJ96], this also results in an increased impact of measurement errors (e.g. discretization and other undeterministic effects) on the estimation.

4.5.1.2 Spherical projection model

We now develop the depth estimation machinery for the spherical projection model. In this case, the following quantity is taken as visual feature measured on the image plane

$$\boldsymbol{s} = \boldsymbol{\eta} = \frac{\boldsymbol{\pi}}{\|\boldsymbol{\pi}\|} = \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|} \in \mathbb{S}^2,$$

where $\mathbb{S}^2$ represents the unit sphere and, as well-known (see [HM02]) and shown in Sect. 2.1.3.3,

$$\dot{\boldsymbol{s}} = \begin{bmatrix} \delta\left(\boldsymbol{s}\boldsymbol{s}^T - \boldsymbol{I}_3\right) & [\boldsymbol{s}]_\times \end{bmatrix} \boldsymbol{u},$$

with $\delta = \frac{1}{\|\boldsymbol{p}\|}$ and

$$\dot{\delta} = -\delta^2 \frac{\mathrm{d}\,\|\boldsymbol{p}\|}{\mathrm{d}t} = -\delta^2 \boldsymbol{s}^T \dot{\boldsymbol{p}} = \delta^2 \boldsymbol{s}^T \boldsymbol{v}. \tag{4.34}$$

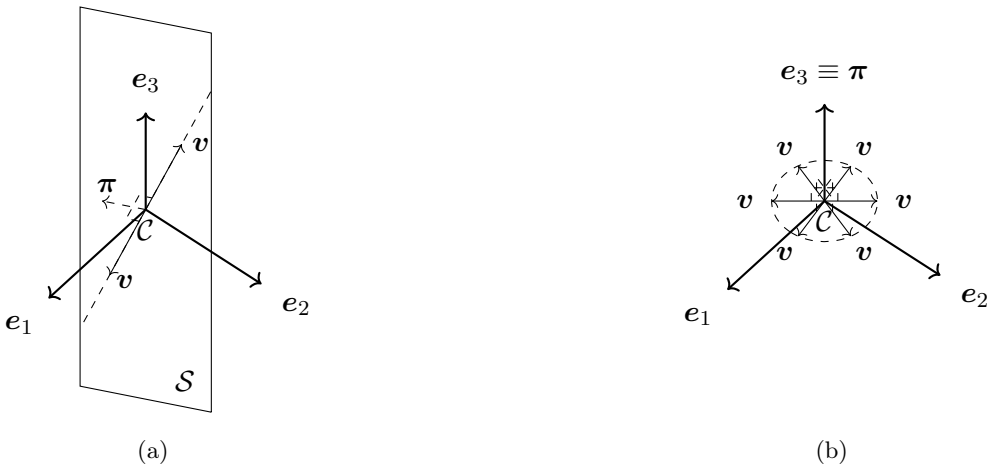


Figure 4.1 – **Optimality conditions for the camera linear velocity $\boldsymbol{v}$ as dictated by system (4.33)**. Fig. (a): when $\boldsymbol{\pi} \neq \boldsymbol{e}_3$, vector $\boldsymbol{v}$ must be orthogonal to $\boldsymbol{\pi}$ and lie on the plane $\mathcal{S}$ spanned by $\boldsymbol{\pi}$ and $\boldsymbol{e}_3$ (that is, $\boldsymbol{v}$ must belong to a specific straight line). Fig. (b): when $\boldsymbol{\pi} = \boldsymbol{e}_3$, any $\boldsymbol{v} \perp \boldsymbol{e}_3$ is a valid solution to (4.33).

Hence by taking $\chi = \delta$ one obtains for (3.10)

$$\begin{cases} \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{s},\, \boldsymbol{u}) = [\boldsymbol{s}]_{\times}\, \boldsymbol{\Omega} \\ \boldsymbol{\Omega}(\boldsymbol{s},\, \boldsymbol{v}) = -\boldsymbol{v}^T \left(\boldsymbol{I}_3 - \boldsymbol{s}\boldsymbol{s}^T\right) \\ f_{\chi}(\boldsymbol{s},\, \chi, \boldsymbol{u}) = \chi^2 \boldsymbol{s}^T \boldsymbol{v} \end{cases} \tag{4.35}$$

with $m = 3$, $p = 1$, and $d(\widetilde{\chi},\, t) = (\widehat{\chi}^2 - \chi^2)\boldsymbol{s}^T\boldsymbol{v}$ for the perturbation term in (3.12). We note that, although in this case $m = 3$, vector $\boldsymbol{s}$ is subject to the constraint $\|\boldsymbol{s}\| = 1$, thus resulting in only two independent measurements (as in the previous case of planar projection). Moreover, from the estimated χ one can easily retrieve $\boldsymbol{p} = \boldsymbol{\eta}/\delta = \boldsymbol{s}/\chi$.

For the spherical projection model, the eigenvalue determining the convergence of the estimation error is

$$\sigma_1^2 = \boldsymbol{\Omega}\boldsymbol{\Omega}^T = \boldsymbol{v}^T\boldsymbol{v} - (\boldsymbol{s}^T\boldsymbol{v})^2,$$

with thus

$$\boldsymbol{J}_{\boldsymbol{v},1} = 2\boldsymbol{v}^T(\boldsymbol{I}_3 - \boldsymbol{s}\boldsymbol{s}^T). \tag{4.36}$$

As before, σ_1^2 does not depend on $\boldsymbol{\omega}$ which can then be exploited to fulfill any additional task of interest (e.g., keeping $\boldsymbol{s} \simeq const$ during motion).

As for the conditions on $\boldsymbol{s}$ and $\boldsymbol{v}$ that yield maximization of σ_1^2, one clearly has

$$\sigma_1^2 = \sigma_{max}^2 = \max_{\boldsymbol{s},\, \boldsymbol{v}} (\sigma_1^2) = \|\boldsymbol{v}\|^2 \tag{4.37}$$

iff $\boldsymbol{s}^T\boldsymbol{v} = 0$ (linear velocity orthogonal to the projection ray passing through $\boldsymbol{p}$). We also note that, in this case, one has $\dot{\chi} = 0$ and $d(\widetilde{\chi}, t) \equiv 0$ (*constant* unknown state and *global* convergence for the estimation error) *regardless* of the location of $\boldsymbol{s}$ on the image plane.

4.5.1.3 Comparison between planar and spherical projection models

For a spherical projection model, maximization of the eigenvalue σ_1^2 imposes only one condition for the linear velocity $\boldsymbol{v}$ ($\boldsymbol{\eta}^T\boldsymbol{v} = 0$). When this condition is met, one has $\sigma_1^2 = \sigma_{max}^2 = \|\boldsymbol{v}\|^2$ and *global* convergence for the estimation error *whatever* the location of the feature point $\boldsymbol{\eta}$. This is equivalent to what was obtained for the planar projection case in the special situation $\boldsymbol{\pi} = \boldsymbol{e}_3$ (indeed the two projection models coincide for $\boldsymbol{\pi} = \boldsymbol{\eta} = \boldsymbol{e}_3$). However, with a spherical projection model one also loses the possibility to increase the estimation convergence rate by suitably positioning the point feature $\boldsymbol{s}$ on the image plane (since in this case σ_{max}^2 does not depend on $\boldsymbol{s}$).

It is then worth noting the complementarity of the two cases: for a given $\|\boldsymbol{v}\|$, and provided the optimal condition $\boldsymbol{\pi}^T\boldsymbol{v} = 0$ is satisfied, the *planar* projection allows obtaining a faster error convergence at the price of local stability (increase of the perturbation d) by suitably positioning $\boldsymbol{s} = [x, y]^T$ (the larger $\|\boldsymbol{s}\|$ the faster the convergence). The *spherical* projection guarantees global error convergence for *any* location of the feature point, but at the price of being always subject to the same convergence rate only function of the control effort $\|\boldsymbol{v}\|$.

4.5.2 Active Structure from Motion for a plane

Plane detection and estimation from raw visual data is a classical problem in sensor-based robot control, especially in the context of mobile robotics. Indeed, planes are widespread in artificial (man-made) and natural environments, and therefore constitute the typical 3-D structure one tries to segment in order to, e.g., plan safe paths among planar obstacles (e.g., vertical walls), or navigate by keeping a desired attitude or distance from special planes (e.g., ground plane for flying robots). The ability to classify and reconstruct planes in the perceived environment is therefore an important feature for several sensor-based applications. When dealing with images taken by a (possibly moving) camera, a number of approaches has been developed for solving the problem of detecting and identifying planes from visual data.

Several methods for instance exploit known correspondences across frames to identify point features (or directly pixels) as whether belonging to a common plane together with the associated plane parameters [ASNM10, LJPS10, GBR12]. These methods usually rely on special geometric constraints linking two views of a pla-

nar scene such as the homography constraint. We have already explained, in Sect. 2.1.3.2, some techniques for exploiting the homography constraint between two views of the same set of points to recover the parameters of the plane and the camera motion up to a scalar factor. We will use the classical 8-point algorithm, in the following referred to as *method A*, as a baseline for comparison and evaluation of the other original ones proposed in this section.

Alternative strategies, such as those suggested in [VBP08, SSN11, BELN11] and references therein, instead, attempt to directly measure (using special sensors such as the RGB-D camera) or recover (exploiting structure from motion algorithms) a 'depth map' of the observed images, for then dealing with the issue of clustering and extracting planes from clouds of 3-D points. In these cases, the problem is rather on how to fit planes to sets of 3-D points and on how to cluster them according to some reasonable 'planarity measure'. We propose, in Sect. 4.5.2.1, a solution that can be ascribed to this second class because it uses a simple least-squares fitting strategy to extract a plane from a point cloud of estimated points. In Sect. 4.5.2.2 we will discuss, instead, an alternative solution that estimates the plane parameters *directly* from a set of discrete image moments. Finally Sect. 4.5.2.4 explains how a similar method could be applied to a set of dense image moments computed on a segmented planar patch.

4.5.2.1 Plane reconstruction from 3-D points

Assume that a visual tracker is able to extract and track, in a sequence of images, the projection of a set of N points $\boldsymbol{p}_k$ belonging to the same plane $\mathcal{P}$.

Using the active strategies discussed in Sect. 4.5.1, one can (optimally) retrieve an estimation $\widehat{Z}_k$ of the unknown depth Z_k of each point (similarly, one could also use a spherical projection model to find an estimate $\widehat{\|\boldsymbol{p}_k\|}$ of the point distances $\|\boldsymbol{p}_k\|$). Then, from each measured point feature $\boldsymbol{\pi}_k$ (or $\boldsymbol{\eta}_k$) one can recover an estimation $\widehat{\boldsymbol{p}}_k = \widehat{Z}_k \boldsymbol{\pi}_k$ (or $\widehat{\boldsymbol{p}}_k = \widehat{\|\boldsymbol{p}\|}_k \boldsymbol{\eta}_k$) of the corresponding 3-D point $\boldsymbol{p}_k$ in the current camera frame. Let $\mathcal{P}: \ \boldsymbol{n}^T \boldsymbol{E} + d = 0$ be the equation of the sought plane, with $\boldsymbol{n} \in \mathbb{S}^2$ and $d \in \mathbb{R}$ representing the unit normal vector and distance in camera frame. For the estimated points $\widehat{\boldsymbol{p}}_k$ to belong to $\mathcal{P}$, it must hold

$$\boldsymbol{n}^T \widehat{\boldsymbol{p}}_k + d = 0, \quad i = 1 \dots N. \tag{4.38}$$

Equation (4.38) can be rearranged in matrix form as

$$\begin{bmatrix} \widehat{\boldsymbol{p}}_1^T & 1 \\ \vdots & \vdots \\ \widehat{\boldsymbol{p}}_N^T & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \boldsymbol{\emptyset}_N \tag{4.39}$$

with $\boldsymbol{A} \in \mathbb{R}^{N\times 4}$. Assuming $N \geq 4$ and $\operatorname{rank}(\boldsymbol{A}) = 3$, the linear system (4.39) has a unique solution (up to a scalar factor) for the pair $(\boldsymbol{n}, d)$ which can be found by standard least-square techniques. Let $\boldsymbol{U}_A \boldsymbol{S}_A \boldsymbol{V}_A^T = \boldsymbol{A}$ be the SVD of matrix $\boldsymbol{A}$, with $\sigma_{1,A} \leq \cdots \leq \sigma_{4,A}$ being the associated singular values. As well-known, a (least-square) solution of the homogeneous system (4.39) is given by $\mathbf{v}_1 = [\mathrm{v}_{11}, \mathrm{v}_{12}, \mathrm{v}_{13}]^T$, the column of $\boldsymbol{V}_A$ associated to $\sigma_{1,A}$. Furthermore, the inverse of the condition number $\sigma_A = \sigma_{1,A}/\sigma_{4,A}$ can be taken as a normalized measure of the planarity of the N points $\widehat{\boldsymbol{p}}_k$, in fact $\operatorname{rank}(\boldsymbol{A}) = 3 \iff \sigma_A = 0$. From $\mathbf{v}_1$ one can then recover

$$\begin{bmatrix} \boldsymbol{n} \\ d \end{bmatrix} = \pm \frac{\mathbf{v}_1}{\sqrt{\mathrm{v}_{11}^2 + \mathrm{v}_{12}^2 + \mathrm{v}_{13}^2}}, \tag{4.40}$$

i.e., by imposing $\|\boldsymbol{n}\| = 1$. The final sign ambiguity can be resolved by fixing the sign of d according to the adopted convention.

As for the issue of optimally recovering the unknown depths Z_k for the N tracked point features $\boldsymbol{\pi}_k$, this can be addressed by exploiting the SfM scheme (3.11). Let $\boldsymbol{s} = [x_1, y_1, \ldots, x_N, y_N]^T \in \mathbb{R}^{2N}$ be the vector of measured visual features, and $\boldsymbol{\chi} = [1/Z_1, \ldots, 1/Z_N]^T \in \mathbb{R}^N$ be the 3-D structure to be estimated (the depths of all tracked points). This choice results in the matrix

$$\boldsymbol{\Omega}\boldsymbol{\Omega}^T = \operatorname{diag}(\sigma_{1,1}^2, \sigma_{1,2}^2, \ldots \sigma_{1,N}^2), \tag{4.41}$$

with

$$\sigma_{1,i}^2 = (x_k v_z - v_x)^2 + (y_k v_z - v_y)^2 \tag{4.42}$$

being the eigenvalue determining the convergence speed of the k-th estimation error $\widetilde{\chi}_k(t) = \widehat{\chi}_k(t) - \chi_k(t) = 1/\widehat{Z}_k(t) - 1/Z_k(t)$ for the k-th feature point. Exploiting (4.22), optimization of the convergence of the whole vector $\widetilde{\boldsymbol{\chi}}(t)$ can then be obtained by, e.g., maximizing the minimum eigenvalue

$$\sigma_m^2 = \min_{i=1\ldots N} \sigma_{1,i}^2 \tag{4.43}$$

w.r.t. the camera linear velocity $\boldsymbol{v}$.

We finally note that this method does not require the exact matching of point features across distant frames (initial and current ones) as it is instead the case for the homography reconstruction method described in Sect. 2.1.3.2, but it only needs a frame-by-frame tracking. As a consequence, the method can straightforwardly cope with loss/gain of feature points because of, e.g., limited FOV: new estimated points $\widehat{\boldsymbol{p}}_k$ can be added to system (4.39) by initializing the corresponding estimated depth $\widehat{Z}_k$ so as to belong to the current estimation of the plane $\mathcal{P}$. The only assumption (common to all the methods) is that all the tracked points seen by the

moving camera belong to a common plane[1]. In the following, this second possibility for recovering $(\boldsymbol{n},\, d)$ will be denoted as *method B*.

4.5.2.2 Plane reconstruction from discrete image moments

Another possibility for estimating the structure of a plane is based on the machinery of *point-based* image moments originally introduced in [TC05]. This method, hereafter denoted as *method C*, can be seen as a further improvement of method B in that it exploits the active estimation scheme (3.11) for directly estimating the pair $(\boldsymbol{n},\, d)$ (3 independent quantities) instead of the N depths Z_k of the N considered point features $\boldsymbol{\pi}_i$ for then algebraically solving system (4.39). Thus, the complexity of the SfM scheme results reduced w.r.t. method B as the number of estimated states is independent of the number of tracked points. Furthermore, since $(\boldsymbol{n}, d)$ are directly estimated via a filtering process, one can expect method C to be more robust than method B w.r.t. non perfectly planar scenes as no algebraic step is involved (contrarily to method B that still requires the least-square solution of the linear system (4.39)). Indeed, these considerations are also supported by the experimental results of Sect. 5.2.

Consider then the $(i,\, j)$-th moment m_{ij} evaluated on the collection of N observed feature points $\boldsymbol{\pi}_k = (x_k,\, y_k,\, 1)$

$$m_{ij} = \sum_{k=1}^{N} x_k^i y_k^j. \tag{4.44}$$

As shown in [Cha04], dividing the plane equation (4.38) by Z_k and d, the depth Z_k on any 3-D point $\boldsymbol{p}_k \in \mathbb{R}^3$ lying on this plane can be expressed in terms of its normalized image coordinates $\boldsymbol{\pi}_k$ as

$$\frac{1}{Z_k} = \zeta_k = -\frac{\boldsymbol{n}^T}{d}\boldsymbol{\pi}_k = \boldsymbol{\nu}^T\boldsymbol{\pi}_k, \tag{4.45}$$

where $\boldsymbol{\nu} = -\boldsymbol{n}/d \in \mathbb{R}^3$ represents an *unmeasurable* 3-D scene structure (as with Z for the point feature case). Exploiting this fact, [TC05] shows that the dynamics of m_{ij} takes the expression

$$\dot{m}_{ij} = f_{m_{ij}}(m_{kl},\, \boldsymbol{\omega}) + \boldsymbol{\varpi}_{m_{ij}}^T(m_{kl},\, \boldsymbol{v})\boldsymbol{\nu} \tag{4.46}$$

where m_{kl} stands for the generic $(k,\, l)$-th moment of order up to $i+j+1$. Analogous considerations hold for the centered moments

$$\mu_{ij} = \sum_{k=1}^{N}(x_k - x_g)^i(y_k - y_g)^j$$

[1]The results of Sect. 5.2 will nevertheless test the robustness of the methods against this hypothesis.

with $x_g = m_{10}/m_{00}$ and $y_g = m_{01}/m_{00}$ being the barycenter coordinates, and $m_{00} = N = const$ in this case. Furthermore, it is (see, e.g., [1])

$$\dot{\boldsymbol{\nu}} = \boldsymbol{\nu}\boldsymbol{\nu}^T\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{\nu} = \boldsymbol{f}_{\boldsymbol{\nu}}(\boldsymbol{\nu}, \boldsymbol{u}). \tag{4.47}$$

The estimation scheme (3.11) can then be directly applied for recovering $\boldsymbol{\chi} = \boldsymbol{\nu}$ by including in $\boldsymbol{s}$ a suitable collection of $m \geq 3$ image moments $\boldsymbol{s} = (m_{i_1 j_1}, \dots, m_{i_m j_m})$, and thus letting $\boldsymbol{\varsigma} = (m_{k_1 l_1}, \dots, m_{k_r l_r})$ be the set of moments that appear in the dynamics of $\boldsymbol{s}$ but are not included in $\boldsymbol{s}$, $\boldsymbol{f}_{\boldsymbol{s}} = [f_{m_{i_1 j_1}} \dots f_{m_{i_m j_m}}]^T \in \mathbb{R}^m$, $\boldsymbol{f}_{\boldsymbol{\chi}}(\boldsymbol{\chi}, \boldsymbol{u}) = \boldsymbol{f}_{\boldsymbol{\nu}}(\boldsymbol{\nu}, \boldsymbol{u})$ and

$$\boldsymbol{\Omega} = [\boldsymbol{\varpi}_{m_{i_1 j_1}} \dots \boldsymbol{\varpi}_{m_{i_m j_m}}] \in \mathbb{R}^{3\times m}. \tag{4.48}$$

This estimation strategy, however, lacks the possibility of taking into account the loss/gain of feature points over time (as it is instead the case with method B). When a feature point leaves visibility, a practical workaround could be to just redefine a moment m_{ij} as the sum over the remaining $N-1$ points and feed the estimation scheme with this new measurement (and analogously for new points entering visibility). However, this would clearly introduce a discontinuity in the measured m_{ij} — a discontinuity not modeled by the dynamics (4.46) which predicts the moment evolution as only a function of the camera own motion $(\boldsymbol{v}, \boldsymbol{\omega})$. Therefore, we now propose a redefinition of *weighted image moments* meant to explicitly cope with this issue.

Assume presence of a countable number of feature points on the plane $\boldsymbol{\pi}_k = [x_k, y_k, 1]^T$, $k = 1 \dots \infty$, and define the (i, j)-th weighted moment as

$$m_{ij} = \sum_{k=1}^{\infty} w(x_k, y_k, t - t_k) x_k^i y_k^j, \tag{4.49}$$

where the *weighting function* $w(x, y, \tau) : \mathbb{R}^3 \mapsto [0, 1]$ is a sufficiently smooth map, and t_k represents the time at which the point feature $\boldsymbol{\pi}_k$ is considered for the first time.

The weight w can be exploited to assign a 'quality' measure to each feature point so as to enforce a smooth change in m_{ij} whenever a tracked feature leaves visibility or a new feature is taken into consideration (regardless of its position on the image plane). In particular, we design the weight $w(x, y, \tau)$ as the product of three scalar functions

$$w(x, y, \tau) = w_1(x) w_2(y) w_3(\tau).$$

Weights $w_1(x)$ and $w_2(y)$ are designed so as to vanish at the image borders and are meant to smoothly take into account features entering/exiting the image plane.

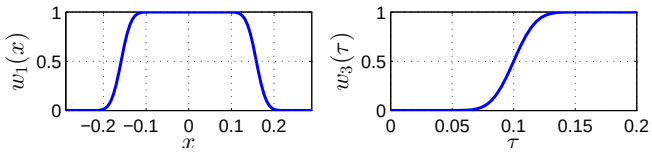


Figure 4.2 – **Possible shape of the weight functions used to cope with a limited camera FOV**. Left: shape of weight $w_1(x)$ with limits $x_{min} = -x_{max} = 0.2884$ (normalized size of the image plane). Right: shape of weight $w_3(\tau)$.

Weight $w_3(\tau)$ is finally intended to smoothly take into account the introduction of a newly detected feature point $\boldsymbol{\pi}_k$ when already within visibility (for instance, when starting to track a new point close to the image center). Figure 4.2 shows a possible shape for $w_1(x)$ (also representative of $w_2(y)$) and $w_3(\tau)$.

Exploiting the definition (4.49), it is easy (although tedious) to obtain an expression conceptually equivalent to (4.46) for the dynamics of the weighted moments $\dot{m}_{ij}$. Some details in this sense are reported in Appendix A.2. This then allows to directly apply the SfM scheme (3.11) to the case of weighted moments. We finally note that, in practice, the summation (4.49) is clearly evaluated only on the (*finite* but time-varying) set of currently tracked point features since $w(x_k,\, y_k,\, t - t_k) = 0$ for any $\boldsymbol{\pi}_k$ not visible or not considered at any time $t \leq t_k$.

As for which moments to consider for the estimation of $\boldsymbol{\chi}$, after some experimental tests we opted for

$$\boldsymbol{s} = (x_g,\, y_g,\, \mu_{20},\, \mu_{02},\, \mu_{11}) \in \mathbb{R}^m, \qquad m = 5. \tag{4.50}$$

This choice is partially motivated by [TC05] which proposed the triple $(x_g, y_g,\, \mu_{20} + \mu_{02})$ as a good set of features for controlling the camera translational DOFs in a VS loop. However, we empirically found this latter set to be ill-conditioned for what concerns the estimation of $\boldsymbol{\chi}$, with instead (4.50) providing enough information (i.e., full rankness of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$) for the estimation convergence. Alternatively, one could also resort to an adaptive/online selection of the best set of image moments as discussed in Sect. 4.5.2.3.

Finally, analogously to the previous cases, optimization of the structure estimation convergence from image moments can be achieved by maximizing w.r.t. $\boldsymbol{v}$ the smallest eigenvalue σ_1^2 of the square 3×3 matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ from (4.48).

4.5.2.3 Optimizing online the selection of moments

As already anticipated in the previous section, the selection of a good set of image moments for visual control or SfM is still an open problem. Ideally, for VS applications, one would like to find a unique set of visual features resulting in the 'most linear' control problem with the largest convergence domain. In case of SfM tasks,

instead, one is in general interested in maximizing observability for a given camera displacement. However, to the best of our knowledge, only local, partial (e.g., depending of the particular shape of the object) or heuristic results are currently available. For instance, [TC05, KDS+07, BCM13] propose different combinations of image moments able to only guarantee *local* stability of the servoing loop around the desired pose, with a basin of attraction to be heuristically determined case by case. As for what concerns the SfM case, the choice of which moments to exploit for allowing a converging estimation of the scene structure is also not straightforward. In [RDO08] the area a and barycenter coordinates $(x_g,\, y_g)$ of a dense region are successfully fed to a SfM scheme based on the (intuitive) motivation that the same set $(a,\, x_g,\, y_g)$ is also the typical choice for *controlling* the camera translational motion in a servoing loop [TC05]. However, this intuition breaks down when considering moments of a discrete point cloud: in this case, the typical choice for *controlling* the camera translational motion, that is, the set $(x_g,\, y_g,\, \mu_{20} + \mu_{02})$ (see [TC05]), is empirically shown in [4] to not provide enough information for allowing a converging estimation of the scene structure.

One could argue that the hope of finding a unique set of visual features optimal in all situations might eventually prove to be unrealistic, if not impossible, while it could just be more appropriate (and reasonable) to rely on an *automatic* and *online* selection of the best feature set (within a given class) tailored to the particular task at hand. Motivated by these considerations, we propose, in this section, a generalization of the image moments definition that aims in this direction.

Let $w = w(x,\, y,\, \boldsymbol{\theta})$ be a smooth function of the coordinates $(x,\, y)$ on the image plane and of a vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^h$. One can generalize (4.44) and define a *weighted parametric* image moment for N observed features $\boldsymbol{\pi}_k$

$$m_w(\boldsymbol{\theta}) = \sum_{k=1}^{N} w(x_k,\, y_k,\, \boldsymbol{\theta}), \tag{4.51}$$

with, obviously, $m_w = m_{ij}$ for $w(x,\, y,\, \boldsymbol{\theta}) = x^i y^j$. Function $w(x,\, y,\, \boldsymbol{\theta})$ can be seen as the *class* of all the considered image moments (e.g., a quadratic form in $x,\, y$) parametrized by vector $\boldsymbol{\theta}$ (e.g., the coefficients of the quadratic form). Consider now the following additional definitions

$$\begin{aligned} m^x_{w_{ij}}(\boldsymbol{\theta}) &= \sum\nolimits_{k=1}^{N} x_k^i y_k^j \left.\frac{\partial w(x,\, y,\, \boldsymbol{\theta})}{\partial x}\right|_{(x_k,\, y_k)} \\ m^y_{w_{ij}}(\boldsymbol{\theta}) &= \sum\nolimits_{k=1}^{N} x_k^i y_k^j \left.\frac{\partial w(x,\, y,\, \boldsymbol{\theta})}{\partial y}\right|_{(x_k,\, y_k)} \\ \boldsymbol{m}^{\boldsymbol{\theta}}_w(\boldsymbol{\theta}) &= \sum\nolimits_{k=1}^{N} \left.\nabla_{\boldsymbol{\theta}} w(x,\, y,\, \boldsymbol{\theta})^T\right|_{(x_k,\, y_k)}, \end{aligned} \tag{4.52}$$

and note that $\boldsymbol{m}_w^{\boldsymbol{\theta}}$ is a row vector of dimension h. Following the derivations in [TC05], it is easy to show that the dynamics of $m_w(\boldsymbol{\theta})$ takes the expression (reminiscent of (4.46))

$$\begin{aligned}\dot{m}_w(\boldsymbol{\theta}) =&[m_A(\boldsymbol{v},\,\boldsymbol{\theta}) \quad m_B(\boldsymbol{v},\,\boldsymbol{\theta}) \quad m_C(\boldsymbol{v},\,\boldsymbol{\theta})]\boldsymbol{\nu} \\ &+ [m_{\omega_x}(\boldsymbol{\theta}) \quad m_{\omega_y}(\boldsymbol{\theta}) \quad m_{\omega_z}(\boldsymbol{\theta})]\boldsymbol{\omega} + \boldsymbol{m}_w^{\boldsymbol{\theta}}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}\end{aligned} \tag{4.53}$$

with

$$\begin{cases} m_A =- m_{w_{10}}^x v_x - m_{w_{10}}^y v_y + (m_{w_{20}}^x + m_{w_{11}}^y)v_z \\ m_B =- m_{w_{01}}^x v_x - m_{w_{01}}^y v_y + (m_{w_{11}}^x + m_{w_{02}}^y)v_z \\ m_C =- m_{w_{00}}^x v_x - m_{w_{00}}^y v_y + (m_{w_{10}}^x + m_{w_{01}}^y)v_z \\ m_{\omega_x} =(m_{w_{11}}^x + m_{w_{02}}^y + m_{w_{00}}^y) \\ m_{\omega_y} =(-m_{w_{00}}^x - m_{w_{20}}^x - m_{w_{11}}^y) \\ m_{\omega_z} =(m_{w_{01}}^x - m_{w_{10}}^y) \end{cases}, \tag{4.54}$$

and $\boldsymbol{\nu} = [A,\, B,\, C]^T = -\boldsymbol{n}/d$.

One can then exploit (4.51–4.54) for implementing a visual control or estimation algorithm as in the classical case, but with the *additional* possibility of acting on vector $\boldsymbol{\theta}$ (a free parameter) for optimizing any criterion of interest, e.g., the norm of the observability matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ during an estimation task.

Clearly, there exist many possibilities for designing the weighting function $w(\cdot)$, i.e., the class of moments spanned by vector $\boldsymbol{\theta}$. A convenient choice, in our opinion, is to take $w(\cdot)$ as some *polynomial* basis in x and y with $\boldsymbol{\theta}$ being the vector of coefficients. Indeed, in this way the weighted moments (4.51), the expressions in (4.52) and, eventually, all the terms in (4.54) will reduce to linear combinations of the *unweighted moments* m_{ij} in (4.44). The overall computational complexity will then result equivalent to the classical case [TC05].

As for which polynomial basis to exploit, many choices are possible depending on the constraints/requirements of the particular application. Within the scope of this work, two possibilities are considered:

Polynomial basis of fixed degree First, one can take $w(\cdot)$ as a polynomial in x and y of a given *degree* $\delta \in \mathbb{N}_+$, that is,

$$w(x,\, y,\, \boldsymbol{\theta}) = \sum_{j=1}^{\delta}\sum_{k=0}^{j} \theta_{T_j+k} x^{(j-k)} y^k \tag{4.55}$$

with $T_j = \binom{j+1}{2}$ and $\boldsymbol{\theta} = (\theta_1, \ldots\ldots \theta_{T_\delta+\delta}) \in \mathbb{R}^{T_\delta+\delta}$. Indeed, this allows (4.51) to span all the moment linear combinations of order up to δ with coefficients in vector

$\boldsymbol{\theta}$. As illustration, by choosing $\delta = 2$ in (4.55), one obtains the following quadratic polynomial

$$w(x,\, y,\, \boldsymbol{\theta}) = \theta_1 x + \theta_2 y + \theta_3 x^2 + \theta_4 xy + \theta_5 y^2$$

that, when plugged in (4.51), yields

$$m_w(\boldsymbol{\theta}) = \theta_1 m_{10} + \theta_2 m_{01} + \theta_3 m_{20} + \theta_4 m_{11} + \theta_5 m_{02}. \tag{4.56}$$

The class (4.56) can then specialize into, e.g., the barycenter coordinate x_g for $\boldsymbol{\theta} = (1/N,\, 0,\, 0,\, 0,\, 0)$, the centered moment μ_{02} for $\boldsymbol{\theta} = (0,\, -y_g,\, 0,\, 0,\, 1)$, and so on. Clearly, the larger the value of the degree δ, the richer the basis representation power in encoding the scene geometry, but at the (well-known) cost of an increasing noise level with the moment order.

Constrained polynomial basis A second possibility is to design a *constrained* polynomial basis for coping with the possible loss/gain of point features during the camera motion because of the limited camera FOV. Indeed, by imposing that $w(\cdot)$ vanishes (with vanishing derivative) at the image borders, any point feature close to the limits will *smoothly* enter or leave the image plane and, thus, prevent any discontinuity in the moment dynamics (4.53).

Let then $x_{min} < x_{max}$ and $y_{min} < y_{max}$ represent the limits of a rectangular image plane, and consider a weighting function $w(\cdot)$ partitioned as

$$w(x,\, y,\, \boldsymbol{\theta}) = w^x(x, \boldsymbol{\theta}^x) w^y(y, \boldsymbol{\theta}^y), \tag{4.57}$$

where $w^x(x, \boldsymbol{\theta}^x)$ and $w^x(x, \boldsymbol{\theta}^y)$ are polynomial bases and $\boldsymbol{\theta} = (\boldsymbol{\theta}^x,\, \boldsymbol{\theta}^y) \in \mathbb{R}^{h_x + h_y}$, $h_x + h_y = h$, is the vector of coefficients. Assuming $h_x \geq 4$ and imposing

$$\begin{cases} w^x(x_{min}, \boldsymbol{\theta}^x) = w^x(x_{max}, \boldsymbol{\theta}^x) = 0 \\ \left.\dfrac{\partial w^x(x, \boldsymbol{\theta}^x)}{\partial x}\right|_{x_{min}} = \left.\dfrac{\partial w^x(x, \boldsymbol{\theta}^x)}{\partial x}\right|_{x_{max}} = 0 \end{cases}, \tag{4.58}$$

one can solve for a set of 4 parameters in vector $\boldsymbol{\theta}^x$ for shaping $w^x(x, \boldsymbol{\theta}^x)$ as desired. For instance, by taking

$$w^x(x,\, \boldsymbol{\theta}^x) = \theta_1^x x^5 + \theta_2^x x^4 + \theta_3^x x^3 + \theta_4^x x^2 + \theta_5^x x + \theta_6^x \tag{4.59}$$

and by (arbitrarily) choosing the pair $(\theta_1^x,\, \theta_2^x)$ as free parameters in vector $\boldsymbol{\theta}^x$,

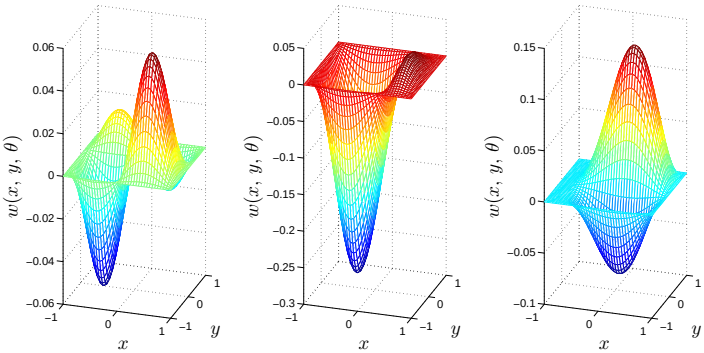


Figure 4.3 – **Three examples of the constrained polynomial basis** $w(x, y, \boldsymbol{\theta})$ **in (4.57–4.60).** Note how $w(x, y, \boldsymbol{\theta})$ smoothly vanishes at the image borders of size $[-1, 1] \times [-1, 1]$

system (4.58) yields

$$\begin{cases} \theta_3^x = (-3x_{max}^2 - 3x_{min}^2 - 4x_{max}x_{min})\theta_1^x + (-2x_{min} - 2x_{max})\theta_2^x \\ \theta_4^x = (2x_{max}^3 + 8x_{max}^2 x_{min} + 8x_{max}x_{min}^2 + 2x_{min}^3)\theta_1^x \\ \qquad + (x_{max}^2 + 4x_{max}x_{min} + x_{min}^2)\theta_2^x \\ \theta_5^x = (-7x_{max}^2 x_{min}^2 - 4x_{min}^3 x_{max} - 4x_{max}^3 x_{min})\theta_1^x \\ \qquad + (-2x_{max}x_{min}^2 - 2x_{max}^2 x_{min})\theta_2^x \\ \theta_6^x = (2x_{max}^3 x_{min}^2 + 2x_{max}^2 x_{min}^3)\theta_1^x + x_{max}^2 x_{min}^2 \theta_2^x. \end{cases} \tag{4.60}$$

Imposing analogous conditions to function $w^y(y, \boldsymbol{\theta}^y)$ at y_{min} and y_{max} (with again $h_y \geq 4$) will then constrain a total of 8 parameters in vector $\boldsymbol{\theta}$, with the remaining $h - 8$ coefficients still free to be exploited for optimization purposes. For the sake of illustration, Figure 4.3 shows three examples of weighting functions $w(\cdot)$ smoothly vanishing at the borders of an image plane of size $[-1, 1] \times [-1, 1]$ and obtained by picking at random three values for the free parameters in vector $\boldsymbol{\theta}$.

We conclude by noting that, compared to the previous case (4.55), this latter possibility necessitates of a polynomial basis (4.57) with a degree of at least 7. Indeed, as explained, the vanishing conditions at the image border will constrain $4 + 4$ coefficients in $\boldsymbol{\theta}^x$ and $\boldsymbol{\theta}^y$, thus forcing both $w^x(x, \boldsymbol{\theta}^x)$ and $w^y(y, \boldsymbol{\theta}^y)$ to have a degree of (at least) 3 for ensuring $h_x \geq 4$ and $h_y \geq 4$ as required (see (4.59)). However, for any *optimization* of the coefficient vector $\boldsymbol{\theta}$ to be possible, either $h_x > 4$ or $h_y > 4$ must hold for allowing presence of at least *one free coefficient* to be optimized besides those already constrained by the vanishing conditions. On the other hand, if either $h_x > 4$ or $h_y > 4$, the final polynomial basis (4.57) will necessarily result of at least degree 7. Therefore, the use of higher-order moments (of at least order 7) is the 'price to pay' for smoothly taking into account the

loss/gain of point features during the camera motion[2]. In contrast, the degree of the polynomial basis in (4.55) can be chosen at will and thus adjusted, if necessary, for limiting the noise level in the measured moments.

Optimization of the weighted parametric image moments Consider now a set of $m \geq 3$ *weighted* moments

$$\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1), \ldots, m_w(\boldsymbol{\theta}_m)) \in \mathbb{R}^m$$

with $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m) \in \mathbb{R}^h$ being the stack of all parameters. Plugging the weighted moment dynamics (4.53–4.54) in the definition (4.48), one has

$$\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}) = \begin{bmatrix} m_A(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_1) & \cdots & m_A(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_m) \\ m_B(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_1) & \cdots & m_B(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_m) \\ m_C(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_1) & \cdots & m_C(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta}_m) \end{bmatrix} \in \mathbb{R}^{3\times m}. \tag{4.61}$$

Therefore, when employing the weighted parametric moments (4.51) instead of the classical moments (4.44), one gains the additional possibility of *also* acting on vector $\boldsymbol{\theta}$ (i.e., on the 'moment shape') for affecting matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$.

By applying (4.26) to matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ and expanding the various terms, one obtains

$$\begin{aligned} \dot{\Phi}_D = &\sum_i \operatorname{tr}\left(\operatorname{adj}\left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right) \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial v_i}\right) \dot{v}_i + \sum_i \operatorname{tr}\left(\operatorname{adj}\left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right) \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial \theta_i}\right) \dot{\theta}_i \\ &+ \sum_i \operatorname{tr}\left(\operatorname{adj}\left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right) \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial s_i}\right) \dot{s}_i = \boldsymbol{J}_{\boldsymbol{v}}\dot{\boldsymbol{v}} + \boldsymbol{J}_{\theta}\dot{\boldsymbol{\theta}} + \boldsymbol{J}_{\boldsymbol{s}}\dot{\boldsymbol{s}} \end{aligned} \tag{4.62}$$

where the Jacobian matrices

$$\begin{aligned} \boldsymbol{J}_{\boldsymbol{v}} &= \left[\ldots \operatorname{tr}\left(\operatorname{adj}\left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right) \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial v_i}\right) \ldots\right] \in \mathbb{R}^{1\times 3} \\ \boldsymbol{J}_{\theta} &= \left[\ldots \operatorname{tr}\left(\operatorname{adj}\left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right) \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial \theta_i}\right) \ldots\right] \in \mathbb{R}^{1\times h} \\ \boldsymbol{J}_{\boldsymbol{s}} &= \left[\ldots \operatorname{tr}\left(\operatorname{adj}\left(\boldsymbol{\Omega}\boldsymbol{\Omega}^T\right) \frac{\partial(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)}{\partial s_i}\right) \ldots\right] \in \mathbb{R}^{1\times m} \end{aligned} \tag{4.63}$$

are function of $(\boldsymbol{s}, \boldsymbol{v}, \boldsymbol{\theta})$ (all available quantities). We stress that all the terms in (4.63) can be computed in closed-form.

The relation (4.62) can then be exploited for affecting $\Phi_D(t)$ over time by acting on $\dot{\boldsymbol{v}}$ (the camera linear acceleration) *and/or* $\dot{\boldsymbol{\theta}}$ (the parameter vector). Among the many possibilities, we considered here the following update rules

$$\begin{cases} \dot{\boldsymbol{v}} = k_v \left(\boldsymbol{I}_3 - \dfrac{\boldsymbol{v}\boldsymbol{v}^T}{\boldsymbol{v}^T\boldsymbol{v}}\right) \boldsymbol{J}_{\boldsymbol{v}}^T \\ \dot{\boldsymbol{\theta}} = k_\theta \left(\boldsymbol{I}_3 - \dfrac{\boldsymbol{\theta}\boldsymbol{\theta}^T}{\boldsymbol{\theta}^T\boldsymbol{\theta}}\right) \boldsymbol{J}_{\theta}^T \end{cases}, \qquad k_{\boldsymbol{v}} > 0,\ k_\theta > 0 \tag{4.64}$$

[2]Of course, the use of different functional bases, also non-polynomial, could be possible.

which are meant to maximize $\Phi_D(t)$ by following its gradient w.r.t. $(\boldsymbol{v}, \boldsymbol{\theta})$ projected on the null-spaces of the constant-norm constraints $\|\boldsymbol{v}(t)\| = const$ and $\|\boldsymbol{\theta}(t)\| = const$. As explained in Sect. 4.4, the constraint $\|\boldsymbol{v}(t)\| = const$ is meant to prevent a better conditioning of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ *only* due to a faster camera motion while observing the scene. The second constraint $\|\boldsymbol{\theta}(t)\| = const$ is motivated by similar arguments: an increasing $\|\boldsymbol{\theta}(t)\|$ would (artificially) magnify $\Phi_D(t)$ at the cost of an increased noise level (all the terms in (4.54) would just result amplified).

The optimization action (4.64) will then maximize the observability measure $\det(\boldsymbol{\Omega}\boldsymbol{\Omega}^T)$ for the SfM task at hand by:(i) adjusting the direction of the camera linear velocity $\boldsymbol{v}$ (as with the other case studies presented so far) and, *at the same time*, (ii) by adapting the shape of the m weighted moments $\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1), \ldots, m_w(\boldsymbol{\theta}_m))$ as only a function of the perceived scene and camera motion. We also remark that (4.64) (or any other equivalent strategy) assumes the possibility of acting at will on the direction of the linear camera velocity $\boldsymbol{v}$. There could be cases where this is not (fully) possible, and $\boldsymbol{v}$ (or components of it) are given (for example during a combined estimation/servoing loop as ii will be discussed in Chapt. 6). In all these cases, it is obviously still possible to just keep on optimizing $\boldsymbol{\theta}(t)$ during the (given/known) camera motion in order to adapt, as best as possible, the moment shape. Finally, since $\boldsymbol{\Omega}(t)$ (and thus $\Phi_D(t)$) does *not* depend on the camera angular velocity, one can freely choose $\boldsymbol{\omega}$ to fulfill any additional goal of interest.

4.5.2.4 Using dense image moments

We conclude the analysis of the plane estimation case study by discussing a possible extension of these techniques to the case of dense image moments. In some context, in fact, the (planar) scene observed by the camera might not be sufficiently textured to allow for the extraction and tracking of a sufficient number of point features in a reliable way. For this kind of situations it might be better to segment and track some objects or "patches" in the images and use a "dense" definition of the (i, j)-th order image moments in which the summation in (4.44) (and similar definitions) is substituted by a continuous integral over the image of object $\mathcal{O}$

$$m_{ij} = \iint_{\mathcal{O}} x^i y^j \, dxdy. \tag{4.65}$$

Still from [Cha04], the dynamics of m_{ij} can be shown to take the expression

$$\dot{m}_{ij} = \boldsymbol{L}_{m_{ij}}(m_{kl}, \boldsymbol{\chi})\, \boldsymbol{u} = f_{m_{ij}}(m_{kl}, \boldsymbol{\omega}) + \boldsymbol{\varpi}^T_{m_{ij}}(m_{kl}, \boldsymbol{v})\boldsymbol{\nu} \tag{4.66}$$

where m_{kl} stands for a generic (k, l)-th moment of order up to $i + j + 1$ and, again, $\boldsymbol{\nu} = -\boldsymbol{n}/d$. Note that (4.66) is linear in the unmeasurable $\boldsymbol{\chi}$, while all the other quantities are available to measurement. Let then $\boldsymbol{s} = [m_{i_1 j_1} \ldots m_{i_m j_m}]^T \in \mathbb{R}^m$

be a collection of m image moments, $\boldsymbol{\varsigma} = [m_{k_1 l_1} \dots m_{k_r l_r}]^T \in \mathbb{R}^r$ be the set of moments that appear in the dynamics of $\boldsymbol{s}$ but are not part of $\boldsymbol{s}$, and $\boldsymbol{\chi} = \boldsymbol{\nu}$. Using again (4.47), formulation (3.10) can be recovered with

$$\begin{cases} \boldsymbol{f_s}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{\omega}) = \left[f_{m_{i_1 j_1}}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{\omega}) \ \dots \ f_{m_{i_m j_m}}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{\omega}) \right]^T \\ \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{v}) = \left[\boldsymbol{\varpi}_{m_{i_1 j_1}}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{v}) \ \dots \ \boldsymbol{\varpi}_{m_{i_m j_m}}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{v}) \right] \\ \boldsymbol{f_\chi}(\boldsymbol{\chi}, \boldsymbol{u}) = \boldsymbol{\chi}\boldsymbol{\chi}^T \boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{\chi} \\ \boldsymbol{d}(\widetilde{\boldsymbol{\chi}}, t) = (\widehat{\boldsymbol{\chi}}\widehat{\boldsymbol{\chi}}^T - \boldsymbol{\chi}\boldsymbol{\chi}^T)\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{e}_3 \end{cases}.$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{p \times m}$, $p = 3$. As before, it is $\boldsymbol{d}(\boldsymbol{\emptyset}_3, t) = \boldsymbol{\emptyset}_3$.

As with the discrete moments case, choice of which moments to include in $\boldsymbol{s}$ is in general not obvious as many possibilities exist (in number and kind). The adaptive techniques, described in the previous sections, for selecting online the set of moments to use and for dealing with the limitations of the camera FOV, can easily be extended to the case of dense image moments. However, for the results in Sect. 5.2.5, we limited our analysis to the use of the lowest-order moments because of their robustness w.r.t. image noise: $\boldsymbol{s} = [a, x_g, y_g]^T$, i.e., the area a and barycenter (x_g, y_g) of the observed object. This choice, originally suggested by [RDO08], implies $m = p = 3$, thus yielding a 'square' problem (square matrix $\boldsymbol{\Omega}$). Note that, in this case, vector $\boldsymbol{\varsigma} = [n_{20}, n_{11}, n_{02}]^T \in \mathbb{R}^3$ (the normalized centered moments of order 2) thus yielding $r = 3$, see [Cha04]. For the case under consideration, matrix $\boldsymbol{\Omega}$ then takes the expression

$$\boldsymbol{\Omega} = \begin{bmatrix} 3ax_g v_z - av_z & \left(x_g^2 + 4n_{20}\right) v_z - x_g v_z & \left(x_g y_g + 4n_{11}\right) v_z - x_g v_y \\ 3ay_g v_z - av_y & \left(x_g y_g + 4n_{11}\right) v_z - y_g v_z & \left(y_g^2 + 4n_{02}\right) v_z - y_g v_y \\ 2av_z & x_g v_z - v_z & y_g v_z - v_y \end{bmatrix} \tag{4.67}$$

We can note that, as in all other cases, $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{v})$ thus allowing to exploit the camera angular velocity $\boldsymbol{\omega}$ for fulfilling additional tasks of interest without affecting observability. For example, analogously to the point feature case, one could try to keep the barycenter at a constant position $(x_g, y_g) \simeq const$ in order to mitigate the effects of $\dot{\boldsymbol{s}}$ in (4.22). Furthermore, it is interesting to note that matrix $\boldsymbol{\Omega}$ (and, therefore, matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ as well) loses rank whenever $v_z = 0$: in order to meet condition (3.13), that is, to keep $\sigma_1^2(t) > 0$, the camera then necessarily needs to translate with a non-zero component along the optical axis regardless of the orientation of the plane[3]. No special insights can be gained, instead, from the inspection of the Jacobian matrix $\boldsymbol{J_v}$.

[3] A requirement not present in the point-feature case where any non-zero linear velocity not aligned with the projection ray could guarantee fulfillment of (3.13).

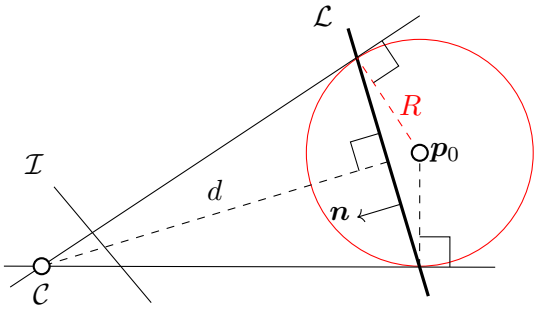


Figure 4.4 – **Geometry of a spherical object**. Spherical target $\mathcal{O}_S$ and planar limb surface $\mathcal{L}$.

We conclude this section by mentioning the interesting possibility of employing the *photometric image moments*, as defined in [BCM13], to estimate the plane structure. The general expression of the $(i,\, j)$-th order photometric image moment is in fact

$$m_{ij} = \iint_{\mathcal{I}} Y(x,y) x^i y^j \, dxdy. \tag{4.68}$$

where $Y(x,y)$ is the intensity level of the image in position (x,y) and the integral is extended to the entire image $\mathcal{I}$. With this new definition, the image processing steps (segmentation, tracking and so on) are simplified even further since it is not necessary to identify neither a specific set of points $\boldsymbol{p}_k$ nor a particular object $\mathcal{O}$ as it is the case, instead for (4.44–4.65) and other similar definitions introduced in this section. Along the same line, Appendix B will present some preliminary original results on the *direct* use of the photometric information for SfM estimation.

4.5.3 Active Structure from Motion for a sphere

We now detail the application of the proposed estimation machinery to the case of a spherical target. Consider a sphere $\mathcal{O}_S$ of radius R and let $\boldsymbol{p}_0 = (X_0,\, Y_0,\, Z_0)$ be the coordinates of its center in the camera frame. Let also

$$\mathcal{L}: \quad \boldsymbol{n}^T \boldsymbol{p} + d = 0$$

represent the *planar limb surface* associated to the sphere in the camera frame, where $\boldsymbol{p} \in \mathbb{R}^3$ is any 3-D point on the plane, $\boldsymbol{n} \in \mathbb{S}^2$ is the plane unit normal vector and $d \in \mathbb{R}$ the plane distance to the camera center [Cha04]. Figure 4.4 shows the quantities of interest.

The depth Z of any point $\boldsymbol{p}$ lying on $\mathcal{L}$ can be expressed in terms of its normalized image coordinates $\boldsymbol{\pi} = (x,\, y,\, 1)$ as

$$\frac{1}{Z} = \frac{X_0}{K} x + \frac{Y_0}{K} y + \frac{Z_0}{K} = \boldsymbol{\nu}^T \boldsymbol{\pi}, \tag{4.69}$$

where $K = \boldsymbol{p}_0^T \boldsymbol{p}_0 - R^2$ and $\boldsymbol{\nu} = \boldsymbol{p}_0/K = -\boldsymbol{n}/d \in \mathbb{R}^3$ represent *unmeasurable* quantities (analogously to Z for the point feature case), see [ECR92] for all the details. As discussed in Sect. 4.5.2, the interaction matrix of a generic (i, j)-th order moment m_{ij} evaluated on the image of $\mathcal{O}_S$ depends linearly on $\boldsymbol{\nu}$, see again [Cha04, RDO08]. Therefore, a first possibility to retrieve the sphere 3-D parameters $(\boldsymbol{p}_0, R)$ would be to implement the estimation scheme (3.11) with $\boldsymbol{s}$ being a suitable collection of image moments (e.g., area and barycenter) and $\boldsymbol{\chi} = \boldsymbol{\nu}$. It is in fact possible to show that (see Appendix A.3)

$$\dot{\boldsymbol{\chi}} = -\frac{\boldsymbol{v}}{K} - [\boldsymbol{\omega}]_\times \boldsymbol{\chi} + 2\boldsymbol{\chi}\boldsymbol{\chi}^T \boldsymbol{v}$$

and that K can be expressed in terms of image moments and of vector $\boldsymbol{\chi}$ itself, so that, having estimated $\boldsymbol{\chi}$, one can consequently retrieve $\boldsymbol{p}_0 = \boldsymbol{\chi} K$ and $R = \sqrt{\boldsymbol{p}_0^T \boldsymbol{p}_0 - K}$.

Although conceptually valid, this solution requires the concurrent estimation of *three time-varying quantities* (vector $\boldsymbol{\chi}(t)$). On the other hand, inspired by [FC09], we now describe a *novel representation* of the sphere projection on the image plane that allows to reformulate the structure estimation task in terms of a *single unknown constant parameter*, i.e., the sphere radius R.

To this end, define vector $\boldsymbol{s} = (s_x, s_y, s_z) \in \mathbb{R}^3$ as

$$\begin{cases} s_x = \dfrac{x_g}{s_z a_1^2} \\ s_y = \dfrac{y_g}{s_z a_1^2} \\ s_z = \sqrt{\dfrac{1 + a_1^2}{a_1^2}} \end{cases}, \tag{4.70}$$

where $(x_g, y_g, n_{20}, n_{11}, n_{02})$ represent the barycenter and normalized centered moments of order 2 measured from the elliptical projection of the sphere $\mathcal{O}_S$ on the image plane, and a_1 is the minor axis of the observed ellipse with [Cha04]

$$a_1^2 = 2\left(n_{20} + n_{02} - \sqrt{(n_{20} - n_{02})^2 + 4n_{11}}\right). \tag{4.71}$$

We thus note that vector $\boldsymbol{s}$ can be directly evaluated in terms of measured image quantities. From [Cha04, FC09] one also has

$$x_g = \frac{X_0 Z_0}{Z_0^2 - R^2}, \quad y_g = \frac{Y_0 Z_0}{Z_0^2 - R^2}, \quad a_1^2 = \frac{R^2}{Z_0^2 - R^2} \tag{4.72}$$

which, when plugged in (4.70–4.71), result in the equivalent expression $\boldsymbol{s} = \boldsymbol{p}_0/R$. Since vector $\boldsymbol{s}$ can be computed from image measurements as in (4.70), estimation of the (unknown) sphere radius R allows to recover the 3-D sphere center as $\boldsymbol{p}_0 = \boldsymbol{s}R$.

Exploiting now the results of [FC09], it is possible to show that

$$\dot{\boldsymbol{s}} = \begin{bmatrix} -\frac{1}{R}\boldsymbol{I}_3 & [\boldsymbol{s}]_\times \end{bmatrix} \boldsymbol{v}. \tag{4.73}$$

Since (4.73) is linear in $1/R$, we can define $\chi = 1/R$, with then $m = 3$ and $p = 1$, and obtain for (3.10–3.12)

$$\begin{cases} \boldsymbol{f}_{\boldsymbol{s}}(\boldsymbol{s}, \boldsymbol{\omega}) = [\boldsymbol{s}]_\times \boldsymbol{\omega} \\ \boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v}) = -\boldsymbol{v}^T \\ f_\chi(\boldsymbol{s}, \chi, \boldsymbol{u}) = 0 \\ d(\widetilde{\mathbf{x}}, t) = 0 \end{cases}. \tag{4.74}$$

We note that in this case it is always possible to obtain *global* convergence for the estimation error since $\dot{\chi} = 0$ and therefore $\boldsymbol{d}(\widetilde{\mathbf{x}}, t) = 0$ by construction (see Remark 3.1). Furthermore, matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ reduces again to its single eigenvalue $\sigma_1^2 = \|\boldsymbol{v}\|^2$ and, if $\sigma_1^2(t) \equiv const > 0$, the 'ideal' estimation error dynamics (4.15) can be exactly obtained. One also has $\boldsymbol{\Omega} = \boldsymbol{\Omega}(\boldsymbol{v})$ and $\boldsymbol{J}_{\boldsymbol{v},1} = 2\boldsymbol{v}^T$.

We finally note the following facts: first of all, contrarily to the previous cases, here $\dot{\boldsymbol{s}}$ has no effect on the regulation of σ_1^2 which is only function of the camera linear velocity $\boldsymbol{v}$. As usual, it is of course still possible to freely exploit the camera angular velocity $\boldsymbol{\omega}$ for, e.g., keeping the sphere at the center of the image by regulating (s_x, s_y) to zero. Second, we note the strong similarities with the previous optimal results obtained for a point feature under a *spherical* projection model (σ_{max}^2 in (4.37)): in both cases the maximum estimation convergence rate for a given $\|\boldsymbol{v}\|$ does not depend on the position of the observed object on the image plane. Differently from the case of the point, however, one now has $\sigma_1^2 = \|\boldsymbol{v}\|^2$ *regardless* of the direction of $\boldsymbol{v}$: due to the spherical symmetry of the observed object any direction of motion is equally informative. Moreover, since the sphere has a finite non-zero dimension, moving in its direction causes a variation of the measurement $\boldsymbol{s}$ (the sphere will appear bigger or smaller in the image) that makes this motion informative for the estimation task, differently from the point feature case.

4.5.4 Active Structure from Motion for a cylinder

We now finally consider the case of SfM for a 3-D cylindrical object. A cylinder $\mathcal{O}_C$ can be described by its radius $R > 0$ and by its main axis $\boldsymbol{a} \in \mathbb{S}^2$ passing through a 3-D point $\boldsymbol{p}_0 = (X_0, Y_0, Z_0)$, with $\|\boldsymbol{a}\| = 1$ and, w.l.o.g., $\boldsymbol{a}^T\boldsymbol{p}_0 = 0$ ($\boldsymbol{p}_0$ can be chosen as the closest point on $\boldsymbol{a}$ to the origin of the camera frame [CBBJ96]). Moreover, analogously to the sphere, a cylinder is also associated with a planar limb

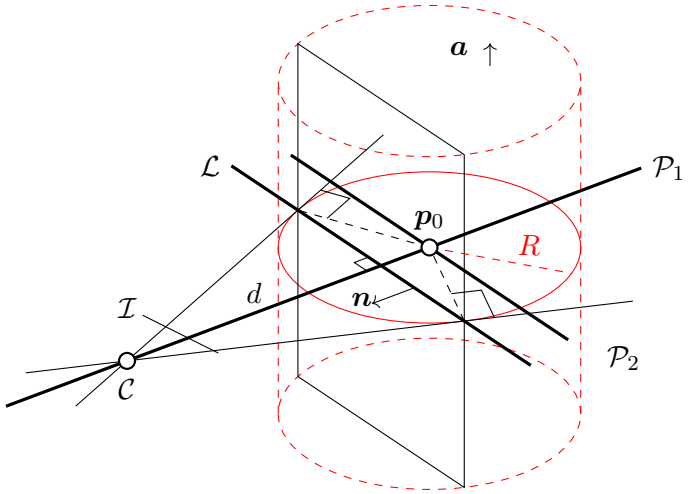


Figure 4.5 – **Geometry of a cylindrical object**. Camera $\mathcal{C}$ and cylindrical target $\mathcal{O}_C$ with the planar limb surface $\mathcal{L}$ and the other planes of interest $\mathcal{P}_1$ and $\mathcal{P}_2$

surface $\mathcal{L}$ such that (4.69) holds for any point on $\mathcal{L}$ with projection $\boldsymbol{\pi} = (x,\, y,\, 1)$. Therefore, as for the case of a sphere, a first possibility is to estimate the three unknown parameters of the limb plane $\mathcal{L}$ (with $\boldsymbol{\chi} = \boldsymbol{\nu}$) by exploiting (at least) three image measurements, see [CBBJ96] and Appendix A.4 for some details in this sense. However, following the previous developments, we now propose a *novel representation* of the cylinder projection on the image plane which, again, allows to obtain the cylinder parameters $(\boldsymbol{p}_0,\, \boldsymbol{a}, R)$ in terms of image measurements and of the *unknown but constant* cylinder radius R which, therefore, represents the only quantity to be estimated.

Let $(\rho_1,\, \theta_1)$ and $(\rho_2,\, \theta_2)$ be the (*measured*) distance/angle parameters of the two straight lines resulting from the projection of the cylinder on the image plane, and

$$\boldsymbol{n}_1 = (\cos\theta_1,\, \sin\theta_1,\, -\rho_1),\ \boldsymbol{n}_2 = (\cos\theta_2,\, \sin\theta_2,\, -\rho_2) \tag{4.75}$$

be the normal vectors to the two planes passing through the origin of the camera frame and the two above-mentioned projected lines[4]. Figure 4.5 gives a graphical representation of the quantities of interest. Note that vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ can be directly evaluated from image measurements (the line parameters). We then define vector $\boldsymbol{s} \in \mathbb{R}^3$ as

$$\boldsymbol{s} = \frac{\boldsymbol{\Delta}}{\|\boldsymbol{\Delta}\|^2} \tag{4.76}$$

with

$$\boldsymbol{\Delta} = \frac{1}{2}\left(\frac{\boldsymbol{n}_1}{\|\boldsymbol{n}_1\|} + \frac{\boldsymbol{n}_2}{\|\boldsymbol{n}_2\|}\right). \tag{4.77}$$

Vector $\boldsymbol{s}$ is, thus, also directly obtainable in terms of image quantities.

[4]The two planes are therefore tangent to the surface of the cylinder.

We now note that, from [Cha94], an equivalent expression for vectors $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ in terms of the cylinder 3-D geometry can be obtained as

$$\boldsymbol{n}_1 = \frac{1}{N_1}\begin{bmatrix} R\dfrac{X_0}{\sqrt{K}} - a \\ R\dfrac{Y_0}{\sqrt{K}} - b \\ R\dfrac{Z_0}{\sqrt{K}} - c \end{bmatrix}, \quad \boldsymbol{n}_2 = \frac{1}{N_2}\begin{bmatrix} R\frac{X_0}{\sqrt{K}} + a \\ R\dfrac{Y_0}{\sqrt{K}} + b \\ R\dfrac{Z_0}{\sqrt{K}} + c \end{bmatrix} \tag{4.78}$$

with

$$\begin{cases} K = \sqrt{\boldsymbol{p}_0^T\boldsymbol{p}_0 - R^2} \\ (a,\, b,\, c) = [\boldsymbol{p}_0]_\times \boldsymbol{a} \\ N_1 = \sqrt{\left(R\dfrac{X_0}{\sqrt{K}} - a\right)^2 + \left(R\dfrac{Y_0}{\sqrt{K}} - b\right)^2} \\ N_2 = \sqrt{\left(R\dfrac{X_0}{\sqrt{K}} + a\right)^2 + \left(R\dfrac{Y_0}{\sqrt{K}} - b\right)^2} \end{cases}, \tag{4.79}$$

thus yielding

$$\begin{cases} \dfrac{\boldsymbol{n}_1}{\|\boldsymbol{n}_1\|} = \dfrac{1}{\boldsymbol{p}_0^T\boldsymbol{p}_0}\begin{bmatrix} RX_0 - a\sqrt{K} \\ RY_0 - b\sqrt{K} \\ RZ_0 - c\sqrt{K} \end{bmatrix} \\ \dfrac{\boldsymbol{n}_2}{\|\boldsymbol{n}_2\|} = \dfrac{1}{\boldsymbol{p}_0^T\boldsymbol{p}_0}\begin{bmatrix} RX_0 + a\sqrt{K} \\ RY_0 + b\sqrt{K} \\ RZ_0 + c\sqrt{K} \end{bmatrix} \end{cases}. \tag{4.80}$$

Plugging (4.80) in (4.77) results in the equivalent expression

$$\boldsymbol{\Delta} = \frac{R^2}{\boldsymbol{p}_0^T\boldsymbol{p}_0}\boldsymbol{s}$$

which, using (4.76), finally yields the following relationship between image quantities and cylinder 3-D structure

$$\boldsymbol{s} = \frac{\boldsymbol{\Delta}}{\|\boldsymbol{\Delta}\|^2} = \frac{\boldsymbol{p}_0}{R}. \tag{4.81}$$

As for the cylinder axis $\boldsymbol{a}$, exploiting (4.78) one has

$$\begin{aligned} [\boldsymbol{n}_2]_\times \boldsymbol{n}_1 &= \frac{2R}{N_1N_2\sqrt{K}}\begin{bmatrix} Z_0b - Y_0c \\ X_0c - Z_0a \\ Y_0a - X_0b \end{bmatrix} = \frac{2R}{N_1N_2\sqrt{K}}\begin{bmatrix} a \\ b \\ c \end{bmatrix}_\times \boldsymbol{p}_0 \\ &= \frac{2R}{N_1N_2\sqrt{K}}\left[[\boldsymbol{p}_0]_\times \boldsymbol{n}\right]_\times \boldsymbol{p}_0 = \frac{2R\boldsymbol{p}_0^T\boldsymbol{p}_0}{N_1N_2\sqrt{K}}\boldsymbol{a} \end{aligned} \tag{4.82}$$

where in the last step the property $\boldsymbol{a}^T\boldsymbol{p}_0 = 0$ was used. Since $\|\boldsymbol{a}\| = 1$, from (4.82) it is

$$\boldsymbol{a} = \frac{[\boldsymbol{n}_2]_\times \boldsymbol{n}_1}{\left\|[\boldsymbol{n}_2]_\times \boldsymbol{n}_1\right\|}. \tag{4.83}$$

The cylinder axis $\boldsymbol{a}$ can then be directly obtained in terms of only measured quantities.

We now note that, as in the sphere case, the only unknown left is the cylinder radius R: once known, the cylinder 3-D structure can be fully recovered from image measurements as $\boldsymbol{p}_0 = R\boldsymbol{s}$ from (4.81) and $\boldsymbol{a}$ from (4.83). An estimation scheme for R can be obtained exploiting the following differential relationship whose derivation is given in Appendix A.5

$$\dot{\boldsymbol{s}} = \begin{bmatrix} -\frac{1}{R}\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right) & [\boldsymbol{s}]_\times \end{bmatrix} \boldsymbol{u}. \tag{4.84}$$

Note the similarity of (4.84) with (4.73) for the sphere case.

Being (4.84) linear in $1/R$, one can then apply observer (3.11) by choosing $\boldsymbol{s} = \boldsymbol{s}$, $\boldsymbol{\varsigma} = \boldsymbol{a}$, $\chi = 1/R$ with $m = 3$, $r = 3$ and $p = 1$, and obtaining

$$\begin{cases} \boldsymbol{f}_{\boldsymbol{s}}\left(\boldsymbol{s}, \boldsymbol{\omega}\right) = [\boldsymbol{s}]_\times \boldsymbol{\Omega} \\ \boldsymbol{\Omega}\left(\boldsymbol{s}, \boldsymbol{\varsigma}, \boldsymbol{v}\right) = -\boldsymbol{v}^T\left(\boldsymbol{I}_3 - \boldsymbol{\varsigma}\boldsymbol{\varsigma}^T\right) \\ f_{\boldsymbol{\chi}}\left(\boldsymbol{s}, \chi, \boldsymbol{u}\right) = 0 \\ d(\widetilde{\mathbf{x}}, t) = 0 \end{cases}. \tag{4.85}$$

Note how, again, being $\dot{\chi} = 0$ it is $d(\widetilde{\mathbf{x}}, t) = 0$ (global convergence for the error system (3.12) as in the sphere case).

Matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ reduces to its single eigenvalue

$$\sigma_1^2 = \boldsymbol{\Omega}\boldsymbol{\Omega}^T = \|\boldsymbol{v}\|^2 - (\boldsymbol{\varsigma}^T\boldsymbol{v})^2. \tag{4.86}$$

It is worth comparing (4.86) with the result obtained for the sphere ($\sigma_1^2 = \|\boldsymbol{v}\|^2$). In the cylinder case, the convergence rate of the estimation error is affected by both the *norm* and the *direction* of the linear velocity $\boldsymbol{v}$. In particular, for a given $\|\boldsymbol{v}\| = const$, the maximum value for σ_1^2 is obtained when $\boldsymbol{v}$ has a null component along the cylinder axis $\boldsymbol{a}$ ($\boldsymbol{a}^T\boldsymbol{v} = 0$) with, in this case, $\sigma_1^2 = \sigma_{max}^2 = \|\boldsymbol{v}\|^2$. Intuitively, any camera motion along the cylinder axis does not provide any useful information to the estimation task. Furthermore, as in all previous cases with $p = 1$, keeping a $\sigma_1^2(t) = const$ allows to exactly enforce the ideal estimation error dynamics (4.15), see Remark 4.1.

Finally, from (4.86) one has

$$\dot{(\sigma_1^2)} = \boldsymbol{J}_{\boldsymbol{v},1}\dot{\boldsymbol{v}} + \boldsymbol{J}_{\boldsymbol{a},1}\dot{\boldsymbol{a}} = \boldsymbol{J}_{\boldsymbol{v},1}\dot{\boldsymbol{v}} + \boldsymbol{J}_{\boldsymbol{a},1}[\boldsymbol{a}]_\times\boldsymbol{\Omega} \tag{4.87}$$

with $\boldsymbol{J}_{\boldsymbol{v},1} = 2\boldsymbol{v}^T\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right)$ and $\boldsymbol{J}_{\boldsymbol{a},1} = 2\boldsymbol{v}^T\boldsymbol{a}\boldsymbol{v}^T$. Although (4.87) also depends on the angular velocity $\boldsymbol{\Omega}$, it is possible to fully compensate for the effects of $\boldsymbol{J}_{\boldsymbol{a},1}[\boldsymbol{a}]_\times\boldsymbol{\Omega}$ (a known quantity) when inverting (4.87) w.r.t. $\dot{\boldsymbol{v}}$ as discussed in Sect. 5.4. Therefore, one can act on $\dot{\boldsymbol{v}}$ to regulate the value of $\sigma_1^2(t)$ and, at the same time and in a decoupled way, exploit the camera angular velocity $\boldsymbol{\Omega}$ for implementing additional tasks of interest such as keeping the cylinder axis $\boldsymbol{a}$ at the center of the image plane by enforcing $(s_x,\, s_y) = \boldsymbol{\emptyset}_2$.

4.6 Conclusions

In this chapter we addressed the problem of active SfM for recovering the 3-D structure of some basic but common geometric primitives: a point feature, a planar object, and a spherical and a cylindrical targets. We proposed a novel active estimation strategy tailored to the four cases under consideration. Using a nonlinear observer we showed how one can impose to the estimation error a transient evolution that is (almost) equivalent to that of a second order linear system. More importantly one can act online on the camera linear velocity to maximize the excitation of the system and thus reduce the estimation error convergence time given some constraints on the maximum allowed camera velocity.

For the depth estimation of a point feature, two possibilities differing in the adopted projection model (planar or spherical) were proposed and critically compared highlighting the complementarity of the two models in terms of attainable convergence rates and basin of attraction for the estimation error.

For the planar case, we first applied a simple standard strategy to extract a best fitting plane, in a least-squares sense, from a (actively) estimated point cloud. Then, we showed how to conveniently exploit image moments (in both their discrete and dense definitions) to estimate *directly* the parameters of the plane. For this second solution, we also explained how the standard definition of image moments can be extended to(i) cope with the possible loss/addition of point features due to a limited camera FOV and, (ii) automate the selection of which image moments order to use for the estimation to further improve the system observability.

Finally, in the spherical and cylindrical cases, we showed how an adequate choice of the measured visual features allows to reduce the SfM task to the estimation of a single unknown constant quantity (the sphere/cylinder radius R) in place of the classical (and time-varying) three parameters (scaled normal vector of the planar limb surface). Availability of this quantity allows to then retrieve the full 3-D structure of the observed targets.

In Chapt. 5 some simulative and experimental results will be reported to fully

confirm the validity of the theoretical analysis presented here and, in particular, the ability of the proposed *active* estimation strategy to improve, in all cases, the transient response of the estimation error.

Chapter 5

Experiments and simulations of active structure from motion

In this chapter we show some experimental results meant to validate the theoretical developments of Chapt. 4. For all of the case studies presented there (point features, planar objects and spherical and cylindrical targets) we show here the advantages of adopting an active strategy for selecting on line the best camera linear velocity direction (the velocity norm is kept constant) for the sake of maximizing the convergence rate of the SfM estimator.

The experiments were run by employing a Point Grey Dragonfly2 greyscale camera (Fig. 5.1(a)). This has a resolution of 640×480 px and a framerate of 30 fps. The open-source ViSP library [MSC05] was used to accurately calibrate the intrinsic parameters of the camera before running the experiments. Also the image processing and feature tracking were implemented using ViSP as it will be detailed

(a) (b)

Figure 5.1 – **Experimental setup.** The Point Grey Dragonfly2 greyscale camera Fig. (a) and the 6-DOFs Gantry robot Fig. (b) used for all experiments.

in the next sections.

The camera was mounted on the end-effector of a 6-DOFs Gantry robot, a picture of which is shown in Fig. 5.1(b). This robot has 3 linear orthogonal axes and a spherical wrist. This configuration is particularly convenient for our goals as it minimizes the number of kinematic singularities. The employment of the presented techniques on other robotic platforms such as, e.g., anthropomorphic serial manipulators, is certainly possible but might require to explicitly deal with the presence of robot kinematic singularities, a topic that is well assessed in the literature, but out of the scope of this thesis. The robot geometric Jacobian can be expressed as:

$$\boldsymbol{J}_{\mathcal{E}} = \begin{bmatrix} s_4 s_5 c_{56} + c_4 s_{56} & -c_4 s_5 c_{56} + s_4 s_{56} & c_5 c_{56} & l_{56} s_5 c_{56} & 0 & 0 \\ -s_4 s_5 s_{56} + c_4 c_{56} & c_4 s_5 s_{56} + s_4 c_{56} & -c_5 s_{56} & -l_{56} s_5 s_{56} & 0 & 0 \\ -s_4 c_5 & c_4 c_5 & s_5 & -l_{56} c_5 & 0 & 0 \\ 0 & 0 & 0 & c_5 c_{56} & s_{56} & 0 \\ 0 & 0 & 0 & -c_5 s_{56} & c_{56} & 0 \\ 0 & 0 & 0 & s_5 & -\gamma_{56} & 1 \end{bmatrix}$$

with $c_i = \cos(q_i)$, $s_i = \sin(q_i)$, $c_{56} = \cos(q_6 - \gamma_{56} q_5)$, $s_{56} = \sin(q_6 - \gamma_{56} q_5)$, $l_{56} = 0.06924\,\mathrm{m}$, and $\gamma_{56} = 0.009091$.

Since the robot only accepts velocity commands, the acceleration signal generated by the proposed control strategies was numerically integrated before being sent to the robot as a joint velocity command. The SfM observers and the controllers (and their internal states) were updated at 1 kHz, while the commands were sent to the robot at 100 Hz. The estimation and control algorithms were implemented in Matlab/Simulink where as the communication with the robot is ensured by ViSP in a compiled C++ executable. The communication between the two software components is ensured by the Robot Operating System (ROS) with the framework described in [10].

Videos representing some of the experiments and simulations shown here can be downloaded from the pages associated with the publications [1, 2, 3, 4, 5, 6] at http://ieeexplore.ieee.org. The videos are also available at the following links:

- for the point feature and for spherical and cylindrical targets: https://www.youtube.com/watch?v=i-9xxNNV82Q;
- for the planar case using point features and discrete image moments: https://www.youtube.com/watch?v=QNXrkZj4NU0;
- for the planar case using adaptive discrete image moments: https://www.youtube.com/watch?v=5PtrbXuhtd0.

5.1 Active structure estimation for a point

5.1.1 Comparison of planar and spherical projection models

We start by comparing via simulation results the effects of adopting a planar and spherical projection model for the depth estimation of a point feature as extensively discussed in Sect. 4.5.1.1 and Sect. 4.5.1.2. We considered three cases differing for the location on the image plane at which the point feature was (purposely) kept exploiting the camera angular velocity $\boldsymbol{\omega}$:

case I: the point feature was kept at the center of the image plane (red line in the following plots);

case II: the point feature was kept at one of the corners of an image plane with the same size of the camera used in the experiments (green line in the following plots);

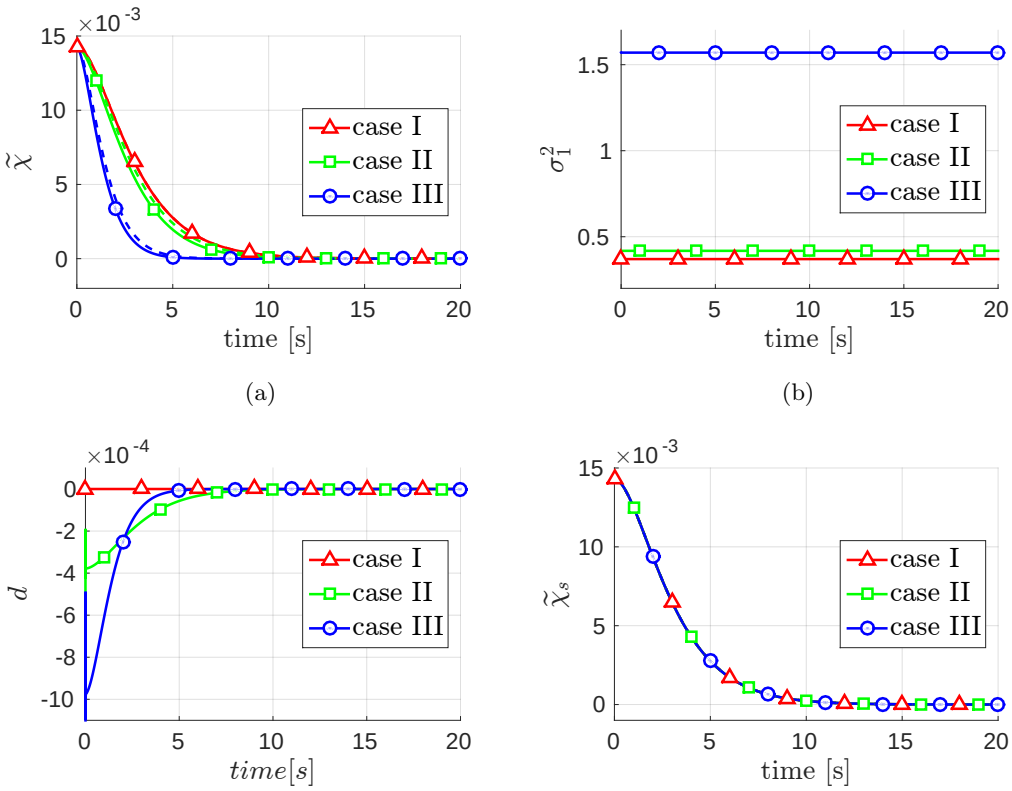
case III: the point feature was kept at one of the corners of an image plane with a size five times larger than case II (blue line in the following plots).

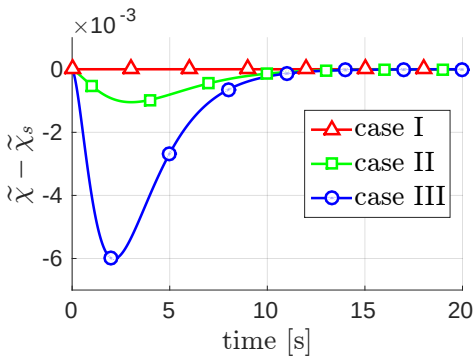
In all cases, a constant camera velocity $\boldsymbol{v}(t) \equiv \boldsymbol{v}(t_0) = const$ was kept during motion, with the initial condition $\boldsymbol{v}(t_0)$ chosen so as to comply with the optimality conditions discussed in Sects. 4.5.1.1 and 4.5.1.2 for letting $\sigma_1^2 = \sigma_{max}^2$ (e.g., with $\boldsymbol{v}(t_0)$ being a solution of (4.33) in the planar projection case).

Figure 5.2(a) shows the behavior of $\widetilde{\chi}(t)$ for the three cases when using a *planar* projection model. We can then note how the convergence rate of the estimation error increases from case I (slowest convergence) to case III (fastest convergence) as predicted by the theory (for the same $\|\boldsymbol{v}\|$ a larger $\|\boldsymbol{\pi}\|$ results in a larger σ_{max}^2). Similarly, Fig. 5.2(b) reports the behavior of $\sigma_1^2(t)$ for the three cases: as expected, $\sigma_1^2(t)$ results largest for case III. Note also how $\sigma_1^2(t)$ for case II (green line) is only *slightly* larger than case I (red line). This is due to relatively small size of the image plane of case II whose dimensions were set as those of the real camera used for the experiments. Finally, Fig. 5.2(c) shows the behavior of the perturbation term $d(\widetilde{\mathbf{x}}, t)$ in the three cases: here, one can verify how $d = 0$ for case I, with then an increasing $|d|$ for cases II and III. Indeed, as discussed in Sect. 4.5.1.1, the 'amplification' effect on σ_{max}^2 obtained by increasing $\|\boldsymbol{\pi}\|$ comes at the price of an increased magnitude of the perturbation d. This is also evident in Fig. 5.2(a) where the ideal response of (4.15) is plotted with dashed lines for the three considered cases. We can thus note how $\widetilde{\chi}(t)$ in case I presents a perfect match with its corresponding ideal response, with then an increasing (albeit very limited) mismatch in the other two cases due to the increased effect of the perturbation d.

As for the spherical projection model, Fig. 5.2(d) reports the behavior of the estimation error $\widetilde{\chi}(t)$ for the three cases under consideration, together with the ideal response (4.15). Here, the symbol $\widetilde{\chi}_s(t)$ is used to denote the estimation error in the spherical projection case in order to distinguish it from the error obtained with the planar projection model. All the plots result perfectly superimposed as expected from the analysis of Sect. 4.5.1.2. Indeed, in the spherical projection case, $\sigma^2_{max} = \|\boldsymbol{v}\|^2$ regardless of the location of $\boldsymbol{\eta}$ and $d(t) \equiv 0$. However, absence of perturbation terms is obtained at the expense of the convergence rate of $\widetilde{\chi}_s(t)$, which indeed results slower or equal to that of $\widetilde{\chi}(t)$ in the planar projection case. This is shown in Fig. 5.2(e) where the behavior of $\widetilde{\chi}(t) - \widetilde{\chi}_s(t)$ is reported for the three cases. We can then note how $\widetilde{\chi}(t) - \widetilde{\chi}_s(t) = 0$ only in case I, as the planar and spherical models coincide when the feature point is at the center of the image plane.

These results then fully confirm the validity of the theoretical analysis reported in Sects. 4.5.1.1 and 4.5.1.2. However, we also note the marginal effects of the two projection models on the estimation performance when applied to an image plane of size comparable to that of the real camera used in our experimental setup. Therefore, in the following experimental results we will only consider the case of planar projection model.





(e)

Figure 5.2 – **Simulation results comparing the planar and spherical projection models for the depth estimation of a point feature**. The following color coding is adopted for the three considered cases: red–case I, green–case II, blue–case III. Fig. (a) behavior of the estimation error $\widetilde{\chi}(t)$ in the planar projection case (solid lines) with superimposed the corresponding ideal response (4.15) (dashed lines). The convergence results slowest in case I and fastest in case III with, however, a corresponding increasing mismatch among $\widetilde{\chi}(t)$ and its ideal response (as expected). Fig. (b) behavior of $\sigma_1^2(t)$ for the three cases with, again, the largest $\sigma_1^2(t)$ in case III. Fig. (c) behavior of the perturbation term $d(\widetilde{\mathbf{x}}, t)$ for the three cases. As expected, $d(t) \equiv 0$ in case I, and it is largest in case III (thus, explaining the increasing mismatch among $\widetilde{\chi}(t)$ and the corresponding ideal response). Fig. (d) behavior of the estimation error $\widetilde{\chi}_s(t)$ for the spherical projection model in the three cases. The three plots result exactly superimposed as predicted by the theory (no influence of the location of $\boldsymbol{\eta}$ on the estimation convergence). Fig. (e) behavior of $\widetilde{\chi}(t) - \widetilde{\chi}_s(t)$. As expected, the two projection models give rise to the same estimation error dynamics only in in case I (feature point at the center of the image plane).

5.1.2 Depth estimation for a point feature

We here report some experimental results for the depth estimation of a point feature under a planar projection model (Sect. 4.5.1.1). The following experiments are meant to demonstrate how the proposed active estimation framework can be exploited to select online the 'best' camera motion. As visual target, we made use of a circular white dot of 5 mm radius painted on a planar black surface and sufficiently far from the camera in order to safely consider it as a 'point feature' (see Fig. 5.3(a)).

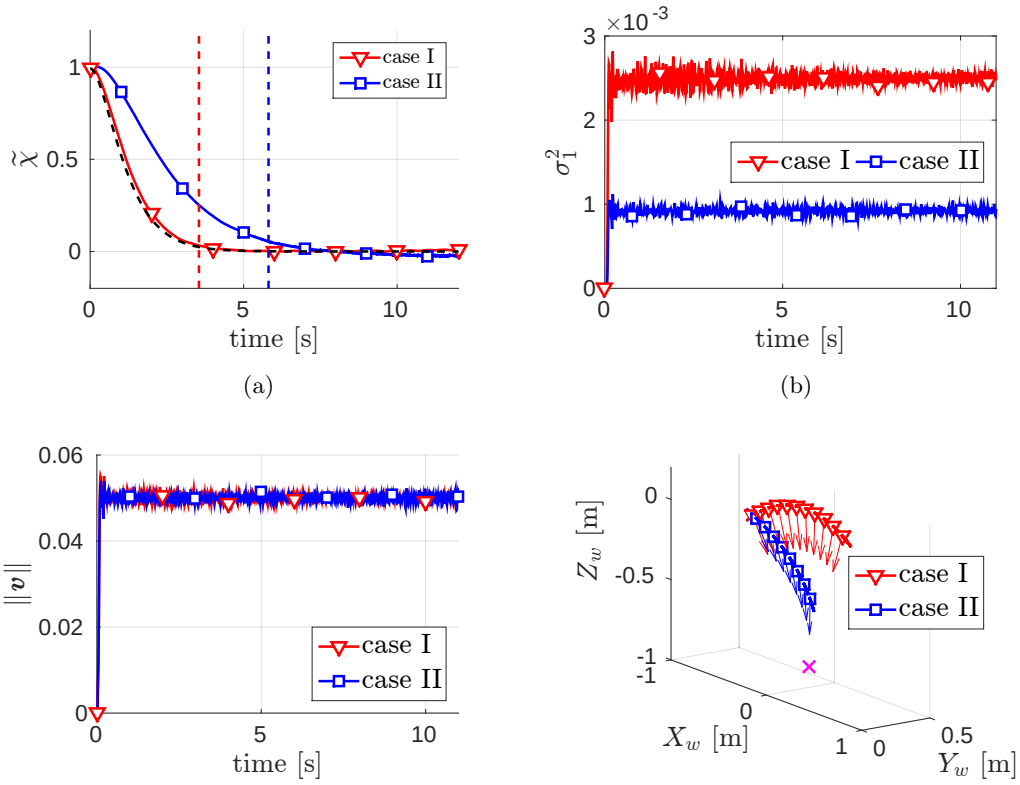
Figure 5.4(a) shows the evolution of the estimation error $\widetilde{\boldsymbol{\chi}}(t) = 1/\widehat{Z}(t) - 1/Z(t)$ for two experiments[1] in which $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$ but with its direction being either

case I: optimized to maximize the estimation convergence rate (red line) or

[1]The ground truth $Z_0(t)$ was obtained from a previous offline estimation of the 3-D position $\boldsymbol{p}$ in the world frame, and by then using the information on the camera position provided by the robot forward kinematics.

(a) (b) (c)

Figure 5.3 – **Camera snapshots** for the point feature Fig. (a), the sphere Fig. (b) and the cylinder Fig. (c) experiment. The result of the image processing is highlighted in red.



(a) (b) (c) (d)

Figure 5.4 – **Experimental results for the depth estimation of a point feature**. Fig. (a): behavior of the estimation error for case I (solid red line) and case II (solid blue line), and for an 'ideal' second order system (4.15) with desired poles at σ^2_{max} (dashed black line). The two vertical dashed lines indicate the times $T_1 = 3.54\,\mathrm{s}$ and $T_2 = 5.81\,\mathrm{s}$ at which the estimation error drops below the threshold of 3 cm. Fig. (b): behavior of $\sigma_1^2(t)$ for case I (red line) and case II (blue line). Fig. (c): camera linear velocity norm $\|\boldsymbol{v}\|$ for case I (red line) and case II (blue line). Fig. (d): Camera trajectories for case I (red line) and case II (blue line) with arrows indicating the direction of the camera optical axis. Note how the use of an active strategy for optimizing the direction of the camera linear velocity results in improved performance of the estimation for the same control effort (i.e. same velocity norm).

case II: kept constant so that $\boldsymbol{v}(t) = \boldsymbol{v}_0 = const$ (blue line).

This effect was obtained by using the following control law (equivalent to (4.23))

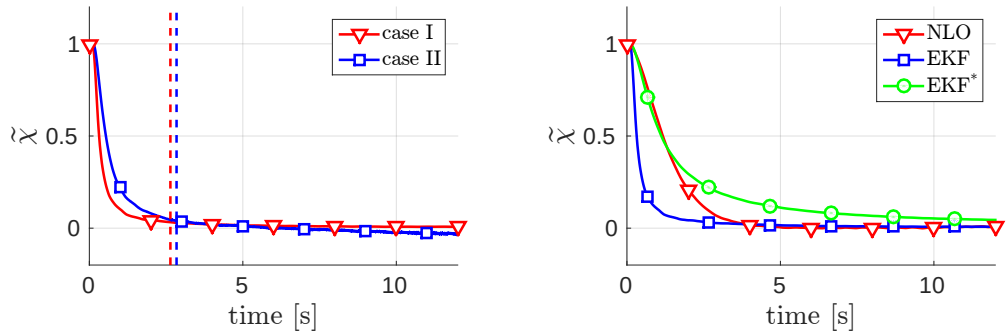
$$\dot{\boldsymbol{v}} = -k_1 \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2} (\kappa - \kappa_{\mathrm{d}}) + k_2 \left(\boldsymbol{I}_3 - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2} \right) \boldsymbol{J}_{\boldsymbol{v},1}^T \tag{5.1}$$

with $k_1 > 0$, $k_2 \geq 0$, $\kappa = \frac{1}{2}\boldsymbol{v}^T\boldsymbol{v}$, $\kappa_{\mathrm{d}} = \frac{1}{2}\boldsymbol{v}_0^T\boldsymbol{v}_0$, and $\boldsymbol{J}_{\boldsymbol{v},1}$ given by (4.31). In fact, the first term in (5.1) enforces the constraint $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$ (same control effort in both cases), while the second term (maximization of σ_1^2 within the null-space of the first constraint) allows to implement either case I ($k_2 > 0$) or case II ($k_2 = 0$). In both cases, the angular velocity $\boldsymbol{\omega}$ was exploited for keeping the point feature at the center of the image plane $(x, y) \to (0, 0)$. We note that, as discussed in Sect. 4.5.1.1, when $(x, y) = (0, 0)$ one has $\sigma_{max}^2 = v_x^2 + v_y^2$ from (4.32) and $\sigma_1^2 = \sigma_{max}^2$ iff $v_z = 0$ (circular motion around the point feature). The experiments were run with the following parameters: $\alpha = 10^3$, $c_1 = c_1^*$ for $\boldsymbol{D}_1$ in (4.8), $\boldsymbol{v}(t_0) = \boldsymbol{v}_0 = (0.03, 0, -0.04)$ m/s, $k_1 = 5$ and $k_2 = 10^4$, thus resulting in the maximum value $\sigma_{max}^2 = 0.0025$ for the eigenvalue σ_1^2.

As clear from Fig. 5.4(d), while in case II the camera gets closer to the point feature, the use of the active strategy of case I results in a null component of $\boldsymbol{v}$ along the projection ray of the point feature (i.e., $v_z = 0$) and in an associated circular trajectory centered on the tracked point as predicted by the theoretical analysis of Sect. 5.1.1. This then allows to move faster in the 'useful' directions (while keeping the same constant $\|\boldsymbol{v}\|$), and, thus, to increase the value of σ_1^2 towards its theoretical maximum $\sigma_{max}^2 = 0.0025$ (Fig. 5.4(b)), resulting in an overall faster convergence for the estimation error (Fig. 5.4(a)). Furthermore, Fig. 5.4(a) also reports the ideal response of (4.15) with desired poles at σ_{max}^2 (dashed black line). We can then note the almost perfect match with case I (solid red line): indeed, as explained in Remark 4.1, imposing a $\sigma_1^2(t) = const$ allows to exactly obtain the ideal behavior governed by (4.15). It is finally worth noting the accuracy of the reconstructed depth: Fig. 5.4(a) reports two vertical dashed lines indicating, for the two cases under consideration, the times $T_1 = 3.54$ s and $T_2 = 5.81$ s at which the estimation error $\widetilde{\chi}(t)$ becomes smaller than 3 cm. We then obtained a standard deviation of approx. 7.5 and 8.4 mm evaluated on a time window of 1 s after the times T_1 and T_2, respectively. These results then also confirm the robustness of the proposed estimation approach despite the unavoidable presence of noise and discretization in the image acquisition. Note also that, as expected, the estimation error in the (active) case I reaches 'convergence' (i.e., drops below the threshold of 3 cm) significantly faster than case II ($T_1 < T_2$).

5.1.3 Comparison between the nonlinear observer and the EKF

In this section we propose a basic comparison between the nonlinear observer scheme introduced in Sect. 3.2.3, and used in the rest of this thesis, and the probabilistic EKF described in Sect. 3.3.3. Figure 5.5(a) shows the behavior of the estimation error when employing an EKF for estimating the (inverse) depth of the point feature using the same experimental data and camera trajectory as in the previous cases I and II. We notice that, also when using the EKF, the estimation performance still benefits from the use of an active strategy for choosing the best camera trajectory in order to improve observability. In fact, the times at which the estimation error $\widetilde{\chi}(t)$ becomes smaller than 3 cm, are $T_1 = 2.64$ s in case I and $T_2 = 2.84$ s in case II (vertical dashed lines in Fig. 5.5(a)). Moreover, we obtained a standard deviation of approx. 3.7 and 6.4 mm evaluated on a time window of 1 s after the times T_1 and T_2, respectively. These results are not surprising since, from the developments in Chapt. 3, observability is a property of the system itself and not of the estimation algorithm and therefore one should expect *any* estimation scheme (EKF included) to benefit from an optimization of the camera trajectory. This is further confirmed by the fact that the control law (5.1) does not depend on any estimated quantity, but only on the measurements. We also notice that, in both cases I and II, the EKF filter outperforms the nonlinear observer for the final covariance. This fact, however, is still expected from the theoretical analysis. Indeed, the use of the measured $\boldsymbol{s}$ (the (x, y) coordinates of the point feature) in place of the estimated $\widehat{\boldsymbol{s}}$ in the prediction step of the EKF transforms the nonlinear system dynamics in a *linear time-varying* one with an additional disturbing term due to the non perfect cancellation of the dynamics of χ, see also [GBSO13]. Since the EKF is optimal for linear time-varying systems, it (correctly) outperforms the nonlinear observer, especially in case I where, thanks to the active camera velocity optimization, the depth of the point feature remains constant and the disturbing term disappears. On the other hand, the EKF has, in general, less predictable stability properties and convergence rate than the nonlinear filter. For instance, one cannot expect, for the EKF, to obtain a good match with the dynamics of a second order system as the one shown in Fig. 5.4(a) for the nonlinear observer. Moreover, the convergence time of the EKF also depends on the initialization of the state covariance matrix $\boldsymbol{\Sigma}_0$. Figure 5.5(b) shows, in fact, with a green line, the results obtained when the state covariance is initialized with a value 10 times smaller than in the previous case I. As one can see, with this initialization, the EKF convergences slower than the nonlinear observer. Other works, such as [GBSR15], have also shown that nonlinear estimation techniques can, in some cases, perform better than the EKF. Because of all these considerations, we decided to use the nonlinear observer (3.11) for the rest of our experiments.



(a) (b)

Figure 5.5 – **Experimental results for the depth estimation of a point feature using an EKF**. Fig. (a): behavior of the estimation error for case I (solid red line) and case II (solid blue line). The two vertical dashed lines indicate the times $T_1 = 2.64\,\mathrm{s}$ and $T_2 = 2.84\,\mathrm{s}$ at which the estimation error drops below the threshold of 3 cm. Note how the use of an active strategy for optimizing the direction of the camera linear velocity results, again, in improved performance of the estimator. Fig. (b): behavior of the estimation error for the nonlinear observer (solid red line) and the EKF (solid blue line) in case I. The green line (labeled EKF* in the legend) finally represents the estimation error for the EKF in case I when the estimator is initialized with a covariance of $\widehat{\chi}(t_0)$ 10 times larger than in the blue line case.

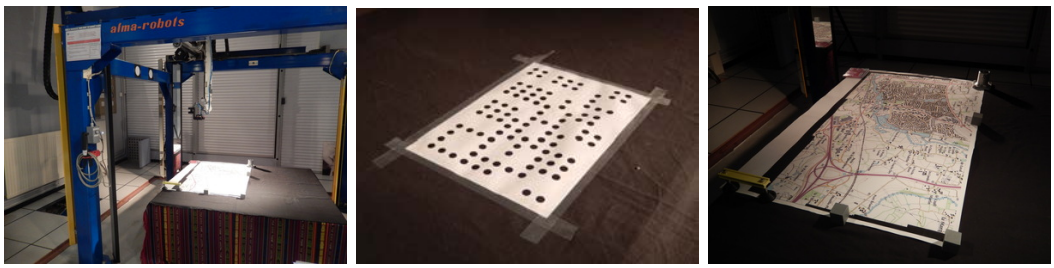


Figure 5.6 – **Experimental set-up for plane estimation** with the dotted pattern and the topographic map used for feature extraction and tracking.

5.2 Active structure estimation for a plane

This section reports some experimental results meant to illustrate and compare the various plane estimation methods introduced in Sect. 4.5.2. In the first set of experiments (Sects. 5.2.1 and 5.2.2) a simple dotted pattern was used for feature extraction and matching. This solution was meant to reduce as much as possible the variability between each experimental run by ensuring tracking of the very same set of points across all trials. In the last experiment (Sect. 5.2.3) a more realistic scene was considered with the Pyramidal Kanade Lucas Tomasi feature tracker (KLT) implemented in OpenCV used for tracking points on the surface of a (planar) topographic map (see Fig. 5.6).

The convergence rate of methods B and C was optimized by *actively* maximizing

the minimum eigenvalue σ_m^2 in (4.43) for method B, and the smallest eigenvalue σ_1^2 of the 3×3 matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ from (4.48) for method C. Exploiting (4.22), and recalling that the Jacobian $\boldsymbol{J_v}$ can be computed in closed form in both cases, the update rule (5.1) was implemented also in this case with $k_1 > 0$ and $k_2 \geq 0$. This allows, again, to optimize the direction of the camera velocity so as to maximize the SfM estimation convergence while keeping $\|\boldsymbol{v}(t)\| = const$.

As for the angular velocity $\boldsymbol{\omega}$, it was exploited, in the experiments in Sects. 5.2.1 and 5.2.2, to keep the centroid of the tracked point features at the center of the image and, in the experiments in Sect. 5.2.3, to align the camera optical axis with the (estimated) plane normal $\widehat{\boldsymbol{n}}$. Indeed, we remark again that matrix $\boldsymbol{\Omega}(\boldsymbol{s}, \boldsymbol{v})$ in (3.10) *does not* depend on $\boldsymbol{\omega}$ and thus one can freely choose the camera angular velocity without affecting the estimation convergence.

5.2.1 Plane estimation from 3-D points (method B)

We report here the results in estimating the plane parameters $(\boldsymbol{n}, d)$ with method B: plane fitting from an estimated point cloud. The experiment started from an initial guess $(\widehat{\boldsymbol{n}}(t_0), \widehat{d}(t_0))$ with an error of 40 deg w.r.t the true $\boldsymbol{n}(t_0)$ and a relative error of 50% w.r.t. the true $d(t_0)$. The initial depths of all the tracked points $\boldsymbol{\pi}_k$ were initialized so as to force $\widehat{\boldsymbol{p}}_k(t_0)$ to belong to the estimated plane described by $(\widehat{\boldsymbol{n}}(t_0), \widehat{d}(t_0))$. In order to demonstrate the importance of the active camera velocity optimization, we first ran a set of four experiments starting from the same initial conditions but using different initial camera velocities with the same norm $\|\boldsymbol{v}(t_0)\| = 0.0224 m/s$. In these experiments we used $k_1 = 10$ in (5.1) but we either substituted $k_2 \boldsymbol{J_v}^T$ with a random acceleration vector (purple dashed line) or we set $k_2 = 0$, thus keeping a $\boldsymbol{v}(t) = \boldsymbol{v}(t_0) = const$ during motion (green, red and cyan dashed lines). Finally we started the experiment again from the same initial camera velocity as in the experiment that performed worst in the previous set (cyan line) and we adopted the update rule (5.1) with $k_1 = 10$ and $k_2 = 1$.

Finally, for the sake of allowing a *fair* comparison between the convergence rates of methods B and C, we first collected all the data during a first execution of all trajectories, and then ran the two estimation schemes offline on the collected dataset by properly adjusting the estimation gains α_B and α_C of both methods[2]. Indeed, let $\overline{\sigma}_m^2 = \frac{1}{T}\int_{t_0}^{t_0+T} \sigma_m^2(\tau)\mathrm{d}\tau$ and $\overline{\sigma}_1^2 = \frac{1}{T}\int_{t_0}^{t_0+T} \sigma_1^2(\tau)\mathrm{d}\tau$ be the average values of the eigenvalues $\sigma_m^2(t)$ and $\sigma_1^2(t)$ during motion in the active estimation cases, with T representing the experiment duration (blue lines in Figs. 5.7 and 5.8). After having computed $\overline{\sigma}_m^2$ and $\overline{\sigma}_1^2$ during the first run, the estimation gains α_B and α_C were chosen so as to satisfy $\alpha_B \overline{\sigma}_m^2 = \alpha_C \overline{\sigma}_1^2$ for imposing the same *closed-loop*

[2]This is possible because the control law (5.1) *does not* depend on any estimated quantity.

dynamics to both methods B and C[3]. This resulted in gain $\alpha_B = 1043.4$ (used in these experiments), and in gain $\alpha_C = 20000$ (used in the experiments of the next Sect. 5.2.2).

Fig. 5.7(a) shows the behavior of the norm of the estimation error $\widetilde{\boldsymbol{\chi}}$ between the real and estimated inverse feature depths, normalized w.r.t. its initial value. The normalization is meant to allow a comparison of this plot with the analogous one in Fig. 5.8(b). The angle between vectors $\boldsymbol{n}(t)$ and $\widehat{\boldsymbol{n}}(t)$ and the relative error $(\widehat{d}(t) - d(t))/d(t)$ are also shown in Fig. 5.7(b). We can then note how the plane estimation task is solved in all cases (the estimation errors converge towards zero) but, clearly, in the active case (blue line) the error convergence is significantly faster than in the other experiments. This is further evident from Fig. 5.7(c) where the value of the $\alpha_B \sigma_m(t)$ is shown for all experiments (same color code): thanks to the active optimization of the direction of $\boldsymbol{v}(t)$, during the active experiment, $\alpha_B \sigma_m(t)$ results approximately 12.5 times larger than in the worst experiment (cyan) which started from the same initial camera velocity.

Finally, Figs. 5.7(e) and 5.7(f) depict the camera trajectory in all cases with arrows indicating the direction of the camera optical axis. The green patch represents the location of the plane to be estimated. We encourage the reader to also look at the attached video for better appreciating the effects of the active strategy on the camera trajectory.

5.2.2 Plane estimation from discrete image moments (method C)

In this second set of experiments we show the results of using the weighted discrete image moments for the estimation of the plane parameters. As before the initial guess for $\widehat{\boldsymbol{\chi}}(t_0)$ has an error of approximately 40° w.r.t. $\boldsymbol{n}(t_0)$ and a relative error of around 50% w.r.t. $d(t_0)$. Again, we first ran a set of four experiments starting from the same initial conditions but using different initial camera velocities with the same norm $\|\boldsymbol{v}(t_0)\| = 0.0206 m/s$ and using (5.1) with $k_1 = 10$ and either substituting $k_2 \boldsymbol{J}_v^T$ with a random acceleration vector (purple dashed line) or setting $k_2 = 0$, thus keeping a $\boldsymbol{v}(t) = \boldsymbol{v}(t_0) = const$ during motion (green, red and cyan dashed lines). Finally, in the active case we started again from the initial camera velocity of the experiment that performed worst in the previous set (cyan line), and we adopted the update rule (5.1) with $k_1 = 10$ and $k_2 = 1$. In all cases, as explained in Sect. 5.2.1, we set $\alpha_C = 20000$.

We show again in Fig. 5.8(a) the behavior of the normalized norm of the estimation error $\widetilde{\boldsymbol{\chi}}$. The angle between the actual and estimated normal direction $\boldsymbol{n}(t)$

[3] As explained in [1], the convergence rate of the SfM scheme (3.11) is actually dictated by the smallest eigenvalue of $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ *times* the chosen estimation gain α.

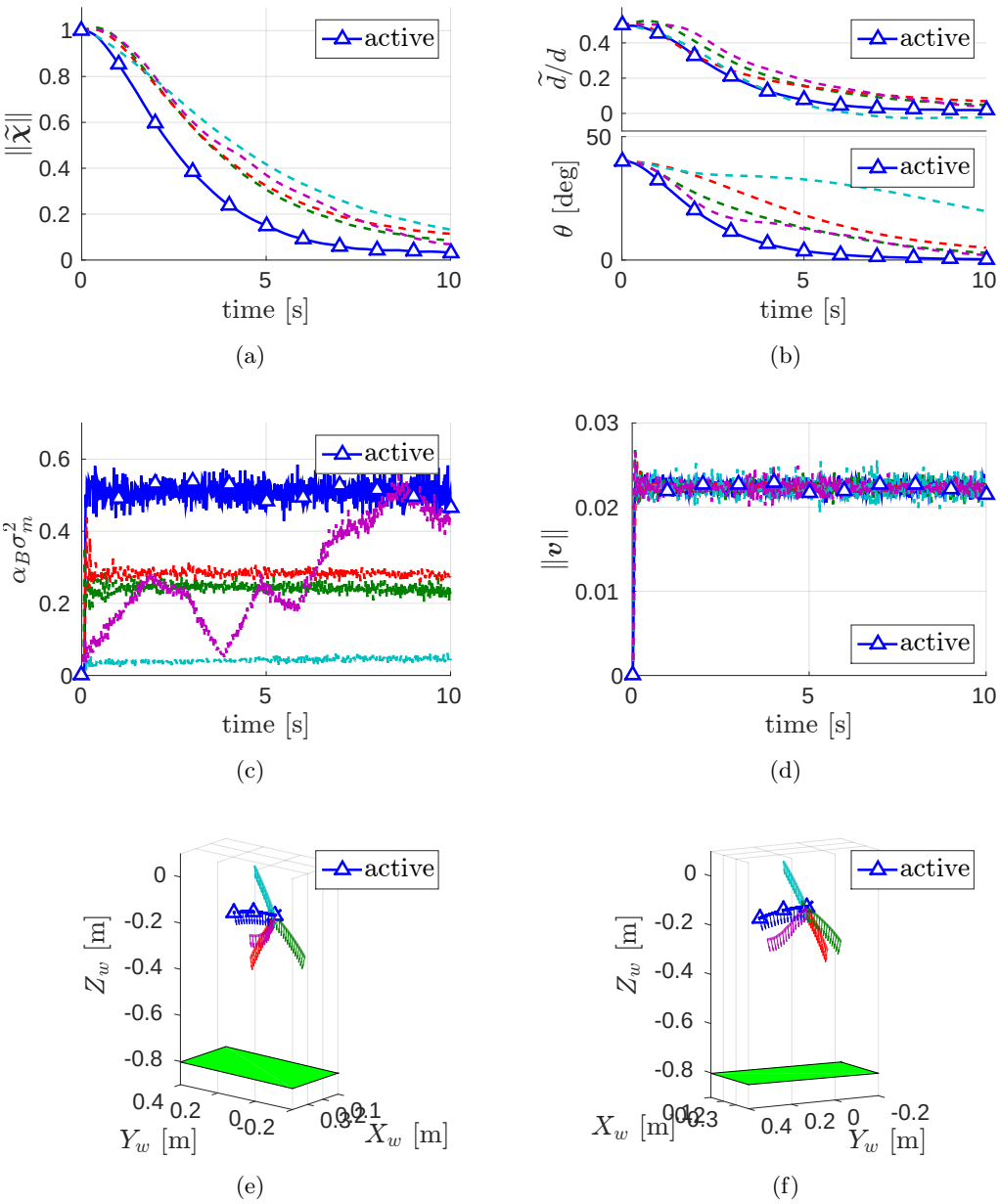


Figure 5.7 – **Experimental results for the estimation of the plane parameters using 3-D points** (method B) with an active strategy (blue lines) or a random acceleration (purple line) or a constant linear velocity (green, red and cyan lines). Fig. (a): normalized norm of the estimation error $\widetilde{\boldsymbol{\chi}}$; Fig. (b): relative error between estimated and actual distance d and angle between the estimated and actual normal $\boldsymbol{n}$. Fig. (c): smallest eigenvalue σ_m^2 of the $N \times N$ matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ multiplied by α_B. Fig. (d): camera linear velocity norm. Fig. (e) and Fig. (f): geometric 3-D trajectory of the camera with arrows indicating the optical axis and a green patch representing the plane to be estimated.

and the relative error on $d(t)$ are shown in Fig. 5.8(b). As evident from the plots, the active strategy results again in a faster convergence of the estimation error, as also clear from Fig. 5.8(c) where the behavior of $\alpha_C \sigma_1^2(t)$ is shown for all cases. The trajectory of the camera in the various experiments is finally shown in Figs. 5.8(e) and 5.8(f).

5.2.3 Comparison of the three methods A, B and C

This set of experiments is meant to provide a comparative analysis of the differences between methods B and C against the classical method A taken as a *baseline condition*. The 'convergence' of method A for the estimation of the plane normal direction is, in general, faster w.r.t. the other two methods due to its algebraic nature (no filtering process is present in this case). On the other hand the use of an estimation scheme in methods B and C allows for the possibility of tuning the estimation gain α (a free parameter) against the noise level present in the system (i.e., trading off convergence speed for noise robustness).

In order to test the three methods in a more challenging scenario, we added to the scene a small planar picture with a non-negligible inclination w.r.t. the main plane (see Fig. 5.9) so as to introduce the presence of some 'outliers' w.r.t. the main dominant plane[4]. The picture was located to be in visibility at the beginning of the experiment and to leave the camera FOV shortly after. The camera linear velocity was optimized via (5.1) by maximizing the smallest eigenvalue σ_1^2 of the matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ for the image moment case, and then the same trajectory, depicted in Figs. 5.10(e) and 5.10(f), was used for the other two methods. This resulted in a non-optimal, but still observable, trajectory for method B.

As done in the previous experimental sections, for the sake of obtaining a fair comparison between the convergence rates of methods B and C, we adjusted the estimation gains of both methods in such a way that $\alpha_B \overline{\sigma}_m^2 = \alpha_C \overline{\sigma}_1^2$, where $\overline{\sigma}_m^2$ and $\overline{\sigma}_1^2$ are the average values of the smallest eigenvalues for the two estimators along the (this time common) trajectory. This resulted in $\alpha_B = 200$ for method B and $\alpha_C = 26179$ for method C.

The behavior of the estimation error on the plane parameters is depicted in Figs. 5.10(a) and 5.10(b) for method A (green lines), method B (blue lines) and method C (red lines). In Fig. 5.10(c) the products $\alpha_B \sigma_m^2(t)$ and $\alpha_C \sigma_1^2(t)$ are plotted.

[4]Of course one could utilize a RANdom SAmple Consensus (RANSAC) based classification (exploiting the homography constraint) for preliminarily segmenting the two planes so as to only consider the points belonging to the main dominant plane for the estimation task. However, in a real situation, the accuracy of any classification method can never be perfect and some outliers will fail to be detected. Therefore, in this experiment we *intentionally* decided to not include any preliminary RANSAC based pruning in order to just assess the "intrinsic" robustness of the proposed algorithms (which would clearly be improved by any preliminary outlier rejection step).

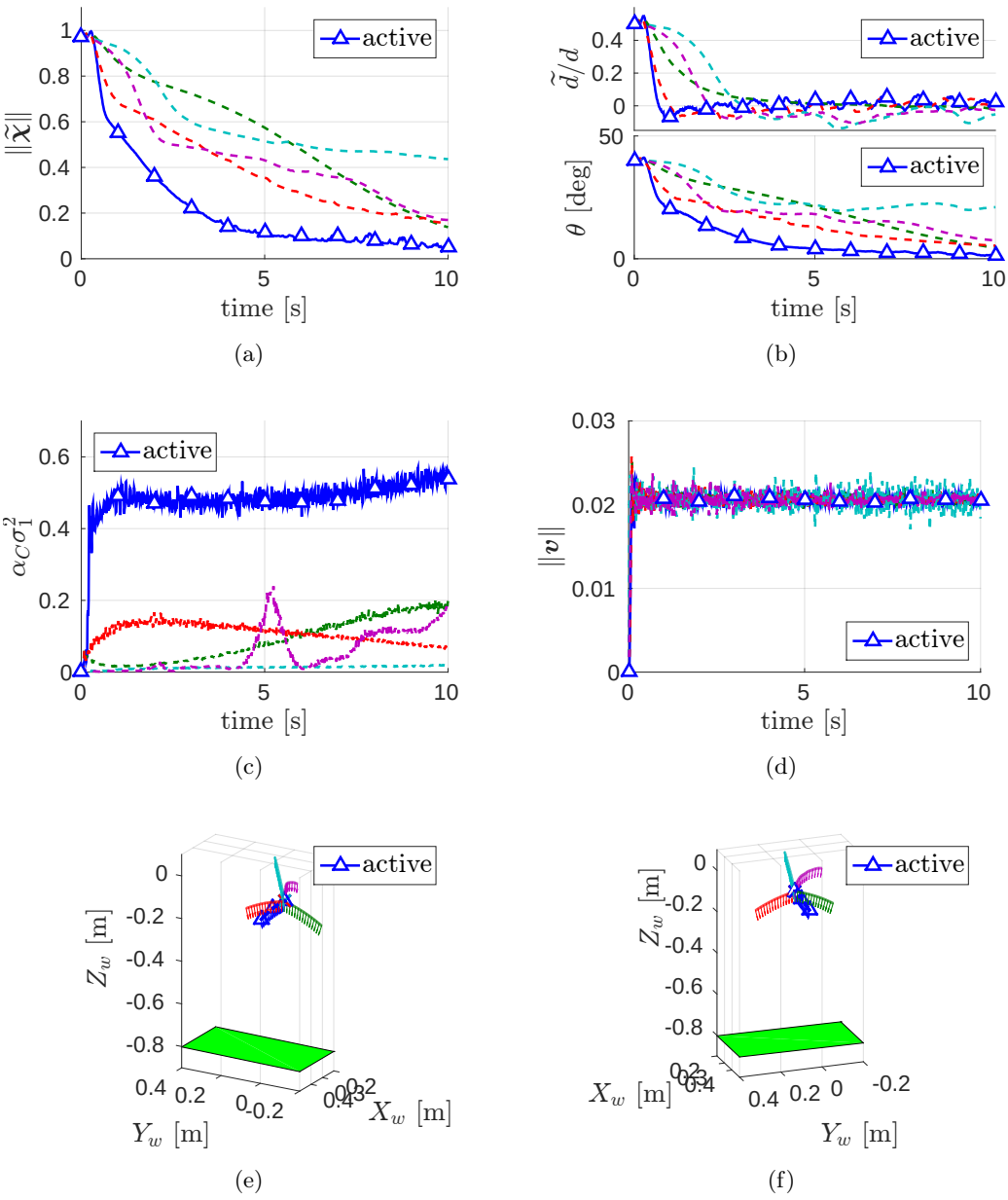


Figure 5.8 – **Experimental results for the estimation of the plane parameters using discrete image moments method C** with an active strategy (blue lines) or a random acceleration (purple line) or a constant linear velocity (green, red and cyan lines). Fig. (a): normalized norm of the estimation error $\tilde{\chi}$. Fig. (b): relative error between estimated and actual distance d and angle between the estimated and actual normal $\boldsymbol{n}$. Fig. (c): smallest eigenvalue σ_1^2 of the 3×3 matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ multiplied by α_C. Fig. (d): camera linear velocity norm. Fig. (e) and Fig. (f): geometric 3-D trajectory of the camera with arrows indicating the optical axis and a green patch representing the plane to be estimated.

Figure 5.9 – **Experimental setup for the estimation of the plane parameters in presence of outlier measurements**. Note the introduction of the inclined picture in the observed scene on the right.

It can be noticed that in all three cases at the beginning of the experiment (i.e. when the 'outlier' effect of the inclined image over the main planar scene is more present) the error in the estimation of the normal is significant although not diverging. All methods estimate a plane with an intermediate normal direction (as one would expect). Subsequently, the estimation errors for method C and method B start converging toward zero at $t \approx 8\,\mathrm{s}$ (first dashed vertical line), that is, when the outlier image starts leaving the image plane. However, note how the homography method still yields a very noisy estimation during this phase. Furthermore, once all the outliers are lost ($t \approx 20.3\,\mathrm{s}$ and second vertical dashed lines in the plots) all the methods yield a converging estimation error. However we can still notice two facts:(i) method C results in the fastest convergence. This is also because the weight w of the outliers starts approaching 0 as they get close to the image border (and thus their disturbing effect is more quickly discarded); (ii) method A has a faster convergence rate w.r.t. method B once all the outliers are lost, but it also yields a noisier estimation until the end of the experiment. In particular one can notice the presence of considerable "jumps" in the estimation of method A due to the reinitialization performed each time the number of matched features falls below a given threshold.

In order to demonstrate the effectiveness of the adopted weighting functions in the computation of the discrete moments, the behavior of $m_{00}(t)$ is shown in Fig. 5.10(d). This is meant to illustrate how the number of active points changes over time due to losses at the image border or detection of new features. The presence of the weighting strategy function guarantees the desired continuity of $m_{00}(t)$ (and similarly of all other image moments not plotted here).

Finally Fig. 5.11 shows the evolution of the individual switching functions $w_1(x)$, $w_2(y)$ and $w_3(\tau)$, and of their product $w = w_1 w_2 w_3$ for three representative point features. At $t \approx 3\,\mathrm{s}$ the red feature starts leaving the image plane (first along the

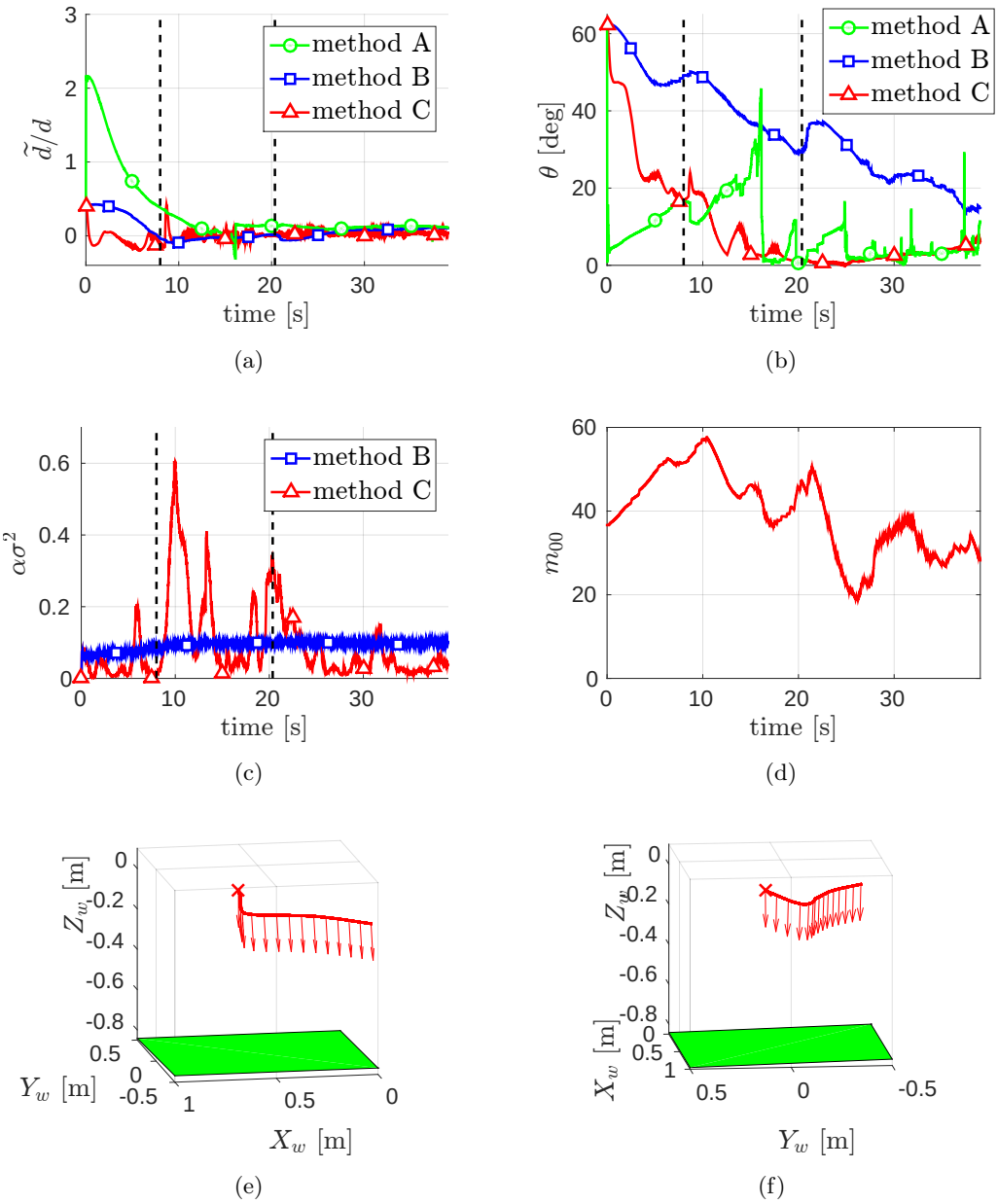


Figure 5.10 – **Experimental results for plane estimation in presence of outlier measurements** using homography decomposition (method A – green lines), 3-D points estimation (method B – blue lines) or image moments (method C – red lines). Fig. (a): relative error between estimated and actual distance d; Fig. (b): angle between the estimated and actual normal $\boldsymbol{n}$; Fig. (c): product $\alpha\sigma$ methods B and C; Fig. (d): evolution of the image moment m_{00}; Fig. (e) and Fig. (f): geometric 3-D trajectory of the camera with arrows indicating the optical axis and a green patch representing the plane to be estimated.

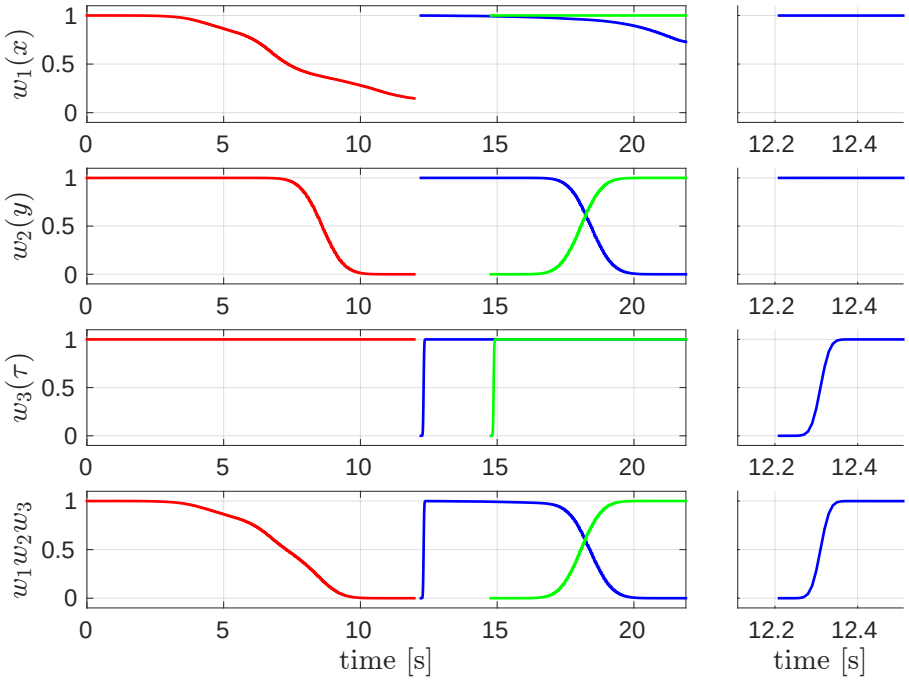
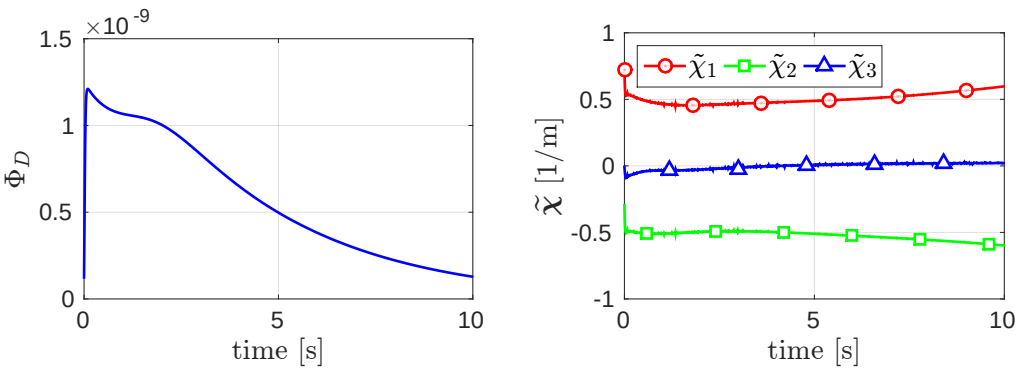


Figure 5.11 – **Evolution of the switching functions and of their product for three representative point features**. On the right: detailed views of the corresponding plots on the left in the time interval immediately following the introduction of the blue feature in the estimator.

x direction and then along the y direction) and its total weight goes to zero by the action of both $w_1(x)$ and $w_2(y)$. After the feature has completely left the plane, the tracker detects a new feature (the blue one) at $t \approx 12\,\mathrm{s}$. Being far from the image border, it is smoothly taken into account thanks to the effect of weight $w_3(t)$ (note the zoomed views on the right side of the plots where the smooth rise of $w_3(t)$ can be seen). Finally, the green feature is close to the border of the image at the time of detection. In this case, even if weight $w_3(t)$ is rising towards 1, the total wight of the feature is kept small by the action of $w_2(y)$.

5.2.4 Simulation results for the use of adaptive moments

In this section we report some numerical simulations concerning the use of the adactive strategy described in Sect. 4.5.2.3 to select *online* the best discrete image moments to use for estimating the structure of a plane. All the following simulations consider a free-flying camera observing a planar scene $\mathcal{P}$ consisting of $N = 30$ points, and with plane parameters $\boldsymbol{n} = (0, 0, -1)$ and $d = 1.5\,\mathrm{m}$ in ${}^0\mathcal{F}_\mathcal{C}$ (the initial camera frame at $t = t_0$). The initial estimations of the plane normal and distance are always taken as $\widehat{\boldsymbol{n}}(t_0) = (-0.87,\ 0,\ -0.49)$ and $\widehat{d}(t_0) = 1\,\mathrm{m}$, thus representing an



(a) (b)

Figure 5.12 – **Simulation results obtained by employing the classical discrete moment set** $(x_g,\, y_g,\, \mu_{20}+\mu_{02})$ and optimizing for the camera linear velocity $\boldsymbol{v}$. In this case $\Phi_D(t)$ keeps very close to zero (the scale of Fig. (a) is 10^{-9}) and as a consequence the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ does not converge (Fig. (b)).

initial incertitude of $\approx 60\,\text{deg}$ on the real normal direction and of $0.5\,\text{m}$ on the real distance to the plane. Finally, the point features $\boldsymbol{p}_k$, $k = 1 \ldots N$, are sampled at $60\,\text{Hz}$ and then corrupted component-wise by a uniformly distributed random noise of magnitude 2 pixels before being processed for evaluating the image moments. The camera motion (and the optimization (4.64)) is instead updated at $100\,\text{Hz}$.

5.2.4.1 Unconstrained polynomial basis

We start with the results obtained by making use of the unconstrained polynomial basis (4.55) of fixed degree δ introduced in Sect. 4.5.2.3. We tested our method by considering a set of $m = 3$ weighted moments $\boldsymbol{s} = (m_w(\boldsymbol{\theta}_1),\, m_w(\boldsymbol{\theta}_2),\, m_w(\boldsymbol{\theta}_3)) \in \mathbb{R}^3$ with degree $\delta = 2$ defined as in (4.56), with then $\boldsymbol{\theta}_i \in \mathbb{R}^5$, $i = 1 \ldots 3$, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,\, \boldsymbol{\theta}_2,\, \boldsymbol{\theta}_3) \in \mathbb{R}^h$, $h = 15$. This choice was meant to provide a direct comparison against the use of:

1. the more 'classical set' $(x_g,\, y_g,\, \mu_{20}+\mu_{02})$ that, as explained, is known to be an optimal choice for *controlling* the camera translational motion but also to yield poor results when employed for SfM purposes;

2. the set of *five* moments $(x_g,\, y_g,\, \mu_{20},\, \mu_{11},\, \mu_{02})$ which, as reported in Sect. 5.2.2, does allow for a converging estimation but at cost of an increased complexity (need of propagating five image moments).

The goal of the comparison is to prove that estimation of vector $\boldsymbol{\chi}$ is, instead, fully possible when a suitable combination of just *three* moments of order up to 2 is selected.

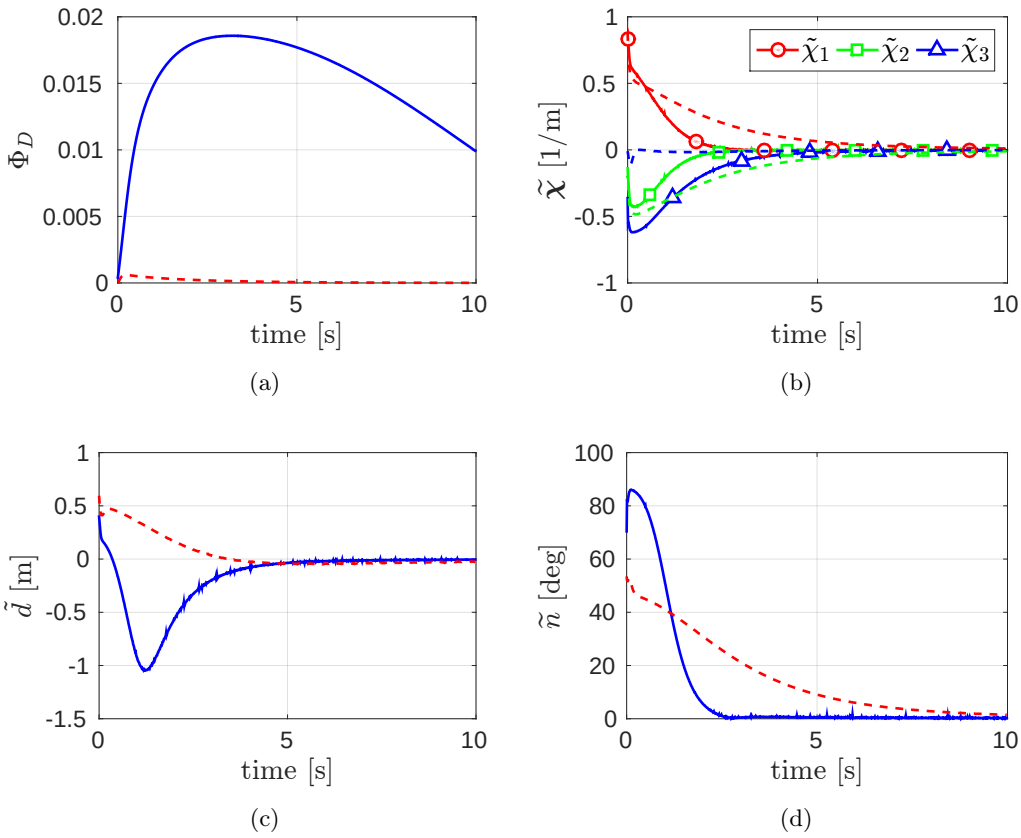


Figure 5.13 – **Simulation results for plane estimation employing three adaptive moments of degree $\delta = 2$ and optimizing for both vector $\boldsymbol{\theta}$ and the camera linear velocity $\boldsymbol{v}$** (*solid* lines). Fig. (a): the value of Φ_D is now $\approx 10^7$ times larger than in the previous case of Fig. 5.12(a). This then allows for a quick convergence of the error quantities $\widetilde{\boldsymbol{\chi}}(t)$ (Fig. (b)), $\widetilde{d}(t)$ (Fig. (c)) and $\widetilde{n}(t) = \arccos(\boldsymbol{n}^T(t)\widehat{\boldsymbol{n}}(t))$ (Fig. (d)). For a comparison, in all plots *dashed* lines correspond to the use of the *five* moments (x_g, y_g, μ_{20}, μ_{11}, μ_{02}): it is worth noting how, despite the increased measurement set (five moments vs. only three), the estimation convergence results still slower than in the weighted moment case.

Figures 5.12(a) and 5.12(b) start showing the results obtained by employing the set (x_g, y_g, $\mu_{20} + \mu_{02}$) for estimating vector $\boldsymbol{\chi}$ while, at the same time, optimizing the camera linear velocity $\boldsymbol{v}$ by implementing the first row of (4.64) with $k_v = 1$ (similarly to previous experiments). The linear velocity was initially set to $\boldsymbol{v}(t_0) = [0\ 0.1\ 0]^T$ m/s with then $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}(t_0)\| = 0.1\,\text{m/s}$ during the camera motion. As expected, and even despite the velocity optimization, the value of $\Phi_D(t)$ keeps (numerically) very close to 0 with a maximum of $\approx 1.2{\cdot}10^{-9}$ (Fig. 5.12(a)). Thus, the chosen set (x_g, y_g, $\mu_{20} + \mu_{02}$) is not able to provide enough information for allowing convergence of the SfM scheme, and indeed the estimation error $\widetilde{\boldsymbol{\chi}}(t) = \widehat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)$ even starts diverging (Fig. 5.12(b)).

On the other hand, exploiting the three weighted moments of degree 2 yields a

much more satisfactory estimation performance: Fig. 5.13 reports in *solid lines* the results obtained by implementing (4.64) with $k_v = 1$ and $k_{\boldsymbol{\theta}} = 3$, and by taking again $\boldsymbol{v}(t_0) = [0\ 0.1\ 0]^T$ m as in the previous case. The parameter vector $\boldsymbol{\theta}$ was instead chosen at random under the constraint $\|\boldsymbol{\theta}_i(t_0)\| = 1$, $i = 1 \ldots 3$.

Looking at Fig. 5.13(a) one can then verify how now $\Phi_D(t)$ attains an overall quite larger value compared to Fig. 5.12(a), with a maximum of $\approx 1.8 \cdot 10^{-2}$ (thus, more than 10^7 times larger than in the previous case). As a result, the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ is able to quickly converge towards $\boldsymbol{\emptyset}_3$ in about 4 s (Fig. 5.13(b)). For a better appreciation of the estimation performance, Figs. 5.13(c) and 5.13(d) also report the behavior of $\widetilde{d}(t) = \widehat{d}(t) - d(t)$ (the error in estimating the plane distance d) and $\widetilde{n} = \arccos(\boldsymbol{n}^T(t)\widehat{\boldsymbol{n}}(t))$ (the angular error in estimating the direction of the plane normal $\boldsymbol{n}$) with $\widehat{d} = 1/\|\widehat{\boldsymbol{\chi}}\|$ and $\widehat{\boldsymbol{n}} = -\widehat{\boldsymbol{\chi}}/\|\widehat{\boldsymbol{\chi}}\|$. Finally, Fig. 5.13 superimposes in *dashed lines* the behavior of $\Phi_D(t)$ and of the estimation errors when instead relying on the set of $m = 5$ moments $(x_g, y_g, \mu_{20}, \mu_{11}, \mu_{02})$ for estimating $\boldsymbol{\chi}$: in this case, the estimation error does actually converge (as expected from the results in Sect. 5.2.2), but nevertheless at a slower rate compared to the weighted moment case (indeed, the maximum value of $\Phi_D(t)$ is now 'only' $\approx 5.9 \cdot 10^{-4}$). We then believe these results clearly show the advantages of the proposed approach: the SfM scheme has its best performance when relying on the optimization (4.64) for automatically selecting (online) the best combination of *three* moments of order up to 2.

As an additional evaluation, Fig. 5.14 shows the results obtained when *only* optimizing the parameter vector $\boldsymbol{\theta}$ while keeping a *constant* linear velocity $\boldsymbol{v}(t) = \boldsymbol{v}(t_0)$ during the whole motion (thus, by setting $k_v = 0$ in (4.64)). This case is meant to assess the optimization performance in a situation in which the camera velocity cannot be arbitrarily adjusted but it must be considered as 'given' by an external source. Thus, the only possibility for improving the conditioning of the observability matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ is to act on vector $\boldsymbol{\theta}$, i.e., on the moment shape. Nevertheless also in this situation $\Phi_D(t)$ still reaches a range of values comparable with the previous case, with indeed $\max \Phi_D(t) \approx 1.05 \cdot 10^{-2}$ against the previous $1.6 \cdot 10^{-2}$ (thus, still $\approx 10^7$ times larger than when employing the classical set $(x_g, y_g, \mu_{20} + \mu_{02})$). As a result, vector $\widetilde{\boldsymbol{\chi}}(t)$ keeps converging to $\boldsymbol{\emptyset}_3$ in about 4 s (Fig. 5.14(b)) even if slightly more slowly w.r.t. the previous case of Fig. 5.13(b) (as one could expect because of the smaller value of $\Phi_D(t)$). In any case, we believe it is worth noting how the *sole* optimization of the moment shape (via vector $\boldsymbol{\theta}$) is still able to yield a very satisfactory SfM performance even for a non-optimal camera motion.

We finally remark that in all simulations the camera angular velocity $\boldsymbol{\omega}$ was exploited for keeping the centroid of the observed point features $\boldsymbol{\pi}_i$ at the center of the image plane (we recall that matrix $\boldsymbol{\Omega}$ and, thus, $\Phi_D(t)$ do not depend on $\boldsymbol{\omega}$

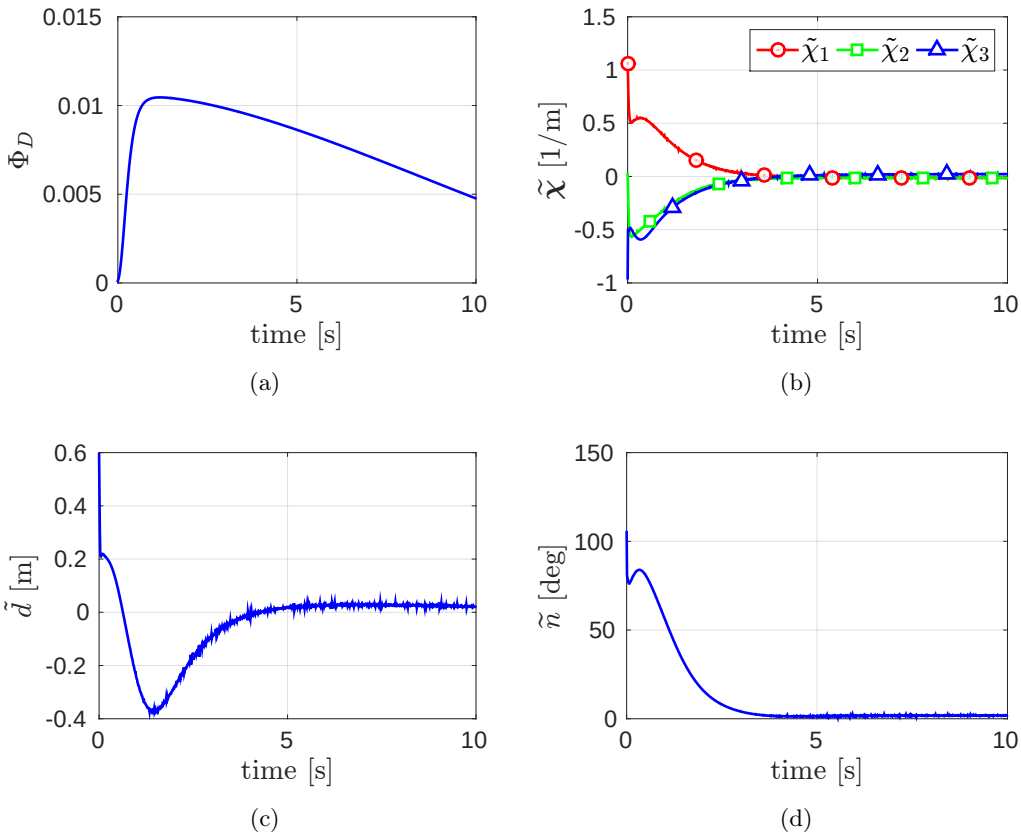


Figure 5.14 – **Simulation results for plane estimation employing three adaptive moments of degree $\delta = 2$ and only optimizing for vector $\boldsymbol{\theta}$.** Fig. (a): note how $\Phi_D(t)$ still reaches a range of values comparable with the previous case of Fig. 5.13(a) despite the lack of any optimization of the camera velocity $\boldsymbol{v}$. The other figures show the behavior of the error quantities $\widetilde{\boldsymbol{\chi}}(t)$ (Fig. (b)), $\widetilde{d}(t)$ (Fig. (c)) and $\widetilde{n}(t)$ (Fig. (d)).

that can then be freely chosen without affecting the estimation performance).

5.2.4.2 Constrained polynomial basis

We now address the case of the constrained polynomial basis (4.57–4.58)) described in Sect. 4.5.2.3 and meant to smoothly take into account the loss/gain of point features because of the camera limited FOV. We consider again a set of $m = 3$ weighted moments $m = (m_w(\boldsymbol{\theta}_1), m_w(\boldsymbol{\theta}_2), m_w(\boldsymbol{\theta}_3)) \in \mathbb{R}^3$ with both functions $w^x(\cdot)$ and $w^y(\cdot)$ taken as the fifth-order polynomials given in (4.59) with, therefore, a total of $h_x + h_y - 8 = 4$ parameters to be optimized. The initial camera velocity $\boldsymbol{v}(t_0)$ was set as in the previous cases, and the optimization action (4.64) was again implemented with $k_v = 1$ and $k_{\boldsymbol{\theta}} = 3$. The camera angular velocity $\boldsymbol{\omega}$ was instead kept null for facilitating the loss or point features during motion.

Figure 5.15(a) shows the camera trajectory during the estimation task, and

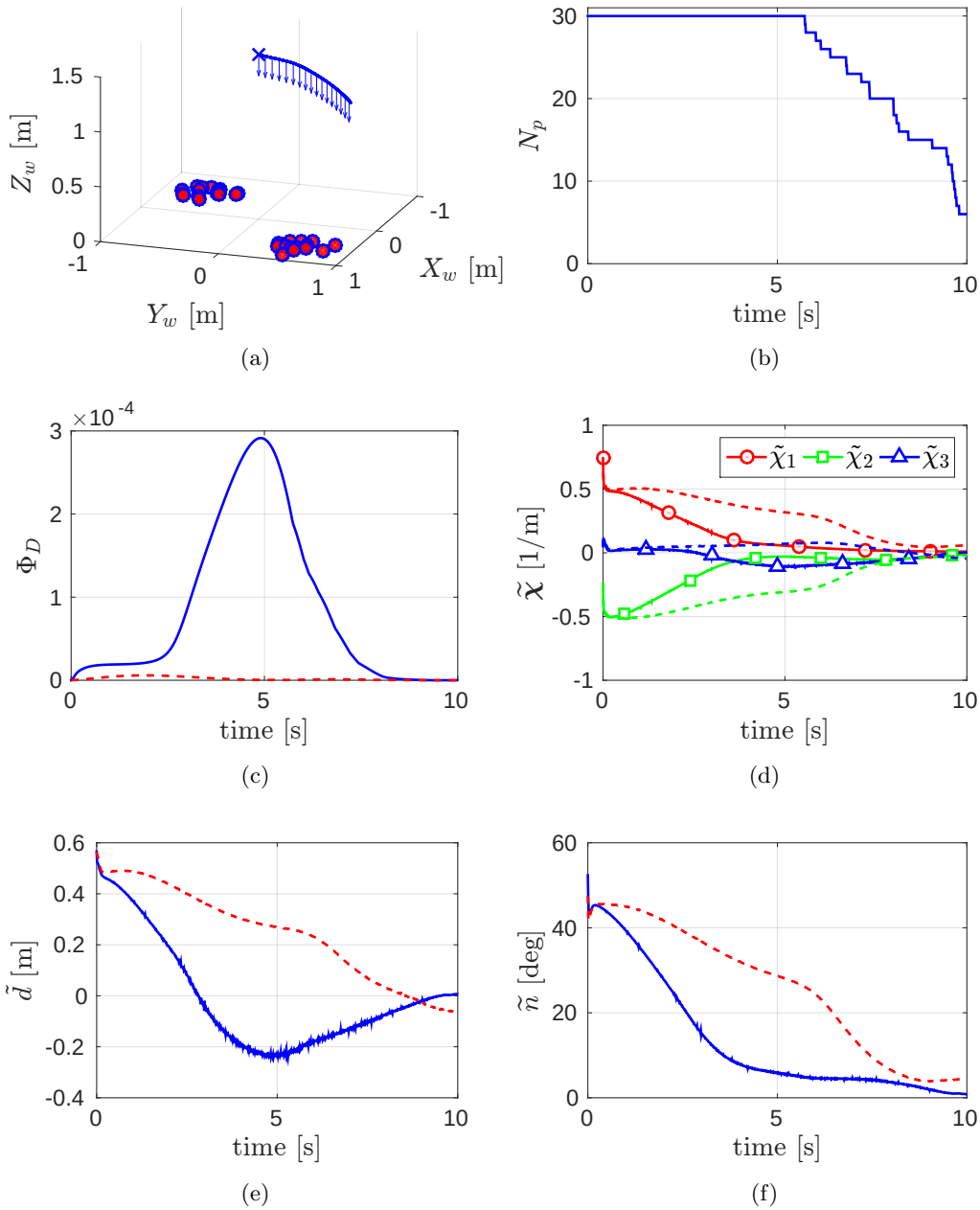


Figure 5.15 – **Simulation results for plane estimation employing three constrained adaptive moments, a camera with limited FOV, and optimizing for both vector $\boldsymbol{\theta}$ and the linear velocity $\boldsymbol{v}$** (*solid* lines). Fig. (a): camera trajectory and direction of the optical axis during the estimation task. Fig. (b): number N_p of tracked point features over time. Fig. (c): behavior of $\Phi_D(t)$ which reaches a maximum of $\approx 2.9 \cdot 10^{-4}$ before starting to decrease because of the fewer tracked points. Finally the other figures show the behavior of the error quantities $\widetilde{\boldsymbol{\chi}}(t)$ (Fig. (d)), $\widetilde{d}(t)$ (Fig. (e)) and $\widetilde{n}(t)$ (Fig. (f). In *dashed* lines, the behavior that all quantities would have had in case no optimization of $\boldsymbol{\theta}$ had been performed.

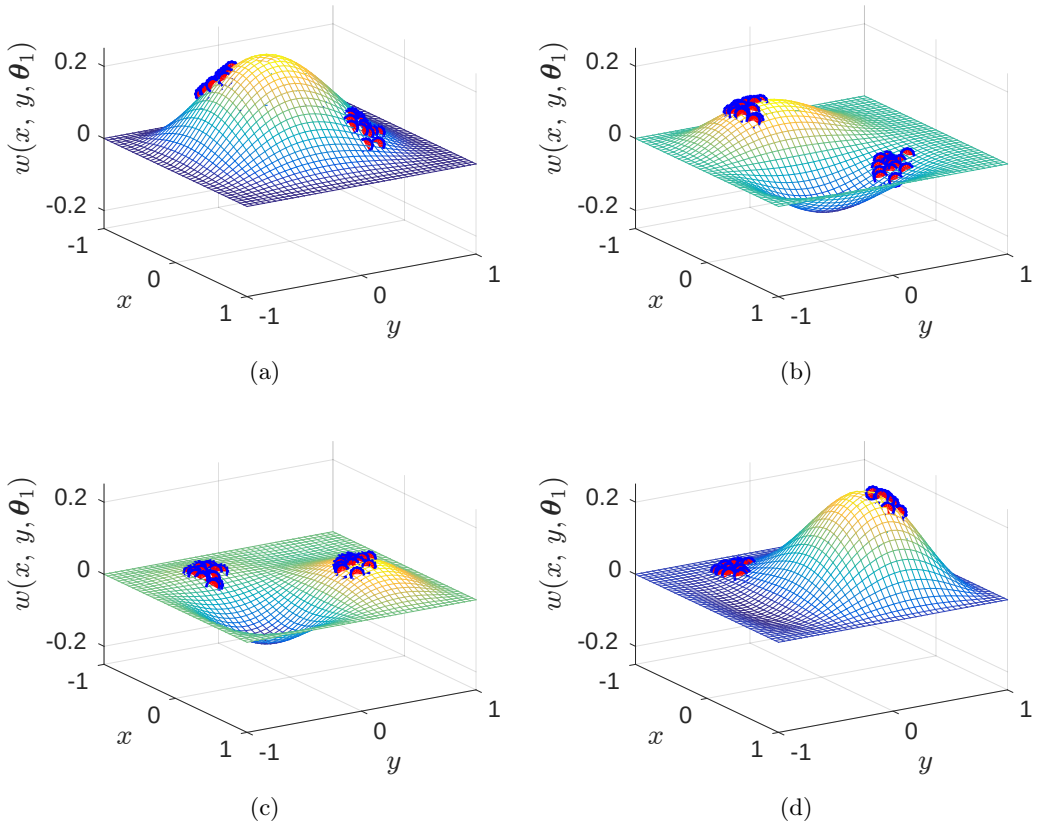


Figure 5.16 – **Four snapshots of the weighting function $w(x, y, \boldsymbol{\theta}_1)$ taken during the camera motion**. It is interesting to visualize how function $w(\cdot)$ automatically adjusts its shape as a function of the observed scene (e.g., it tends to peak around clusters of points).

Fig. 5.15(b) reports the number $N_p(t)$ of tracked features over time: after about 5 s some points start being lost, dropping from a total of 30 to a minimum of 6 at $t = 10$ s. Nevertheless, thanks to the adopted *constrained* weighted moments, the scene structure is still correctly estimated without suffering from discontinuities or numerical instabilities because of the lost features. Figure 5.15(c) reports again the behavior of $\Phi_D(t)$ (blue solid line) that reaches a maximum of $\approx 2.9 \cdot 10^{-4}$ before starting to decrease at $t \approx 5$ s because of the fewer tracked points. As a comparison, Fig. 5.15(c) also reports the superimposed behavior of $\Phi_D(t)$ in case no optimization of vector $\boldsymbol{\theta}$ had been performed (the almost horizontal red dashed line). In this case, the maximum attained value for $\Phi_D(t)$ would have been $\approx 1.2 \cdot 10^{-6}$ (100 times smaller), thus proving again the importance of properly optimizing the shape of the chosen weighting function $w(\cdot)$. Figures 5.15(d) to 5.15(f) then show (in *solid* lines) the behavior of the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ and of the corresponding quantities $\widetilde{d}(t)$ and $\widetilde{n}(t)$ that *smoothly* reach convergence in about 10 s of motion despite the loss of point features. Again, for a comparison, Figs. 5.15(d) to 5.15(f) also report

the superimposed behavior (in *dashed* lines) of the estimation errors in case of no optimization of vector $\boldsymbol{\theta}$ (all quantities have a slower convergence rate as expected).

Finally, Fig. 5.16 depicts four snapshots of the shape of function $w(x, y, \boldsymbol{\theta}_1)$ used to compute the first constrained weighted moment. One can then appreciate how the function shape evolves over time and, in particular, *automatically* polarizes its peaks around the location of the tracked point features.

5.2.5 Simulation results of plane estimation from dense image moments

We conclude the analysis of the planar case by showing some preliminary simulative results of the active estimation framework applied to the case of *dense* image moments discussed in Sect. 4.5.2.4. As explained, in this case the problem is 'square': 3 available measurements (area and barycenter coordinates) for 3 unknowns (vector $\boldsymbol{\nu}$). Let $\boldsymbol{J}_{v,1}^T$ be the Jacobian associated with the first (smallest) eigenvalue σ_1^2 of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ from (4.67), the control law (5.1) can be used also in this case to maximize σ_1^2 while keeping a constant norm of $\boldsymbol{v}$. Similarly to what done for the point feature case, two simulations were run in which $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$ but with its direction being either

case I: optimized to maximize the estimation convergence rate (red line) using (5.1) or

case II: kept constant so that $\boldsymbol{v}(t) = \boldsymbol{v}_0 = const$ (blue line).

In particular, we used: $\alpha = 5 \cdot 10^4$, $\widetilde{\boldsymbol{s}}(t_0) = \boldsymbol{\emptyset}_3$, $\widetilde{\boldsymbol{\chi}}(t_0) = [-2.16\ 4.5\ 1.27]^T$, $\boldsymbol{v}(t_0) = [0.02\ \ -0.05\ \ -0.005]^T$, $k_1 = 10$, $k_2 = 100$ in case I and $k_2 = 0$ in case II, and $\kappa_{\mathrm{d}} = \frac{1}{2}\boldsymbol{v}_0^T\boldsymbol{v}_0$. The moments were generated from a planar circle of radius $R = 0.2$ (see Fig. 5.17(c)).

For this simulation, we also exploited the camera angular velocity $\boldsymbol{\omega}$ in order to keep the observed barycenter (x_g, y_g) stationary during motion. As explained at the end of Sect. 4.4, this was meant to (partially) mitigate the effects of a non-zero $\dot{\mathbf{y}}$ when inverting (4.22), with $\mathbf{y} = [a\ x_g\ y_g\ n_{20}\ n_{11}\ n_{02}]$ for the moment case.

Figure 5.17(b) depicts the behavior of the three eigenvalues $\sigma_1^2(t)$, $\sigma_2^2(t)$, and $\sigma_3^2(t)$ over time for case I (red lines) and case II (blue lines). We can then note how, while the smallest eigenvalue $\sigma_1^2(t)$ is correctly maximized, in case I, by the effect of the control law (5.1), the other two eigenvalues $(\sigma_2^2(t), \sigma_3^2(t))$ (which are not being controlled) are larger in case II. The consequence of this is that, at the beginning, the error convergence rate in case II is faster than in case I, see Fig. 5.17(a). Nevertheless, once the fastest modes have converged, the slowest one, associated with

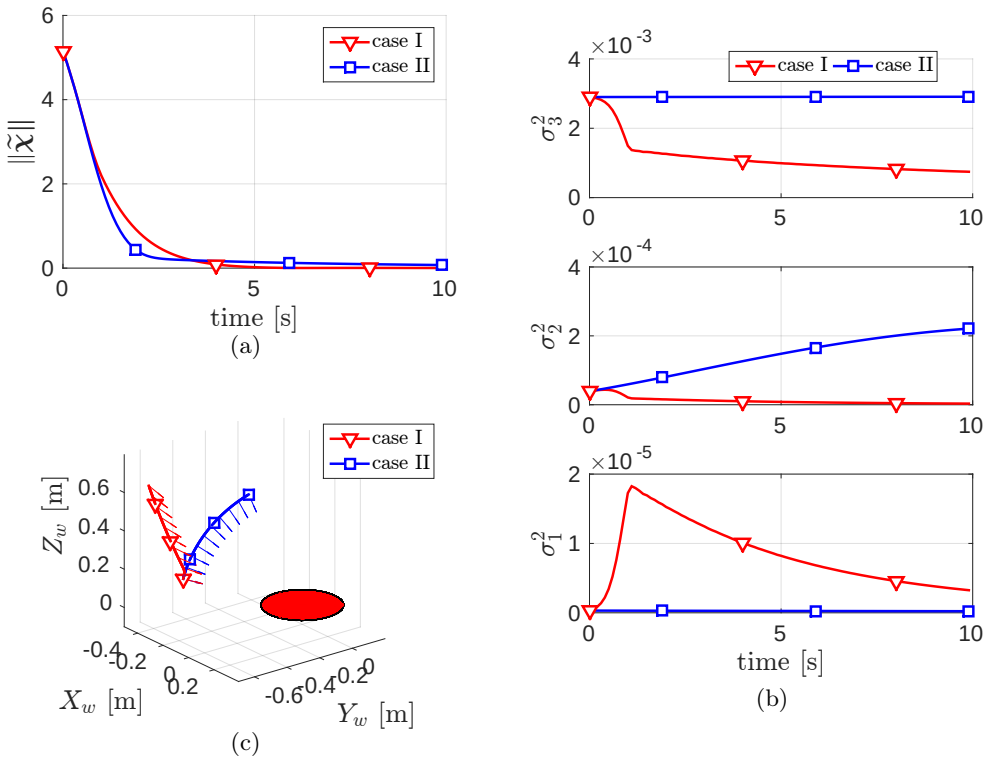


Figure 5.17 – **Simulation results for the dense moments case**. Fig. (a): behavior of $\|\widetilde{\chi}(t)\|$ for case I (red lines) and case II (blue lines). Fig. (b): behavior of the three eigenvalues of matrix $\mathbf{\Omega}\mathbf{\Omega}^T$ for case I (red lines) and case II (blue lines). Note how the convergence rate of case II is initially faster than in case I due to a higher value of the two largest eigenvalues σ_2^2, σ_3^2 (see Fig. (b)) which are not being controlled. However, once the smallest eigenvalue σ_1^2 becomes dominant, the maximization of σ_1^2 operated in case I results, as expected, in a faster convergence rate of the estimation error. Fig. (c): 3-D Cartesian trajectory followed by the camera for case I (red line) and case II (blue line) together with the planar circle used to generate image moments. In case I, the camera has a larger velocity along the Z-axis that results in a larger σ_1^2.

$\sigma_1^2(t)$, dominates the error behavior and, as a consequence, having maximized $\sigma_1^2(t)$ results in an overall faster convergence in case I. This then confirms the validity of our analysis also in the case of dense image moments. Finally, Fig. 5.17(c) shows the 3-D trajectory followed by the camera for case I (red line) and case II (blue line): one can note how(i) the camera keeps looking at the barycenter as expected, and how (ii) in case I the camera velocity has a larger component along the Z-axis that results in an increase of $\sigma_1^2(t)$.

5.3 Active structure estimation for a sphere

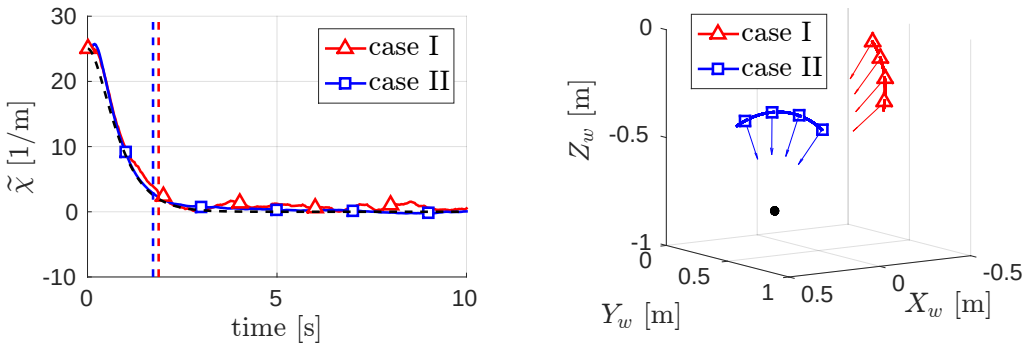
We now discuss some experimental results concerning the estimation of the radius of a spherical target: indeed, as explained in Sect. 4.5.3, estimation of R allows to

fully recover the sphere 3-D position $\boldsymbol{p}_0 = \boldsymbol{s}R$ where vector $\boldsymbol{s}$ is directly obtainable from image measurements, see (4.70).

As object to be tracked, we made use of a white table tennis ball placed on a black table and with a radius of 1.9 cm (see Fig. 5.3(b)). As explained in Sect. 4.5.3, the convergence rate of the estimation error for the sphere case only depends on the norm of the linear velocity $\|\boldsymbol{v}\|$ and not on its direction. This fact is proven by the first experiment where the estimation task is run twice starting from two different positions and imposing two different camera velocities but with same norm. These values were used during the experiments: $\alpha = 2 \cdot 10^3$, $c_1 = c_1^* = 2\sqrt{\alpha}\sigma$ for $\boldsymbol{D}_1$ in (4.8), and $\boldsymbol{v} = (-0.05,\, 0,\, 0)$ m/s for case I and $\boldsymbol{v} = (0,\, 0.045,\, 0.02)$ m/s for case II, with $\|\boldsymbol{v}\| = 0.05$ m/s in both cases. The camera angular velocity $\boldsymbol{\omega}$ was exploited to keep $(s_x,\, s_y) \simeq (0,\, 0)$ (centered sphere).

Figure 5.18(a) shows the behavior of the estimation errors (solid blue and red lines): note how the error transient response for the two cases is essentially coincident, and also equivalent to that of the *reference second order system* (4.15) with the desired poles, i.e., by setting $\sigma_1^2 = \|\boldsymbol{v}\|^2 = const$ and $c_1 = c_1^*$ in (4.15) (dashed black line). The higher noise level in case II (red line) is due to the larger distance between the camera and the spherical target (see Fig. 5.18(b)) which negatively affects the estimation task. The standard deviation of the radius estimation error, computed on a time window of 1 s after $\widetilde{\chi}(t)$ has become smaller than 1 mm (vertical dashed lines in the plot), is 0.3 mm for case I and 0.2 mm for case II: we can note, again, the very satisfactory results obtained with the proposed estimation scheme in terms of accuracy of the reconstructed sphere radius. Note also how, in the two cases, the estimation error $\widetilde{\chi}(t)$ drops below the threshold of 1 mm at essentially the same time, as expected (same error transient response).

Since the direction of the velocity does not play any role in this case, no optimization of σ_1^2 can be performed under the constraint $\|\boldsymbol{v}\| = const$. On the other hand, the analysis of Sect. 4.3 clearly indicates the importance of choosing a proper value of c_1 for the damping matrix $\boldsymbol{D}_1$ in (4.8). To show this fact, we report here three experiments characterized by the same camera trajectory of the previous case I, but by employing three different values for c_1, that is, c_1^*, $2c_1^*$ and $0.5c_1^*$. These correspond to a critically damped, overdamped and underdamped response for the ideal system (4.15), respectively. The experimental results reported in Fig. 5.19 show that the behavior of the estimation error $\widetilde{\chi}$ (solid lines) has an excellent match with that of (4.15) (represented by dashed lines), thus fully confirming(i) the validity of the proposed theoretical analysis, and (ii) the importance of choosing the 'right' damping matrix $\boldsymbol{D}_1$ for optimizing the convergence speed in addition to a proper regulation of σ_1^2.



(a) (b)

Figure 5.18 – **Experimental results for the estimation of the radius of a sphere using different constant camera velocities with the same norm**. Fig. (a): behavior of the estimation error $\widetilde{\chi}(t)$ for the two cases (solid blue and red lines), and for an 'ideal' second order system with poles at the desired locations (dashed black line). The vertical dashed lines indicate the times at which the estimation error $\widetilde{\chi}(t)$ drops below the threshold of 1 mm. Fig. (b): camera trajectories for case I (blue line) and case II (red line) with arrows indicating the direction of the camera optical axis.

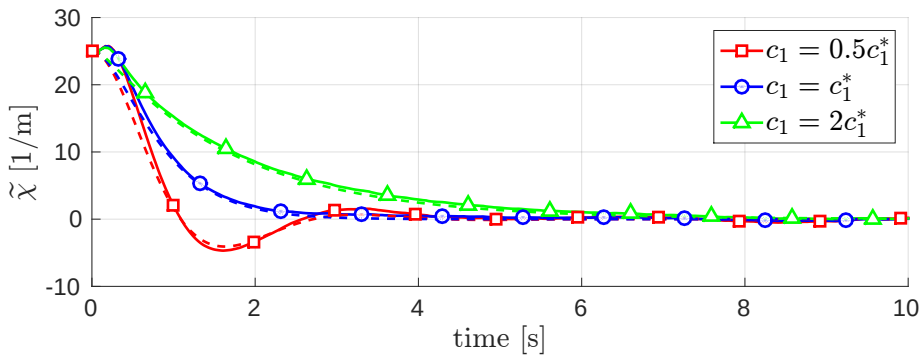


Figure 5.19 – **Experimental results for the estimation of the radius of a sphere with different damping factors**: $c_1 = c_1^*$ (blue line), $c_1 = 2c_1^*$ (green line) and $c_1 = 0.5c_1^*$ (red line). The dashed lines represent the response of an 'ideal' second order system with the corresponding poles. Note again the almost perfect match between the plots.

5.4 Active structure estimation for a cylinder

In this final section we report some experimental results concerning the active estimation of the radius of a cylindrical object. Indeed, as in the sphere case, knowledge of R allows to fully recover the 3-D point $\boldsymbol{p}_0 = R\boldsymbol{s}$, with vector $\boldsymbol{s}$ from (4.81) and the cylinder axis $\boldsymbol{a}$ in (4.83) being directly obtainable from image measurements. For these experiments we used a white cardboard cylinder placed on a black table (see Fig. 5.3(c)). The radius of the cylinder was approximately 4.2 cm.

In the cylinder case, the convergence rate of the estimation error depends both on the norm of the camera linear velocity $\boldsymbol{v}$ and on its direction w.r.t. the cylinder axis $\boldsymbol{a}$, see (4.86). It is then interesting to optimize the direction of $\boldsymbol{v}$ under the constraint $\|\boldsymbol{v}\| = const$ for maximizing the eigenvalue σ_1^2 (i.e., so as to obtain the fastest convergence rate for a given 'control effort' $\|\boldsymbol{v}\|$).

From (4.87), maximization of $\sigma_1^2(t)$ w.r.t. vector $\boldsymbol{v}$ can be obtained by choosing

$$\dot{\boldsymbol{v}} = \boldsymbol{J}_{v,1}^T - \boldsymbol{J}_{v,1}^\dagger \boldsymbol{J}_{a,1} \left[\boldsymbol{a}\right]_\times \boldsymbol{\omega}, \tag{5.2}$$

with, i.e., by following the gradient of σ_1^2 w.r.t. $\boldsymbol{v}$ and by compensating for the (known) effects of input $\boldsymbol{\omega}$. In order to additionally enforce the constraint $\|\boldsymbol{v}\| = const$ during the eigenvalue maximization, (5.2) can be modified, similarly to (5.1), as

$$\dot{\boldsymbol{v}} = -k_1 \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|^2} (\kappa - \kappa_\mathrm{d}) + k_2 \left(\boldsymbol{I}_3 - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^2} \right) (\boldsymbol{J}_{v,1}^T - \boldsymbol{J}_{v,1}^\dagger \boldsymbol{J}_{a,1} \left[\boldsymbol{a}\right]_\times \boldsymbol{\omega}), \tag{5.3}$$

with $k_1 > 0$ and $k_2 > 0$. Analogously to the point feature case, the first term in (5.3) asymptotically guarantees $\|\boldsymbol{v}(t)\| = \|\boldsymbol{v}_0\|$ while the second term projects (5.2) onto the null-space of the constraint $\|\boldsymbol{v}(t)\| = const$. As for the angular velocity $\boldsymbol{\omega}$, we exploited it for keeping the axis of the cylinder at the center of the image plane by regulating (s_x, s_y) to $(0, 0)$.

We now present three experimental results structured as follows:

case I: in the first experiment, the update rule (5.3) is fully implemented ($k_1 > 0$, $k_2 > 0$) for actively optimizing the direction of $\boldsymbol{v}$;

case II: in the second experiment, the camera starts from the same initial pose and velocity as in case I, but (5.3) is implemented with $k_1 > 0$ and $k_2 = 0$, i.e., without performing any optimization of σ_1^2;

case III: finally, in the third experiment, the camera starts from a different initial pose and with a different velocity direction (but same norm) w.r.t. the previous two cases, and (5.3) is again fully implemented. This last case

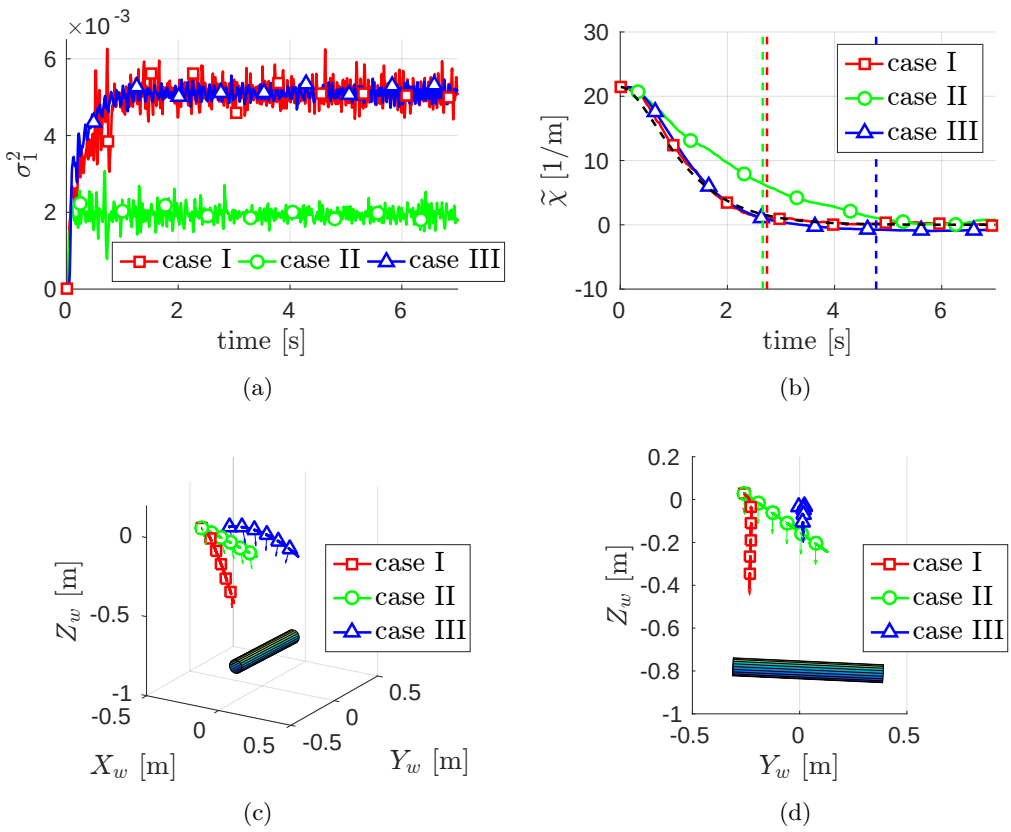


Figure 5.20 – **Experimental results for the estimation of the radius of a cylinder** with the following color coding: blue – case I, red – case II, green – case III. Fig. (a): behavior of $\sigma_1^2(t)$ for the three cases (coincident for cases I and III and larger than in case II). Fig. (b): behavior of $\widetilde{\chi}(t)$. The three vertical dashed lines indicate the times $T_1 = 2.74\,\mathrm{s}$, $T_2 = 4.78\,\mathrm{s}$ and $T_3 = 2.66\,\mathrm{s}$ at which the estimation error drops below the threshold of 2 mm. Note how $T_1 \approx T_3$ and $T_1 < T_2$ as expected. Fig. (c): two views of the camera trajectories for the three cases with arrows indicating the direction of the camera optical axis.

is meant to show how the convergence properties of the estimator are not affected by the direction of the camera linear velocity as long as it stays orthogonal to the cylinder axis $\boldsymbol{a}$.

The experiments were run with the following conditions: $\alpha = 500$, $c_1 = c_1^*$ for $\boldsymbol{D}_1$ in (4.8), $k_1 = 10$, $k_2 = 1$ for cases I and III, and $k_2 = 0$ for case II. As for the linear velocity, we set $\boldsymbol{v}(t_0) = \boldsymbol{v}_0 = (-0.01, 0.05, 0.05)$ m/s for cases I and II, and $\boldsymbol{v}(t_0) = \boldsymbol{v}_0 = (-0.05, 0.05, 0.01)$ m/s for case III (note how $\|\boldsymbol{v}_0\|^2 = 5.1 \times 10^{-3}$ $\mathrm{m}^2/\mathrm{s}^2$ in all three cases).

The behavior of $\sigma_1^2(t)$ is shown in Fig. 5.20(a): as explained at the end of Sect. 4.5.4, under the constraint $\|\boldsymbol{v}\| = const$, one has $\max_{\boldsymbol{v}} \sigma_1^2 = \|\boldsymbol{v}\|^2$ as the largest possible value for σ_1^2 (obtained when $\boldsymbol{v}^T\boldsymbol{a} = 0$). It is then possible to verify

that, indeed, $\sigma_1^2(t) \to \|\boldsymbol{v}_0\|^2$ in cases I and III despite the different initial conditions of the experiments (different camera pose and direction of $\boldsymbol{v}$). The optimization in (5.3) results in a null component of $\boldsymbol{v}$ along $\boldsymbol{a}$, thus allowing to move faster in the 'useful' directions (while keeping a constant $\|\boldsymbol{v}\|$), and to increase the value of σ_1^2 to its maximum possible value. Also, note how the value of $\sigma_1^2(t)$ for case II results smaller than in the other two cases (as expected) since no optimization is present in this case.

The behavior of the estimation error $\widetilde{\chi}(t)$ is shown in Fig. 5.20(b): again, we can note that the transient response for cases I and III results essentially coincident and in almost perfect agreement with that of the reference system (4.15) with desired poles (dashed black line). As expected, the response for case II (red line) is slower than in cases I and III. As in the point feature case, Fig. 5.20(b) reports, for the three cases under consideration, the times $T_1 = 2.74\,\mathrm{s}$, $T_2 = 4.78\,\mathrm{s}$ and $T_3 = 2.66\,\mathrm{s}$ at which the estimation error drops below the threshold $2\,\mathrm{mm}$ (vertical dashed lines). The standard deviation of the error evaluated on a time window of $1\,\mathrm{s}$ after convergence has been 'reached' resulted in the values of 0.4, 0.6 and $0.7\,\mathrm{mm}$, respectively. We can then appreciate, again, the high accuracy of the proposed approach in estimating the cylinder radius R while also optimizing online for the camera motion. The higher estimation error in case III can be ascribed to the larger distance between the camera and the observed target, which increases the effect of discretization errors. Note also how $T_1 \approx T_3 < T_2$ thanks to the active optimization of the error convergence rate. Finally, Fig. 5.20(c) depicts the camera trajectories for the three experiments with arrows indicating the direction of the optical axis. In case II the camera simply travels along a straight line ($\boldsymbol{v}(t) \equiv \boldsymbol{v}_0$), while in cases I and III the direction of $\boldsymbol{v}$ is suitably modified resulting in a trajectory lying on a plane orthogonal to $\boldsymbol{a}$.

5.5 Conclusions

In this chapter we reported a large collection of simulation and experimental trials meant to validate the theoretical results of Chapt. 4 in the context of active structure estimation from motion.

All of the geometric primitives considered in Sect. 4.5 (i.e. point features, planes, spheres and cylinders) were re-discussed, in this chapter, from a numerical/experimental point of view. The experiments were run on a 6-DOFs manipulator equipped with a camera in-hand. We showed that, even in non ideal experimental conditions, the use of the nonlinear observer described in Sect. 3.2.3 allows to obtain a good match between the dynamics of the SfM estimation error and that of a linear second order system with assigned poles when the quantity to be estimated is a single

scalar (i.e. for the point feature, sphere and cylinder cases) in agreement with Remark 4.1. More importantly, we experimentally demonstrated that the use of an active strategy for selecting the camera motion policy can, as predicted by the theoretical analysis, increase the convergence rate of the estimation and reduce the final estimation covariance for the same control effort (same camera linear velocity norm). This latter result is also true regardless of the chosen estimation scheme and, in particular, a brief comparison between the adopted nonlinear observer and a standard EKF scheme demonstrated that the camera trajectory optimization is also beneficial for this latter alternative estimation algorithm.

For the point feature case, we also compared, in simulation, the two different estimation schemes, proposed in Sect. 4.5.1.1, based on either a planar or a spherical projection model. We pointed out the differences between the two approaches in terms of robustness and estimation convergence rate. We also demonstrated however, that for our real camera parameters, the difference between the two projection models is negligible.

For the planar case, we experimentally compared the active plane estimation strategies introduced in Sects. 4.5.2.1 and 4.5.2.2 with a more standard homography based reconstruction technique showing the advantages of the former in terms of intrinsic robustness w.r.t. the presence of outliers. Moreover we presented some simulations in which the use of an adaptive strategy for selecting the image moments to use for plane estimation was investigated showing very promising results in terms of additional improvement of the estimation convergence rate. Finally, we reported some simulation results showing how the proposed machinery could be extended to the case of dense image moments.

For the sphere case, we showed that, due to the symmetry of the scene, the direction of the camera linear velocity does not affect the convergence rate of the estimation, however it is still possible to tune the observer gains online so as to fix a desired damping coefficient of the error dynamics.

Finally, for the cylinder case, we experimentally proved that, similarly to the sphere, as long as the camera does not translate along the direction of the cylinder axis, the direction of the camera velocity does not affect the estimation error converge rate.

The results reported in this chapter confirm, in our opinion, the theoretical analysis of Chapt. 4.

Part III

Information aware Visual Servoing

Chapter 6

Coupling active SfM and IBVS

In the first part of this thesis, we introduced the basic concepts behind computer vision, robot control and state estimation. We showed that a camera is a scale invariant sensor in the sense that the perspective projection that it operates, transforms the 3-D world in a 2-D image in which the information about the scale is lost. In general, if one wishes to use a camera to *control* the full pose of a robot, knowledge of this scale will become necessary for either reconstructing the complete camera state (in PBVS frameworks) or for correctly estimating the feature interaction matrix of a IBVS control scheme (see Sect. 2.1.3.3 and Sect. 2.3.2 and [MMR10]). The scale can be estimated by taking multiple images from different camera positions and feeding them to, e.g., one of the estimation schemes discussed in Chapt. 3. Due to the nonlinearity of the problem, however, the estimation is only possible if the camera trajectory is sufficiently persistently *exciting* and thus enough information is collected. A IBVS controller, e.g., should then be able to realize the main visual task while, *at the same time*, ensuring a sufficient level of information gain for allowing an accurate state estimation.

In Part II we put the classical IBVS control problem to the side and we concentrated on the estimation problem. In particular we devised and experimentally validated an estimation/control strategy that gives full control over the eigenvalues of the estimation error dynamics (which is made equivalent to that of a second order linear system) by acting online on the estimation gains and, more importantly, on the camera velocity. While doing this, we considered the structure estimation as the main task that had to be accomplished by the robot and we did not constrain the camera velocity in any way, apart from requiring a constant norm.

In this chapter, and in the next one, we consider, instead, a more common situation in which the estimation is not directly the main task that the robot is required to accomplish but still some estimated quantities are used to compute the robot control law and thus the accuracy of the estimation affects the control

performance. In particular, we investigate the possible coupling between the framework for active SfM introduced in Part II and the execution of a standard IBVS task. The main idea is to project any optimization of the camera motion (aimed at improving the SfM performance) within the null-space of the considered task in order to not degrade the servoing execution (see Sect. 2.2.4.3). For any reasonable IBVS application, however, a simple null-space projection of a camera trajectory optimization (as in (2.26)) turns out to be ineffective because of a structural lack of redundancy. Therefore, in order to gain the needed freedom for implementing the SfM optimization, we suitably exploit and extend the redundancy framework introduced in [MC10] which grants a *large* projection operator by considering the *norm* of the visual error as main task. In addition, an adaptive mechanism is also introduced with the aim of activating/deactivating online the camera trajectory optimization as a function of the accuracy of the estimated 3-D structure. Thanks to this addition, it is then possible to enable the SfM optimization only when strictly needed such as, e.g., when the 3-D estimation error grows larger than some desired minimum threshold.

As discussed in depth in Sect. 3.4, there exists a vast literature on the topic of *trajectory optimization* for improving the identification/estimation of some unknown parameters/states. Similarly, a large number of works has addressed the so-called Next Best View (NBV) problem, i.e., loosely speaking, how to actively control the motion of a vision sensor so as to reconstruct the 3-D shape of an object of interest or of the surrounding environment while, e.g., minimizing the estimation uncertainty, the traveled distance, the number of acquired images, or any other meaningful criterion. However, many of these strategies are meant for an *offline* use (a whole trajectory is planned, executed, and then possibly re-planned based on the obtained results), and, in any case, do not take into account the *online* realization of a visual task *concurrently* to the optimization of the estimation performance. At the other end of the spectrum, several works have already investigated how to plug the online estimation of the 3-D structure into a IBVS loop (i.e., considering the realization of a visual task), but *without* any concurrent optimization of the camera motion for improving the estimation performance, see, e.g., [MHMM09, FKS07, DOR08, PCCB10, Cor10, MS12]. In all of these works, the SfM scheme is just fed with the camera trajectory generated by the IBVS controller which, on the other hand, has no guarantee of generating a sufficient level of excitation w.r.t. the estimation task.

With respect to this previous literature, this chapter provides, instead, a *unified* and *online* solution to the problem of concurrently optimizing execution of a IBVS task (visual control) and performance of the 3-D structure estimation (active perception). We also wish to stress that the proposed machinery is not restricted to the sole class of IBVS problems considered in this work: indeed, one can easily gen-

eralize the reported ideas to other servoing tasks, or apply them to other contexts not necessarily related to visual control (as long as the chosen robot trajectory has an effect on the state estimation performance).

The rest of the chapter is organized as follows: after a short introduction in Sect. 6.1, Sect. 6.2 details the machinery needed for coupling IBVS execution and optimization of the 3-D structure estimation, with a particular emphasis on the second-order extension of the strategy described in [MC10] for increasing the redundancy w.r.t. the considered visual task. Subsequently, Sect. 6.3 introduces an extension of the strategy detailed in Sect. 6.2 for allowing a smooth activation/deactivation of the camera trajectory optimization as a function of the current estimation accuracy. Finally, Sect. 6.4 concludes the chapter with some final considerations.

Some (quite) preliminary results in this context were presented in [7]. W.r.t. this previous work, we provide, in this chapter, a more complete analysis from both a theoretical and experimental point of view. Most of the material presented here can be found in [9] which, at the time of writing, is under consideration for publication. Part of these more recent results were also presented in [8].

6.1 Problem description

Let us consider again the classical situation of a robot manipulator with joint configuration vector $\boldsymbol{q} \in \mathbb{R}^n$ carrying an eye-in-hand camera that measures a set of visual features $\boldsymbol{s} \in \mathbb{R}^m$ (e.g., the x and y coordinates of a point feature). We have already discussed in Sect. 2.3.2 some control strategies that allow to regulate $\boldsymbol{s}$ to a desired constant value $\boldsymbol{s}_\mathrm{d}$. As already anticipated in Remark 2.1, whatever the particular case (redundant/non-redundant, feasible/non-feasible), any implementation of (2.32–2.33) (or variants) must also deal with the lack of a direct measurement of vector $\boldsymbol{\chi}$ which prevents the exact on-line computation of $\boldsymbol{J}$. A common workaround is to replace the exact task Jacobian $\boldsymbol{J}(\boldsymbol{s}, \boldsymbol{\chi}, \boldsymbol{q})$ with an estimation $\widehat{\boldsymbol{J}} = \boldsymbol{J}(\boldsymbol{s}, \widehat{\boldsymbol{\chi}}, \boldsymbol{q})$ evaluated on some *approximation* $\widehat{\boldsymbol{\chi}}$ of the unknown true vector $\boldsymbol{\chi}$, for instance the value at the desired pose $\widehat{\boldsymbol{\chi}} = \boldsymbol{\chi}_\mathrm{d} = \mathit{const}$ [CH06], assuming that this is known or easily measurable in advance by additional sensors. In this approximated case, assuming for simplicity $\ddot{\boldsymbol{q}}_w \equiv \boldsymbol{\emptyset}_m$, the closed-loop dynamics, obtained by, plugging (2.32–2.33) into (2.31), becomes

$$\dot{\boldsymbol{e}} = -\lambda \boldsymbol{J} \widehat{\boldsymbol{J}}^\dagger \boldsymbol{e}. \tag{6.1}$$

Local stability of (6.1) in a neighbourhood of $\boldsymbol{e} = \boldsymbol{\emptyset}_m$ is then determined by the eigenvalues of matrix $\boldsymbol{J}(\boldsymbol{s}_\mathrm{d}, \boldsymbol{\chi}_\mathrm{d}, \boldsymbol{q})\boldsymbol{J}(\boldsymbol{s}_\mathrm{d}, \widehat{\boldsymbol{\chi}}, \boldsymbol{q})^\dagger$.

In case the approximation $\widehat{\boldsymbol{\chi}} = \boldsymbol{\chi}_\mathrm{d}$ is used, then (6.1) will be locally asymptotically stable around $\boldsymbol{e} = \boldsymbol{\emptyset}_m$ (with, however, a possibly small convergence domain).

Nonetheless, in this case, the ideal closed-loop behavior (2.20) will no longer be obtained (even in the feasible case $\rho = m$) because of the approximation in evaluating $\widehat{\boldsymbol{J}}$ away from the desired pose. In addition to this shortcoming, different choices of $\widehat{\boldsymbol{\chi}}$ (including *rough* estimations of the true $\boldsymbol{\chi}_{\mathrm{d}}$ at the final pose) may also move (a subset of) the eigenvalues of (6.1) to the right-half complex plane, and thus yield an *unstable* closed-loop system (and failure of the servoing) even when starting *arbitrarily close* to the final pose. An illustrative example of such instability is demonstrated in Sect. 7.1.3, while further discussions about the robustness of visual servoing schemes against uncertainties on $\widehat{\boldsymbol{\chi}}$ can be found in [MMR10].

Special approximations such as $\widehat{\boldsymbol{\chi}} = \boldsymbol{\chi}_{\mathrm{d}}$ can then, at best, only guarantee local stability in a neighbourhood of $\boldsymbol{s}_{\mathrm{d}}$ and, in any case, require some prior knowledge on the scene (the value of $\boldsymbol{\chi}_{\mathrm{d}}$ must be obtained independently from the execution of the servoing task). Additionally, too rough estimations of the final $\boldsymbol{\chi}_{\mathrm{d}}$ (or other approximation choices for $\widehat{\boldsymbol{\chi}}$) may result in a poor, or even unstable, closed-loop behavior for the servoing causing, e.g., loss of feature tracking. In this context, the use of an incremental filtering (SfM) scheme able to generate a converging $\widehat{\boldsymbol{\chi}}(t) \to \boldsymbol{\chi}(t)$ from (ideally) any initial approximation $\widehat{\boldsymbol{\chi}}(t_0)$ can represent an effective alternative. Indeed, a SfM scheme can improve the servoing execution by approximating the ideal closed-loop behavior (2.20) also when *far* from the desired pose and without needing special assumptions/approximations of $\boldsymbol{\chi}$, since as $\widehat{\boldsymbol{\chi}}(t) \to \boldsymbol{\chi}(t)$ one obviously has $\widehat{\boldsymbol{J}} \to \boldsymbol{J}$ (see, e.g., [DOR08]).

We now note that, in this conceptual framework, the estimation of the 3-D structure $\boldsymbol{\chi}$ can then take place only during the *transient phase* of the servoing task, i.e., while the camera is in motion towards its goal location. Being this phase of limited duration, with the camera reaching a full stop at the end of the servoing, one should clearly aim at imposing the *fastest* possible convergence to the estimation error for a given camera displacement (from initial to final pose). As demonstrated in the previous chapters, other factors (e.g., estimation gains) being equal, the convergence rate of a SfM scheme is mainly affected by the particular trajectory followed by the camera w.r.t. the observed scene, with some trajectories being more informative/exciting than other ones. Therefore, the IBVS controller should select (online) the 'most informative' camera trajectory, among all the possible ones solving the visual task, for the sake of obtaining the fastest possible SfM convergence during the servoing transient. Chapter 4 described a strategy for generating such an informative camera motion policy. The rest of this chapter will, instead, detail how to exploit the results of Chapt. 4 for improving the performance of a IBVS controller by suitably taking advantage of (and possibly maximizing) the redundancy of the considered visual task.

6.2 Plugging active sensing in IBVS schemes

The execution of a servoing task can be naturally coupled with the (concurrent) optimization of the estimation of vector $\boldsymbol{\chi}$ by exploiting vector $\ddot{\boldsymbol{q}}_w$ in (2.32) for projecting any action aimed at maximizing, e.g., the smallest eigenvalue σ_1^2 of the PE matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ in the null-space of the visual task. The expression in (4.22) shows that optimization of $\sigma_1^2(t)$ requires an action at the *joint acceleration level*. In particular, since

$$\nabla_{\dot{\boldsymbol{q}}}\sigma_1^2 = \left(\nabla_{\boldsymbol{v}}\sigma_1^{2^T}\nabla_{\dot{\boldsymbol{q}}}\boldsymbol{v}^T\right)^T = \boldsymbol{J}_{\boldsymbol{v}}^T\boldsymbol{J}_{\sigma_v}^T \tag{6.2}$$

(where the relationship $\boldsymbol{v} = \boldsymbol{J}_{\boldsymbol{v}}(\boldsymbol{q})\dot{\boldsymbol{q}}$ was used), local maximization of σ_1^2 can be achieved by just following its positive gradient via a joint acceleration vector

$$\ddot{\boldsymbol{q}}_\sigma = k_\sigma\nabla_{\dot{\boldsymbol{q}}}\sigma_1^2 = k_\sigma\boldsymbol{J}_{\boldsymbol{v}}^T\boldsymbol{J}_{\sigma_v}^T, \qquad k_\sigma > 0. \tag{6.3}$$

Following the developments of Sect. 2.2.4.4 and being $\dot{\boldsymbol{e}} = \boldsymbol{J}\dot{\boldsymbol{q}}$ and, thus, $\ddot{\boldsymbol{e}} = \boldsymbol{J}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}$, the *second-order/acceleration level* counterpart of the classical law (2.32) for regulating the error vector $\boldsymbol{e}(t)$ to $\boldsymbol{\emptyset}_m$ is simply

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_e = \boldsymbol{J}^\dagger(-k_v\dot{\boldsymbol{e}} - k_p\boldsymbol{e} - \dot{\boldsymbol{J}}\dot{\boldsymbol{q}}) + \boldsymbol{P}\ddot{\boldsymbol{q}}_w \tag{6.4}$$

with $k_p > 0$ and $k_v > 0$. Therefore, by setting $\ddot{\boldsymbol{q}}_w = \ddot{\boldsymbol{q}}_\sigma$ in (6.4), one would obtain the desired maximization of σ_1^2 (i.e., of the convergence rate of the 3-D estimation error) concurrently to the execution of the main visual task. This straightforward strategy, although appealing for its simplicity, is unfortunately not viable in most practical situations because of the structural lack of enough *redundancy* for implementing action (6.3) (or any equivalent one) in the null-space of the main visual task in (6.4). This important limitation is illustrated by the following Proposition.

Proposition 6.1. *Assume* $\text{rank}(\boldsymbol{L}_{\boldsymbol{s}}) = 6$*, that is, the chosen set of visual features* $\boldsymbol{s}$ *allows controlling the full* 6*-DOFs camera pose. Then, for any scalar function* $w(\boldsymbol{v}(\dot{\boldsymbol{q}}))$*, one has* $\boldsymbol{P}\nabla_{\dot{\boldsymbol{q}}}w = \boldsymbol{\emptyset}_n$*.*

Proof. We first note that, thanks to the assumption $\text{rank}(\boldsymbol{L}_{\boldsymbol{s}}) = 6$ (which implies $m \geq 6$ and, thus, full column-rank for $\boldsymbol{L}_{\boldsymbol{s}}$), one has $\ker(\boldsymbol{L}_{\boldsymbol{s}}\boldsymbol{J}_{\mathcal{C}}) = \ker(\boldsymbol{J}_{\mathcal{C}})$. Being $\boldsymbol{P}$ the orthogonal projector on $\ker(\boldsymbol{J})$, the following then holds

$$\boldsymbol{P}\boldsymbol{J}^T = \boldsymbol{P}\boldsymbol{J}_{\mathcal{C}}^T\boldsymbol{L}_{\boldsymbol{s}}^T = \boldsymbol{P}\boldsymbol{J}_{\mathcal{C}}^T = \boldsymbol{\emptyset}_n.$$

On the other hand, $\nabla_{\dot{\boldsymbol{q}}}w = \boldsymbol{J}_{\boldsymbol{v}}^T\nabla_{\dot{\boldsymbol{q}}}w^T$. Since $\boldsymbol{J}_{\boldsymbol{v}}^T$ belongs to the range space of $\boldsymbol{J}_{\mathcal{C}}^T$ (see (2.29)), it follows that $\boldsymbol{P}\nabla_{\dot{\boldsymbol{q}}}w = \boldsymbol{\emptyset}_n$ as claimed. □

Proposition 6.1 formalizes an intuitive consideration: if the feature set $\boldsymbol{s}$ is rich enough to constrain all the camera DOFs, then no optimization of the camera linear velocity $\boldsymbol{v}$ can be performed via the null-space projector operator $\boldsymbol{P}$ since (obviously) the camera motion is already fully specified by the chosen visual task. Being classical VS tasks *purposely* built upon a feature set $\boldsymbol{s}$ able to ensure full (visual) control over the free camera DOFs, Prop. 6.1 clearly prevents any optimization meant to affect the value of σ_1^2 during the servoing execution[1]. This fundamental limitation motivates the development of the alternative strategy presented in the following section.

6.2.1 Second-order VS using a Large Projection Operator

An alternative control strategy that is able to circumvent the limitations imposed by Prop. 6.1 can be devised by suitably exploiting the redundancy framework originally proposed in [MC10]. In this work it is shown how regulation of the full visual error vector $\boldsymbol{e}$ (a m-dimensional task) can be replaced by the regulation of its norm $\|\boldsymbol{e}\|$ (a 1-dimensional task). This manipulation results in a null-space of (maximal) dimension $n-1$ available for additional optimizations including, in our case, those prevented by Prop. 6.1.

The machinery presented in [MC10] can be exploited as follows: by letting $\nu = \|\boldsymbol{e}\|$, we have

$$\dot{\nu} = \frac{\boldsymbol{e}^T \boldsymbol{J}}{\|\boldsymbol{e}\|}\dot{\boldsymbol{q}} = \boldsymbol{J}_\nu \dot{\boldsymbol{q}}, \qquad \boldsymbol{J}_\nu \in \mathbb{R}^{1\times n},$$

and, at second-order,

$$\ddot{\nu} = \boldsymbol{J}_\nu \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_\nu \dot{\boldsymbol{q}}.$$

Regulation of $\nu(t) \to 0$ can then be achieved by the following control law

$$\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}_\nu = \boldsymbol{J}_\nu^\dagger(-k_v\dot{\nu} - k_p\nu - \dot{\boldsymbol{J}}_\nu\dot{\boldsymbol{q}}) + \boldsymbol{P}_\nu \ddot{\boldsymbol{q}}_w, \tag{6.5}$$

with $k_p > 0$, $k_v > 0$, $\boldsymbol{J}_\nu^\dagger = \frac{\|\boldsymbol{e}\|}{\boldsymbol{e}^T\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{e}}\boldsymbol{J}^T\boldsymbol{e}$ and $\boldsymbol{P}_\nu = \boldsymbol{I}_n - \frac{\boldsymbol{J}^T\boldsymbol{e}\boldsymbol{e}^T\boldsymbol{J}}{\boldsymbol{e}^T\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{e}}$ being the null-space projection operator of the error norm with rank $n-1$ (see [MC10]).

By implementing controller (6.5) in place of (6.4) one can thus still obtain regulation of the whole visual task error since, obviously, $\nu(t) = \|\boldsymbol{e}(t)\| \to 0$ implies $\boldsymbol{e}(t) \to \boldsymbol{\emptyset}_m$. However, contrarily to (6.4), the new null-space projector $\boldsymbol{P}_\nu$ allows implementing a broader range of optimization actions including (6.3) or equivalent ones. In this respect, next Sect. 6.2.2 addresses the design of a suitable cost

[1]Note that, if $n > m \geq 6$, there would still be room in (6.4) for implementing an action in the null-space of the main task. However, presence of this redundancy would be useless for the sake of optimizing the camera trajectory as it would only result in an internal reconfiguration of the robotic arm without effects on the camera motion.

function $\mathcal{V}(\boldsymbol{v}(\dot{\boldsymbol{q}}))$ able to trade off maximization of σ_1^2 with the boundedness of the camera/robot self-motions in the null-space of the main visual task.

We also note, however, that controller (6.5) suffers from some shortcomings: in particular, the Jacobian $\boldsymbol{J}_\nu$ is singular for $\|\boldsymbol{e}\| = 0$, while the projection matrix $\boldsymbol{P}_\nu$ and the pseudoinverse $\boldsymbol{J}_\nu^\dagger$ are not well-defined for $\|\boldsymbol{e}\| = 0$ and for $\boldsymbol{e} \in \ker(\boldsymbol{J}^T)$. As discussed in [MC10], the singularity at $\|\boldsymbol{e}\| = 0$ can be avoided by switching from controller (6.5) to the classical law (6.4) when 'close enough' to convergence (i.e., when $\|\boldsymbol{e}\|$ becomes sufficiently small). However, since the 'first-order' switching strategy proposed in [MC10] is not directly transposable to the second-order case, Sect. 6.2.3 details a suitable 'second-order' strategy able to guarantee a proper switching from (6.5) to the classical law (6.4).

Remark 6.1. *We wish to emphasize the different roles of the two singularities* $\|\boldsymbol{e}\| = 0$ *and* $\boldsymbol{e} \in \ker(\boldsymbol{J}^T)$ *affecting controller* (6.5)*. The singularity occurring at* $\|\boldsymbol{e}\| = 0$ *is a consequence of the choice of controlling the norm of the error vector and, thus, it does not affect other schemes such as the classical one* (6.4) *(therefore one can safely switch to* (6.4) *when* $\|\boldsymbol{e}\| \to 0$*). The other singularity* $\boldsymbol{e} \in \ker(\boldsymbol{J}^T)$ *corresponds, instead, to a* local minimum *for the servoing itself since, if* $\boldsymbol{e} \in \ker(\boldsymbol{J}^T)$*, no camera motion can* instantaneously *realize the task. Therefore,* any *'local' control action (including* (6.4) *and* (6.5)*) would be equally affected by the condition* $\boldsymbol{e} \in \ker(\boldsymbol{J}^T)$*, and no simple switching strategy could be employed in this case (local minima escaping strategies, such as random walks or global optimizations, are out of the scope of this thesis).*

6.2.2 Optimization of the 3-D Reconstruction

Being the convergence rate of the 3-D estimation error $\widetilde{\boldsymbol{\chi}}(t) = \widehat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)$ determined by the eigenvalue σ_1^2, a straightforward choice is to attempt maximization of a cost function in the form

$$\mathcal{V}(\dot{\boldsymbol{q}}) = k_\sigma \sigma_1^2(\boldsymbol{v}(\dot{\boldsymbol{q}})), \qquad k_\sigma > 0, \tag{6.6}$$

in a similar way to what done in Part II. This would result in the joint acceleration $\ddot{\boldsymbol{q}}_\sigma$ in (6.3) to be plugged in vector $\ddot{\boldsymbol{q}}_w$ in (6.5). However, a cost function such as (6.6) is unbounded from above w.r.t. $\dot{\boldsymbol{q}}$ ($\sup_{\|\dot{\boldsymbol{q}}\|} \mathcal{V}(\dot{\boldsymbol{q}}) = \infty$) since, in general, the faster the camera motion the larger σ_1^2. Indeed, on the one hand, $\sigma_1^2 \propto \|\boldsymbol{v}\|^2$ (see (4.31) for the point feature case) and, on the other hand, $\|\boldsymbol{v}\|^2 \propto \|\dot{\boldsymbol{q}}\|^2$ being $\|\boldsymbol{J}_{\boldsymbol{v}}\|$ in (2.29) a bounded quantity. Therefore, maximization of (6.6) via (6.3) would simply make the camera velocity to grow *unbounded* in the null-space of the main servoing task.

In order to cope with this issue, it is then necessary to modify (6.6) for allowing existence of a finite upper bound w.r.t. $\|\dot{\boldsymbol{q}}\|$. The addition of a classical quadratic

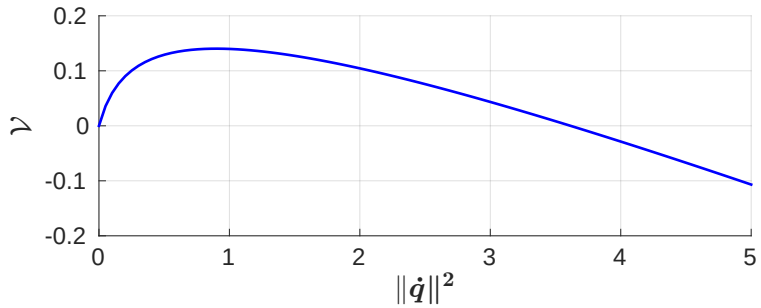


Figure 6.1 – **A representative graph of the cost function used to calculate the IBVS secondary task** in (6.8) plotted against $\|\dot{\boldsymbol{q}}\|$ for $k_\sigma = 1$, $k_d = 0.2$, $\gamma = 0.1$, and assuming $\sigma_1^2 = \|\dot{\boldsymbol{q}}\|^2$. Note the presence of a finite upper bound for $\mathcal{V}(\dot{\boldsymbol{q}})$ as desired.

penalty (damping) term in the form

$$\mathcal{V}(\dot{\boldsymbol{q}}) = k_\sigma \sigma_1^2(\dot{\boldsymbol{q}}) - \frac{k_d}{2}\|\dot{\boldsymbol{q}}\|^2, \qquad k_d > 0, \tag{6.7}$$

is, nevertheless, still not a valid solution: as both terms in (6.7) are proportional to $\|\dot{\boldsymbol{q}}\|^2$, existence of a finite (non-zero) upper bound would depend, case by case, by the particular combination of gains and of the camera trajectory. Because of these considerations, we then opted for the following cost function

$$\mathcal{V}(\dot{\boldsymbol{q}}) = k_\sigma \gamma \log\left(\frac{\gamma + \sigma_1^2(\dot{\boldsymbol{q}})}{\gamma}\right) - \frac{k_d}{2}\|\dot{\boldsymbol{q}}\|^2, \qquad \gamma > 0, \tag{6.8}$$

for which a representative graph is depicted in Fig. 6.1. This choice is motivated by considering that, for large velocities ($\|\dot{\boldsymbol{q}}\| \to \infty$), the damping term $\frac{k_d}{2}\|\dot{\boldsymbol{q}}\|^2$ will always be dominant w.r.t. the first term regardless of the choice of gains or of the particular camera trajectory. Indeed, since $\sigma_1^2 \propto \|\dot{\boldsymbol{q}}\|^2$ and since $\log(x) = o(g(x))$ for any polynomial function $g(x)$, it follows that

$$\lim_{\|\dot{\boldsymbol{q}}\| \to \infty} \mathcal{V}(\dot{\boldsymbol{q}}) = -\infty.$$

Therefore, maximization of $\mathcal{V}(\dot{\boldsymbol{q}})$ in (6.8) will guarantee maximization of σ_1^2 with, embedded, a bound on the maximum allowed joint velocity $\|\dot{\boldsymbol{q}}\|$. The location of this maximum and of the slope of (6.8) for $\|\dot{\boldsymbol{q}}\| \to 0$ and $\|\dot{\boldsymbol{q}}\| \to \infty$ is determined by the gains k_σ, γ and k_d. Maximization of $\mathcal{V}(\dot{\boldsymbol{q}})$ is then obtained by plugging in vector $\ddot{\boldsymbol{q}}_w$ in (6.5) the following joint acceleration vector

$$\ddot{\boldsymbol{q}}_{\mathcal{V}} = \nabla_{\dot{\boldsymbol{q}}}\mathcal{V} = \frac{k_\sigma \gamma}{\gamma + \sigma_1^2}\nabla_{\dot{\boldsymbol{q}}}\sigma_1^2 - k_d \dot{\boldsymbol{q}}^T. \tag{6.9}$$

6.2.3 Second-order Switching Strategy

We now discuss a second-order switching strategy meant to avoid the singularity of controller (6.5) when $\nu(t) = \|\boldsymbol{e}(t)\| \to 0$. We start noting that, in closed-loop, controller $\ddot{\boldsymbol{q}}_\nu$ in (6.5) imposes the following second-order dynamics to the error norm

$$\ddot{\nu} + k_v\dot{\nu} + k_p\nu = 0. \tag{6.10}$$

Define $\nu_{\|\boldsymbol{e}\|}(t)$ as the solution of (6.10) for a given initial condition $(\nu(t_0), \dot{\nu}(t_0))$: $\nu_{\|\boldsymbol{e}\|}(t)$ thus represents the 'ideal' evolution of the error norm, that is, the behavior one would obtain if controller (6.5) could be implemented $\forall t \geq t_0$.

Let now $t_1 > t_0$ be the time at which the switch from controller (6.5) to the classical law $\ddot{\boldsymbol{q}}_{\boldsymbol{e}}$ in (6.4) occurs (e.g., triggered by some threshold on $\|\boldsymbol{e}\|$ as proposed in [MC10]). For $t \geq t_1$, controller $\ddot{\boldsymbol{q}}_{\boldsymbol{e}}$, under the assumption[2] $\text{rank}(\boldsymbol{J}) = \rho = m$, yields in closed-loop

$$\ddot{\boldsymbol{e}} + k_v\dot{\boldsymbol{e}} + k_p\boldsymbol{e} = 0. \tag{6.11}$$

Let $\boldsymbol{e}^*(t)$ be the solution of (6.11) with initial conditions $(\boldsymbol{e}(t_1), \dot{\boldsymbol{e}}(t_1))$, and let $\nu^*(t) = \|\boldsymbol{e}^*(t)\|$ be the corresponding behavior of the error norm. Ideally, one would like to have

$$\nu^*(t) \equiv \nu_{\|\boldsymbol{e}\|}(t), \quad \forall t \geq t_1. \tag{6.12}$$

In other words, the behavior of the error norm should not be affected by the control switch at time t_1, but $\nu^*(t)$ (obtained from (6.11)) should exactly match the 'ideal' evolution $\nu_{\|\boldsymbol{e}\|}(t)$ generated by (6.10) as if no switch had taken place.

While condition (6.12) is easily satisfied at first-order [MC10], this is not necessarily the case at the second-order level. Indeed, when moving to the second-order the following result can be shown (see Appendix A.5.1)

Proposition 6.2. *For the second-order error dynamics (6.10–6.11), condition* (6.12) *holds iff, at the switching time t_1, vectors $\boldsymbol{e}(t_1)$ and $\dot{\boldsymbol{e}}(t_1)$ are* parallel.

It is then necessary to introduce an intermediate phase, before the switch, during which any component of $\dot{\boldsymbol{e}}$ orthogonal to $\boldsymbol{e}$ is made negligible.

To this end, let

$$\boldsymbol{P}_e = \left(\boldsymbol{I}_m - \frac{\boldsymbol{e}\boldsymbol{e}^T}{\boldsymbol{e}^T\boldsymbol{e}}\right) \in \mathbb{R}^{m\times m}$$

be the null-space projector spanning the $(m-1)$-dimensional space orthogonal to vector $\boldsymbol{e}$. Let also

$$\boldsymbol{\delta} = \boldsymbol{P}_e\dot{\boldsymbol{e}} = \boldsymbol{P}_e\boldsymbol{J}\dot{\boldsymbol{q}}. \tag{6.13}$$

[2] If $\rho < m$, as in the case studies reported in Chapt. 7, the ideal behavior (6.11) can, in general, only be approximately imposed, see Sect. 2.3.2.

The scalar quantity $\boldsymbol{\delta}^T\boldsymbol{\delta} \geq 0$ provides a measure of the misalignment among the directions of vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ ($\boldsymbol{\delta}^T\boldsymbol{\delta} = 0$ iff $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ are parallel, $\forall \boldsymbol{e} \neq \boldsymbol{\emptyset}_m$, $\dot{\boldsymbol{e}} \neq \boldsymbol{\emptyset}_m$). One can then minimize $\boldsymbol{\delta}^T\boldsymbol{\delta}$ compatibly with the main task (regulation of the error norm) by choosing vector $\ddot{\boldsymbol{q}}_w$ in (6.5) as

$$\ddot{\boldsymbol{q}}_{\boldsymbol{\delta}} = -\frac{k_\delta}{2}\nabla_{\dot{\boldsymbol{q}}}(\boldsymbol{\delta}^T\boldsymbol{\delta}) = -k_\delta \boldsymbol{J}^T\boldsymbol{P}_{\boldsymbol{e}}\boldsymbol{J}\dot{\boldsymbol{q}} = -k_\delta \boldsymbol{J}_\delta^T \tag{6.14}$$

where the properties $\boldsymbol{P}_{\boldsymbol{e}} = \boldsymbol{P}_{\boldsymbol{e}}^T = \boldsymbol{P}_{\boldsymbol{e}}\boldsymbol{P}_{\boldsymbol{e}}$ were used.

A possible switching strategy, shown in the flowchart in Fig. 6.2, then consists of the following three different control phases:

phase 1): apply the norm controller $\ddot{\boldsymbol{q}}_\nu$ given in (6.5) with the null-space vector $\ddot{\boldsymbol{q}}_w = \ddot{\boldsymbol{q}}_{\mathcal{V}}$ as defined in (6.9) as long as $\nu(t) \geq \nu_T$, with $\nu_T > 0$ being a suitable threshold on the error norm. During this phase, the error norm will be governed by the closed-loop dynamics (6.10) and the convergence rate in estimating $\widehat{\boldsymbol{\chi}}$ will be maximized thanks to (6.9);

phase 2): when $\nu(t) = \nu_T$, keep applying controller $\ddot{\boldsymbol{q}}_\nu$ but replace (6.9) with (6.14) in vector $\ddot{\boldsymbol{q}}_w$. Stay in this phase as long as some terminal condition on the minimization of $\boldsymbol{\delta}^T\boldsymbol{\delta}$ is reached. In our case, we opted for a threshold δ_T on the minimum norm of vector $\|\boldsymbol{P}_\nu\boldsymbol{J}_\delta\|$ as an indication of when no further minimization of $\boldsymbol{\delta}^T\boldsymbol{\delta}$ is possible in the null-space of the error norm. Note also that, during this second phase, $\nu(t)$ keeps being governed by the closed-loop dynamics (6.10) since $\ddot{\boldsymbol{q}}_w$ acts in the null-space of the error norm (i.e., no distorting effect is produced on the behavior of $\nu(t)$ by the change in $\ddot{\boldsymbol{q}}_w$);

phase 3): when $\boldsymbol{\delta}^T\boldsymbol{\delta}$ has been minimized, switch to the classical controller $\ddot{\boldsymbol{q}}_{\boldsymbol{e}}$ given in (6.4) until completion of the task. The minimization of $\boldsymbol{\delta}^T\boldsymbol{\delta}$ will ensure a smooth switch as per Prop. 6.2.

We finally note that this strategy could cause a discontinuity in the commanded *acceleration* $\ddot{\boldsymbol{q}}$ when moving from phase 1) to phase 2) because of the instantaneous change of vector $\ddot{\boldsymbol{q}}_w$ from (6.9) to (6.14). If needed, this discontinuity could be easily dealt with by resorting to a smoothing procedure as that proposed in [MC10]. As for the switch from phase 2) to phase 3), discontinuities in the commanded $\ddot{\boldsymbol{q}}$ are, instead, avoided thanks to the alignment among vector $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ occurring during phase 2).

Before concluding this section we remark that the proposed scheme (active SfM (3.11) coupled to the second-order VS (6.4–6.5)), null-space terms (6.9–6.14) and associated switching strategy of Fig. 6.2) only requires, as measured quantities,

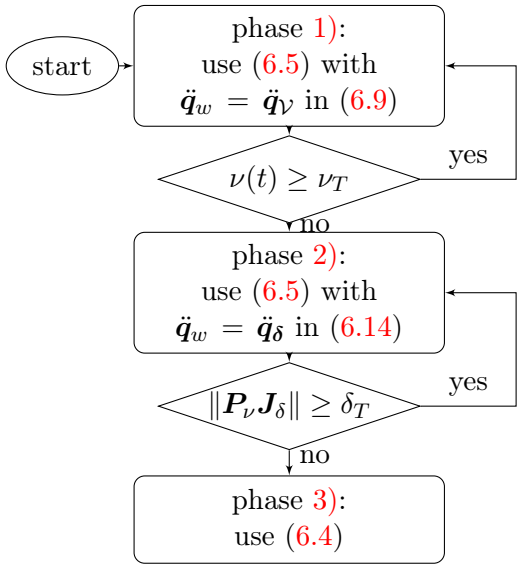


Figure 6.2 – **Flowchart representation of the basic switching control strategy**.

the visual features $\boldsymbol{s}$, the robot joint configuration vector $\boldsymbol{q}$ and the joint velocities $\dot{\boldsymbol{q}}$ (in addition to the usual assumption of known intrinsic and camera-to-robot parameters). Indeed from the estimated $\widehat{\boldsymbol{\chi}}$, a (possibly approximated) evaluation of *all* the other quantities entering the various steps of the second-order control strategy (e.g., the task Jacobians $\boldsymbol{J}$, $\boldsymbol{J}_\nu$ and their time derivatives $\dot{\boldsymbol{J}}$, $\dot{\boldsymbol{J}}_\nu$) can be obtained from ($\boldsymbol{s}$, $\widehat{\boldsymbol{\chi}}$, $\boldsymbol{q}$) and $\dot{\boldsymbol{q}}$ (the only 'velocity' information actually needed). We also note that the level of approximation is clearly a monotonic function of $\|\widehat{\boldsymbol{\chi}} - \boldsymbol{\chi}\|$ (i.e., the uncertainty in knowing $\boldsymbol{\chi}$): thus, all the previous quantities will asymptotically match their real values as the estimation error $\widetilde{\boldsymbol{\chi}}(t) = \widehat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)$ converges to zero (thus, the faster the convergence of $\widetilde{\boldsymbol{\chi}}(t)$, the sooner the ideal closed-loop behaviors (6.10–6.11) will be realized).

We finally remark that, due to the nonlinear nature of the estimation and servoing schemes, stability of each individual block does not imply, in general, stability of their composition since one cannot invoke the separation principle only valid for linear time-invariant systems[3]. Nevertheless, the experimental results reported in Chapt. 7 show a promising level of robustness in this sense.

6.3 Adaptive switching

We now propose a further improvement to the strategy detailed in Sect. 6.2. The goal is to introduce an *automatic* mechanism for adaptively *activating/deactivat-*

[3]Analogous theoretical difficulties affect any (non-trivial) robotic application in which an estimation step is plugged into a control loop (e.g., whenever exploiting an EKF for feeding online a motion controller with the estimated state).

ing optimization of the 3-D structure estimation as a function of the accuracy in estimating $\boldsymbol{\chi}(t)$. This modification is motivated by the following considerations w.r.t. the flowchart of Fig. 6.2:

- the optimization of σ^2 is active during the whole phase 1), i.e., as long as the error norm is larger than some predefined threshold (i.e., $\nu(t) \geq \nu_T$). However, this is obtained at the expense of a possible distortion of the camera trajectory as it will be shown in, e.g., Figs. 7.1(d) and 7.3(d) which depict the camera spiralling motion due to action (7.1) while approaching the final pose. Clearly, a more efficient strategy would implement (7.1) *only* when strictly needed, e.g., as long as the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\| = \|\widehat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)\|$ is larger than some minimum threshold;

- similarly, once in phases 2) and 3), the flowchart of Fig. 6.2 does not allow any reactivation of the optimization of σ^2. On the other hand, a reactivation could be necessary in case of unforeseen events such as, e.g., an unpredictable motion of the target that would make the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\|$ to abruptly increase.

We then now detail a modification of the previous strategy of Sect. 6.2 for addressing these issues. To this end, we first introduce a way to quantify the uncertainty level in the estimation of the unknown vector $\boldsymbol{\chi}(t)$. Since the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ is (obviously) not directly measurable, we consider instead the following *measurable* quantity

$$E(t) = \frac{1}{T}\int_{t-T}^{t} \widetilde{\boldsymbol{s}}^T(\tau)\widetilde{\boldsymbol{s}}(\tau)\mathrm{d}\tau, \qquad T \geq \epsilon > 0, \tag{6.15}$$

where T represents the integration window and $\widetilde{\boldsymbol{s}} = \widehat{\boldsymbol{s}} - \boldsymbol{s}$ is the feedback term driving observer (3.11). Indeed, as discussed in Appendix A.5.2, $E(t)$ plays a role comparable with the unmeasurable $\widetilde{\boldsymbol{\chi}}(t)$, that is, it provides a measure of the uncertainty of the estimated $\widehat{\boldsymbol{\chi}}$ vs. the actual $\boldsymbol{\chi}$. In particular, provided the camera trajectory is sufficiently exciting (i.e., $\sigma_1^2(t) > 0$ during motion), $E(t) \equiv 0$ iff $\|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$ (i.e., the estimation has converged) and $E(t) > 0$ otherwise . One can then leverage knowledge of $E(t)$ for, e.g.,(i) automatically switching from phase 1) to phase 2) when the estimation error becomes smaller than a desired threshold, (ii) automatically switching from phase 3) back to phase 1) when the estimation error grows larger than a desired threshold, and (iii) adaptively weighting the first term in action (6.9) for smoothly activating/deactivating the optimization of σ_1^2.

Let then $0 \leq \underline{E} < \overline{E}$ be a fixed minimum/maximum threshold for $E(t)$ and define

$$k_E(E) : [\underline{E}, \overline{E}] \mapsto [0, 1]$$

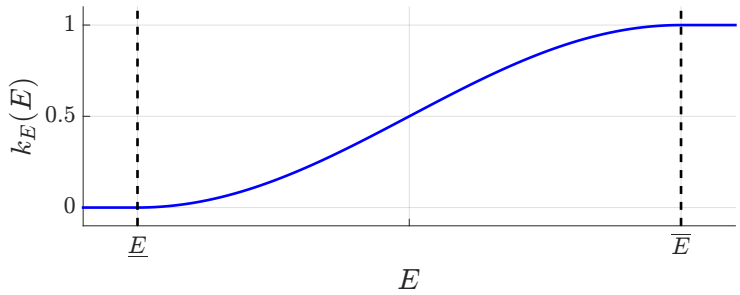


Figure 6.3 – **Qualitative plot of the adaptive gain** $k_E(E)$ for $\underline{E} = 1 \times 10^{-5}$ and $\overline{E} = 1 \times 10^{-4}$ used to tune the effect of the active camera velocity optimization depending on the current status of the structure from motion estimation.

as a monotonically increasing smooth map from the interval $[\underline{E}, \overline{E}]$ to the interval $[0, 1]$. For instance, one could take

$$k_E(E) = \begin{cases} 0 & \text{if } E \leq \underline{E} \\ \frac{1}{2} - \frac{1}{2}\cos\left(\pi \frac{E - \underline{E}}{\overline{E} - \underline{E}}\right) & \text{if } \underline{E} < E < \overline{E} \\ 1 & \text{if } E \geq \overline{E} \end{cases} \tag{6.16}$$

for which the graph for $\underline{E} = 1 \times 10^{-5}$ and $\overline{E} = 1 \times 10^{-4}$ is depicted in Fig. 6.3. Function $k_E(E)$ can be exploited for suitably weighting the optimization of σ_1^2: a simple but effective possibility is to just modify the cost function (6.8) as

$$\mathcal{V}_E(\dot{\boldsymbol{q}}, E) = k_\sigma k_E(E)\gamma \log\left(\frac{\gamma + \sigma_1^2(\dot{\boldsymbol{q}})}{\gamma}\right) - \frac{k_d}{2}\|\dot{\boldsymbol{q}}\|^2, \tag{6.17}$$

resulting in the new optimization action

$$\ddot{\boldsymbol{q}}_{\mathcal{V}_E} = \nabla_{\dot{\boldsymbol{q}}}\mathcal{V}_E = \frac{k_\sigma k_E(E)\gamma}{\gamma + \sigma_1^2}\boldsymbol{J}_{\boldsymbol{v}}^T \boldsymbol{J}_{\sigma_v}^T - k_d\dot{\boldsymbol{q}} \tag{6.18}$$

to be plugged in vector $\ddot{\boldsymbol{q}}_w$ in (6.5). This modification clearly grants a *smooth modulation* of the first term in (6.18) from a full activation in case of large estimation inaccuracies ($k_E(E) = 1$ for $E \geq \overline{E}$), to a full deactivation in case of small estimation inaccuracies ($k_E(E) = 0$ for $E \leq \underline{E}$).

Exploiting $E(t)$ and the modified optimization action (6.18), we then propose the new (adaptive) switching strategy depicted in Fig. 6.4. This new strategy consists of the same three phases of Sect. 6.2.3, but it now exploits knowledge of $E(t)$ for implementing an improved switching policy among the various phases:

- at the beginning of the servoing task start in phase 3) (instead of phase 1)), thus implementing the classical controller (6.4). Remain in this phase while $E(t) \leq \underline{E}$ or $\nu(t) \leq \nu_T$, otherwise switch to phase 1);

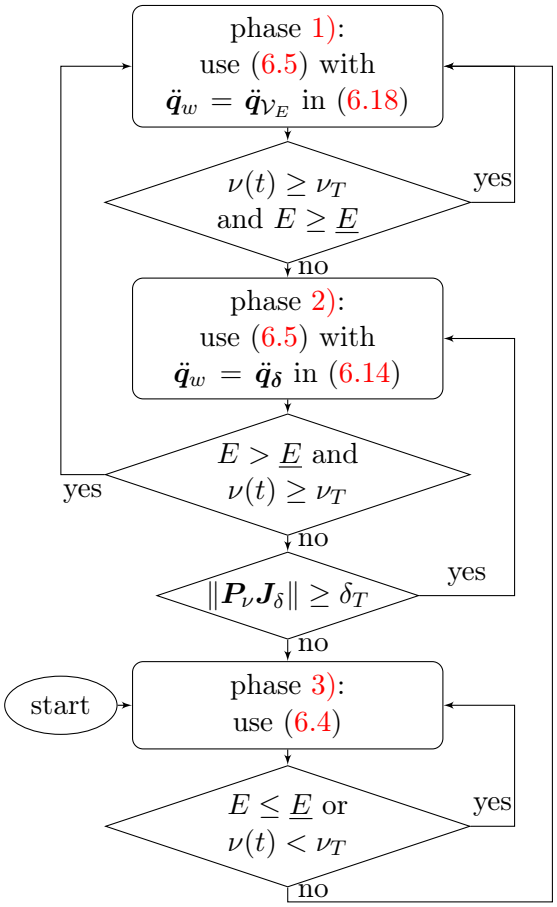


Figure 6.4 – **Flowchart representation of the adaptive switching strategy** exploiting the measurable error energy for triggering changes of status.

- when in phase 1), implement the norm controller (6.5) with the new optimization action (6.18). Stay in phase 1) as long as $E(t) \geq \underline{E}$ and $\nu(t) \geq \nu_T$, otherwise switch to phase 2);

- when in phase 2), implement the norm controller (6.5) with the optimization action (6.14). If $E(t) \geq \underline{E}$ and $\nu(t) \geq \nu_T$, switch back to phase 1), otherwise, if $\|\boldsymbol{P}_\nu \boldsymbol{J}_\delta\| \leq \delta_T$ (terminating condition for the alignment among vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$) switch to phase 3).

We highlight the following features of this new adaptive strategy: first of all, the initial (possible) switch from phase 3) to phase 1) is performed only if $E(t) \geq \underline{E}$ (the estimation error is large enough for justifying an optimization of the camera motion) *and* $\nu(t) \geq \nu_T$ (the visual error norm is large enough for preventing singularities in (6.5)). As illustration, two scenarios will typically trigger this switch:(i) a camera starting far enough from the desired pose and with a poor enough initial estimation $\widehat{\boldsymbol{\chi}}(t_0)$, or ($ii$) an unpredicted motion of the target object during the servoing task

that causes an increase in the error norm *and* in the estimation uncertainty. The experiments of the next Sects. 7.2 and 7.3 will indeed address these two practical cases. Furthermore, while in phase 1), the optimization of the camera motion will be performed only until either a good enough accuracy has been reached ($E(t) < \underline{E}$), or controller (6.5) is close to become singular ($\nu(t) < \nu_T$). The new switching condition $E(t) < \underline{E}$ will then help in minimizing the distortion of the camera trajectory by allowing a quick switch to phase 2) as soon as the estimation accuracy reaches a satisfactory level (see again the experiments in Sects. 7.2 and 7.3).

As a final step, we comment about the choice of the two thresholds $\underline{E}$ and $\overline{E}$ exploited for triggering the various switches and for modulating the activation of the optimization of σ_1^2 in (6.18). Assume the range of possible values of $E(t)$ during the camera motion can be lower/upper bounded as $0 \leq E_{min} \leq E(t) \leq E_{max}$. It would obviously be meaningful to choose $\underline{E}$ and $\overline{E}$ such that $E_{min} \leq \underline{E} < \overline{E} \leq E_{max}$ for properly tuning the adaptive switching strategy.

Concerning the lower bound E_{min}, being $E(t) \geq 0$, a straightforward choice would be $E_{min} = 0$. However, presence of measurement noise and other non-idealities can, in practice, prevent $E(t)$ to fall below some minimum value even after convergence of the estimation error (up to some residual noise). If needed, this minimum value can be, e.g., experimentally determined by simply averaging, across a sufficient number of different camera trajectories, the (steady-state) value reached by $E(t)$ once the estimation has converged[4]. As for E_{max}, any (arbitrarily large) positive value would in principle be a valid choice since, the larger the initial approximation error $\|\widetilde{\boldsymbol{\chi}}(t_0)\| = \|\widehat{\boldsymbol{\chi}}(t_0) - \boldsymbol{\chi}(t_0)\|$, the wider the possible range of $E(t)$. It is, however, possible to show that, exploiting the properties of observer (3.11) and, in particular, its port-Hamiltonian nature, the following bound holds (see Appendix A.5.2)

$$E(t) \leq \frac{\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2}{\alpha}. \tag{6.19}$$

Therefore, if an upper bound $\|\widetilde{\boldsymbol{\chi}}(t_0)\| \leq z_{max}$ on the initial estimation error can be assumed (as in most practical situations), one can exploit (6.19) and set

$$E_{max} = \frac{z_{max}^2}{\alpha}. \tag{6.20}$$

6.4 Conclusions

In this chapter we investigated how to couple the execution of a VS task with an active SfM strategy meant to optimize the reconstruction of the 3-D scene structure. This result was achieved by projecting the active SfM action within the null-space

[4]This is indeed the solution adopted for the experiments in Sects. 7.2 and 7.3.

of the considered IBVS task. In general IBVS, however, tasks are intentionally constructed in such a way that all the camera DOFs are constrained and, therefore, they typically lack any additional redundancy to exploit for optimizing the estimation convergence using, e.g., some of the techniques proposed in Chapt. 4. To cope with this problem, we proposed to suitably extend to the second order the framework originally introduced in [MC10] for granting the needed redundancy for an effective optimization of the camera motion by controlling the task error norm instead of the task itself. A thorough analysis of the closed-loop convergence performance, including a switching strategy meant to avoid some structural singularities of [MC10], was also provided. As an additional contribution, we detailed an adaptive strategy able to *automatically* activate/deactivate and, more in general, tune the optimization of the SfM convergence as a function of the current accuracy of the estimated 3-D structure. This was obtained by exploiting the prediction error $\widetilde{\boldsymbol{s}}$ (in particular the average of its norm over a finite time interval) as an indication of the current convergence status of the estimation relying, for this, the prediction error and the pH nature of the error system dynamics.

The next Chapt. 7 will report a thorough experimental validation of all of the theoretical results obtained in this chapter.

Chapter 7

Experimental results of coupling active SfM and IBVS

This section reports the results of several experiments meant to validate the approach proposed in Chapt. 6 for coupling the execution of a VS task with the concurrent optimization of the 3-D structure estimation. All experiments were run by making use of the same experimental setup described in Chapt. 5.

In the reported experiments, we considered, as visual task, the regulation of $N = 4$ point features $\boldsymbol{\pi}_i$ with, thus, $\boldsymbol{s} = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N) \in \mathbb{R}^m$, and $\boldsymbol{L_s} = (\boldsymbol{L}_{\boldsymbol{s}_1}, \ldots, \boldsymbol{L}_{\boldsymbol{s}_N}) \in \mathbb{R}^{m\times 6}$, $m = 8$, with $\boldsymbol{L}_{\boldsymbol{s}_i}$ being the standard 2×6 interaction matrix for a point feature (2.10). As for vector $\boldsymbol{\chi}$, we then have $\boldsymbol{\chi} = (\chi_1, \ldots, \chi_N) \in \mathbb{R}^p$, $p = 4$, where $\chi_i = 1/Z_i$ as explained in Sect. 4.5.1. For most of the experiments, the tracked points were black non-coplanar dots belonging to the surface of a white cube and with relative 3-D positions ${}^O\boldsymbol{p}_i$ (w.r.t. the center of the cube)

$$
\begin{aligned}
&{}^O\boldsymbol{p}_1 = [-0.03,\ -0.03,\ -0.0575]^T, {}^O\boldsymbol{p}_2 = [-0.03,\ 0.03,\ -0.0575]^T,\\
&{}^O\boldsymbol{p}_3 = [0.03,\ 0.03,\ -0.0575]^T, {}^O\boldsymbol{p}_4 = [0.03,\ 0.0575,\ 0.03]^T.
\end{aligned}
$$

Knowledge of these 3-D coordinates was exploited in a standard pose estimation algorithm for obtaining the ground truth value of $\boldsymbol{\chi}(t)$ from the known object model and the measured features $\boldsymbol{s}(t)$.

Because of the high contrast between black dots and white cube surface, the segmentation and tracking of the N points could be easily obtained at video-rate via the blob tracker available in ViSP. Besides easing the image processing step, this experimental setting made it also possible the reproduction of (practically) identical initial experimental conditions across the several trials illustrated in the following sections. The results reported in Sect. 7.3 will instead resort to a KLT tracker for segmenting and tracking a generic set of points lying on a much less structured

target object in order to show the viability of our method also in more realistic situations.

As for what concerns the optimization of the 3-D reconstruction, we note that each feature point is characterized by its own (independent) eigenvalue $\sigma_{1,i}^2$. Optimization of the estimation of the whole vector $\boldsymbol{\chi}$ (i.e., of the depth of all points) was then obtained by considering the average of the N eigenvalues

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}\sigma_{1,i}^2$$

as quantity to be optimized. This obviously corresponds to using the A-optimality condition introduced in (4.24), normalized w.r.t. the number of points. Being, obviously,

$$\nabla_{\dot{\boldsymbol{q}}}\sigma^2 = \frac{1}{N}\boldsymbol{J}_{\boldsymbol{v}}^T\sum_{i=1}^{N}\boldsymbol{J}_{\sigma_{v_i}}^T$$

the acceleration command (6.9) was then simply replaced by

$$\ddot{\boldsymbol{q}}_{\mathcal{V}} = \frac{1}{N}\frac{k_\sigma\gamma}{\gamma+\sigma^2}\boldsymbol{J}_{\boldsymbol{v}}^T\sum_{i=1}^{N}\boldsymbol{J}_{\sigma_{v_i}}^T - k_d\dot{\boldsymbol{q}} \tag{7.1}$$

during phase 1) of all the following experiments.

The rest of the chapter is organized as follows: Sect. 7.1 is meant to validate the basic machinery, described in Sect. 6.2, for coupling IBVS execution and optimization of the 3-D structure estimation. Subsequently, in Sect. 7.2, we will report the results of some experiments using the strategy detailed in Sect. 6.3 for allowing a smooth activation/deactivation of the camera trajectory optimization as a function of the current estimation accuracy. This extension is then further validated in Sect. 7.3 in a more typical experimental scenario. Section 7.4 concludes the chapter with some final considerations.

Videos for the experiments shown in this chapter can be downloaded from the web page associated with the publication [7] at `http://ieeexplore.ieee.org`. Additional videos are also available at `https://www.youtube.com/watch?v=IYX6C2qYInA` and `https://www.youtube.com/watch?v=kgoWUu-9fhs`.

7.1 Using a basic switching strategy

7.1.1 First set of experiments

In this first set of experiments we aim at illustrating the benefits arising from the coupling between the execution of a VS task and the concurrent active optimization of the 3-D structure estimation. To this end, we consider the following four different cases, all starting from the same initial conditions:

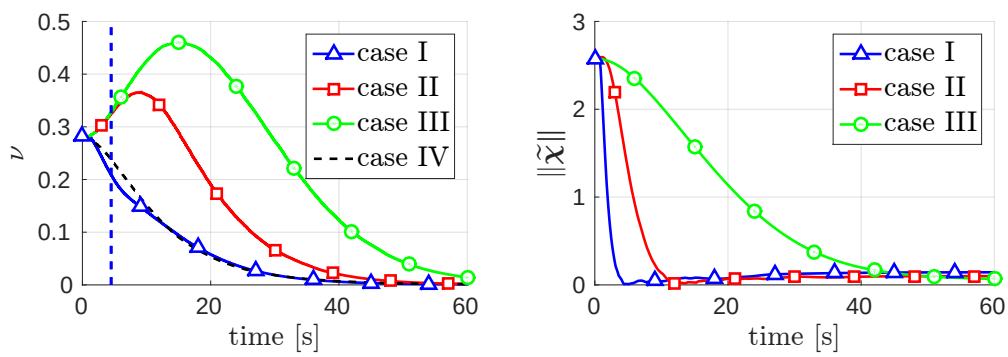
case I: the full strategy (three phases) illustrated in Sect. 6.2 and Fig. 6.2 is implemented for regulating the visual error $\boldsymbol{e}(t)$. The estimator (3.11) is run in parallel to the servoing task for generating the estimated $\widehat{\boldsymbol{\chi}}(t)$ fed to all the various control terms. The active optimization of the camera motion (7.1) takes place for the whole duration of phase 1);

case II: the classical control law (6.4) is implemented for regulating the visual error $\boldsymbol{e}(t)$. The estimator (3.11) is *still* run in parallel to the servoing task for generating the estimated $\widehat{\boldsymbol{\chi}}(t)$ fed to all the various control terms. However, in this case, no optimization of the estimation error convergence is performed;

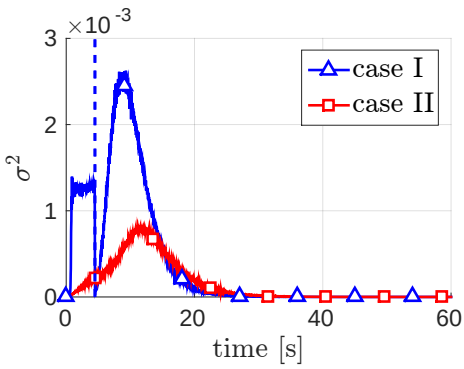
case III: the classical control law (6.4) is again implemented for regulating the visual error $\boldsymbol{e}(t)$. However, the estimator (3.11) is *not* run and vector $\widehat{\boldsymbol{\chi}}(t)$ is taken coincident with its value at the desired pose, i.e., $\widehat{\boldsymbol{\chi}}(t) = \boldsymbol{\chi}_{\mathrm{d}} = const$, as customary in many VS applications;

case IV: finally, as a reference 'ground truth', the classical control law (6.4) is again implemented but by exploiting knowledge of the ground truth value $\widehat{\boldsymbol{\chi}}(t) = \boldsymbol{\chi}(t)$ during the whole servoing execution. This case then represents the 'ideal' behavior one could obtain were $\boldsymbol{\chi}(t)$ available from direct measurement.

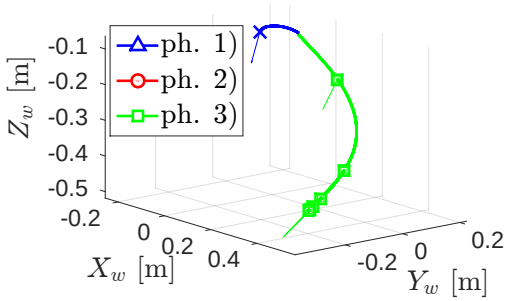
The following gains and thresholds were used in the experiments: $\alpha = 2000$ in (3.11), $k_p = 0.0225$ and $k_v = 0.3$ in (6.4–6.5). Moreover, only for case I, we used $k_\sigma = 20$, $\gamma = 0.001$ and $k_d = 18$ in (7.1), $\nu_T = 0.21$ and $\delta_T = 0.004$ in the flowchart of Fig. 6.2 and finally $k_\delta = 100$ in (6.14). Furthermore, in cases I and II, vector $\widehat{\boldsymbol{\chi}}$ was initialized as $\widehat{\boldsymbol{\chi}}(t_0) = \boldsymbol{\chi}_{\mathrm{d}}$, that is, starting from the (assumed known) value at the desired pose $\boldsymbol{\chi}_{\mathrm{d}}$ also exploited in case III.



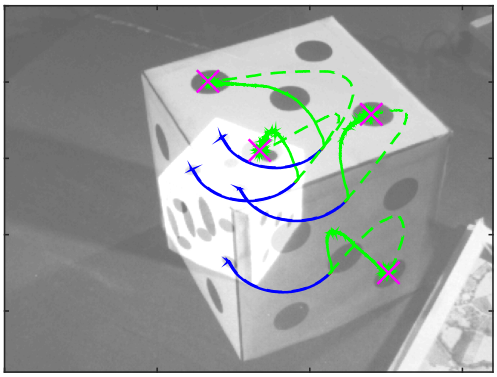
(a) (b)



(c)



(d)



(e)

Figure 7.1 – **Regulation of 4 point features**. Fig. (a): behavior of the error norm $\nu(t)$ when using the full strategy of Sect. 6.2 and the estimated $\widehat{\boldsymbol{\chi}}(t)$ (case I – blue line); the classical controller (6.4) and the estimated $\widehat{\boldsymbol{\chi}}(t)$ (case II – red line); the classical controller (6.4) by employing $\widehat{\boldsymbol{\chi}}(t) = \boldsymbol{\chi}_{\mathrm{d}}$ (case III – green line); the classical controller (6.4) by using the ground truth $\boldsymbol{\chi}(t)$ (case IV – black dashed line). Fig. (b): behavior of the norm of the approximation error $\|\widetilde{\boldsymbol{\chi}}(t)\| = \|\widehat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)\|$ with the same color code. Fig. (c): behavior of $\sigma^2(t)$ when actively optimizing the camera motion (case I – blue line) or not performing any optimization (case II – red line). In the previous plots, the (practically coincident) vertical dashed blue lines represent the switching times between the various control phases used in case I. Fig. (d): 3-D camera trajectory during case I with arrows representing the camera optical axis and square and circular markers representing the camera initial and final poses respectively. The three phases of Sect. 6.2.3 are denoted by the following color code: blue – phase 1), red – phase 2), green – phase 3). Fig. (e): trajectory of the four point features in the image plane during case I using the same color code, and with crosses indicating the desired feature positions. Superimposed, the initial and final camera images. Finally, solid lines represent the result of (correctly) implementing phase 2), while dashed lines represent the effects of a direct switch from phase 1) to phase 3) without the action of vector $\ddot{\boldsymbol{q}}_w = \ddot{\boldsymbol{q}}_{\boldsymbol{\delta}}$ in (6.14).

Figures 7.1(a) and 7.1(b) show the evolution of the error norm $\nu(t)$ and of the

estimation error norm $\|\widetilde{\boldsymbol{\chi}}(t)\| = \|\widehat{\boldsymbol{\chi}}(t) - \boldsymbol{\chi}(t)\|$ for the four cases. Figure 7.1(c) reports, instead, the evolution of the average eigenvalue $\sigma^2(t)$ for case I (blue line) and case II (red line), and finally Figs. 7.1(d) and 7.1(e) depict the camera and feature trajectory for case I.

Let us first focus on Fig. 7.1(b): from the plots one can note how the use of observer (3.11) in cases I and II makes it possible for the estimation/approximation error $\|\widetilde{\boldsymbol{\chi}}(t)\|$ to converge faster than in case III where convergence is reached only at the end of the task, when $\boldsymbol{\chi}(t) \to \widehat{\boldsymbol{\chi}} = \boldsymbol{\chi}_{\mathrm{d}}$ (as obvious). Furthermore, the convergence of $\|\widetilde{\boldsymbol{\chi}}(t)\|$ is clearly faster in case I (about 4 s) than in case II (about 12 s, thus three times slower). This improvement is due to the *active* optimization of the camera velocity occurring during phase 1) of case I. Indeed, looking at Fig. 7.1(c), one can note how the value of $\sigma^2(t)$ of case I (blue line) is approximately 4 times larger than in case II (red line) for the whole duration of phase 1) as a result of the more 'exciting' trajectory performed by the camera under the action of the optimization term (7.1).

A similar pattern can also be found in Fig. 7.1(a): indeed, the behavior of $\nu(t)$ for case I (blue line)(i) quickly reaches a good match with the ideal behavior of case IV (dashed black line), and, more importantly, (ii) keeps *monotonically* decreasing during all the various phases. This is clearly achieved thanks to the fast convergence of $\|\widetilde{\boldsymbol{\chi}}(t)\| \to 0$ that translates into a fast accurate evaluation of the task Jacobian $\widehat{\boldsymbol{J}}$ and any related quantity. On the other hand, due to the larger error in estimating $\boldsymbol{\chi}(t)$, both cases II and III present an initial *divergence* of the error norm $\nu(t)$ that starts increasing (rather than decreasing as in case I) because of the poorer approximation in the evaluation of the task Jacobian $\widehat{\boldsymbol{J}}$. It is worth noting how this initial divergent phase has, nevertheless, a shorter duration for case II w.r.t. case III thanks, again, to the use of observer (3.11) which is eventually able to provide a sufficiently accurate estimation of $\boldsymbol{\chi}(t)$ starting from $t \approx 12$ s.

The camera trajectory, depicted in Fig. 7.1(d), is also helpful for better understanding the effects of the active optimization of the camera motion during phase 1) of case I. Note, indeed, how the camera initially moves along an approximately circular path (blue line) because of the null-space term (7.1) that generates an 'exciting' motion for the estimation of the four point depths Z_i. This then allows to quickly reduce the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\|$ as reported in Fig. 7.1(b). It is also possible to, again, appreciate the benefits of having employed the norm controller (6.5) during phase 1): indeed, it is only thanks to the large redundancy granted by controller (6.5) that the camera is made able to follow a quite 'unusual' trajectory while, *at the same time*, ensuring a convergent behavior for the error norm $\nu(t)$ (Fig. 7.1(a)). For completeness, the red line in Fig. 7.1(d) represents (the quite short) phase 2) of the switching strategy (i.e., the alignment among vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$),

while the green line represents phase 3), i.e., the use of the classical controller (6.4) for completing the servoing task. Finally in Fig. 7.1(e) the image plane trajectory of the four point features is reported exploiting the same color code of Fig. 7.1(d).

As a supplementary evaluation of the theoretical analysis of Sect. 6.2.3, we now report, for case I only, an additional experiment aimed at assessing the importance of having introduced phase 2) in the switching strategy of Sect. 6.2.3 (i.e., of having enforced the alignment of $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ before switching to the classical controller (6.4). To this end, Fig. 7.2(a) shows the behavior of the error norm $\nu(t)$ for the previous case I (blue line) together with the behavior of $\nu(t)$ when *not* implementing phase 2) but, instead, directly switching from phase 1) to phase 3) (cyan line). The two (almost coincident) blue vertical lines represent the switch from phase 1) to phase 2) and then phase 3) for the first experiment, and the direct switch from phase 1) to phase 3) for the second experiment. One can then note how, in the second experiment, the error norm $\nu(t)$ has a large overshoot when switching to phase 3) because of the misalignment of vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ at the switching time. In particular, this overshoot makes the error norm temporarily increase instead of monotonically decrease as desired. This overshoot is, instead, clearly not present in the first experiment where $\nu(t)$ keeps (correctly) converging during all phases.

Similarly, Fig. 7.2(b) reports the behavior of $\|\boldsymbol{\delta}\|$ from (6.13), i.e., the measure of misalignment among vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ minimized during phase 2). One can then verify how, in the first experiment, $\|\boldsymbol{\delta}\|$ is correctly (and very quickly) minimized at the end of phase 2) thanks to the action of (6.14). Finally, the effects of implementing/non-implementing phase 2) are also illustrated in Fig. 7.1(e), where the point feature trajectory with phase 2) *activated* (solid lines) and *deactivated* (dashed lines) are reported. One can again note how, in this latter case, the point features are subject to a significant overshoot at the switching time which is, instead, avoided when implementing phase 2).

7.1.2 Second set of experiments

We now discuss a second set of experiments that involve the same four cases I to IV introduced at the beginning of the previous section but with the camera starting from a different initial pose and with a different desired configuration $\boldsymbol{s}_{\mathrm{d}}$ w.r.t. the previous run. The results are reported in Fig. 7.3 by following the same pattern and color codes of the previous Fig. 7.1.

As compared to Fig. 7.1, it is worth noting how the sole case I (blue line in Fig. 7.3(a)) results in a successful regulation of the visual task error $\boldsymbol{e}(t)$ thanks, again, to the fast convergence of the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\|$ during the active optimization of phase 1) (blue line in Fig. 7.3(b)). The servoing fails instead in case II

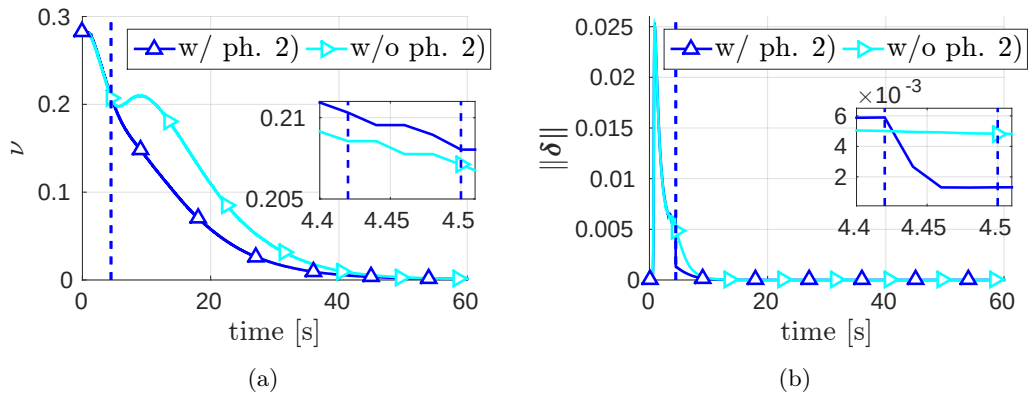
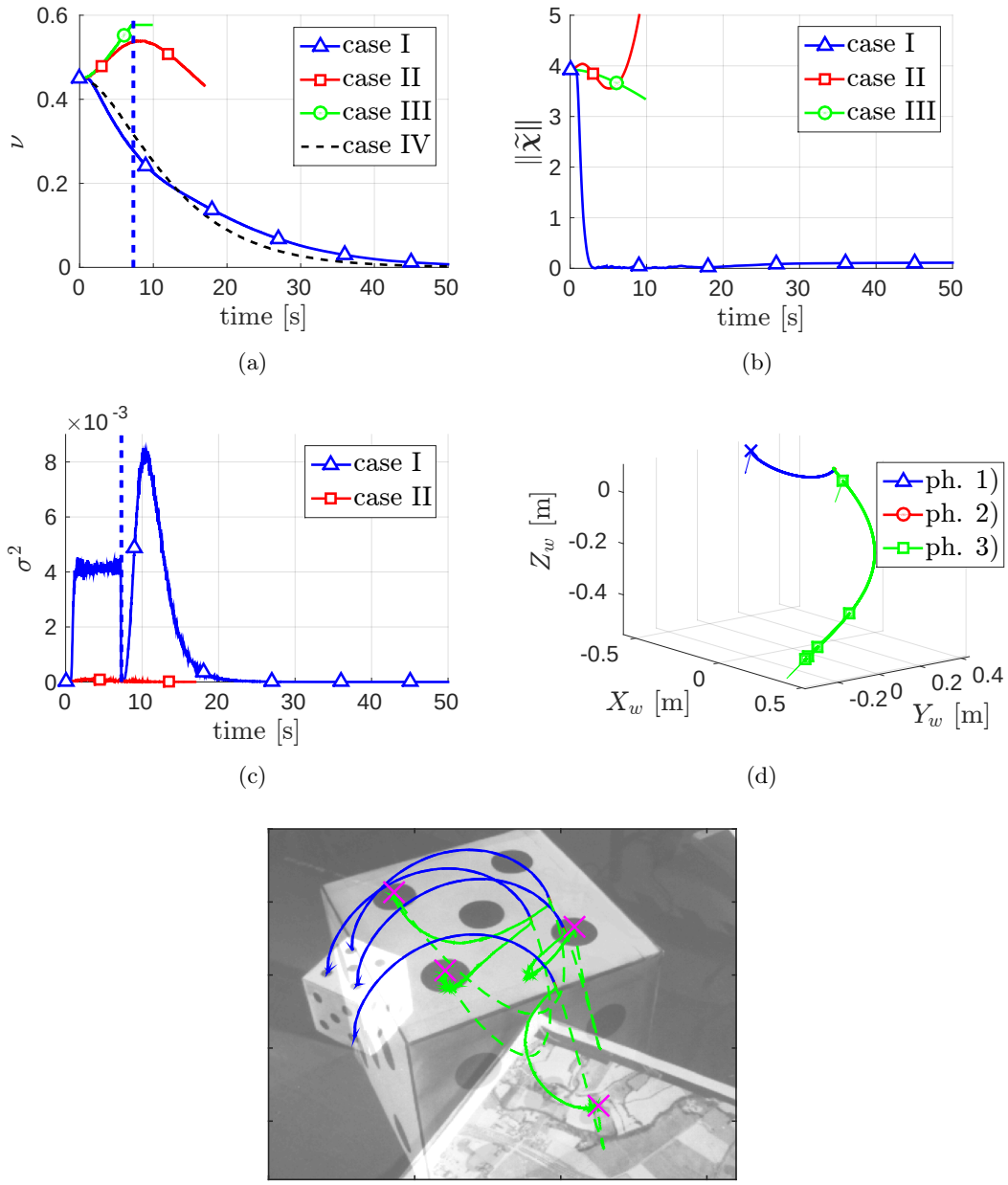


Figure 7.2 – **Importance of implementing phase 2) of Fig. 6.2 in the regulation of 4 point features**. Behavior of the error norm $\nu(t)$ (Fig. (a)) and of $\|\boldsymbol{\delta}\|$, the measure of misalignment between vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ (Fig. (b)). In both plots, the blue lines represent the behavior of case I (full implementation of the switching strategy of Sect. 6.2.3), while cyan lines represent the direct switch from phase 1) to phase 3) without the action of vector $\ddot{\boldsymbol{q}}_w = \ddot{\boldsymbol{q}}_{\boldsymbol{\delta}}$ in (6.14). The small picture-in-picture plots provide a zoomed view of the switching phase.

(red line in Fig. 7.3(a)), i.e., when coupling the classical controller (6.4) with observer (3.11) but *without* optimizing for the convergence rate of $\|\widetilde{\boldsymbol{\chi}}(t)\|$. In fact, in this case, the very small value of $\sigma(t)$ during the camera motion (red line in Fig. 7.1(c)) makes the estimation task ill-conditioned w.r.t. noise and other unmodeled effects (including the disturbance $\boldsymbol{d}(\widetilde{\boldsymbol{\chi}}, t)$ in (3.12)), resulting in a divergence of the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\|$ at $t \approx 9\,\mathrm{s}$ (red line in Fig. 7.3(b)). On the other hand, the active optimization of case I is able to increase $\sigma(t)$ by approximately a factor of 40 w.r.t. case II, thus ensuring a sufficiently high level of excitation for the camera motion and, consequently, a quick convergence of the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\|$. Failure of the servoing is finally obtained also in case III, i.e., when exploiting the *exact* final value $\widehat{\boldsymbol{\chi}}(t) = \boldsymbol{\chi}_{\mathrm{d}}$, because of the large initial error of the visual task that causes the features to leave the camera FOV (green line in Fig. 7.3(a)).

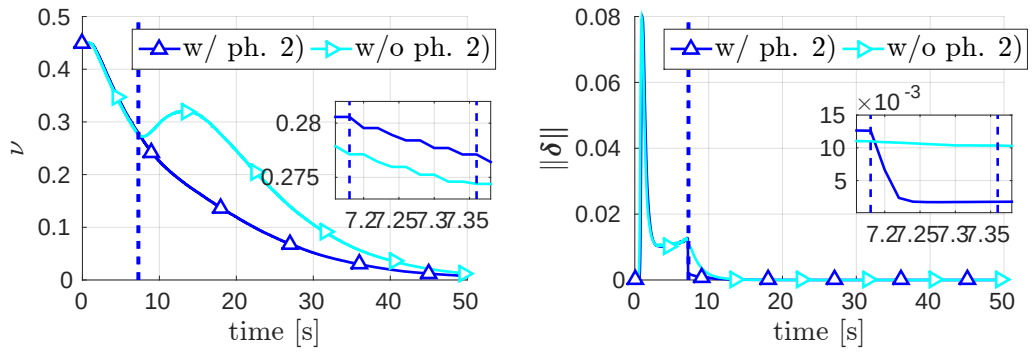
Finally, Figs. 7.3(d) and 7.3(e) depict the camera and feature trajectories during case I. One can again appreciate, in Fig. 7.3(d), the initial spiralling motion of the camera that allows the increase of $\sigma(t)$ during phase 1). It is also worth noting how, in case I, the error norm $\nu(t)$ keeps a *monotonic* decrease during the whole motion (as desired) despite the various switches among the three phases and the 'unusual' initial camera trajectory (blue line in Fig. 7.3(a)).

Analogously to what done in Sect. 7.1.1, we conclude by highlighting again the fundamental role played by phase 2) for ensuring a monotonic convergence of the error norm $\nu(t)$. To this end, Fig. 7.4(a) shows the behavior of $\nu(t)$ when activating (blue line) or not activating (cyan line) phase 2). One can note, again,



(e)

Figure 7.3 – **Regulation of** 4 **point features starting from a different initial camera pose** w.r.t. the experiments reported in Fig. 7.1. The plot pattern and color codes are the same as in Fig. 7.1. Note how, this time, a correct realization of the servoing task is obtained only in case I (blue line in Fig. 7.3(a)). In case II, the estimation error $\|\widetilde{\chi}(t)\|$ diverges (red line in Fig. 7.3(b)) because of the too small value of $\sigma^2(t)$ (red line in Fig. 7.3(c)) during the camera motion which makes the estimation task ill-conditioned. In case III, the visual task error starts diverging because of the too rough approximation $\widehat{\chi} = \chi_{\mathrm{d}}$ and the point features leave the camera FOV (green line in Fig. 7.3(a)). Note also the initial spiralling motion of the camera (blue line in Fig. 7.3(d)) that allows maximization of the value of $\sigma^2(t)$ during phase 1) of case I.



(a) (b)

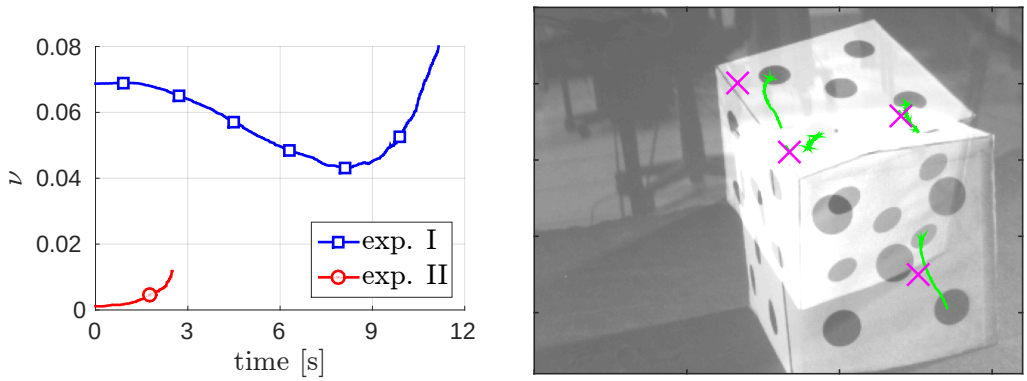
Figure 7.4 – **Importance of implementing phase 2) of Fig. 6.2 in the regulation of 4 point features starting from a different initial camera pose**. Behavior of the error norm $\nu(t)$ (Fig. (a)) and of $\|\boldsymbol{\delta}(t)\|$, the measure of misalignment between vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ (Fig. (b)). In both plots, the blue lines represent the behavior of case I (full implementation of the switching strategy of Sect. 6.2.3), while cyan lines represent the direct switch from phase 1) to phase 3) without the action of vector $\ddot{\boldsymbol{q}}_w = \ddot{\boldsymbol{q}}_{\boldsymbol{\delta}}$ in (6.14). The small picture-in-picture plots provide a zoomed view of the switching phase.

the large overshoot (and temporary divergence) introduced at the switching time when parallelism among vectors $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$ is not enforced. Figure 7.4(b) depicts the associated behavior of the misalignment measure $\|\boldsymbol{\delta}(t)\|$ that proves, again, how the action (6.14) during phase 2) (blue line) is able to quickly minimize $\|\boldsymbol{\delta}(t)\|$ as desired.

7.1.3 Third set of experiments

In this section, we report the results of two experiments meant to show how even relatively small inaccuracies in determining the value $\boldsymbol{\chi}_\mathrm{d}$ at the desired pose can cause failure of the servoing when setting $\widehat{\boldsymbol{\chi}}(t) = \boldsymbol{\chi}_\mathrm{d}$ as classically done in many visual servoing applications. The two experiments involve the same setting of the previous cases (regulation of 4 point features) and differ from the starting location of the camera w.r.t. the target object: in the first experiment, the camera starts (relatively) far from the desired pose while, in the second experiment, the camera starts at almost the desired pose. In both cases, the classical second order control (6.4) by taking $\widehat{\boldsymbol{\chi}} = (\boldsymbol{I}_p + \mathrm{diag}(\boldsymbol{\epsilon}))\boldsymbol{\chi}_\mathrm{d}$ with $\boldsymbol{\epsilon} \in \mathbb{R}^p$ being a random vector with magnitude 0.09 (thus, simulating an uncertainty of 9% in the accuracy of $\boldsymbol{\chi}_\mathrm{d}$).

Figure 7.5(a) shows the behavior of the error norm $\nu(t)$ for both cases: in the first experiment (blue line), the visual error starts converging from its initial (large) value but then, at about $t \approx 8\,\mathrm{s}$, the servoing diverges and the features leave the camera FOV. An even more interesting result is obtained in the second experiment (red line): in this case, the error $\nu(t)$ starts at a very small value since the camera



(a) (b)

Figure 7.5 – **IBVS of 4 point features using a constant approximation $\boldsymbol{\chi}(t) = \boldsymbol{\chi}_{\mathrm{d}}$** where the value of $\boldsymbol{\chi}_{\mathrm{d}}$ is corrupted by a relative error of 9%. In the first experiment the camera starts far from the desired pose, while in the second experiment the camera starts at almost the desired pose. Fig. (a): behavior of the error norm $\nu(t)$ for the first (blue line) and second (red line) experiments. Because of the approximated value of $\boldsymbol{\chi}_{\mathrm{d}}$, the servoing is not able to converge in both cases (thus also when starting very close to the desired pose), resulting in a loss of tracking for the point features. Fig. (b): image plane trajectory of the 4 point features during the first experiment. The four crosses indicate the desired feature positions, and the initial and final (i.e., until loss of tracking) camera images are superimposed.

is already quite close to its desired pose. However, controller (6.4) is not able to impose a stable closed-loop behavior, and the error norm starts diverging until loss of tracking of the feature points at about $t \approx 2.5\,\mathrm{s}$.

These results then provide an experimental demonstration of the effects discussed in Sect. 2.3.2 and originally introduced in [MMR10]: a (rather small) error in approximating $\boldsymbol{\chi}_{\mathrm{d}}$ can be sufficient to move part of the eigenvalues of matrix $-\boldsymbol{J}(\boldsymbol{s}_{\mathrm{d}}, \boldsymbol{\chi}_{\mathrm{d}}, \boldsymbol{q})\widehat{\boldsymbol{J}}(\boldsymbol{s}_{\mathrm{d}}, \widehat{\boldsymbol{\chi}}, \boldsymbol{q})^{\dagger}$ to the right-half complex plane, thus resulting in an unstable closed-loop dynamics even when starting arbitrarily close to the desired pose. By, instead, resorting to an online (and optimized) estimation of $\boldsymbol{\chi}(t)$, one can obtain a considerably larger degree of robustness against uncertainties in evaluating $\boldsymbol{\chi}_{\mathrm{d}}$ or similar approximation errors as extensively shown by the previous results.

7.2 Using an adaptive switching strategy

For the experiments in this section, we considered the same experimental setup of Sect. 7.1, that is, regulation of $N = 4$ point features belonging to the surface of a cube. However, we now employed the full adaptive strategy described in Sect. 6.3 for triggering the activation/deactivation of the optimization of the 3-D structure estimation (switch from phase 3) to phase 1) and vice-versa) and, more in general,

for tuning its effect (when in phase 1)) as a function of the accuracy in estimating $\boldsymbol{\chi}(t)$

As done in the previous experiments, we initialized the SfM observer (3.11) with vector $\widehat{\boldsymbol{\chi}}(t_0)$ taken coincident with the (assumed known) $\boldsymbol{\chi}_{\mathrm{d}}$ at the final pose, and $\widehat{\boldsymbol{s}}(t_0) = \boldsymbol{s}(t_0)$. As explained in Appendix A.5.2, this results in a bound $\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2/\alpha = 5.3e{-}3$ in (6.19). As for the adaptive strategy thresholds, we set $\underline{E} = 10^{-5}$ and $\overline{E} = 10^{-4}$. The results of the experiment are reported in Fig. 7.6: during the trial, the target object is purposely displaced at $t \approx 5.9\,\mathrm{s}$, $t \approx 10.6\,\mathrm{s}$ and $t \approx 17.2\,\mathrm{s}$ for introducing an "external disturbance" able to increase both the servoing and the estimation errors above their minimum thresholds with a corresponding (re-)activation of the camera motion optimization.

At the beginning of the motion (phase 3)), the eigenvalue σ_1^2 is considerably small due to the low information content of the camera trajectory (Fig. 7.6(c)) and, analogously to case II in Sect. 7.1.2, the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ even starts diverging because of measurement noise, the disturbance term $\boldsymbol{d}$ in (3.12), and other non-idealities (Fig. 7.6(b)). At time $t \approx 1.1\,\mathrm{s}$, however, the quantity $E(t)$ increases over the threshold $\underline{E}$ because of the high uncertainty in the estimated $\widehat{\boldsymbol{\chi}}$ (Fig. 7.6(d)), thus triggering the switch to phase 1) and the corresponding optimization of the camera motion. The optimization action (6.18) results in a fast increase of the mean eigenvalue $\sigma(t)$ (Fig. 7.6(c)) and, as a consequence, in a fast convergence of the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ (Fig. 7.6(b)) that practically vanishes at time $t \approx 4\,\mathrm{s}$. As a consequence, $E(t)$ decreases again below the minimum threshold $\underline{E}$ indicating that a sufficient level of accuracy has been reached. This then triggers the (very quick) switch to phase 2) and, then, the switch back to phase 3) at $t \approx 4.4\,\mathrm{s}$. Note how the adaptive gain $k_E(E)$ used in (6.18) correctly (and smoothly) activates and deactivates the optimization of σ^2 during phase 1) as clear from Fig. 7.6(e).

It is worth noting that the switch from phase 1) to phase 3) occurs when the error norm $\nu(t)$ is still well above the threshold ν_T indicating singularity of controller (6.5). Therefore, the distortion of the camera trajectory (depicted in Fig. 7.6(f)), needed to maximize σ^2, lasts considerably less than in the non-adaptive case where the switch would have occurred only at $\nu(t) = \nu_T$. Finally, one can also appreciate how the error norm $\nu(t)$ correctly converges monotonically towards zero once the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ becomes small enough, i.e., for $t \geq 4\,\mathrm{s}$, see Fig. 7.6(a). At $t \approx 5.9\,\mathrm{s}$ the target object is purposely displaced, as explained, causing both the servoing and the estimation error to grow with a corresponding increase of $E(t)$ above the threshold $\underline{E}$. This, in turn, triggers the switch to phase 1) at $t \approx 6.1\,\mathrm{s}$ for (re-)activating the optimization of the camera motion until convergence of the estimation error is again reached at $t \approx 9.1\,\mathrm{s}$. The same pattern then repeats two more times at $t \approx 10.6\,\mathrm{s}$ and $t \approx 17.2\,\mathrm{s}$ because of the two additional displacements

of the target object during the camera motion.

As explained in Sect. 6.3, the switch from phase 1) to phase 3) (and vice-versa) is also a function of the current value of the error norm $\nu(t)$ for avoiding possible singularities in (6.5). This is, indeed, the case of the third switch from phase 1) to phase 3) triggered at $t \approx 13.3\,\mathrm{s}$ by the error norm falling below the threshold ν_T with $E(t)$ still above the minimum value $\underline{E}$. Similarly, the fourth switch from phase 3) to phase 1) at $t \approx 17.9\,\mathrm{s}$ is triggered only when $\nu(t) \geq \nu_T$ even though $E(t)$ has already grown over the threshold $\underline{E}$. By looking at Fig. 7.6(d), it is finally worth noting how $E(t)$ always keeps below the theoretical bound $\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2/\alpha = 5.3e{-}3$ given in (6.19) despite the three intentional target displacements occurred during the servoing (see Appendix A.5.2).

7.3 Using a standard Kanade Lucas Tomasi feature tracker

This last experiment is meant to illustrate the feasibility of our approach in more realistic conditions compared to the use of simple black dots on a white background as done so far. To this end, we considered regulation of 10 point features belonging to a much less structured object, that is, the shrunken piece of textured paper shown in Fig. 7.7(g). Extraction and tracking of the 10 features was achieved by exploiting the well-known KLT algorithm implemented in OpenCV. Finally, we made use of the threshold $\underline{E} = 0.0015$ and $\overline{E} = 0.03$, and initialized $\widehat{\boldsymbol{\chi}}(t_0) = \boldsymbol{\chi}_\mathrm{d}$ as before, with, in this case, $\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2/\alpha = 6.3e{-}3$ for bound (6.19).

Figure 7.7 reports the results of the experiment: the robot starts in phase 3) driven by the classical law (6.4) but, being the mean eigenvalue σ^2 rather small during this phase, the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ does not converge, and likewise the error norm $\nu(t)$ because of the too rough approximation in $\widehat{\boldsymbol{\chi}}$. However, the quantity $E(t)$ starts to grow and, at $t \approx 1\,\mathrm{s}$, it exceeds the threshold $\underline{E}$ triggering the switch to phase 1) (Fig. 7.7(d)). During this phase (which lasts until $t \approx 5\,\mathrm{s}$) the optimization of the camera motion is then able to maximize the eigenvalue σ^2 resulting in a quick convergence of the estimation error that practically vanishes at $t \approx 4.5\,\mathrm{s}$. Similarly, the quantity $E(t)$ first reaches a maximum peak value (which is anyway lower than the theoretical bound (6.19) as expected), and then starts decreasing back to zero thus allowing a smooth deactivation of the optimization action thanks to the adaptive gain k_E (Fig. 7.7(e)). Finally, at $t \approx 5\,\mathrm{s}$ the error norm $\nu(t)$ falls below the threshold ν_T inducing a quick switch to phase 2) (alignment of $\boldsymbol{e}$ and $\dot{\boldsymbol{e}}$) followed by a last switch back to phase 3) until completion of the servoing task.

Looking at these results, it is then possible to appreciate how the overall behavior

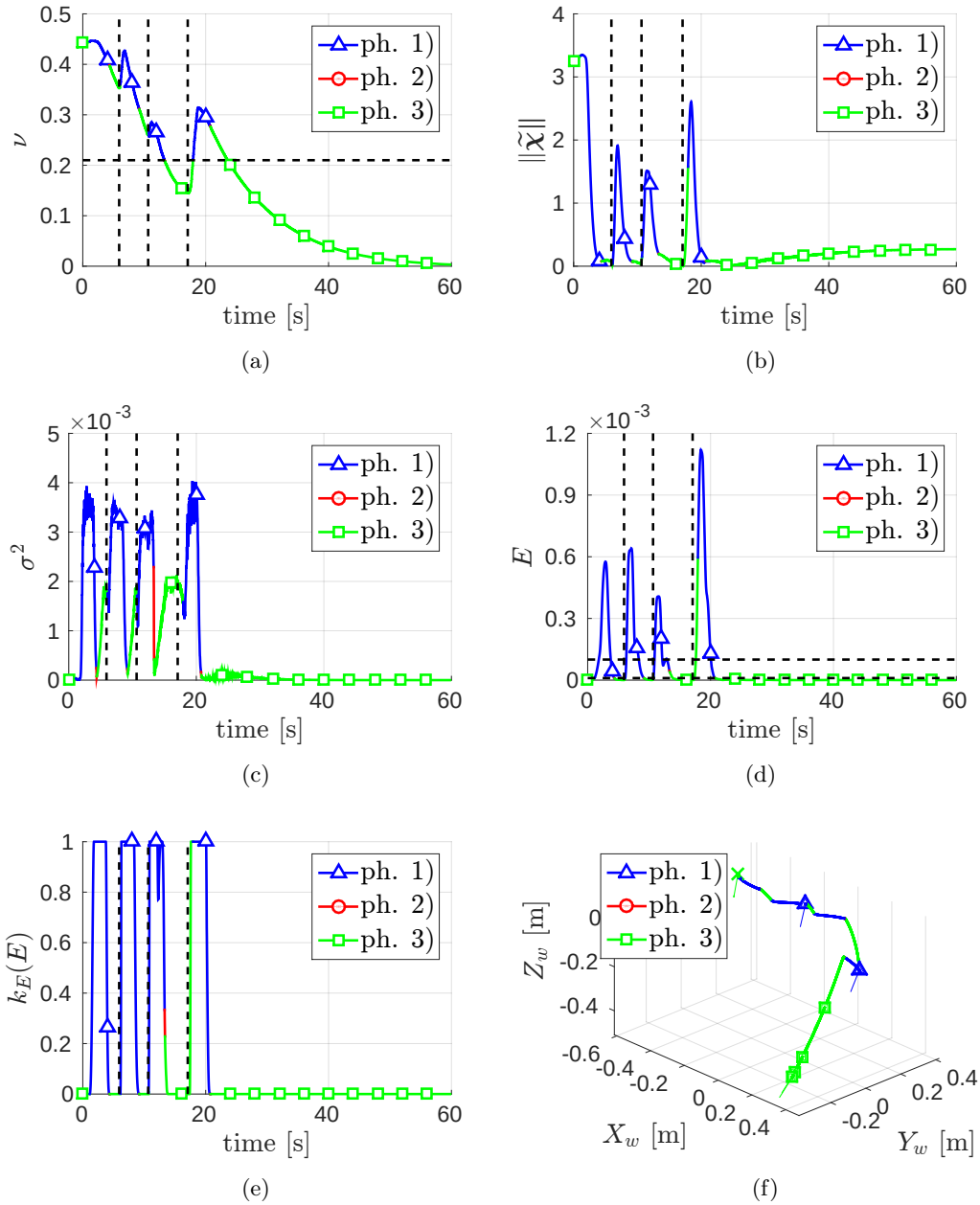
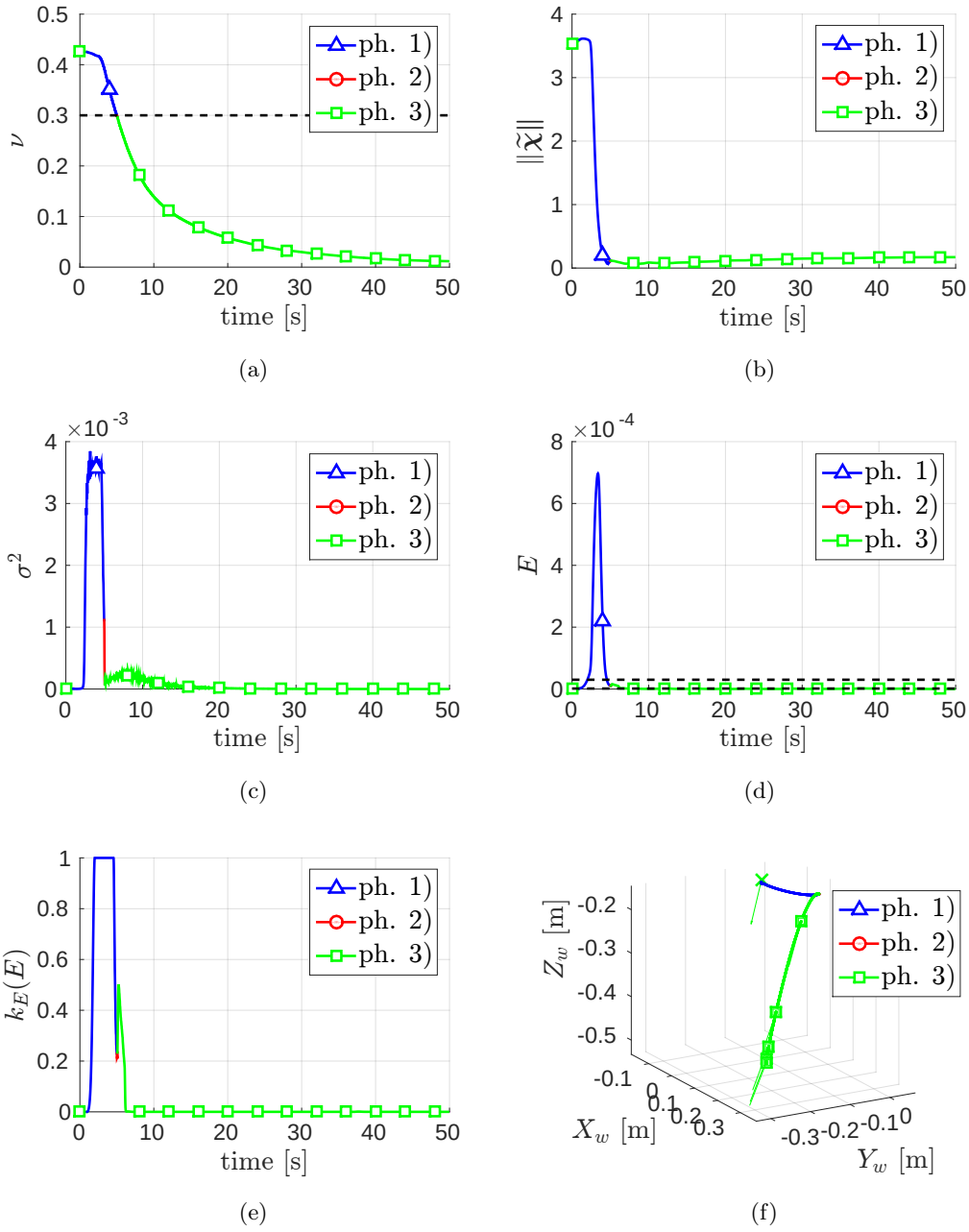
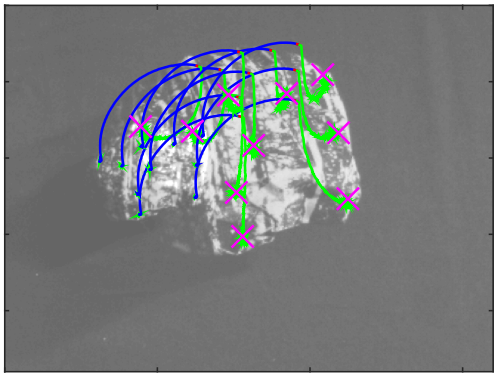


Figure 7.6 – **Regulation of** 4 **point features using the adaptive strategy of Sect. 6.3**. The three phases of Fig. 6.4 are denoted by the following color code: blue – phase 1), red – phase 2), green – phase 3). Fig. (a): behavior of the error norm $\nu(t)$ with superimposed a horizontal dashed black line indicating the threshold ν_T. Fig. (b): behavior of the norm of the estimation error $\|\widetilde{\boldsymbol{\chi}}(t)\| = \|\boldsymbol{\chi}(t) - \widehat{\boldsymbol{\chi}}(t)\|$. Fig. (c): behavior of the mean eigenvalue σ^2. Fig. (d): behavior of $E(t)$ with, superimposed, two dashed horizontal lines indicating the minimum and maximum thresholds $\underline{E}$ and $\overline{E}$. Fig. (e): behavior of the adaptive gain $k_E(E)$. In all of the previous plots, vertical dashed lines represent the times at which the target object was intentionally displaced. Fig. (f): camera 3-D trajectory with arrows representing the camera optical axis and square and circular markers representing the camera initial and final poses, respectively.

of the adaptive strategy is essentially equivalent to what obtained in the previous case studies, thus confirming that the proposed approach can be seamlessly applied to more complex/realistic situations. We also believe it is particularly worth noting how, again, the error norm $\nu(t)(i)$ starts decreasing as soon as the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ becomes small enough in $t \approx 3\,\mathrm{s}$ (Fig. 7.7(a)), and (ii) keeps a *monotonic* convergence (as desired) until the end of the servoing despite the various switches among the phases and the 'distorted' camera motion during phase 1).





(g)

Figure 7.7 – **Regulation of** 10 **point features on an unstructured object** using a KLT tracker and the adaptive strategy of Sect. 6.3. The same quantities of the previous Fig. 7.6 are reported here with the only exception of Fig. (g) that depicts the trajectory of the 10 point features on the image plane with crosses indicating the desired feature position and, superimposed, two (semi-transparent) camera screenshots taken at the initial and final robot configuration.

7.4 Conclusions

This chapter was entirely dedicated to the experimental validation of the strategy introduced in Chapt. 6 for coupling the execution of a IBVS task with the concurrent active optimization of the camera velocity meant to maximize performance in the reconstruction of 3-D scene. The reported experimental campaign clearly shows, in our opinion, the benefits gained by using the proposed strategy in terms of:(*i*) obtaining a faster convergence of the structure estimation error during the servoing transient w.r.t. non-active cases, (*ii*) imposing an improved closed-loop IBVS behavior by significantly mitigating the negative effects of an inaccurate knowledge of the scene structure, (*iii*) minimizing the deformation of the camera trajectory (consequence of the active SfM action) thanks to the adaptive activation/deactivation of the SfM optimization. The last point, in particular, allows to seamlessly deal with the unexpected growth of the estimation error due to unpredicted disturbances (the target object being purposely displaced). The experiments were run under different initial camera configurations and using different feature trackers (a simple dot tracker and a more standard KLT one) demonstrating the robustness of our approach.

Chapter 8

Conclusions and future work

In this last chapter of the thesis, we wish to summarize the main theoretical and experimental contributions of this work highlighting some of the issues that still remain unsolved and suggesting, whenever possible, some directions for improvement and further investigation.

8.1 Summary and contributions

The goal of this thesis was to investigate the problem of active sensing and control of robotic systems, in particular by emphasizing the relation between the two aspects. After a short introduction to the basic tools that would serve our analysis, we started the thesis by focusing on the estimation problem. In this context, we proposed to employ a nonlinear observer stemming from the adaptive control literature (see, e.g., [MT95]) and already exploited in [RDO08, DOR08] for SfM applications. The advantage of this choice lies in the fact that, for such an observer, one can fully characterize and control the dynamics of the estimation error. In particular we demonstrated that:

(i) the dynamics of the estimation error has a clear port-Hamiltonian form;

(ii) by tuning the observer gains online as a function of measurable quantities only, one can impose to the estimation error a dynamics (approximately) equivalent to that of a second order linear system with assigned poles and, in particular,

(iii) by actively controlling the direction of the camera linear velocity one can maximize the performance of the estimation for a given threshold on the maximum admissible control effort (the camera velocity norm);

(iv) in addition to this, one can also gain additional performance by adaptively changing the measurements that are used for the estimation, e.g., by using

a generalized definition of image moments with some weight parameters that can be changed online to maximize the "excitation" of the system.

The theoretical results were then applied to the reconstruction of the structure for different geometric primitives hereafter summarized:

(v) point features: we compared the effects of using a planar and a spherical projection model showing how the former can potentially lead to faster convergence of the estimation error at the cost of a higher sensitivity to disturbance and noise w.r.t. the latter.

(vi) planar scenes: we proposed a comparison between the standard homography based reconstruction technique and two "active" strategies based on either the extraction of a plane from an actively estimated point cloud (using least-squares techniques) or on the direct estimation of the plane parameters from discrete or dense image moments. Finally, as already mentioned, we suggested an adaptive strategy to automatically select on line the best image moments to use for the estimation.

(vii) spherical objects: we showed that, using a suitable parametrization suggested by [FC09], the estimation problem can be recast in such a way that the only unknown is the constant sphere radius, thus granting globally exponential stability properties of the observer. Due to the spherical symmetry in this case, the direction of the camera motion does not have any effect on the estimation performance, however, it is still possible to tune at will the "damping factor" of the error dynamics.

(viii) cylindrical objects: we proposed a novel parametrization that, similarly to the sphere case, reduces the estimation problem to that of retrieving a constant parameter: the cylinder radius. This results, again, in a globally exponential stability of the observer. Differently from the spherical case, however, here the estimation performance does depend on the camera motion and thus our active strategy allows, again, to maximize the convergence rate.

We reported an extensive simulative and experimental validation of our approach that, in our opinion, fully confirmed the validity of the theoretical claims.

After this, we moved our attention to the more challenging, but also more interesting, problem of simultaneous structure estimation and control of a robot. We considered, in particular, the case of a IBVS control task for which we showed that:

(ix) the performance of a IBVS scheme clearly benefits from the use of an active strategy that deforms online the trajectory of the camera, during the servo-

ing transient, so as to maximize the observability of the unmeasurable 3-D quantities that appear in the feature interaction matrix;

(x) the redundancy of the system can be conveniently maximized by using a second order extension of the strategy originally proposed in [MC10] which grants a large null-space projection operator by considering the regulation of the visual task norm instead of the task itself;

(xi) the deforming effects on the camera trajectory of the active observability maximization strategy can be effectively reduced by using an adaptive technique that allows to activate/deactivate, and, more in general, tune it as a function of the current estimation status.

Again, all of the theoretical claims were supported by a number of different experiments run on a real robotic manipulator equipped with a in-hand camera. The results demonstrated, in particular, the robustness of the proposed approach w.r.t. unmodeled disturbances such as sudden unexpected displacements of the target object used for the definition of the visual task.

8.2 Open issues and future perspectives

The results of this thesis, although encouraging, also show a number of limitations that affect our approach.

First of all the camera trajectory optimization strategy that we propose is *local*. In fact it is based on the greedy optimization of the *instantaneous* version of the PE condition (3.13). This has two drawbacks:(*i*) it can, in general, only produce locally optimal solutions and might fail to produce a globally optimal camera trajectory and, more importantly, (*ii*) it does not allow to tackle applications in which the number of available measurements is smaller than the number of unknown since, in this case, (3.13) is never full-rank and its smallest eigenvalue is identically zero. To overcome this limitation one should consider the original integral version of the PE condition (3.9) (we remind the reader that this is the only *necessary* condition for convergence of the estimation, with (3.13) being, instead, a *sufficient* but not necessary one). The error transient dynamics analysis proposed in Sect. 4.2 could be generalized by using *averaging techniques* [SB11], i.e. by considering the system:

$$\begin{cases} \dot{\widetilde{\boldsymbol{s}}}_{av} = \lim\limits_{T\to\infty} \displaystyle\int_t^{t+T} \left(-\boldsymbol{H}\widetilde{\boldsymbol{s}} + \boldsymbol{\Omega}^T\widetilde{\boldsymbol{\chi}}\right) \mathrm{d}\tau = \boldsymbol{f}_{\widetilde{\boldsymbol{s}}_{av}}(\widetilde{\boldsymbol{s}}_{av}, \widetilde{\boldsymbol{\chi}}_{av}) \\ \dot{\widetilde{\boldsymbol{\chi}}}_{av} = \lim\limits_{T\to\infty} \displaystyle\int_t^{t+T} \left(-\alpha\boldsymbol{\Omega}\widetilde{\boldsymbol{s}} + \boldsymbol{d}\right) \mathrm{d}\tau = \boldsymbol{f}_{\widetilde{\boldsymbol{\chi}}_{av}}(\widetilde{\boldsymbol{s}}_{av}, \widetilde{\boldsymbol{\chi}}_{av}) \end{cases}$$

instead of (3.12). In fact, we expect the eigenvalues of this system to be related to the PE integral condition (3.9). Note however that the maximization of (some norm of) (3.9) requires, as already stressed, the evaluation of matrix $\boldsymbol{\Omega}(\tau)$ in the future, i.e. for $\tau > t$. This, in turn, requires a prediction of the future measurements $\boldsymbol{s}(\tau)$ for $\tau > t$ which is only possible if the structure of the scene ($\boldsymbol{\chi}$) is known. In some works, such as, e.g. [WM13, WSM14] this issue is solved by planning and executing multiple experimental trajectories, each time leveraging the information retrieved during the previous experiment to obtain a better estimation of the unknown quantities that is then exploited in the new planning stage. The possible solution that we envision, instead, would employ a Model Predictive Control (MPC)-like strategy [GPM89, LHD06]: the optimization of the camera trajectory could be done, at time t, by maximizing (3.9) over a finite horizon T, using the current estimation $\widehat{\boldsymbol{\chi}}(t)$ to predict the future measurements $\boldsymbol{s}(\tau)$, $\tau \in [t, t+T]$; only a small portion (lasting, say $\Delta t \ll T$) of the resulting trajectory would, however, actually be executed, this would allow to collect some additional information about $\boldsymbol{\chi}$ that would result in an improved estimation $\widehat{\boldsymbol{\chi}}(t+\Delta t)$; the optimization process would then be repeated again, based on this more accurate estimation of the scene 3-D geometry, until full convergence of the estimation error.

Another limitation of our approach is the fact that we do not explicitly model the effect of noise. To cope with this, one could adopt a probabilistic estimation/control framework instead of the fully deterministic one proposed in this work. On the one hand this would potentially result in improved robustness w.r.t. non deterministic disturbances; on the other hand, however, one would probably have to give up on a formal characterization of the estimation error dynamics such as the one proposed here. An approach that we find particularly interesting in this context, because of its potential to address a large number of applications, is that of POMDPs. The significant required computational effort remains, however, and to the best of our knowledge, a major limitation of this approach.

As for the estimation problem, we are currently investigating novel strategies for eliminating (or reducing to the minimum) the need of preliminary image processing techniques for extraction and tracking of features. We are, in particular, interested in the class of direct/photometric methods [HS81, MKS89, CM11, BCM13] in which the camera images (i.e. the intensity level of each pixel) are used *directly* "as they are" to estimate the scene or control the robot motion. We believe that the use of photometric information directly could allow the estimation of a dense pixel-level depth map in an extremely efficient way. In fact, if one can ensure that the estimation algorithm can be run for each pixel using information that is both local in space (e.g. it uses the intensity level of the neighbour pixels only) and in time (using recursive estimation techniques) then the resulting observer would be perfectly

suitable for a highly parallel implementation on parallel hardware architectures such as FPGAs or GPUs with potentially extremely high performance both in terms of computation time and in terms of energy efficiency, see [AAM14]. These factors are particularly crucial for embedded systems applications. Some preliminary results in this context can be found in Appendix B.

As for the last part of this thesis, we note that the stability properties of the coupling between active SfM and IBVS are still to be investigated. In fact, due to the nonlinear dynamics of the system, one can not invoke, as in the linear case, the principle of separation: the fact that both the SfM and IBVS algorithms are separately stable does not imply that the overall system will converge. Although the experimental results are encouraging from this point of view, a more formal analysis should be done in the future. In a similar way, we are also concerned about the possibility that the active optimization of the estimation performance might excessively deform the camera trajectory to the point that, e.g. the camera loses track of the features or the robot reaches one of the joint limits. In this respect two possible approaches could be taken. The first possibility would be to use a more advanced kinematics resolution framework such as, e.g., the ones proposed in [MC07, MKK09], that allow to deal with the presence of multiple motion constraints that can be activated/deactivate online based on priorities and on the system status. Another possibility for dealing with the limited camera FOV would be to employ one of the techniques proposed in, e.g., [FC05, CDW+14] to allow for a *temporary* loss of the feature track by resorting on a prediction of the feature motion. We note that such a prediction could obviously benefit from the observability maximization strategy proposed here since it would be based on the most recent estimation of the 3-D parameters associated with the features that went out of the FOV.

Another interesting direction, certainly worth further investigation, is the adaption of the proposed machinery to mobile and flying robots. We have already briefly commented that the difficulty with this kind of platforms mainly comes from the presence of non-holonomic constraints and non negligible dynamics as well as from the higher level of non idealities that affect their actuation which results in the difficulty of retrieving a reliable estimation of the camera velocity, especially for its linear component (for the angular one, the use of good quality gyroscopes can alleviate this issue). Usually a measurement of the camera acceleration is, instead, more accessible thanks to the presence of on-board accelerometers. The nonlinear observer exploited in this work can be extended to use acceleration measurements instead of the camera linear velocity (see [GBSR15]). Note, however, that, in this case, the PE condition would impose constraints on the camera acceleration (e.g. the robot needs to keep accelerating during the estimation, see again [GBSR15]) that are, in general, harder to deal with from a practical point of view than the

velocity ones, especially if the instantaneous version of the PE condition (3.13) is considered.

Finally, the use of mobile and flying robots also opens the way to a number of new potential applications of the proposed machinery in the context of multi-robot systems. In many situations, in fact, a team of robot is required to perform some collaborative estimation task concerning either the internal formation state (e.g. estimation of the degree of connectivity [RFSB13] or of the rigidity [ZFBR14] or of the scale [ZFR14] of the formation) or some external quantity (e.g. localization of the formation in the environment [NRM09] and/or reconstruction of a 3-D scene [TL05] and so on). In these applications, the estimation problem is typically nonlinear and the trajectory followed by the agents does have an effect on the observability of the system so that the use of an active strategy for deciding an optimal motion policy can have a significant impact, especially considering the high number of DOFs that these systems typically have. For instance, the authors of [MM13] use simplified process and measurement models to address the problem of controlling the motion of a flock of aerial vehicles in order to minimize the uncertainty in both agent self-localization and multiple target tracking. The case of multi-robot systems also introduces an additional challenge: for most applications, due to technological limitations and scalability considerations, it is not possible (or at least not desirable) to rely on a central node that processes all the information collected by the agents. Each of the agents should instead be able to independently perform the estimation task and take decisions as to which trajectory to follow in a completely decentralized way using only local information collected by its own sensors or by a subset of agents in its neighbourhood.

Appendix **A**

Technical details

This appendix includes additional technical details about the derivations and results contained in the rest of thesis. The material included here is not essential for the understanding of the main results, but provides further insights to the interested reader.

A.1 Derivation of the optimal Kalman-Bucy filter

This section contains some additional technical details concerning the derivation of the optimal Kalman Filter (KF) equations for continuous-time, linear time-varying systems. The developments in this section are mainly based upon [AT67] which is extended here to the case of correlated input and measurement noise. Alternative derivations of the optimal KF equations can also be found in the original work [KB61] as well as in many other optimal estimation textbooks, such as [LXP07].

A.1.1 Propagation equation for the error covariance matrix

We are interested in finding the dynamic differential equation governing the evolution of the estimation error covariance matrix $\boldsymbol{\Sigma}(t) = \mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\widetilde{\mathbf{x}}(t)^T\right\}$. Thanks to the linearity of the derivative and expected value operations one can write

$$\begin{aligned}\dot{\boldsymbol{\Sigma}}(t) &= \mathrm{E}\left\{\frac{\mathrm{d}}{\mathrm{d}t}\left[\widetilde{\mathbf{x}}(t)\widetilde{\mathbf{x}}(t)^T\right]\right\} = \mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\dot{\widetilde{\mathbf{x}}}(t)^T + \dot{\widetilde{\mathbf{x}}}(t)\widetilde{\mathbf{x}}(t)^T\right\} \\ &= \mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\dot{\widetilde{\mathbf{x}}}(t)^T\right\} + \left(\mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\dot{\widetilde{\mathbf{x}}}(t)^T\right\}\right)^T\end{aligned} \tag{A.1}$$

Plugging (3.29) in (A.1), and dropping time dependence, one obtains

$$\begin{aligned}\mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\dot{\widetilde{\mathbf{x}}}(t)^T\right\} &= \mathrm{E}\left\{\widetilde{\mathbf{x}}[(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})\,\widetilde{\mathbf{x}} - (\boldsymbol{G}-\boldsymbol{K}\boldsymbol{H})\boldsymbol{w} + \boldsymbol{K}\boldsymbol{v}]^T\right\} \\ &= \boldsymbol{\Sigma}(\boldsymbol{A}-\boldsymbol{K}\boldsymbol{C})^T - \mathrm{E}\left\{\widetilde{\mathbf{x}}\boldsymbol{w}^T\right\}(\boldsymbol{G}-\boldsymbol{K}\boldsymbol{H})^T + \mathrm{E}\left\{\widetilde{\mathbf{x}}\boldsymbol{v}^T\right\}\boldsymbol{K}^T.\end{aligned}$$

The solution of (3.29) can be written as

$$\widetilde{\mathbf{x}}(t) = \boldsymbol{\Phi}(t,t_0)\widetilde{\mathbf{x}}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\{\boldsymbol{K}(\tau)\boldsymbol{v}(\tau) - [\boldsymbol{G}(\tau) - \boldsymbol{K}(\tau)\boldsymbol{H}(\tau)]\,\boldsymbol{w}(\tau)\}_{\tau}\,\mathrm{d}\tau$$

where $\boldsymbol{\Phi}(t,\tau)$ is the state transition matrix of $(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})$, therefore one can write

$$\begin{aligned}\mathrm{E}\left\{\widetilde{\mathbf{x}}(t)\boldsymbol{w}(t)^T\right\} &= \boldsymbol{\Phi}(t,t_0)\,\mathrm{E}\left\{\widetilde{\mathbf{x}}(t_0)\boldsymbol{w}(t)^T\right\} \\ &+ \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\boldsymbol{K}(\tau)\,\mathrm{E}\left\{\boldsymbol{v}(\tau)\boldsymbol{w}(t)^T\right\}\mathrm{d}\tau \\ &- \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\,[\boldsymbol{G}(\tau) - \boldsymbol{K}(\tau)\boldsymbol{H}(\tau)]\,\mathrm{E}\left\{\boldsymbol{w}(\tau)\boldsymbol{w}(t)^T\right\}\mathrm{d}\tau.\end{aligned} \tag{A.2}$$

Since $\mathbf{x}(t_0)$ and $\boldsymbol{w}(t)$ were assumed to be independent, $\mathrm{E}\left\{\widetilde{\mathbf{x}}(t_0)\boldsymbol{w}(t)^T\right\} = \boldsymbol{\varnothing}_{q\times w}$ and the first term in (A.2) disappears. Using (3.27), equation (A.2) reduces to

$$\begin{aligned}\mathrm{E}\left\{\widetilde{\mathbf{x}}\boldsymbol{w}^T\right\} &= \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\left\{\boldsymbol{K}(\tau)\boldsymbol{M}(t) - [\boldsymbol{G}(\tau) - \boldsymbol{K}(\tau)\boldsymbol{H}(\tau)]\,\boldsymbol{Q}(t)\right\}\delta(t-\tau)\,\mathrm{d}\tau \\ &= \frac{1}{2}\boldsymbol{K}\boldsymbol{M} - \frac{1}{2}\left(\boldsymbol{G} - \boldsymbol{K}\boldsymbol{H}\right)\boldsymbol{Q}\end{aligned}$$

because the integral interval stops *exactly* at the impulse position and therefore we consider only half of the impulse weight. A similar strategy can be used to show that

$$\mathrm{E}\left\{\widetilde{\mathbf{x}}\boldsymbol{v}^T\right\} = \frac{1}{2}\boldsymbol{K}\boldsymbol{R} - \frac{1}{2}\left(\boldsymbol{G} - \boldsymbol{K}\boldsymbol{H}\right)\boldsymbol{M}^T$$

so that one can conclude

$$\begin{aligned}\dot{\boldsymbol{\Sigma}} = &\left(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C}\right)\boldsymbol{\Sigma} + \boldsymbol{\Sigma}(\boldsymbol{A} - \boldsymbol{K}\boldsymbol{C})^T + \left(\boldsymbol{G} - \boldsymbol{K}\boldsymbol{H}\right)\boldsymbol{Q}(\boldsymbol{G} - \boldsymbol{K}\boldsymbol{H})^T \\ &+ \boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^T - \boldsymbol{K}\boldsymbol{M}(\boldsymbol{G} - \boldsymbol{K}\boldsymbol{H})^T - \left(\boldsymbol{G} - \boldsymbol{K}\boldsymbol{H}\right)\boldsymbol{M}^T\boldsymbol{K}^T.\end{aligned} \tag{A.3}$$

A.1.2 Derivation of the optimal Kalman Filter gain

Given the matrix differential equation (A.3) with the initial condition $\boldsymbol{\Sigma}(t_0) = \boldsymbol{\Sigma}_0$, we seek to find the optimal gain matrix $\boldsymbol{K}(t)$, $t \in [t_0, t_f]$ that minimizes, at the terminal time t_f, the cost functional

$$J = \mathrm{E}\left\{\widetilde{\mathbf{x}}(t_f)^T\widetilde{\mathbf{x}}(t_f)\right\} = \mathrm{E}\left\{\mathrm{tr}\left[\widetilde{\mathbf{x}}(t_f)\widetilde{\mathbf{x}}(t_f)^T\right]\right\} = \mathrm{tr}\left(\boldsymbol{\Sigma}(t_f)\right),$$

where $\mathrm{tr}\left(\cdot\right)$ indicates the trace operation. As suggested in [AT67], this problem can be solved by regarding it as an optimal control problem where the elements of matrix $\boldsymbol{K}(t)$ are the "control variables" to be optimized w.r.t. a cost functional defined in terms of the "state variables" given by the elements of $\boldsymbol{\Sigma}(t)$. One can then resort on the Pontryagin's minimum principle [PBGM62] to find the optimal

solution. To this end we define a *co-state* matrix $\boldsymbol{P}(t) \in \mathbb{R}^{q\times m}$ and the Hamiltonian function

$$\mathcal{H}(t) = \operatorname{tr}\left[\dot{\boldsymbol{\Sigma}}(t)\boldsymbol{P}(t)^T\right]$$

and the we impose the Pontryagin's necessary optimality conditions

$$\left.\frac{\partial \mathcal{H}(t)}{\partial \boldsymbol{K}(t)}\right|_{\substack{\boldsymbol{K}(t)=\mathring{\boldsymbol{K}}(t)\\ \boldsymbol{\Sigma}(t)=\mathring{\boldsymbol{\Sigma}}(t)}} = \boldsymbol{\varnothing}_{q\times m} \tag{A.4a}$$

$$\dot{\mathring{\boldsymbol{P}}}(t) = -\left.\frac{\partial \mathcal{H}(t)}{\partial \boldsymbol{\Sigma}(t)}\right|_{\substack{\boldsymbol{K}(t)=\mathring{\boldsymbol{K}}(t)\\ \boldsymbol{\Sigma}(t)=\mathring{\boldsymbol{\Sigma}}(t)}} \tag{A.4b}$$

$$\mathring{\boldsymbol{P}}(t_f) = \left.\frac{\partial J}{\partial \boldsymbol{\Sigma}(t_f)}\right|_{\substack{\boldsymbol{K}(t)=\mathring{\boldsymbol{K}}(t)\\ \boldsymbol{\Sigma}(t)=\mathring{\boldsymbol{\Sigma}}(t)}} \tag{A.4c}$$

where $\mathring{(\cdot)}$ indicates the optimal value of the corresponding quantity and we used the concept of gradient matrix as in [AT67], e.g.

$$\frac{\partial \mathcal{H}(t)}{\partial \boldsymbol{K}(t)} = \begin{bmatrix} \frac{\partial \mathcal{H}(t)}{\partial K_{1,1}(t)} & \cdots & \frac{\partial \mathcal{H}(t)}{\partial K_{1,m}(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{H}(t)}{\partial K_{q,1}(t)} & \cdots & \frac{\partial \mathcal{H}(t)}{\partial K_{q,m}(t)} \end{bmatrix} \in \mathbb{R}^{q\times m}.$$

From (A.3) and using the matrix differentiation rules in [AT67], (A.4b) returns the matrix Ordinary Differential Equation (ODE)

$$\dot{\mathring{\boldsymbol{P}}}(t) = -\mathring{\boldsymbol{P}}(t)^T\left[\boldsymbol{A}(t) - \mathring{\boldsymbol{K}}(t)\boldsymbol{C}(t)\right] - \left[\boldsymbol{A}(t) - \mathring{\boldsymbol{K}}(t)\boldsymbol{C}(t)\right]^T\mathring{\boldsymbol{P}}(t). \tag{A.5}$$

Moreover, from (A.4c), one obtains that $\mathring{\boldsymbol{P}}(t_f) = \boldsymbol{I}_q$. Since (A.5) is linear and $\mathring{\boldsymbol{P}}(t_f)$ is symmetric and positive definite, then one also has $\mathring{\boldsymbol{P}}(t) = \mathring{\boldsymbol{P}}(t)^T \succ 0, \forall t \in [t_0,\, t_f]$. Finally, using again (A.3), the optimality condititon (A.4a) results in

$$\mathring{\boldsymbol{P}}\left[-\boldsymbol{\Sigma}\boldsymbol{C}^T - \boldsymbol{G}\left(\boldsymbol{Q}\boldsymbol{H}^T + \boldsymbol{M}^T\right) + \mathring{\boldsymbol{K}}\left(\boldsymbol{R} + \boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^T + \boldsymbol{H}\boldsymbol{M}^T + \boldsymbol{M}\boldsymbol{H}^T\right)\right] = \boldsymbol{\varnothing}_{q\times m}.$$

Since we have already demonstrated that $\mathring{\boldsymbol{P}}(t)$ is positive definite, we canclude that the optimal gain is given by

$$\mathring{\boldsymbol{K}} = \left[\boldsymbol{\Sigma}\boldsymbol{C}^T + \check{\boldsymbol{M}}\right]\check{\boldsymbol{R}}^{-1} \tag{A.6}$$

with

$$\check{\boldsymbol{M}} = \boldsymbol{G}\left[\boldsymbol{Q}\boldsymbol{H}^T + \boldsymbol{M}^T\right],$$
$$\check{\boldsymbol{R}} = \boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^T + \boldsymbol{R} + \boldsymbol{H}\boldsymbol{M}^T + \boldsymbol{M}\boldsymbol{H}^T.$$

Note that the Pontryagin's minimum principle imposes only *necessary* optimality conditions. Nevertheless, in the case of Kalman filtering, these conditions are also *sufficient* to prove optimality as discussed in [AT67].

A.2 Dynamics of the weighted image moments

Let

$$\begin{cases} m_{ij}^x = \sum_{k=1}^{\infty} \frac{\partial w}{\partial x}(x_k,\, y_k,\, t-t_k) x_k^i y_k^j \\ m_{ij}^y = \sum_{k=1}^{\infty} \frac{\partial w}{\partial y}(x_k,\, y_k,\, t-t_k) x_k^i y_k^j \\ m_{ij}^t = \sum_{k=1}^{\infty} \frac{\partial w}{\partial t}(x_k,\, y_k,\, t) x_k^i y_k^j \end{cases} \tag{A.7}$$

and $\boldsymbol{\chi} = \boldsymbol{n}/d = (A,\, B,\, C)$. By leveraging on the developments of [TC05], the dynamics of the $(i,\, j)$-th weighted moment (4.49) is given by

$$\dot{m}_{ij} = [m_{vx}\; m_{vy}\; m_{vz}\; m_{wx}\; m_{wy}\; m_{wz}] \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{\omega} \end{bmatrix} + m_{ij}^t \tag{A.8}$$

with

$$\begin{aligned} m_{vx} &= A(-i m_{i,j} - m_{i+1,j}^x) + B(-i m_{i-1,j+1} - m_{i,j+1}^x) \\ &\quad + C(-i m_{i-1,j} - m_{i,j}^x) \\ m_{vy} &= A(-j m_{i+1,j-1} - m_{i+1,j}^y) + B(-j m_{i,j} - m_{i,j+1}^y) \\ &\quad + C(-j m_{i,j-1} - m_{i,j}^y) \\ m_{vz} &= A(j m_{i+1,j} + i m_{i+1,j} + m_{i+2,j}^x + m_{i+1,j+1}^y) \\ &\quad + B(i m_{i,j+1} + j m_{i,j+1} + m_{i+1,j+1}^x + m_{i,j+2}^y) + \\ &\quad + C(j m_{i,j} + i m_{i,j} + m_{i+1,j}^x + m_{i,j+1}^y) \\ m_{wx} &= j m_{i,j+1} + i m_{i,j+1} + j m_{i,j-1} + m_{i+1,j+1}^x + m_{i,j}^y + m_{i,j+2}^y \\ m_{wy} &= -\, i m_{i+1,j} - j m_{i+1,j} - i m_{i-1,j} - m_{i,j}^x - m_{i+1,j+1}^y - m_{i+2,j}^x \\ m_{wz} &= i m_{i-1,j+1} - j m_{i+1,j-1} - m_{i+1,j}^y + m_{i,j+1}^x. \end{aligned}$$

We can note that the dynamics (A.8) involves moments of order up to $(i+j+2)$ associated to the terms m_{ij}^x and m_{ij}^y. Also, it is easy to check that in the unweighted case ($w \equiv 1$ and $m_{ij}^x = m_{ij}^y = m_{ij}^t \equiv 0$), one obviously recovers the classical moment dynamics reported in [TC05].

A.3 Time-derivative of the limb surface parameters for a spherical target

Differentiation of $\boldsymbol{\chi}$ from (4.69) w.r.t. time yields

$$\dot{\boldsymbol{\chi}} = \frac{\dot{\boldsymbol{p}}_0 K - \boldsymbol{p}_0 \dot{K}}{K^2} = \frac{\dot{\boldsymbol{p}}_0 K - 2\boldsymbol{p}_0 \boldsymbol{p}_0^T \dot{\boldsymbol{p}}_0}{K^2} \tag{A.9}$$

which, being $\dot{\boldsymbol{p}}_0 = -\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{p}_0$ and exploiting the property $\boldsymbol{p}_0^T [\boldsymbol{\omega}]_\times \boldsymbol{p}_0 = 0$, can be rewritten as

$$\dot{\boldsymbol{\chi}} = -\frac{\boldsymbol{v}}{K} - \frac{[\boldsymbol{\omega}]_\times \boldsymbol{p}_0}{K} + 2\frac{\boldsymbol{p}_0 \boldsymbol{p}_0^T \boldsymbol{v}}{K^2} = -\frac{\boldsymbol{v}}{K} - [\boldsymbol{\omega}]_\times \boldsymbol{\chi} + 2\boldsymbol{\chi}\boldsymbol{\chi}^T \boldsymbol{v}. \tag{A.10}$$

Letting $s_z = Z_0/R$ ($s_z > 1$), one also has

$$\boldsymbol{\chi}^T \boldsymbol{\chi} - \frac{1}{s_z^2}\chi_z^2 = \frac{X_0^2 + Y_0^2 + Z_0^2}{K^2} - \frac{R^2}{Z_0^2}\frac{Z_0^2}{K^2} = \frac{1}{K}. \tag{A.11}$$

This then shows how $1/K$ can be expressed in terms of $\boldsymbol{\chi}$ and of s_z^2, with s_z being directly obtainable from image measurements, as shown in [FC09] and further discussed in (4.70). Having estimated $\boldsymbol{\chi}$, one can consequently retrieve $\boldsymbol{p}_0 = \boldsymbol{\chi} K$ and $R = \sqrt{\boldsymbol{p}_0^T \boldsymbol{p}_0 - K}$.

A.4 Estimation of the limb surface parameter for a cylindrical target

In order to estimate the parameters of the limb surface associated to a cylindrical object, one could consider as measurement the $2 + 2$ angle-distance parameters (θ_i, ρ_i) of the straight lines resulting from the projection of the cylinder on the image plane. From [CBBJ96, Cha04], the interaction matrix in this case is given by:

$$\boldsymbol{L} = \begin{bmatrix} \lambda_{\rho_1}\mathrm{c}_1 & \lambda_{\rho_1}\mathrm{s}_1 & -\lambda_{\rho_1}\rho_1 & (1+\rho_1^2)\mathrm{s}_1 & -(1+\rho_1^2)\mathrm{c}_1 & 0 \\ \lambda_{\theta_1}\mathrm{c}_1 & \lambda_{\theta_1}\mathrm{s}_1 & -\lambda_{\theta_1}\rho_1 & -\rho_1\mathrm{c}_1 & -\rho_1\mathrm{s}_1 & -1 \\ \lambda_{\rho_2}\mathrm{c}_2 & \lambda_{\rho_2}\mathrm{s}_2 & -\lambda_{\rho_2}\rho_2 & (1+\rho_2^2)\mathrm{s}_2 & -(1+\rho_2^2)\mathrm{c}_2 & 0 \\ \lambda_{\theta_2}\mathrm{c}_2 & \lambda_{\theta_2}\mathrm{s}_2 & -\lambda_{\theta_2}\rho_2 & -\rho_2\mathrm{c}_2 & -\rho_2\mathrm{s}_2 & -1 \end{bmatrix} \tag{A.12}$$

with $\mathrm{s}_i = \sin\theta_i$, $\mathrm{c}_i = \cos\theta_i$, and

$$\begin{cases} \lambda_{\rho_i} &= -\left(\chi_x \rho_i \mathrm{c}_i + \chi_y \rho_i \mathrm{s}_i + \chi_z\right) \\ \lambda_{\theta_i} &= \chi_y \mathrm{c}_i - \chi_x \mathrm{s}_i \end{cases}. \tag{A.13}$$

Therefore, being (A.12–A.13) linear in the unknown $\boldsymbol{\chi}$, one can again apply the estimation scheme (3.11) with $\boldsymbol{s}$ taken as the vector of measured quantities on the image plane, i.e., $\boldsymbol{s} = (\rho_1, \theta_1, \rho_2, \theta_2)$.

As for the dynamics of $\boldsymbol{\chi}$, since (4.69) still holds for a cylindrical object (see [Cha04]), one can again exploit (A.9) with, however, in this case

$$\dot{\boldsymbol{p}}_0 = -\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right)\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{p}_0$$

and thus

$$\dot{\boldsymbol{\chi}} = -\left(\frac{1}{K}\boldsymbol{I}_3 - 2\boldsymbol{\chi}\boldsymbol{\chi}^T\right)\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right)\boldsymbol{v} - [\boldsymbol{\omega}]_\times \boldsymbol{\chi}.$$

Finally, one can invoke (A.11) in order to express $1/K$ as a function of $\boldsymbol{\chi}$ and s_z^2, with s_z being the third element of vector $\boldsymbol{s}$ in (4.81).

A.5 Derivation of equation (4.84)

We note that the cylinder axis $\boldsymbol{a}$ can be determined by the intersection of two planes $\mathcal{P}_i : \boldsymbol{n}_i^T \boldsymbol{X} - d_i = 0$, $i = 1, 2$, with

$$\boldsymbol{n}_1 = \frac{[\boldsymbol{a}]_\times \boldsymbol{p}_0}{\|\boldsymbol{p}_0\|},\ d_1 = 0, \quad \boldsymbol{n}_2 = -\frac{\boldsymbol{p}_0}{\|\boldsymbol{p}_0\|},\ d_2 = \|\boldsymbol{p}_0\|, \tag{A.14}$$

see Fig. 4.5. In particular, plane $\mathcal{P}_1$ passes through the camera optical center, it is orthogonal to plane $\mathcal{P}_2$, and both planes contain the axis $\boldsymbol{a}$ passing through $\boldsymbol{p}_0$ (by construction).

Since $R\boldsymbol{s} = \boldsymbol{p}_0$ and $\boldsymbol{p}_0$ belongs to the cylinder axis $\boldsymbol{a}$, we have $R\boldsymbol{n}_i^T \boldsymbol{s} - d_i = 0$, $i = 1, 2$ (the point $R\boldsymbol{s}$ belongs to both planes $\mathcal{P}_i$). Taking the time derivative of these latter constraints (with $R = const$), one has

$$\boldsymbol{n}_i^T \dot{\boldsymbol{s}} = \frac{1}{R}\dot{d}_i - \boldsymbol{s}^T \dot{\boldsymbol{n}}_i, \quad i = 1, 2. \tag{A.15}$$

Since $\dot{\boldsymbol{n}}_i = [\boldsymbol{n}_i]_\times \boldsymbol{\omega}$ and $\dot{d}_i = \boldsymbol{n}_i^T \boldsymbol{v}$ (see [RDO08]), eq. (A.15) can be rewritten as

$$\boldsymbol{n}_i^T \dot{\boldsymbol{s}} = \frac{1}{R}\boldsymbol{n}_i^T \boldsymbol{v} - \boldsymbol{s}^T [\boldsymbol{n}_i]_\times \boldsymbol{\omega}, \quad i = 1, 2. \tag{A.16}$$

Finally, from $\boldsymbol{a}^T \boldsymbol{p}_0 = 0$ and $\boldsymbol{p}_0 = R\boldsymbol{s}$ we have $\boldsymbol{a}^T \boldsymbol{s} = 0$ implying that

$$\boldsymbol{a}^T \dot{\boldsymbol{s}} = -\boldsymbol{s}^T \dot{\boldsymbol{a}} = -\boldsymbol{s}^T [\boldsymbol{a}]_\times \boldsymbol{\omega}. \tag{A.17}$$

We now note that equations (A.16–A.17) provide three linear constraints for $\dot{\boldsymbol{s}}$ which, by using (A.14), can be rearranged in matrix form as the following linear system

$$\begin{bmatrix} \dfrac{\boldsymbol{p}_0^T}{\|\boldsymbol{p}_0\|} \\ \boldsymbol{a}^T \\ \dfrac{\left([\boldsymbol{a}]_\times \boldsymbol{p}_0\right)^T}{\|\boldsymbol{p}_0\|} \end{bmatrix} \dot{\boldsymbol{s}} = \frac{1}{R} \begin{bmatrix} \dfrac{\boldsymbol{p}_0^T}{\|\boldsymbol{p}_0\|}\boldsymbol{v} \\ -\boldsymbol{p}_0^T [\boldsymbol{a}]_\times \boldsymbol{\omega} \\ \|\boldsymbol{p}_0\| \boldsymbol{a}^T \boldsymbol{\omega} + \dfrac{\left([\boldsymbol{a}]_\times \boldsymbol{p}_0\right)^T}{\|\boldsymbol{p}_0\|}\boldsymbol{v} \end{bmatrix}. \tag{A.18}$$

It is easy to verify that the 3×3 matrix on the left hand side of (A.18) is orthonormal: by then solving (A.18) for $\dot{\boldsymbol{s}}$ and performing some simplifications we finally obtain the sought result

$$\dot{\boldsymbol{s}} = \begin{bmatrix} -\dfrac{1}{R}\left(\boldsymbol{I}_3 - \boldsymbol{a}\boldsymbol{a}^T\right) & [\boldsymbol{s}]_\times \end{bmatrix} \mathbf{u}.$$

A.5.1 Proof of Prop. 6.2

Let $\boldsymbol{\Phi}(t) = [\Phi_{ij}(t)] \in \mathbb{R}^{2\times2}$ be the state-transition matrix associated to the linear time-invariant system (6.10). From classical system theory [Kai98], we have

$$\nu_{\|\boldsymbol{e}\|}(t) = \Phi_{11}(t - t_1)\nu_1 + \Phi_{12}(t - t_1)\dot{\nu}_1, \quad \forall t \geq t_1, \tag{A.19}$$

where we set $\nu_1 = \nu(t_1)$ and $\dot{\nu}_1 = \dot{\nu}(t_1)$ for simplicity.

We also note that (6.11) is governed, component-wise, by the same dynamics of (6.10). Therefore, the solution of (6.11) is

$$\boldsymbol{e}^*(t) = \Phi_{11}(t-t_1)\boldsymbol{e}_1 + \Phi_{12}(t-t_1)\dot{\boldsymbol{e}}_1, \quad \forall t \geq t_1, \tag{A.20}$$

where, again, $\boldsymbol{e}_1 = \boldsymbol{e}(t_1)$ and $\dot{\boldsymbol{e}}_1 = \dot{\boldsymbol{e}}(t_1)$.

If $\boldsymbol{e}_1$ and $\dot{\boldsymbol{e}}_1$ are parallel then (6.12) holds: assuming $\boldsymbol{e}_1$ and $\dot{\boldsymbol{e}}_1$ are parallel, vector $\dot{\boldsymbol{e}}_1$ can be expressed as

$$\dot{\boldsymbol{e}}_1 = \|\dot{\boldsymbol{e}}_1\| \frac{\boldsymbol{e}_1}{\|\boldsymbol{e}_1\|} = \|\dot{\boldsymbol{e}}_1\| \frac{\boldsymbol{e}_1}{\nu_1}. \tag{A.21}$$

Therefore, (A.20) becomes

$$\boldsymbol{e}^*(t) = \left(\Phi_{11}(t-t_1) + \Phi_{12}(t-t_1)\frac{\|\dot{\boldsymbol{e}}_1\|}{\nu_1}\right)\boldsymbol{e}_1, \quad \forall t \geq t_1,$$

resulting in an error norm $\|\boldsymbol{e}^*(t)\|$

$$\begin{aligned}\|\boldsymbol{e}^*(t)\| = \nu^*(t) &= \left(\Phi_{11}(t-t_1) + \Phi_{12}(t-t_1)\frac{\|\dot{\boldsymbol{e}}_1\|}{\nu_1}\right)\|\boldsymbol{e}_1\| \\ &= \left(\Phi_{11}(t-t_1) + \Phi_{12}(t-t_1)\frac{\|\dot{\boldsymbol{e}}_1\|}{\nu_1}\right)\nu_1 \\ &= \Phi_{11}(t-t_1)\nu_1 + \Phi_{12}(t-t_1)\|\dot{\boldsymbol{e}}_1\|, \quad \forall t \geq t_1.\end{aligned} \tag{A.22}$$

Now, being $\nu = \|\boldsymbol{e}\|$ one has

$$\dot{\nu}_1 = \frac{\boldsymbol{e}_1^T\dot{\boldsymbol{e}}_1}{\nu_1} \tag{A.23}$$

which, exploiting (A.21), yields $\dot{\nu}_1 = \|\dot{\boldsymbol{e}}_1\|\boldsymbol{e}_1^T\boldsymbol{e}_1/\nu_1^2 = \|\dot{\boldsymbol{e}}_1\|$. Plugging $\|\dot{\boldsymbol{e}}_1\| = \dot{\nu}_1$ in (A.22) finally results in

$$\nu^*(t) = \Phi_{11}(t-t_1)\nu_1 + \Phi_{12}(t-t_1)\dot{\nu}_1, \quad \forall t \geq t_1,$$

thus showing that $\nu^*(t) \equiv \nu_{\|\boldsymbol{e}\|}(t)$, i.e. fulfilment of condition (6.12) as claimed.

If (6.12) holds then $\boldsymbol{e}_1$ and $\dot{\boldsymbol{e}}_1$ are parallel: from (A.19–A.20) we have (omitting the time dependency for brevity)

$$\nu_{\|\boldsymbol{e}\|}^2 = \Phi_{11}^2\nu_1^2 + 2\Phi_{11}\Phi_{12}\nu_1\dot{\nu}_1 + \Phi_{12}^2\dot{\nu}_1^2 \tag{A.24}$$

and

$$\begin{aligned}\|\boldsymbol{e}^*(t)\|^2 =& \Phi_{11}^2\boldsymbol{e}_1^T\boldsymbol{e}_1 + 2\Phi_{11}\Phi_{12}\boldsymbol{e}_1^T\dot{\boldsymbol{e}}_1 + \Phi_{12}^2\dot{\boldsymbol{e}}_1^T\dot{\boldsymbol{e}}_1 \\ =& \Phi_{11}^2\nu_1^2 + 2\Phi_{11}\Phi_{12}\nu_1\dot{\nu}_1 + \Phi_{12}^2\dot{\boldsymbol{e}}_1^T\dot{\boldsymbol{e}}_1\end{aligned} \tag{A.25}$$

where (A.23) was used. By imposing condition (6.12) to (A.24–A.25) we then have

$$\nu_{\|e\|}^2 \equiv \|\boldsymbol{e}^*(t)\|^2 \implies \Phi_{12}^2 \dot{\nu}_1^2 \equiv \Phi_{12}^2 \dot{\boldsymbol{e}}_1^T \dot{\boldsymbol{e}}_1 \implies \dot{\nu}_1 = \|\dot{\boldsymbol{e}}_1\|. \tag{A.26}$$

Since $\dot{\nu}_1$ is just the projection of vector $\dot{\boldsymbol{e}}_1$ along the direction of $\boldsymbol{e}_1$ (see again (A.23)), condition (A.26) necessarily requires vectors $\boldsymbol{e}_1$ and $\dot{\boldsymbol{e}}_1$ to be parallel as claimed.

A.5.2 Properties of $E(t)$

Relationship between $E(t)$ and the estimation error $\widetilde{\boldsymbol{\chi}}(t)$: if $\sigma_1^2(t) > 0$ during the camera motion then $E(t) \equiv 0$ iff $\|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$ (i.e., the estimation has converged) and $E(t) > 0$ otherwise (i.e., the estimation has not yet converged).

In order to prove this claim, we start by showing the following facts:

Proposition A.1. *If the camera motion is exciting (i.e., $\sigma_1^2(t) > 0$), then $\|\widetilde{\boldsymbol{s}}(t)\| \equiv 0 \iff \|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$ and $\|\widetilde{\boldsymbol{s}}(t)\| > 0$ a.e. $\iff \|\widetilde{\boldsymbol{\chi}}(t)\| > 0$ a.e.*

Proof. Being σ_1^2 the smallest eigenvalue of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$, the hypothesis $\sigma_1^2 > 0$ implies full row-rankness of the (low-rectangular) $p \times m$ matrix $\boldsymbol{\Omega}$. Considering now the error dynamics (3.12), the following holds

- $\|\widetilde{\boldsymbol{s}}(t)\| \equiv 0 \implies \|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$: if $\|\widetilde{\boldsymbol{s}}(t)\| \equiv 0$ then $\widetilde{\boldsymbol{s}}(t) \equiv \boldsymbol{\emptyset}_m$ and $\dot{\widetilde{\boldsymbol{s}}}(t) \equiv \boldsymbol{\emptyset}_m$. The first row of (3.12) then reduces to $\boldsymbol{\Omega}^T \widetilde{\boldsymbol{\chi}} \equiv \boldsymbol{\emptyset}_m$ which implies $\|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$ since matrix $\boldsymbol{\Omega}$ is full row-rank by hypothesis;

- $\|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0 \implies \|\widetilde{\boldsymbol{s}}(t)\| \equiv 0$: if $\|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$, the first row of (3.8) reduces to $\dot{\widetilde{\boldsymbol{s}}} = -\boldsymbol{H}\widetilde{\boldsymbol{s}}$. Being the matrix gain $\boldsymbol{H}$ positive definite, it follows that, at steady-state, the only possible solution is $\widetilde{\boldsymbol{s}}(t) \equiv \boldsymbol{\emptyset}_m$.

These two implications then prove the first item of the Proposition, that is, $\|\widetilde{\boldsymbol{s}}(t)\| \equiv 0 \iff \|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$. The proof is concluded by noting that the remaining two (reverse) implications $\|\widetilde{\boldsymbol{\chi}}(t)\| > 0$ a.e. $\implies \|\widetilde{\boldsymbol{s}}(t)\| > 0$ a.e. and $\|\widetilde{\boldsymbol{s}}(t)\| > 0$ a.e. $\implies \|\widetilde{\boldsymbol{\chi}}(t)\| > 0$ a.e. (needed for proving the second item of the Proposition) are just the logical negations the two ones listed above. □

Prop. A.1 can now be exploited for proving the initial main claim. Indeed, since $E(t)$ is defined as the moving average of signal $\|\widetilde{\boldsymbol{s}}(t)\|^2$ (see (6.15)), it follows that $E(t) = 0$ if $\|\widetilde{\boldsymbol{\chi}}(t)\| \equiv 0$ over (at least) the integration window T. Therefore, convergence of the estimation error $\widetilde{\boldsymbol{\chi}}(t)$ will necessarily make the quantity $E(t)$ vanish as desired. It now remains to show that the reverse condition $\|\widetilde{\boldsymbol{\chi}}(t)\| > 0$ a.e. $\implies E(t) > 0$ holds as well, i.e., that $E(t) = 0$ *only if* convergence of the estimation error has been reached. This again easily follows from Prop. A.1: since $\|\widetilde{\boldsymbol{\chi}}(t)\| > 0$

a.e. $\implies$ $\|\widetilde{\boldsymbol{s}}(t)\| > 0$ a.e., the moving average (6.15) over any non-infinitesimal integration window $T \geq \epsilon > 0$ will necessarily stay positive, thus implying that $E(t) > 0$.

We conclude with the following remarks: since $E(t) > 0$ as long as the estimation error has not converged, the adaptive gain $k_E(E)$ in (6.18) is also guaranteed to never vanish during the estimation transient (by properly placing, if needed, the minimum threshold $\underline{E}$). As a consequence, the optimization of the camera motion (i.e., of $\sigma_1^2(t)$) will always be active during phase 1). We also note that, in general, no special characterization is possible for the behavior of $E(t)$ during the estimation transient (apart from the above-mentioned condition $E(t) > 0$). Nevertheless, if $\sigma_1^2(t) \approx const > 0$ during motion, then the error system (3.12) behaves (in its dominant dynamics) as a second-order critically-damped linear system, with $\widetilde{\boldsymbol{\chi}}(t)$ playing the role of the 'position variables' and $\widetilde{\boldsymbol{s}}(t)$ that of 'velocity variables', see Sect. 4.2. In this situation, $\|\widetilde{\boldsymbol{s}}(t)\|^2$ (and, thus, $E(t)$ as well) will approximate a 'bell-shaped' profile with a monotonic increase towards a maximum value followed by a monotonic decrease towards zero. By looking at Figs. 7.6(d) and 7.7(d), it is worth noticing that this is indeed the profile followed by $E(t)$ during the active phases of all the reported experiments, since maximization of (6.17) does result (as a byproduct) in an approximately constant $\sigma_1^2(t) \approx const$.

Proof of bound (6.19): this bound can be easily proven by exploiting the pH interpretation of the error dynamics (3.8) briefly introduced in Sect. 4.1. The Hamiltonian function (4.3) decreases over time towards its global minimum at $(\widetilde{\boldsymbol{s}}, \widetilde{\boldsymbol{\chi}}) = (\boldsymbol{\emptyset}_m, \boldsymbol{\emptyset}_p)$, provided the usual hypothesis of an exciting camera motion ($\sigma_1^2(t) > 0$) is satisfied. Therefore, along the trajectories of (3.8) it is

$$0 \leq \mathcal{H}(\widetilde{\boldsymbol{s}}(t), \widetilde{\boldsymbol{\chi}}(t)) \leq \mathcal{H}(\widetilde{\boldsymbol{s}}(t_0), \widetilde{\boldsymbol{\chi}}(t_0)), \qquad \forall t \geq t_0. \tag{A.27}$$

We now note that, being the feature vector $\boldsymbol{s}$ a measurable quantity, one can *always* initialize $\widehat{\boldsymbol{s}}(t_0) = \boldsymbol{s}(t_0)$ resulting in $\widetilde{\boldsymbol{s}}(t_0) = \boldsymbol{\emptyset}_m$. By employing this initialization (adopted in all the reported case studies), and exploiting (4.3–A.27), the following bound easily follows

$$\frac{1}{2}\|\widetilde{\boldsymbol{s}}(t)\|^2 \leq \mathcal{H}(\widetilde{\boldsymbol{s}}(t), \widetilde{\boldsymbol{\chi}}(t)) \leq \mathcal{H}(\widetilde{\boldsymbol{s}}(t_0), \widetilde{\boldsymbol{\chi}}(t_0)) = \frac{1}{2\alpha}\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2. \tag{A.28}$$

The proof is then completed by noting that, from standard calculus,

$$E(t) = \frac{1}{T}\int_{t-T}^{t} \widetilde{\boldsymbol{s}}^T(\tau)\widetilde{\boldsymbol{s}}(\tau)\mathrm{d}\tau \leq \max_{\tau\in[t-T,\,t]} \left(\widetilde{\boldsymbol{s}}^T(\tau)\widetilde{\boldsymbol{s}}(\tau)\right) \leq \frac{\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2}{\alpha}. \tag{A.29}$$

For the interested reader, this result can be given an interesting energetic interpretation as a consequence of the pH structure of the dynamics (3.8). In this

interpretation, the Hamiltonian (4.3) represents the total energy of system (3.8) and consists of two energy storages: $\mathcal{H}_m(\widetilde{\boldsymbol{s}}) = \frac{1}{2}\widetilde{\boldsymbol{s}}^T\widetilde{\boldsymbol{s}}$ (the energy of the measurable states) and $\mathcal{H}_u(\widetilde{\boldsymbol{\chi}}) = \frac{1}{2\alpha}\widetilde{\boldsymbol{\chi}}^T\widetilde{\boldsymbol{\chi}}$ (the energy of the unmeasurable states). Matrix $\boldsymbol{\Omega}$ in (3.8) modulates the (power-preserving) interconnection between the two storages, while matrix $\boldsymbol{H}$ implements a dissipative action on the storage $\mathcal{H}_m(\widetilde{\boldsymbol{s}})$. Full-rankness of matrix $\boldsymbol{\Omega}\boldsymbol{\Omega}^T$ (i.e., the usual condition $\sigma_1^2(t) > 0$) then translates into requiring a persistent energy exchange among $\mathcal{H}_m(\widetilde{\boldsymbol{s}})$ and $\mathcal{H}_u(\widetilde{\boldsymbol{\chi}})$ until full depletion of the initial stored energy $\mathcal{H}(\widetilde{\boldsymbol{s}}(t_0), \widetilde{\boldsymbol{\chi}}(t_0))$ via the dissipation induced by $\boldsymbol{H}$. The bound (A.28) then simply states that, over time, the energy stored in $\mathcal{H}_m(\widetilde{\boldsymbol{s}})$ cannot exceed the total initial energy at time $t = t_0$ which, thanks to the initialization $\widehat{\boldsymbol{s}}(t_0) = \boldsymbol{s}(t_0)$, takes the expression $\frac{1}{2\alpha}\|\widetilde{\boldsymbol{\chi}}(t_0)\|^2$.

We conclude by noting that (A.27) (and, as a consequence, (A.28–A.29) as well) is obviously no longer valid in presence of (unmodeled) perturbations such as the several target displacemets discussed in Sect. 7.2. In this case, an external amount of energy could (in general) be injected into system (3.8) with a consequent increase of the total energy $\mathcal{H}(t)$ and violation of bound (A.27) (a violation which, nevertheless, did not occur in the experimental results of Sect. 7.2 because of the limited 'extra' energy produced by the unmodeled target motion).

Appendix B

Dense photometric structure estimation from motion

All of the structure estimation problems analyzed so far in this thesis concerned a limited number of 3-D geometric structures (e.g. points, planes and so on) whose projection was assumed to correspond to a certain (limited) amount of visual features that could be identified and tracked on each image of a video sequence. In practice feature identification, tracking and matching are significantly complex tasks. Although very effective strategies have been proposed in the literature (which explains the successful experimental results reported in this thesis) the process is still prone to failure. In this appendix we wish to report some preliminary results in the context of *photometric structure estimation from motion*, where the term "photometric" is used with reference to a set of visual estimation and control techniques that are based on the *direct* use of the images (intended as a pixel light intensity map) without any (or limiting as much as possible the need for) preliminary processing step. Photometric techniques have been shown to be effective for the visual control of robotic manipulators in [CM11, BCM13]. As for the estimation problem, a certain amount of literature has been produced in this context, especially concerning the evaluation of optical flow from image sequences. The seminal works [HS81] and [LK81] demonstrated, for example, how a dense optical flow can be calculated using variational methods or local least squares optimization techniques respectively.

Note that, in principle, once the optical flow has been calculated from an image sequence, then, if the camera velocity is known, a dense depth map can be estimated by, e.g., inverting (2.10) (in a least-squares sense). One difficulty that arises when trying to estimate optical flow, however, is that the problem is highly ill-posed due to the fact that the brightness constancy condition (typically used as a model for the image brightness propagation) imposes only one constraint which is not sufficient to

completely determine the two dimensional optical flow vector field. The introduction of priors or regularization terms, usually based on the L_2 or L_1 norms of the optical flow gradient, is therefore necessary, see [BJB94, BSL+11].

Variational methods can also be used to estimate the disparity map *directly* from two consecutive images and knowledge of the camera velocity as shown in [ZAR12a]. Even in this case, regularization terms based on the L_2 or L_1 norm of the disparity map gradient can be used to "fill-in" the areas where the problem is ill-conditioned due to scarcely textured images, and to increase robustness w.r.t. noise.

Rough disparity measurements based on differentiating consecutive images can be incorporated in the calculation of innovation terms in asymptotic incremental depth estimators based on Kalman Filters [MSK88] or deterministic observers [ZAR12b]. The prediction, in this case, is based on the known camera velocity. This strategy allows to reduce the effect of noise and discretization errors.

All these methods require the differentiation of consecutive images over time which potentially introduces noise. An improvement in this sense can be obtained by considering a larger number of sampled images when performing the differentiation [BJB94].

Another possibility is to use region-based matching methods [BJB94] that try to find the image displacement that maximizes some similarity measure between the current image and the previous one. Similarity criteria include Sum of Squared Difference (SSD) and mutual information [XMX+10].

The main difference between the method discussed in this appendix and the ones mentioned above is that we avoid a direct differentiation of the video sequence, but we still keep the calculation local in time (we only consider the last received image in the calculation of the update term) by exploiting a recursive observer similar to the ones used in the rest of the thesis. Our strategy is, therefore, mainly inspired to the one reported in [AAM14] for the real-time computation of a dense optical flow. By doing so, we expect to obtain an estimation less sensitive to discretization noise, especially at a higher frame rate, but still very efficient (we do not use any batch technique).

The rest of the appendix is organized as follows. First we derive the system of partial differential equations governing the dynamics of the image and the disparity map using a planar (Appendix B.1) or a spherical (Appendix B.2) projection model. In Appendix B.3 we propose an asymptotic observer for estimating the dense disparity map using the photometric information and the camera velocity and we informally discuss observability and stability issues. In Appendix B.4 we introduce a strategy for dealing with the lack of observability in certain areas of the image by introducing an adaptive smoothing term in the estimated disparity

map propagation dynamics. Appendix B.5 discusses how to model and propagate depth discontinuities. In Appendix B.6 we provide details on the actual numerical implementation of the observer Partial Differential Equations (PDEs). Finally Appendix B.7 concludes the chapter with comments on current results and future research directions.

This work was done in collaboration with Prof. Robert Mahony at the Australian National University branch of the Australian Research Council Centre of Excellence for Robotic Vision[1]. We also wish to thank Juan David Adarve for his fruitful suggestions.

B.1 System dynamics with planar projection

As discussed in the introduction of this chapter, we now consider as measurement the luminance $Y(t, \boldsymbol{\pi})$ of a pixel located at a point $\boldsymbol{\pi}$ (in homogeneous coordinates) at a time t. This is a scalar mapping:

$$Y : \mathbb{R} \times \mathbb{P}^2 \mapsto \mathbb{R}_+, \quad (t, \boldsymbol{\pi}) \mapsto Y(t, \boldsymbol{\pi}).$$

The luminance map represents the flux of the (light) electro-magnetic power, at time t, through an infinitesimal area region of the image plane centered in $\boldsymbol{\pi}$. The value of $Y(t, \boldsymbol{\pi})$ is determined by both geometric (shape and position of the objects at time t) and physical (absorption, reflectivity and transmissivity of the materials as well as position and intensity of light sources) properties of the environment. The exact physics of the image formation process is extremely complex to model but different approximations have been proposed in the literature. In particular, we make the assumption that the materials in the scene obey a *Lambertian* reflectance model [Lam60, BJ03], i.e., they absorb and reflect (part of) the light that hits their surface with equal (isotropic) intensity in all directions. This simple model approximates reasonably well the physics of image formation for "matte" objects and has been used in many vision applications. Finally, we assume that the environment is static. A direct consequence of these choices is the constant brightness assumption

$$\frac{\mathrm{d}}{\mathrm{d}t} Y(t, \boldsymbol{\pi}) = 0, \tag{B.1}$$

that essentially states that, as the image of the environment moves (due to camera motion), its brightness does not change, see, e.g. [HS81]. We are interested in the way the intensity changes, in a fixed position on the image, due to camera motion, i.e. in the quantity:

$$\lim_{\Delta t \to 0} \frac{Y(t + \Delta t, \boldsymbol{\pi}) - Y(t, \boldsymbol{\pi})}{\Delta t} = \frac{\partial Y}{\partial t}. \tag{B.2}$$

[1] http://roboticvision.org/

Using (B.1), one can write:

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = \nabla_{\boldsymbol{\pi}} Y^T \dot{\boldsymbol{\pi}} + \frac{\partial Y}{\partial t} = 0 \iff \frac{\partial Y}{\partial t} = -\nabla_{\boldsymbol{\pi}} Y^T \dot{\boldsymbol{\pi}} \tag{B.3}$$

If $\boldsymbol{\pi}$ is associated with a point of constant luminance, then it follows that its time derivative is determined by the optic flow Φ (see (2.10))

$$\begin{aligned} \dot{\boldsymbol{\pi}} = \Phi(t, \boldsymbol{\pi}) = \Psi(t, \boldsymbol{\pi}) + \Theta(t, \boldsymbol{\pi}) &= \frac{1}{Z(t, \boldsymbol{\pi})} \psi(t, \boldsymbol{\pi}) + \Theta(t, \boldsymbol{\pi}) \\ &= \zeta(t, \boldsymbol{\pi}) \psi(t, \boldsymbol{\pi}) + \Theta(t, \boldsymbol{\pi}) \end{aligned} \tag{B.4}$$

where $Z(t, \boldsymbol{\pi})$ is the depth map, $\zeta(t, \boldsymbol{\pi})$ is its inverse, also called the *disparity map*, and $\Theta(t, \boldsymbol{\pi})$ and $\psi(t, \boldsymbol{\pi})$ are, respectively, the angular and scaled linear components of the optical flow, given by (see, again, (2.10)):

$$\begin{aligned} \psi(t, \boldsymbol{\pi}) &= [\boldsymbol{e}_3]_\times [\boldsymbol{\pi}]_\times \boldsymbol{v}(t) \\ \Theta(t, \boldsymbol{\pi}) &= -[\boldsymbol{e}_3]_\times [\boldsymbol{\pi}]_\times^2 \boldsymbol{\omega}(t). \end{aligned} \tag{B.5}$$

The depth and disparity maps must be considered as scalar functions

$$\begin{aligned} Z : t \times \mathbb{P}^2 &\mapsto \mathbb{R}_+, \quad (t, \boldsymbol{\pi}) \mapsto Z(t, \boldsymbol{\pi}) \\ \zeta : t \times \mathbb{P}^2 &\mapsto \mathbb{R}_+, \quad (t, \boldsymbol{\pi}) \mapsto \zeta(t, \boldsymbol{\pi}), \end{aligned}$$

constituting an *ego-centric* representation of the environment: for each direction $\boldsymbol{\pi}$ at each time t there is a ray radiating out from the optic center that intercepts the closest point of the environment in the 3-D point $\boldsymbol{p}(t, \boldsymbol{\pi}) = Z(t, \boldsymbol{\pi})\boldsymbol{\pi}$. Substituting (B.4) in (B.3) one finally obtains

$$\frac{\partial Y}{\partial t} = -\nabla_{\boldsymbol{\pi}} Y^T \Theta(t, \boldsymbol{\pi}) - \zeta(t, \boldsymbol{\pi}) \nabla_{\boldsymbol{\pi}} Y^T \psi(t, \boldsymbol{\pi}). \tag{B.6}$$

Now let us compute, similarly to (B.2), how the disparity map changes in time, in a fixed position on the image plane, due to camera motion, i.e. the quantity:

$$\lim_{\Delta t \to 0} \frac{\zeta(t + \Delta t, \boldsymbol{\pi}) - \zeta(t, \boldsymbol{\pi})}{\Delta t} = \frac{\partial \zeta}{\partial t}.$$

We start by noting that, as already discussed (4.27),

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = -\boldsymbol{e}_3^T \boldsymbol{v} + Z \boldsymbol{e}_3^T [\boldsymbol{\pi}]_\times \boldsymbol{\omega}$$

and hence we can write

$$\frac{\mathrm{d}\zeta}{\mathrm{d}t} = -\frac{1}{Z^2} \dot{Z} = \zeta^2 \boldsymbol{e}_3^T \boldsymbol{v} - \zeta \boldsymbol{e}_3^T [\boldsymbol{\pi}]_\times \boldsymbol{\omega}. \tag{B.7}$$

Then, analogously to the case of the luminance, we can write

$$\frac{\mathrm{d}\zeta}{\mathrm{d}t} = \nabla_{\boldsymbol{\pi}} \zeta^T \dot{\boldsymbol{\pi}} + \frac{\partial \zeta}{\partial t}, \tag{B.8}$$

and, comparing (B.8) with (B.7) and introducing (B.4), we conclude:

$$\frac{\partial \zeta}{\partial t} = -\nabla_{\boldsymbol{\pi}} \zeta^T \Theta(t, \boldsymbol{\pi}) - \zeta(t, \boldsymbol{\pi}) \nabla_{\boldsymbol{\pi}} \zeta^T \psi(t, \boldsymbol{\pi}) + \zeta(t, \boldsymbol{\pi})^2 \boldsymbol{e}_3^T \boldsymbol{v}(t) - \zeta(t, \boldsymbol{\pi}) \boldsymbol{e}_3^T \left[\boldsymbol{\pi}\right]_\times \boldsymbol{\omega}(t)$$

Dropping function arguments, we finally write the system dynamics as:

$$\begin{cases} \dfrac{\partial Y}{\partial t} = -\nabla_{\boldsymbol{\pi}} Y^T \Theta - \zeta \nabla_{\boldsymbol{\pi}} Y^T \psi & \text{(B.9a)} \\ \dfrac{\partial \zeta}{\partial t} = -\nabla_{\boldsymbol{\pi}} \zeta^T \Theta - \zeta \nabla_{\boldsymbol{\pi}} \zeta^T \psi + \zeta \boldsymbol{e}_3^T \left(\zeta \boldsymbol{v} - \left[\boldsymbol{\pi}\right]_\times \boldsymbol{\omega}\right). & \text{(B.9b)} \end{cases}$$

B.2 System dynamics with spherical projection

A similar strategy can be applied to the spherical projection model. In this case we consider as measurement the (scalar) luminance $Y(t, \boldsymbol{\eta})$ of a pixel located in $\boldsymbol{\eta}$ (in spherical coordinates) at a time t. Again, this is a scalar mapping:

$$Y : \mathbb{R} \times \mathbb{S}^2 \mapsto \mathbb{R}_+, \quad (t, \boldsymbol{\eta}) \mapsto Y(t, \boldsymbol{\eta}).$$

Following identical steps to the planar projection case, one can use the constant brightness assumption (B.1), the dynamics of the inverse range map $\delta = 1/\left\|\boldsymbol{p}\right\|$ in (4.34) and the spherical version of the point feature interaction matrix (2.11) to show that

$$\begin{cases} \dfrac{\partial Y}{\partial t} = -\nabla_{\boldsymbol{\eta}} Y^T \Theta - \delta \nabla_{\boldsymbol{\eta}} Y^T \psi & \text{(B.10a)} \\ \dfrac{\partial \delta}{\partial t} = -\nabla_{\boldsymbol{\eta}} \delta^T \Theta - \delta \nabla_{\boldsymbol{\eta}} \delta^T \psi + \delta^2 \boldsymbol{\eta}^T \boldsymbol{v}. & \text{(B.10b)} \end{cases}$$

where $\Theta(t, \boldsymbol{\eta})$ and $\psi(t, \boldsymbol{\eta})$ are, respectively, the angular and scaled linear components of the optical flow, given by (see, again, (2.11))

$$\begin{aligned} \psi(t, \boldsymbol{\eta}) &= -\left(\boldsymbol{I}_3 - \boldsymbol{\eta}\boldsymbol{\eta}^T\right) \boldsymbol{v}(t) \\ \Theta(t, \boldsymbol{\eta}) &= \left[\boldsymbol{\eta}\right]_\times \boldsymbol{\omega}(t). \end{aligned} \tag{B.11}$$

B.3 A nonlinear observer for photometric Structure from Motion

We can immediately notice that both (B.9) and (B.10) present a similar structure to the, well known by now, dynamics (3.10) with $s = Y$ and $\chi = \zeta$ or $\chi = \delta$ and hence $m = p = 1$. In particular, for the planar projection case (similar results can be found for the spherical one), defining

$$\begin{cases} f_s(s, \boldsymbol{\omega}(t)) = -\nabla_{\boldsymbol{\pi}} s^T \Theta(\boldsymbol{\omega}(t), \boldsymbol{\pi}) \\ \Omega(s, \boldsymbol{v}(t)) = -\nabla_{\boldsymbol{\pi}} s^T \psi(\boldsymbol{v}(t), \boldsymbol{\pi}) \\ f_\chi(\zeta, \boldsymbol{v}(t), \boldsymbol{\omega}(t)) = -\nabla_{\boldsymbol{\pi}} \zeta^T \Phi(\boldsymbol{v}(t), \boldsymbol{\omega}(t), \boldsymbol{\pi}) + \zeta \boldsymbol{e}_3^T \left(\zeta \boldsymbol{v}(t) - \left[\boldsymbol{\pi}\right]_\times \boldsymbol{\omega}(t)\right), \end{cases} \tag{B.12}$$

we can rewrite the system equations (B.9), as[2]

$$\begin{cases} \dfrac{\partial s}{\partial t} = f_s(s, \boldsymbol{\omega}) + \Omega(s, \boldsymbol{v})^T \zeta \\ \dfrac{\partial \chi}{\partial t} = f_\chi(\zeta, \boldsymbol{v}, \boldsymbol{\omega}) \end{cases} \tag{B.13}$$

and devise an observer as:

$$\begin{cases} \dfrac{\partial \widehat{s}}{\partial t} = f_s(s, \boldsymbol{\omega}) + \Omega(s, \boldsymbol{v})^T \widehat{\zeta} - h\, \widetilde{s} \\ \dfrac{\partial \widehat{\chi}}{\partial t} = f_\chi(\widehat{\chi}, \boldsymbol{v}, \boldsymbol{\omega}) - \alpha \Omega(s, \boldsymbol{v}) \widetilde{s}. \end{cases} \tag{B.14}$$

where h and α are positive gain (maps). Following the developments of Sect. 4.3, the gain h can be chosen as:

$$h(t) = h(\boldsymbol{\pi}, \boldsymbol{v}) = 2|\Omega(\boldsymbol{\pi}, \boldsymbol{v})| \tag{B.15}$$

so that the dynamics results critically damped with natural frequency $\alpha\Omega(\boldsymbol{\pi}, \boldsymbol{v})^2$.

Remark B.1. *To completely cancel the image dynamics, the gradient needed to calculate $f_s(s, \boldsymbol{\omega})$ and $\Omega(s, \boldsymbol{v})$ in (B.14) should be computed on $s = Y$ (the last measured image) and not on $\widehat{s}$ (the estimated image). This has some consequences on the numerical implementation of the integration scheme, as it will be discussed in Appendix B.6.*

Remark B.2. *The similarity between (B.14) and (3.11) should be regarded with care and "suspicion". In fact one should never forget that, while (3.11) is a finite dimensional system, (B.14) is an infinite dimensional one. Some of the considerations, and in particular, the stability proof, that hold for the former may not trivially extend to the latter. At the moment, we have not yet worked out a formal stability analysis for system (B.14). However we wish to immediately highlight one important difference w.r.t (3.11) for what concerns the effects of the disturbing term $d = f_\chi(\widehat{\chi}) - f_\chi(\chi)$: from (B.12) to (B.14), one has*

$$\begin{aligned} & d(\boldsymbol{\pi}, \zeta, \widehat{\zeta}, \boldsymbol{v}, \boldsymbol{\omega}) = f_\chi(\boldsymbol{\pi}, \widehat{\zeta}, \boldsymbol{v}, \boldsymbol{\omega}) - f_\chi(\boldsymbol{\pi}, \zeta, \boldsymbol{v}, \boldsymbol{\omega}) \\ & = -\nabla_{\boldsymbol{\pi}} \widetilde{\zeta}^T \Theta - \left(\nabla_{\boldsymbol{\pi}} \widehat{\zeta}^T \widehat{\zeta} - \nabla_{\boldsymbol{\pi}} \zeta^T \zeta \right) \psi + \left(\widehat{\zeta}^2 - \zeta^2 \right) \boldsymbol{e}_3^T \boldsymbol{v} - \widetilde{\zeta} \left[\boldsymbol{\pi} \right]_\times \boldsymbol{\omega} \\ & = -\nabla_{\boldsymbol{\pi}} \widetilde{\zeta}^T \Theta - \frac{1}{2} \nabla_{\boldsymbol{\pi}} \left(\widehat{\zeta}^2 - \zeta^2 \right)^T \psi + \left(\widehat{\zeta}^2 - \zeta^2 \right) \boldsymbol{e}_3^T \boldsymbol{v} - \widetilde{\zeta} \left[\boldsymbol{\pi} \right]_\times \boldsymbol{\omega}. \end{aligned}$$

Due to the presence of some gradients in this expression, for the disturbance d to be vanishing, one needs the estimation error $\widetilde{\zeta}(\boldsymbol{\pi}) \to 0$, to go to zero everywhere, *i.e. for all image points $\boldsymbol{\pi}$. In other words, the presence of spacial derivatives (disparity*

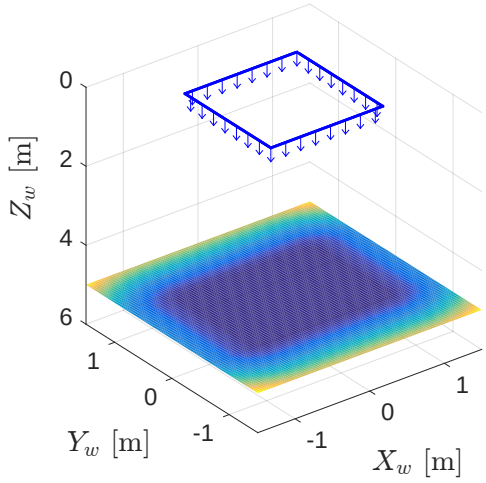
[2]Note that Ω is actually a scalar map, but the transpose was included for a better visual resemblance with (3.10).

map gradients) makes the estimation error on a certain image point $\boldsymbol{\pi}$ have some effects on a neighbourhood of $\boldsymbol{\pi}$. As a consequence, we expect the disturbing term to have a more significant impact on the estimation performance w.r.t. the finite dimensional cases considered in the rest of this thesis.

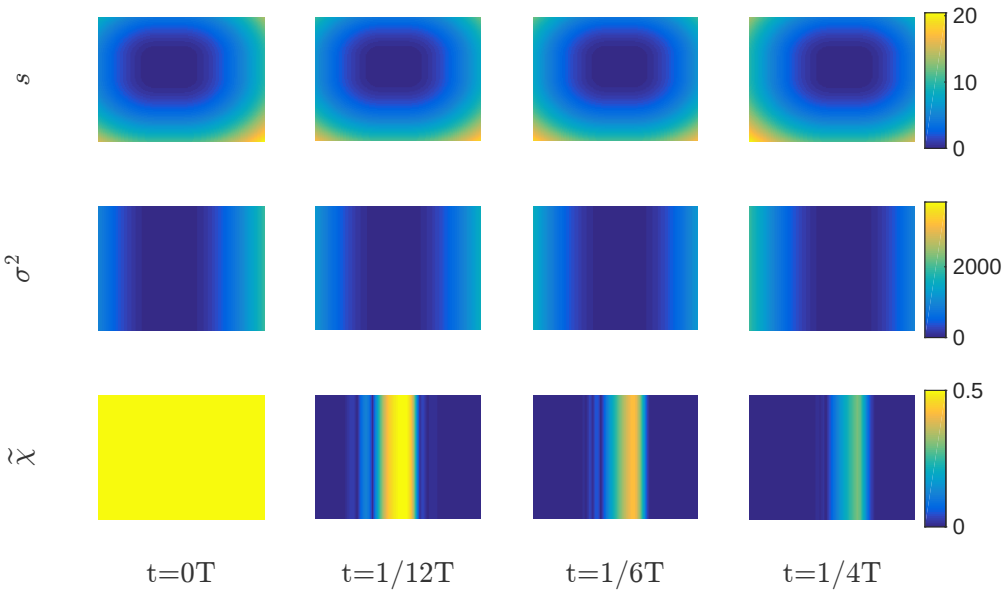
Remark B.3. *By looking at the PE 1×1 matrix $\Omega(s, \boldsymbol{v})\Omega(s, \boldsymbol{v})^T = \sigma^2(s, \boldsymbol{v}) = (\nabla_{\boldsymbol{\pi}} s^T \boldsymbol{l}_{\boldsymbol{v}} \boldsymbol{v})^2$, one can notice the following (intuitive) facts:*

- *one cannot estimate the depth map if the camera does not translate ($\|\boldsymbol{v}\| = 0$);*
- *one cannot estimate the depth map in areas of the environment that are not sufficiently textured ($\|\nabla_{\boldsymbol{\pi}} Y\| \approx 0$);*
- *the "informative" camera velocities are those that make the image move in the direction of the image gradient, i.e. orthogonally to the luminance level sets (the objects contours);*
- *if one has control over the camera linear velocity $\boldsymbol{v}$, an optimization strategy, similar to the ones used for the other SfM case studies, can be devised to maximize observability. Note however that care must be taken to ensure that the optimization does not attempt to increase "excitation" for areas of the image that are scarcely textured (this would happen, e.g. if one tried to naively maximize $\min_{\boldsymbol{\pi}} \sigma^2(t, \boldsymbol{\pi})$) and therefore are intrinsically less observable. In this context it would probably make more sense, instead, to try to maximize the excitation level for the highly textured areas and then rely on some regularization term, as those discussed in Appendix B.4, to "fill-in" the untextured parts. One also needs to make sure that the entire image is sufficiently excited at least for a short time. In this context, finite horizon planning techniques would probably be more suited than the instantaneous greedy optimization strategy used for the other case studies.*

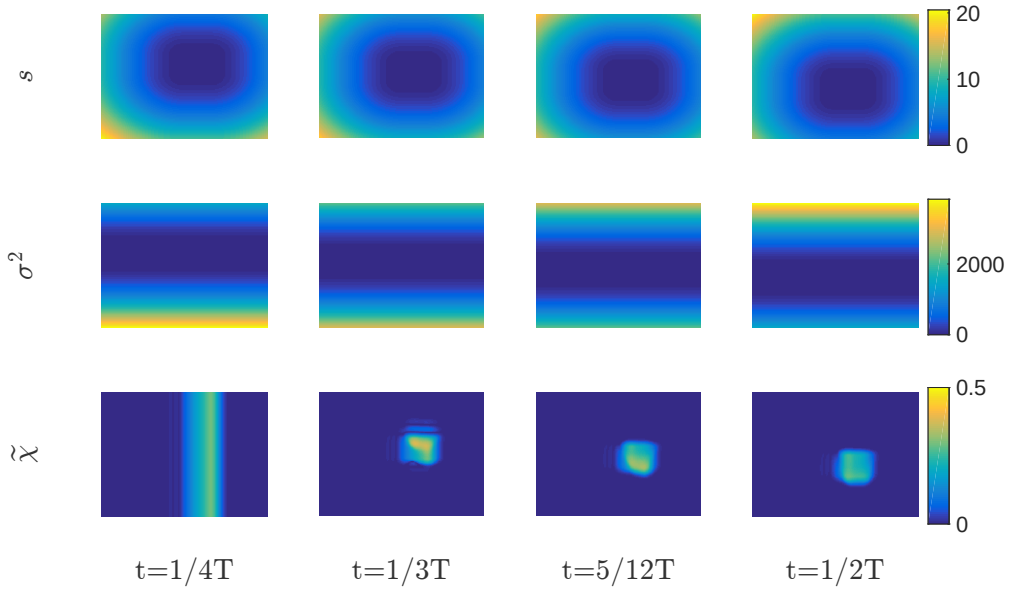
Figure B.1 shows some simulation results obtained by applying observer (B.14) to estimate the structure of a simple scene. The simulated scenario contains only a plane with a surface texture that is uniform in a central rectangular area and grows parabolically from its borders, see Fig. B.1(a). As a consequence the image resulting from the projection process on the simulated camera, contains uniform region, see the top lines of Figs. B.1(b) to B.1(d). The camera moves along a square trajectory whose sides are parallel to the sides of the uniform region (see, again, Fig. B.1(a)) at a constant velocity norm $\|\boldsymbol{v}(t)\| = 1\,\mathrm{m/s}$ and for a total time $T = 6\,\mathrm{s}$. By doing so, it maintains a constant distance of $5\,\mathrm{m}$ from the plane.


0
2
4
6
Z_w [m]
1
0
-1
Y_w [m]
-1
0
1
X_w [m]

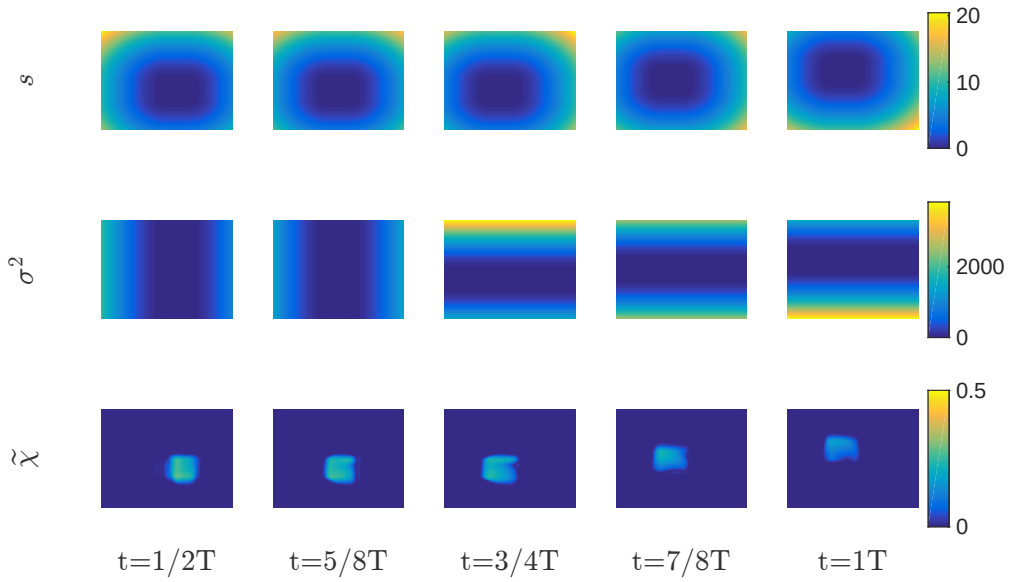
(a)


s
σ^2
$\tilde{\chi}$
20
10
0
2000
0
0.5
0
t=0T
t=1/12T
t=1/6T
t=1/4T

(b)



(c)



(d)

Figure B.1 – **Photometric depth estimation for a planar scene**. Fig. (a): camera trajectory with arrows indicating the direction of the camera optical axis and the observed textured plane. In the other plots: evolution of the image $s = Y$, observability eigenvalue σ^2, and disparity estimation error $\widetilde{\chi}$ maps for the first (Fig. (b)) and second (Fig. (c)) quarters of period and for the second half of the time (Fig. (d)). Note that a RGB color map was used for these plots for better readability, however all of these maps are scalar.

As a result, the camera image first translates horizontally to the left (Fig. B.1(b)), then vertically to the bottom (Fig. B.1(c)), and then again horizontally and vertically but in the opposite direction (Fig. B.1(d)). In other words, for each quarter of a period of the whole trajectory (lasting $T/4 = 1.5\,\mathrm{s}$), the image translates parallel to either the horizontal or the vertical sides of the central uniform region. The error on χ was initialized to a constant value of $\widetilde{\chi}(0, \boldsymbol{\pi}) = 0.5\,\forall \boldsymbol{\pi}$. The observer (B.14) was used with h chosen as in (B.14) and $\alpha = 5$. The image gradient $\nabla_{\boldsymbol{\pi}} s$ was calculated exploiting the algebraic model of the parabolic texture where as the gradient of the estimated disparity map $\nabla_{\boldsymbol{\pi}} \widehat{\chi}$ was approximated using first order upwind finite differences, see [Tho13b]. Finally we used a simulation time-step of 1 ms.

As predicted by the theory, in the first part of the trajectory (Fig. B.1(b)), the observability index σ^2 (second series of plots) is different from zero only on the left and right side of the uniform region. In fact(i) in the central area the image gradient is zero, and (ii) in the top and bottom areas the image gradient is orthogonal to the direction of the linear optical flow ψ. As a result only in the observable regions the error correctly starts converging towards zero. In the second quarter of trajectory (Fig. B.1(c)), the camera changes direction of motion by 90 deg and, consequently, now the areas on the top and on the bottom of the uniform region are "excited" (see σ^2 in the second series of plots) and the error starts converging there too. The areas on the left and right side of the uniform region are not observable in this case because the image gradient, in these areas, is orthogonal to the direction of the linear optical flow ψ. Thanks to the practically ideal conditions of the simulation (e.g. absence of noise), however, the error does not significantly grow. In the second half of trajectory the above process is repeated again and, if one kept using this trajectory, for this particular texture, almost the entire image would be observable for at least half of the time. Nevertheless due to the uniformity of the central area, the geometry of the environment is *never* observable in this region and by no meas one can expect to ever obtain a correct estimation of the depth in this part, unless some additional prior is considered. This is, in fact, the topic addressed in the next Appendix B.4.

B.4 Surface regularization and smoothing

Regardless of the specific observer, as underlined in Remark B.3, convergence properties will depend, for each pixel, on the level of "excitation" measured as:

$$\sigma^2(t, \boldsymbol{\pi}) = \left(\nabla_{\boldsymbol{\pi}} Y^T \psi(t, \boldsymbol{\pi})\right)^2, \text{or } \sigma^2(t, \boldsymbol{\eta}) = \left(\nabla_{\boldsymbol{\eta}} Y^T \psi(t, \boldsymbol{\eta})\right)^2.$$

In general, there will always be areas of the image where σ^2 is small due to low texture level or (possibly temporarily) non-optimal camera velocity (image sliding

along luminance level contours). In such areas the estimation will be undetermined and the observer will be highly affected by noise and other unmodeled effects. To overcome this problem it is necessary to introduce some smoothing/regularization based on priors on the underlying 3-D structure of the scene. Some regularization could also be introduced, if desired, in the areas with high level of information to increase robustness w.r.t. noise and discretization effects. More in general, the amount of regularization can be locally fine tuned based on the value of σ^2 in each pixel position.

One way of addressing the regularization problem, is to model it as a *heat diffusion process* (see, e.g. [Wid76]): intuitively the information about the disparity map $\widehat{\zeta}$ should be "diffused" from the areas in which the estimation is well defined (that can be thought of as "sources of disparity", similarly to heat sources), to those in which it is unobservable. This is similar to the way heat is diffused in an object from some sources, displaced in certain positions, to the rest of the material. As well known the two-dimensional heat diffusion, in absence of sources, is described by the parabolic PDE

$$\frac{\mathrm{d}u}{\mathrm{d}t} = q\nabla^2_{(x,y)}u \tag{B.16}$$

where $u(x, y)$ is the temperature map and $q > 0$ represents the material *thermal diffusivity*. Equation (B.16) converges to an equilibrium state in which,

$$\nabla^2_{(x,y)}u = 0. \tag{B.17}$$

This equilibrium condition is particularly interesting because it allows us to give another physically intuitive interpretation of the regularization process. Let us first consider a one-dimensional case: that of an elastic band stretched between two fixed points in $x = 0$ and $x = l$, see Fig. B.2 and [CH66]. In the assumption that the band is infinitely thin, one can neglect the forces associated with the torsion of the band and only consider the elastic forces due to stretching. In other words the band can be modeled as a sequence of elementary springs connected to each other. As well known the potential energy of a spring is associated with its length and thus, intuitively, the equilibrium position will correspond to the minimum length of the band which is obtained when the band is straight. As a matter of fact, let us assume that the band is curved as in the red line in Fig. B.2 and let us call $u(x)$ the amount of deflection from the straight configuration. If the deformation is small, the length of an elementary component of the band going from x to $x + \Delta x$ is given by:

$$\Delta L \approx \sqrt{[u(x+\Delta x) - u(x)]^2 + \Delta x^2} \approx \sqrt{[u(x) + u_x\Delta x - u(x)]^2 + \Delta x^2} = \Delta x^2\sqrt{1 + {u_x}^2}$$

where u_x is the derivative of u w.r.t. x. For $\Delta x \to 0$, and summing the contribution

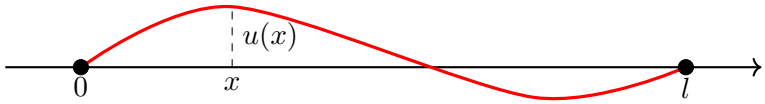


Figure B.2 – **Simple model of a deformed elastic band**. The band is rigidly attached at the two extrema and, in the equilibrium condition, it assumes a straight configuration between the two (black line) thus having a length l. The red line represents a deformed elastic band with $u(x)$ representing the amount of deflection w.r.t. the straight configuration.

of each elementary term, one concludes:

$$L \approx \int_0^l \sqrt{1 + u_x^2}\, \mathrm{d}x.$$

The difference of potential energy w.r.t. the equilibrium configuration is proportional to the change in length

$$L - l = \int_0^l \left(\sqrt{1 + u_x^2} - 1\right) \mathrm{d}x \approx \frac{1}{2} \int_0^l u_x^2 \mathrm{d}x$$

where we used a Taylor series expansion to the first order of $\sqrt{1 + u_x^2}$. One can then conclude that the potential energy is given by:

$$\mathcal{U} = \frac{1}{2} q \int_0^l u_x^2 \mathrm{d}x$$

where q is now the elastic coefficient of the band.

The two-dimensional equivalent of the elastic band is the thin membrane under tension (see [CH66]). In this case one can assume the potential energy to be proportional to the surface area of the membrane which can be computed as

$$\iint_{\mathcal{S}} \sqrt{1 + u_x^2 + u_y^2}\, \mathrm{d}x\, \mathrm{d}y \approx \iint_{\mathcal{S}} 1 + \frac{1}{2}(u_x^2 + u_y^2)\, \mathrm{d}x\, \mathrm{d}y$$

where $\mathcal{S}$ represents the membrane domain. Apart from a constant factor that can be ignored, the potential energy is then given by

$$\mathcal{U} = \frac{1}{2} q \iint_{\mathcal{S}} \left\| \nabla_{(x,y)} u \right\|^2 \mathrm{d}x\, \mathrm{d}y. \tag{B.18}$$

The equilibrium configuration for the membrane is given by the deformation $u(x, y)$ that minimizes the potential energy (B.18) and satisfies additional boundary conditions assigned on the boundary of $\mathcal{S}$. It can be shown (see [CH66]) that finding such minimum is equivalent to solving the boundary value problem for the PDE (B.17) with the assigned boundary conditions. One can conclude that, if (B.16) were used as an additional term to update the estimated disparity map, the solution would tend to a membrane-like shape of the depth map: in the observable regions the

value of $\widehat{\zeta}$ would be imposed by the estimation algorithm; in the non observable regions the disparity map $\widehat{\zeta}$ would tend to the minimum surface solution which is the minimum curvature interpolation of the conditions imposed at its boundaries (by the observable areas). This regularization term, then, has similar effects to the classical minimization of the L_2 norm of the disparity map (or, equivalently, of the optical flow) gradient exploited in many other works, e.g. [HS81]. The surface with minimum curvature is the *plane* and therefore the solution will tend, as much as possible and compatibly with the boundary conditions, to be linear in $\boldsymbol{\pi}$. Note that, assuming a linear ζ in $\boldsymbol{\pi}$ (or equivalently a linear δ in $\boldsymbol{\eta}$) is equivalent to assuming that the unobservable object is a plane in the 3-D space. In fact, as shown in (4.45), the variation of ζ for a planar object is linear in the image coordinates.

As for the regularization gain q, it would make sense to construct it as a function of σ^2 in such a way that in areas with more information a good match between the estimation and actual geometry is obtained and, conversely, smoothness is favoured in areas with poor information. One possibility would be, e.g.,

$$q(t, \boldsymbol{\pi}) = \text{sat}\left(\frac{1}{\sigma^2(t, \boldsymbol{\pi})}\right) = q(\boldsymbol{\pi}, \boldsymbol{v}, \nabla_{\boldsymbol{\pi}} Y^T)$$

where sat refers to a generic monotonic and saturated function of the argument. This choice might however result in a strongly time-varying q. A better solution might be to use an averaged value of σ^2 over a finite time window.

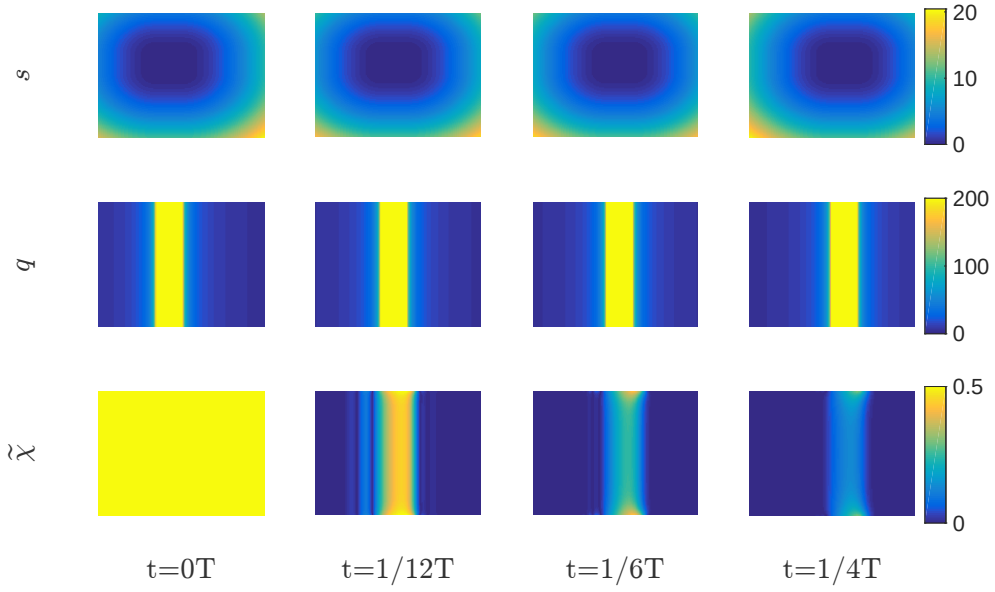
$$q(t, \boldsymbol{\pi}) = \text{sat}\left(\frac{1}{\frac{1}{T}\int_{t-T}^{t} \sigma^2(\tau, \boldsymbol{\pi})\mathrm{d}\tau}\right)$$

Another problem arises when the camera does not translate ($\|\boldsymbol{v}\| = 0$). In this case, obviously $\sigma^2 = 0$ *everywhere* and the regulation would be active on the entire disparity map thus making the estimation converge to a solution that is "flat" everywhere. To avoid this, one could also consider the norm of the camera linear velocity in the calculation of q, e.g.:

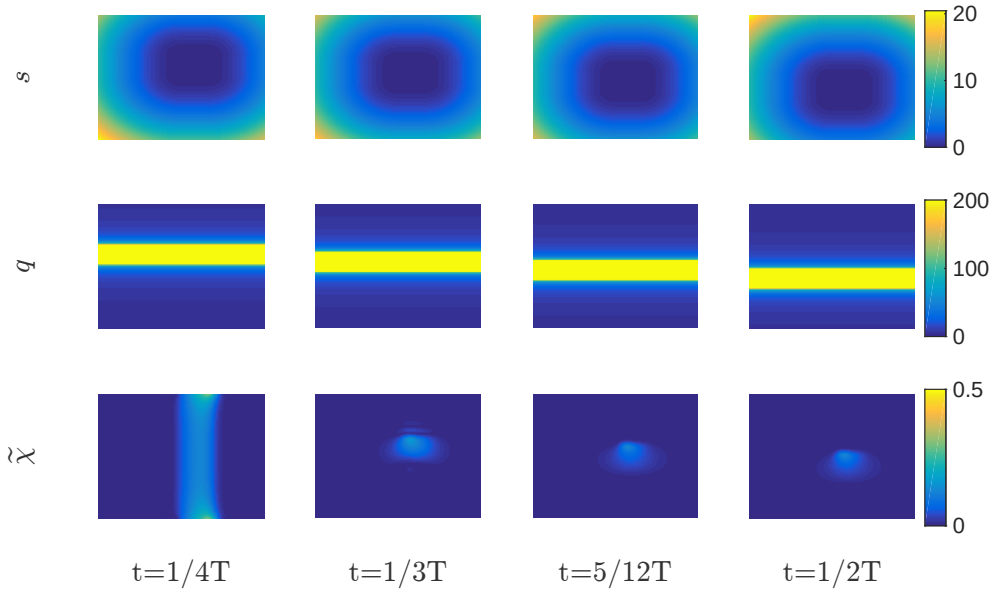
$$q(t, \boldsymbol{\pi}) = \text{sat}\left(\frac{\|\boldsymbol{v}\|}{\sigma^2(t, \boldsymbol{\pi})}\right) = q(\boldsymbol{\pi}, \boldsymbol{v}, \nabla_{\boldsymbol{\pi}} Y^T).$$

Finally, from a computation point of view, it might be convenient to use $|\Omega| = \sqrt{\sigma^2}$, instead of σ^2, since this is less expensive to compute and it is already involved in the calculation of h in (B.15). Adding this kind of regularization term to (B.8) one obtains a new observer in the form:

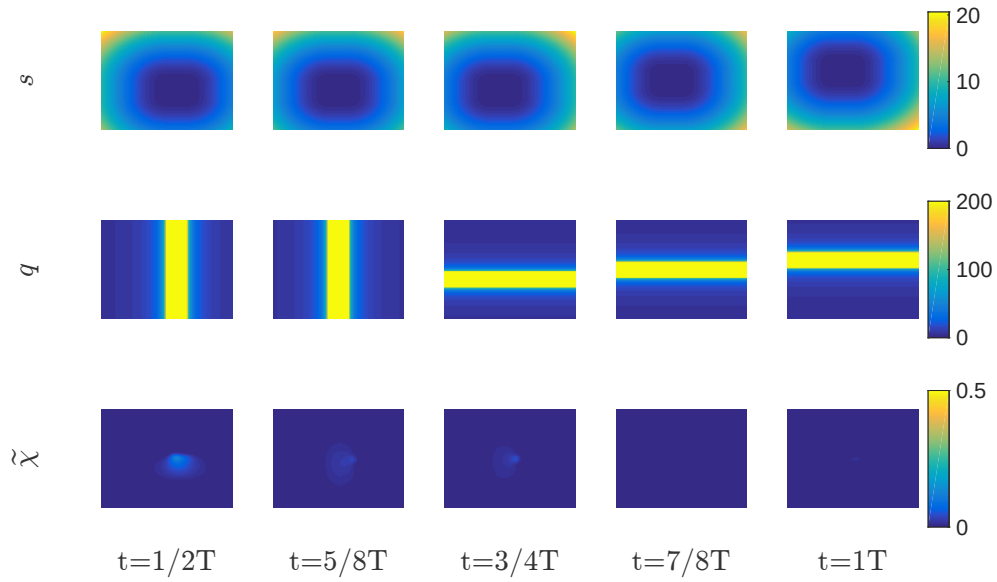
$$\begin{cases} \dfrac{\partial \widehat{s}}{\partial t} = f_s(s, \boldsymbol{\omega}) + \Omega(s, \boldsymbol{v})\widehat{\zeta} - h(t)\widetilde{s} \\ \dfrac{\partial \widehat{\chi}}{\partial t} = f_\chi(\widehat{\chi}, \boldsymbol{v}, \boldsymbol{\omega}) - \alpha\Omega(s, \boldsymbol{v})\widetilde{s} + q(t, \boldsymbol{\pi})\nabla^2_{\boldsymbol{\pi}}\widehat{\chi}. \end{cases} \tag{B.19}$$


s
q
χ̃
20
10
0
200
100
0
0.5
0
t=0T
t=1/12T
t=1/6T
t=1/4T

(a)


s
q
χ̃
20
10
0
200
100
0
0.5
0
t=1/4T
t=1/3T
t=5/12T
t=1/2T

(b)



(c)

Figure B.3 – **Photometric depth estimation for a planar scene with regularization of the disparity map**. Evolution of the image $s = Y$, regularization gain q, and disparity estimation error $\widetilde{\chi}$ maps for the first (Fig. (a)) and second (Fig. (b)) quarters of period and for the second half of the time (Fig. (c)). A RGB color map was used for these plots for better readability, however all of these maps are scalar. Note how the regularization term is mostly active in the unobservable areas (compare q here with the plots of σ^2 in Fig. B.1).

In Fig. B.3 we report the result obtained for the same simulation of Fig. B.1, but now introducing a regularization term as in (B.19) and with

$$q(t, \boldsymbol{\pi}) = 200 \, \|\boldsymbol{v}\| \, \frac{1}{h+1},$$

where h is computed as in (B.15). With this choice, we clearly have $q \approx 200$ if $h \ll 1$ and $q \approx 0$ if $h \gg 1$. The evolution of the smoothing gain q during the simulation is reported in the central plot sequences of Figs. B.3(a) to B.3(c). Note how this is "complementary" to the observability index σ^2 in Figs. B.1(b) to B.1(d). The use of the regularization term allows to(i) increase the converge rate of the areas that are only temporarily (50% of the time) observable and, more importantly, (ii) attain convergence in the central uniform area. Note, however, that the exact convergence, in this central area, is only possible because the observed scene, in this simulation, is actually planar in this region. For a general 3-D structure, the estimation error will not converge to zero in unobservable areas as expected from Remark B.3.

Before concluding this section, we briefly note that this solution could also be extended to operate an *anisotropic regularization* by using a smoothing term in the

form:

$$(\nabla_{\boldsymbol{\pi}}{}^T \boldsymbol{Q} \nabla_{\boldsymbol{\pi}})\widehat{\chi}$$

where $\boldsymbol{Q} \in \mathbb{R}^{2\times 2}$ is a symmetric gain. This should allow, e.g., to smoothen the disparity map only (or mainly) in the direction orthogonal to edges/discontinuities.

B.5 Propagation of depth discontinuities

Equations (B.9–B.10) represent the correct evolution of the system only under the assumption that all involved derivatives are defined and continuous. In the general case, it is hard to make any smoothness assumption on the luminance Y since the image texture can present discontinuities (contours). However, in principle, one might be able to apply the same model to a low pass filtered version of the input image, thus assuring continuity of the inputs. Moreover, in the actual implementation of the observer, the continuous space gradients are substituted by discrete numerical differences which will, in fact, induce some low-pass filtering effect.

The satisfaction of the continuity and differentiability assumptions for ζ, instead, depends on the convexity of the observed scene. As it is shown in Fig. B.4(a), the presence of an occlusion will, in fact, induce a discontinuity in Z from Z_h to Z_l and hence in ζ from $\zeta_h = 1/Z_h$ to $\zeta_l = 1/Z_l$. In general we can expect $\zeta(\boldsymbol{\pi}, t)$ to present some jump discontinuities in presence of occlusions. Note that, differently from $Z(\boldsymbol{\pi}, t)$, which can grow to infinity for far objects, if we assume that $Z(\boldsymbol{\pi}, t) > 0$, the disparity map $\zeta(\boldsymbol{\pi}, t)$ will always remain limited, therefore $\zeta(\boldsymbol{\pi}, t)$ will not present asymptotic discontinuities.

To understand how to model the evolution of ζ in presence of such discontinuities, it is convenient to restrict our attention to a simpler one dimensional case: let us assume that $\boldsymbol{v} = (v_x(t), 0, 0)$ and $\boldsymbol{\omega} = (0, 0, 0)$ with then $\dot{\boldsymbol{\pi}} = (-\zeta v_x, 0, 0)$. Let us also consider the behavior of $\zeta(\boldsymbol{\pi}, t)$ along the image line $\boldsymbol{\pi} = (x, 0, 1)$, i.e. the horizontal line passing through the center of the image. In this case (B.9b) can be rewritten as:

$$\frac{\partial}{\partial t}\zeta(x,t) - v_x(t)\zeta(x,t)\frac{\partial}{\partial x}\zeta(x,t) = 0. \tag{B.20}$$

If $\zeta \in C^1$, the equation can also be rewritten as:

$$\frac{\partial}{\partial t}\zeta(x,t) + \frac{\partial}{\partial x} f(\zeta(x,t), t) = 0. \tag{B.21}$$

with

$$f(\zeta(x,t), t) = -\frac{1}{2} v_x(t)\zeta(x,t)^2. \tag{B.22}$$

Equation (B.21) is called the *conservative form* of (B.20) and $f(\zeta, t)$ in (B.22) is the *flux functional*. From a numerical point of view, it is usually better to deal

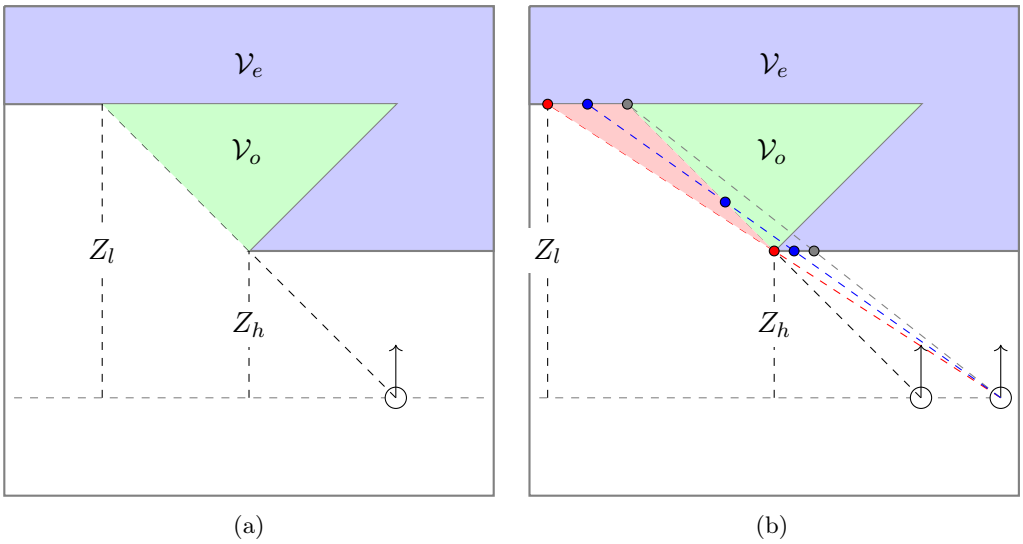


Figure B.4 – **Depth discontinuities in presence of an occlusion**. Fig. (a): depth discontinuity in presence of an occlusion. The environment $\mathcal{V}_e$ is represented in blue. The green area $\mathcal{V}_o$ is occluded. The camera is represented by a circle with an arrow indicating the direction of the optical axis which is assumed coincident with the Z-axis. Fig. (b): shock and rarefaction waves interpretation in the geometry reconstruction. The red region represents the area that is being uncovered or occluded depending on whether the camera moves from right to left or from left to right respectively. The blue dashed line and the blue dots represent a triple valued solution and its geometrical interpretation. The gray line and grey dots represent the first double valued solution in Fig. B.5(b). Finally the red line and red dots represent the correct solution.

with (B.21) rather than (B.20), see [Tho13b, Tho13a]. In particular (B.21) leads to the *conservation law* associated with (B.20). This latter is obtained by integrating (B.21) in the interval $[x_1, x_2]$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2}\zeta(x,t)\,\mathrm{d}x = \frac{\mathrm{d}}{\mathrm{d}t}\overline{\zeta}(t) = -\,f(\zeta(x,t),t)\Big|_{x_1}^{x_2}, \tag{B.23}$$

where $\overline{\zeta}(t)$ is the average value of $\zeta(x,t)$ over the interval $[x_1, x_2]$. Equation (B.23) signifies that the total variation of the average inverse depth over time in the interval is only due to the difference of "flux of inverse depth" through the left and the right boundaries of the interval. Equations in the form of (B.21) appear very often in physics for instance when modelling fluid dynamics, see [Tho13a, LeV02] for some examples.

By differentiating the flux function $f(\zeta, t)$, one obtains the *quasi-linear* form of the conservation law (B.21):

$$\frac{\partial}{\partial t}\zeta(x,t) + f'(\zeta(x,t),t)\frac{\partial}{\partial x}\zeta(x,t) = 0, \tag{B.24}$$

where $f'(\zeta,t) = \frac{\partial}{\partial\zeta} f(\zeta,t)$. One can easily verify that, along any curve $X(t)$ satisfying the ODE

$$\dot{X}(t) = f'(\zeta(X(t),t)) = v_x(t)\zeta(X(t),t) \tag{B.25}$$

one has $\dot{\zeta}(X(t),t) = 0$. Equation (B.25) is called the *characteristic curve* of the PDE. If $v_x(t) = v_x = const$, since ζ is constant along the curve $X(t)$, then, from (B.25), also $\dot{X}(t)$ is constant and therefore the characteristic curve is a straight line in the plane (x,t) starting from $X(0)$ and with slope $v_x\zeta(X(0),0)$. If v_x is time-varying, instead, then the characteristic lines are generic curves described by the integral of (B.25) over time, but one always has $\zeta(X(0)+\int_0^t \dot{X}(\tau)\mathrm{d}\tau, t) = \zeta(X(0),0)$.

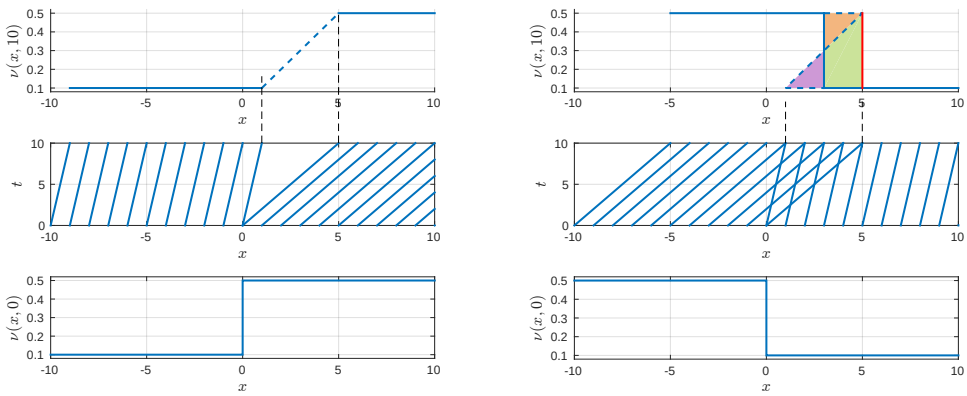
In a *linear* conservation law one has $f(\zeta,t) = a(t)\zeta + b(t)$ and hence all the characteristic curves, starting from any point $X(0)$, are parallel since $\dot{X}(t) = a(t)$ does not depend on X. In this case, therefore, the solution ζ is simply translating in space-time. If a is also constant, one simply has $\zeta(x,t+T) = \zeta(x-aT,t)$. In our case, however, we have $f' = f'(\zeta(x,t),t)$, i.e. the flux derivative depends on ζ. Assuming, for simplicity $v_x = const$, for each $X(t)$, the characteristic lines will still be straight, but with a (possibly) different slope, determined by $v_x\zeta(X(0),0)$. In other words, in the nonlinear case, the characteristic lines are not all parallel, and, as a consequence, they can diverge or intersect each other. To understand what happens in this case, it is convenient to consider a more intuitive physical example sharing the same dynamics of our case. Equations (B.21–B.22), with constant v_x, are a particular example of the so called inviscid Burgers' equation [Bur74], a simplification of the Navier-Stokes equation describing the dynamics of a free incompressible fluid when neglecting pressure. In this case, the average momentum of fluid particles ν in a space region, only changes, over time, due to the difference between the kinetic energy $f = \frac{1}{2m}\nu^2$ flowing to/from the left and the right boundary of the region, therefore, assuming unitary particles mass, one has

$$\frac{\partial\nu}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}\nu^2\right) = 0, \tag{B.26}$$

or, equivalently

$$\frac{\partial\nu}{\partial t} + \nu\frac{\partial\nu}{\partial x} = 0.$$

Equation (B.21) can be written as (B.26) by defining $\nu = v_x\zeta$. Burgers' equation (B.26) is probably the most typical textbook example of nonlinear flux conservation law, see, e.g. [LeV02, Tho13a]. As already mentioned, due to the nonlinearity, the characteristic curves are not parallel since $\dot{X}(t) = \nu(X(t))$. The solution of such equation does not simply translate uniformly in space-time but, instead, it deforms. In particular, since the characteristic curves are not parallel they can converge to the same position or diverge from a point in the plane (x,t). These two phenomenons are called *shock* and *rarefaction* waves respectively.



(a) (b)

Figure B.5 – **Time-space evolution of a rarefaction and shock wave**. Fig. (a): evolution of a rarefaction wave. Fig. (b): evolution of a shock wave. For both figures, from bottom to top we show: (i) the initial condition $\nu(x,0)$, (ii) some of the characteristic curves emanating from $x = -10 + k$ with $k = 0, 1, \ldots, 20$, and (iii) the final condition $\nu(x,10)$. This picture is similar to some reported in [LeV02].

B.5.1 Rarefaction waves

Figure B.5(a) shows some of the characteristic curves of (B.26) starting from $X(0) = -10 + k$ with $k = 1, 2, \ldots, 20$ and with an initial condition:

$$\nu(x,0) = \begin{cases} \nu_r = .5 & \text{if } x > 0 \\ \nu_l = .1 & \text{if } x < 0 \end{cases}.$$

One can notice that these curves diverge from the point $x = 0$, $t = 0$.

This effect is called a *rarefaction wave*. It presents itself whenever the camera is moving in the direction that *reduces* an occluded region as it is the case in Fig. B.4(b) if the camera moves from the right to the left configuration. In this situation new parts of the environment (the red region in Fig. B.4(b)) appear in the camera FOV and the information contained in the initial condition is not sufficient to reconstruct ζ in the discovered area. Simply integrating (B.21) will result in the solution represented by the dashed line in the top plot of Fig. B.5(a). This solution would correspond to the actual evolution of ζ only if the initially occluded area (green plus red region in Fig. B.4(b)) were actually part of the environment (i.e. if nothing is actually being discovered). However the actual depth map, for the case depicted in Fig. B.4(b), instead, remains discontinuous in the new camera position. We stress the fact that there is not much one can do to fix this situation: since the information about the occluded area is not present in the initial condition, it is impossible, in general, to correctly reconstruct the depth map in the new camera

configuration by simply integrating the initial depth map. Given these premises, we can accept the natural hypothesis produced by the simple propagation of the initial condition as a plausible solution for the actual depth map, and rely on the innovation part of the observer for improving the depth map if such hypothesis turns out to be wrong.

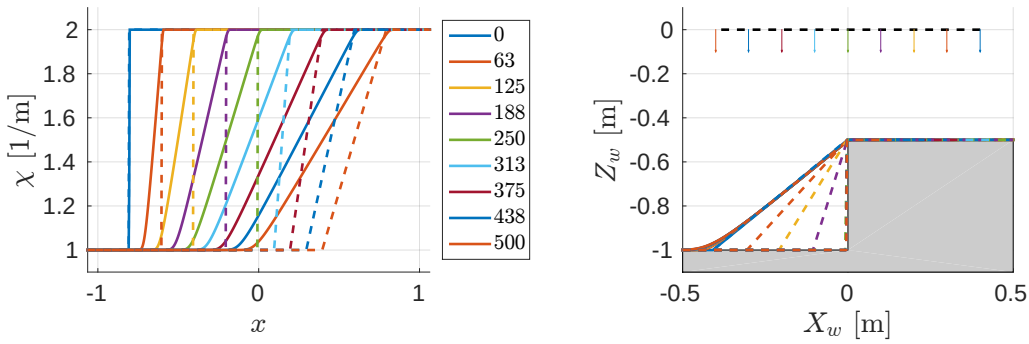
To give an example, in Fig. B.6 we report some simulation results for the propagation ($\alpha = h = q = 0$ in (B.19)) of a disparity rarefaction wave. For this simulation we considered a one-dimensional image and we constrained the camera to move only on the (X, Z) plane in the camera reference frame. The simulated camera had a resolution of 640 px and intrinsic parameters $fd_x = 300$, $j_{o_\mathcal{C}} = 319.5$ in (2.4). Finally, we used an integration time step of 0.002 s. The camera is observing a simple environment that contains a single jump discontinuity, see the gray area in Fig. B.6(b). The trajectory of the camera, represented by a black dashed line in Fig. B.6(b), is traveled at a constant velocity $\boldsymbol{v} = [.8\,0\,0]^T$m/s and $\|\boldsymbol{\omega}\| = 0$ from the right to the left of the figure. The position of the camera at some equally spaced iterations is represented by coloured arrows, indicating the direction of the camera optical axis. For the same iterations, and with the same color code, we represented, with dashed lines, in Fig. B.6(a), the actual value of the disparity map in the camera normalized coordinate x. The dashed lines in Fig. B.6(a) represent, instead, a reconstruction of the environment from the above disparity map, calculated using the parametric expression

$$^{\mathcal{W}}\boldsymbol{p}(\boldsymbol{\pi}, t) = \frac{1}{\chi(\boldsymbol{\pi}, t)}\,^{\mathcal{C}}\boldsymbol{R}_{\mathcal{W}}\boldsymbol{\pi} + {}^{\mathcal{W}}\boldsymbol{t}_{\mathcal{C}}(t). \tag{B.27}$$

Note that the dependence of this reconstructed environment on t is due to the fact that, depending on the camera position (that changes with time) different portions of the environment are occluded as it can be seen in Fig. B.6(b) (the dashed lines are not all coincident). In other words, in the discontinuity, (B.8) is violated and new source terms (i.e. contributions to the total time derivative of the disparity map) appear. This effect will also characterize the shock waves. The solid lines in Figs. B.6(a) and B.6(b) finally represent the estimation of the depth map and the corresponding environment reconstruction, when using (B.19) with $\alpha = h = q = 0$ from an initial value of $\widehat{\chi}(\boldsymbol{\pi}, 0) = \chi(\boldsymbol{\pi}, 0)$. As a numerical scheme, we opted for an upwind finite difference method (see [LeV02]) using the conservative version of the PDE for χ, i.e.

$$\frac{\partial \widehat{\chi}}{\partial t} = -\frac{\partial}{\partial x}\left(\frac{1}{2}v_x\widehat{\chi}^2\right).$$

As expected from the developments of this section, the propagation maintains the initial hypothesis about the structure of the environment in the area that was occluded in the first frame (the one used for initialization). The environment reconstruction (solid lines in Fig. B.6(b)) remains (almost) constant in accordance



(a) (b)

Figure B.6 – **Photometric depth estimation for a rarefaction wave**. Fig. (a) evolution of the actual (dashed) and numerically propagated (solid) disparity map at equally spaced iterations (check the legend). Fig. (b) reconstruction of the environment from the actual (dashed) and numerically propagated depth maps with arrows representing the camera position and the direction of the optical axis, and using the same color code as in Fig. (a).

with the fact that, in these conditions, one obtains, from (B.7), $\frac{\mathrm{d}\zeta}{\mathrm{d}t} = 0$. The only small (but rather important) variation is due to the dissipation introduced by the numerical differentiation which smoothens the depth map at each iteration.

B.5.2 Shock waves

A dual situation to the rarefaction waves is the one represented in Fig. B.5(b), which shows some of the characteristic curves starting from $X(0) = -10 + k$, with $k = 1, 2, \ldots, 20$, but now with:

$$\nu(x, 0) = \begin{cases} \nu_r = .1 & \text{if } x > 0 \\ \nu_l = .5 & \text{if } x < 0 \end{cases}.$$

In this case the characteristic curves intersect and the solution becomes triple-valued (dashed line in the top plot of Fig. B.5(b)). This effect is called a *shock wave*. Obviously such a solution is not physically acceptable for the particles dynamics case: fluid particles can only have one momentum in each position. When modeling physical systems, such non-physical solutions in general appear when some viscosity is neglected in the modeling phase. Moreover, the use of a numerical scheme to integrate (B.21) will always introduce some level of numerical diffusion or dissipation which is due to the approximation of the derivatives with finite differences. Finally, we have already commented that some diffusion term can be added to smoothen the depth map, especially in the areas that are scarcely observable. The PDE that one ends up solving is then:

$$\frac{\partial}{\partial t}\zeta(x,t) + \frac{\partial}{\partial x} f(\zeta(x,t)) = \epsilon \frac{\partial^2}{\partial x^2}\zeta(x,t). \tag{B.28}$$

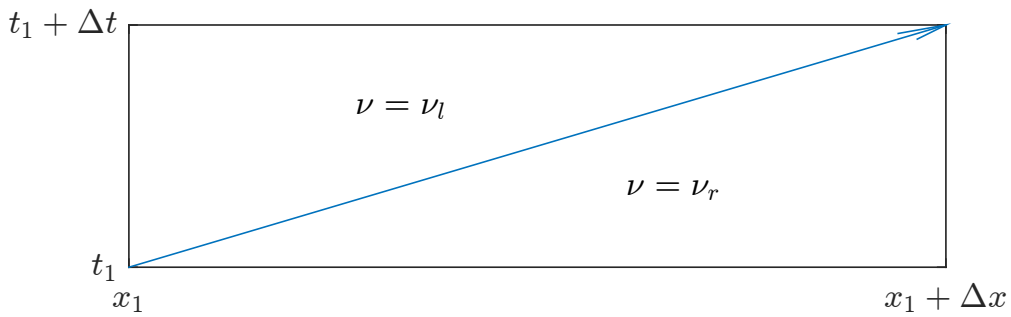


Figure B.7 – **Shock speed propagation on the plane** (x, t). Figure modified from [LeV02].

with $\epsilon > 0$. Equation (B.28) is also called the *viscous* form of the differential equation and a solution to (B.28) for $\epsilon \to 0$ is called a *vanishing-viscosity* solution for (B.26), see [LeV02]. Using the vanishing-viscosity approach, it can be shown that the correct solution can be obtained from the nonphysical multi-valued one buy applying the *equal area rule*: this rule is a consequence of the conservation law and imposes the physical solution to have a discontinuity in a position such that the area "under" the multi-valued solution is the same as the one under the single-valued one or, equivalently, that the areas added (violet) and removed (orange) from the nonphysical solution are equal. A direct consequence of this is that the discontinuity will travel at a velocity v_s given by the average between the velocities (the slope in the plane (x, t)) of the characteristic lines to the right and to the left of the discontinuity. This last result can also be obtained by looking at the integral of the conservation law (B.23) as shown in [LeV02]. Suppose that the shock wave is moving at a constant velocity v_s from x_1 to $x_1 + \Delta x$ in a time Δt (see Fig. B.7). Integrating the conservation law (B.23) in the region $[x_1, x_1 + \Delta x] \times [t_1, t_1 + \Delta t]$ one obtains

$$\int_{x_1}^{x_1+\Delta x} \nu(x, t_1 + \Delta t)\,\mathrm{d}x - \int_{x_1}^{x_1+\Delta x} \nu(x, t_1)\,\mathrm{d}x =$$
$$\int_{t_1}^{t_1+\Delta t} f(\nu(x_1, t))\,\mathrm{d}t - \int_{t_1}^{t_1+\Delta t} f(\nu(x_1 + \Delta x, t))\,\mathrm{d}t.$$

The space-time region $[x_1, x_1 + \Delta x] \times [t_1, t_1 + \Delta t]$ is divided (see Fig. B.7) into a left and a right triangles in which the value of ν is roughly equal to ν_l and ν_r respectively, therefore one obtains

$$\Delta x \nu_l - \Delta x \nu_r \approx \Delta t f(\nu_l) - \Delta t f(\nu_r),$$

and, in the limit $\Delta t \to 0$, one has, for (B.26)

$$\frac{\Delta x}{\Delta t} \approx \dot{X} = v_s = \frac{f(\nu_r) - f(\nu_l)}{\nu_r - \nu_l},$$

which, for (B.26), results in

$$v_s = \frac{1}{2}\frac{\nu_r^2 - \nu_l^2}{\nu_r - \nu_l} = \frac{\nu_r + \nu_l}{2}. \tag{B.29}$$

In the camera projection system (B.9), shock waves appear whenever some part of the environment is occluded as it is the case in Fig. B.4(b) if the camera moves from the left to the right position. The dashed blue line in Fig. B.4(b) represents the double valued solution returned by the equal area rule that intersects the initial environment hypothesis (given by $\mathcal{V}_e \cup \mathcal{V}_o$) in the two blue dots in the figure. Note that, differently from the particles case, however, the disparity dynamics does not contain any viscosity (apart from the artificial one due to numerical discretization or regularization) and the two points in which the camera intersects the environment in presence of an occlusion have completely independent dynamics. Hence the solution of (B.28) does not represent correctly the evolution of the system in this situation. The correct solution in the projection case is clearly the one represented in red in Figs. B.4(b) and B.5(b), which preserves the position of the occlusion. To obtain this solution we need to introduce a "source" term (an additional contribution to the total derivative of ζ) in the PDE that corresponds to the areas represented in green and orange in Fig. B.5(b). For an integration time Δt, this area is given by

$$\Delta A = (\nu_l - \nu_r)[f'(\nu_l) - v_s]\Delta t = \frac{1}{2}(\nu_l - \nu_r)^2 \Delta t$$

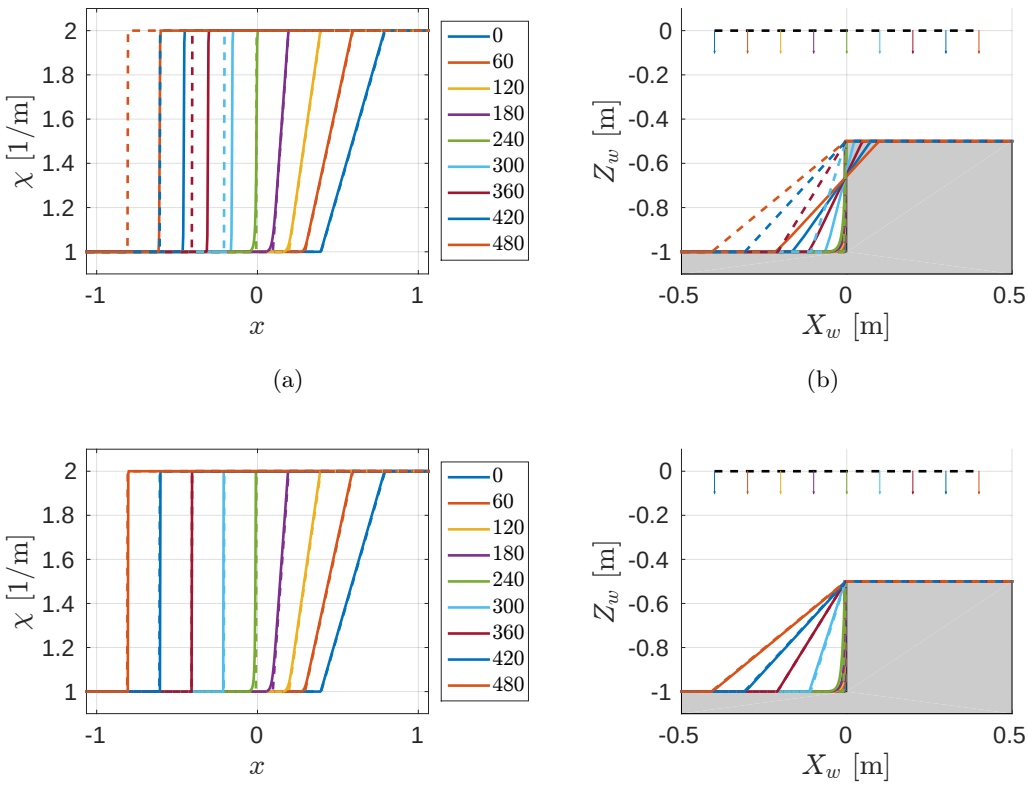
and hence for $\Delta t \to 0$ one has the source term

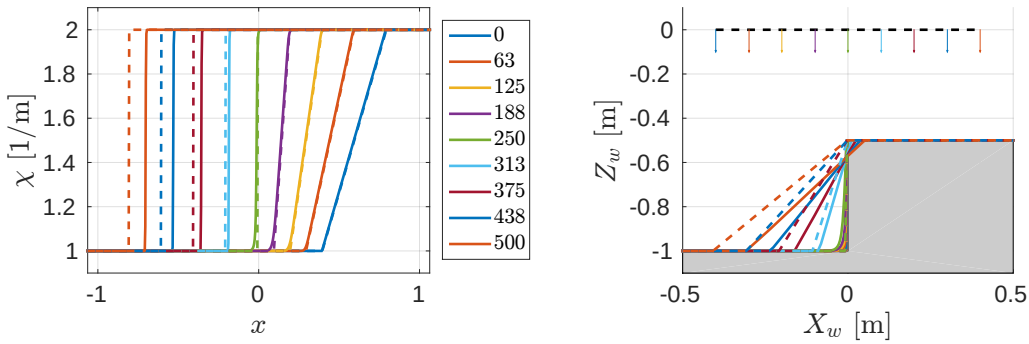
$$\frac{\Delta A}{\Delta t} \to \frac{\mathrm{d}A}{\mathrm{d}t} = \frac{1}{2}(\nu_l - \nu_r)^2. \tag{B.30}$$

We note that the introduction of this term requires, in principle, the identification and tracking of the shock-type depth discontinuities which would make the implementation more complex and less suited for parallelization. However, since the additional term is proportional to the squared first order difference between the disparity levels of neighbour pixels (the ones on each side of the discontinuity), its effect should be negligible in the areas in which the disparity map is smooth. Therefore we expect that introducing this additional source term *everywhere* in the image (and not only on the sides of a discontinuity of the disparity map) should not affect the propagation in a unacceptable way, with, on the other hand, significant advantages from the point of view of the simplicity and parallelizability of the algorithm. The only problematic areas, could be the rarefaction-type occlusions that present depth discontinuities but do not require the addition of this source term. This areas, should, however, be easy to recognize since one has, for them, $\frac{\partial \zeta}{\partial t} = -\frac{\partial f(\zeta)}{\partial \boldsymbol{\pi}} \leq 0$ (i.e. the disparity map does not grow in these region) and checking this condition requires, again, only local information.

We demonstrate the effects of this additional source term by running a similar simulation as in Fig. B.6, but now with the camera traveling in the opposite direction (i.e. from left to right), see Fig. B.8(b).

The results of the simulation are reported in Fig. B.8 with, as before, dashed lines representing the ground truth and solid lines representing the results of the pure propagation ($\alpha = h = q = 0$) of (B.19) from an initial estimation error $\widetilde{\chi} = 0$. The first series of plots, Figs. B.8(a) and B.8(b), show the results obtained *without* the source term (B.30). As one can see, the propagation is correct before the formation of the shock wave (around iteration 240, green lines in the plots). After this, the propagation starts accumulating delays w.r.t. the ground truth. This is due to the fact that the shock wave is propagated at a velocity $v_s = \frac{1}{2}(\chi_l + \chi_r)\psi$, as predicted using (B.29), instead of $\chi_r\psi$, where χ_r and χ_l are the values of χ at the right and left of the discontinuity. The point in which the environment reconstruction lines intersect in Fig. B.8(b) corresponds to the depth value of $Z = \left[\frac{1}{2}(\chi_l + \chi_r)\right]^{-1} \approx 0.66\,\mathrm{m}$, i.e. to the point that is actually supposed to travel at the shock speed v_s.





(e) (f)

Figure B.8 – **Photometric depth estimation for a shock wave**. Fig. (a) evolution of the actual (dashed) and numerically propagated (solid) disparity map at equally spaced iterations (check the legend) and for a time step calculated in such a way that the shock wave moves by exactly one pixel per iteration. Fig. (b) reconstruction of the environment from the actual (dashed) and numerically propagated depth maps with arrows representing the camera position and the direction of the optical axis, and using the same color code as in Fig. (a). The same kind of plots are reported in Fig. (c) and Fig. (d) for the case in which the additional source term (B.30) is used to improve the shock wave propagation. Finally Fig. (e) and Fig. (f) report the results obtained by still using the source term (B.30) but with a time step of 2 ms, i.e. smaller than in the previous case.

Figures B.8(c) and B.8(d) represent the results obtained by introducing the additional source term (B.30) in the propagation equation for $\widehat{\chi}$. One can note that now the propagation is correct both before and after the formation of the shock wave. In this case, however, (as in the previous one) the time step for the propagation was chosen in such a way that the shock wave moved by exactly one pixel per iteration. This was obtained with $\delta t \approx 2.08\,\text{ms}$. By repeating the simulation with a time-step of $\delta t = 2\,\text{ms}$, one can see that the propagation of the shock wave is again not perfect, see Figs. B.8(e) and B.8(f) even though it is still significantly better than in Figs. B.8(c) and B.8(d). We want to stress that the fact that the source term (B.30) seems to only work correctly when the discontinuity moves by *exactly* one pixel per iteration is a significant limitation to its use. In fact, in general, the observed scene will contain more that one shock-type depth discontinuities each moving at different speeds and, therefore, we cannot expect this condition to be realized in practice. The reasons why this happens are still not clear, but we suspect that this might be related to numerical discretization.

B.6 Notes on the numerical implementation

In an actual implementation, the continuous time observer equations of (B.14) or (B.19) are approximated using a numerical resolution scheme. The discussion about the choice of the particular method to use could take an entire thesis, and, in fact, many books have been written on the topic, see again [Tho13b, Tho13a, LeV02]. We are, in particular, interested in finite difference and finite volume methods for their potential to be implemented in a highly parallel architecture. To give just a simple example, a naive numerical scheme for integrating (B.14) or (B.19) could be obtained by using first order central differences in space and forward differences in time to approximate all the derivatives. Considering just a one dimensional case one could then write, e.g., for the first row of (B.14)

$$\frac{\widehat{s}_i^{k+1} - \widehat{s}_i^k}{\delta t} = -\frac{\widehat{s}_{i+1}^k - \widehat{s}_{i-1}^k}{2\delta x}(\Theta_i^k + \psi_i^k \widehat{\zeta}_i^k) - h\widetilde{s}_i^k$$

where $s_i^k = s(x = x_k, t = t_k)$ (and similarly for the other quantities) and δx and δt are the spatial and temporal discretization steps respectively. One can then "isolate" the new estimation as

$$\widehat{s}_i^{k+1} = \widehat{s}_i^k + \left(-\frac{\widehat{s}_{i+1}^k - \widehat{s}_{i-1}^k}{2\delta x}(\Theta_i^k - \psi_i^k \widehat{\zeta}_i^k) - h\widetilde{s}_i^k \right) \delta t.$$

Note how the amount of calculation required is always constant at each iteration and it is identical for each pixel. Moreover, for each pixel, the calculation only requires to access the observer state and the measurements in a small neighbourhood of the pixel (in this case only the pixels to the left and to the right) and only at the time t_k. This properties make it possible to envision an implementation of this technique on highly parallel computer architectures such as GPUs and FPGAs, with potentially extremely high performance, see [AAM14].

In general, an important aspect to consider is that the stability of numerical methods for solving PDEs depends, in general on the relationship between the spatial and temporal discretization steps. A large class of methods, e.g., becomes numerically unstable if the characteristic curves move by more than one pixel per iteration, i.e. in the one-dimensional case, if

$$\dot{X} = f' > \frac{\delta x}{\delta t}.$$

Increasing δx introduces a larger discretization error thus affecting the overall accuracy. It might seem obvious that δx (and similarly for δ_y) should correspond to the actual camera pixel distances, however the spacial resolution can be both increased and reduced at will with some effects on the performance of the estimation. On the other hand, reducing δt clearly has a positive impact on the stability at the cost

of a higher computational effort. Note that, as for the spatial one, the time step for the time discretization does not have to correspond to the camera frame rate. In fact, depending on the camera velocity and on the distance of the environment (which enter in the calculation of f'), the time Δt passing between two frames might be considerably larger than the minimum integration time step necessary to obtain a stable numerical scheme. Given these considerations it is useful to distinguish in (B.19) a *propagation* term and an *innovation* component. The former is meant to align the previous estimate to the latest measurement while the latter uses the actual measurement to update the estimation. Let us indicate with $(\widehat{s}^k, \widehat{\chi}^k)$ the state of the observer at time $t = T^k = t_0 + k\Delta t$ when the image s^k becomes available. Let us also indicate with $(\widehat{s}^{k+}, \widehat{\chi}^{k+})$ the value obtained by integrating the following PDE obtained by setting $h = \alpha = 0$ in (B.19)

$$\begin{cases} \dfrac{\partial \widehat{s}}{\partial t} = f_s(s^?, \boldsymbol{\omega}) + \Omega(s^?, \boldsymbol{v})\widehat{\zeta} \\ \dfrac{\partial \widehat{\chi}}{\partial t} = f_\chi(\widehat{\chi}, \boldsymbol{v}, \boldsymbol{\omega}) + q(t, \boldsymbol{\pi})\nabla^2_{\boldsymbol{\pi}}\widehat{\chi}. \end{cases} \tag{B.31}$$

from the initial condition $(\widehat{s}^k, \widehat{\chi}^k)$ for a time ΔT. When a new camera image is available, we calculate the new estimate at time T^{k+1} as

$$\begin{cases} \widehat{s}^{k+1} = \widehat{s}^{k+} - h\Delta T\left(\widehat{s}^{k+} - s^{k+1}\right) \\ \widehat{\chi}^{k+1} = \widehat{\chi}^{k+} - \alpha\Delta T\Omega(s^?, \boldsymbol{v})\left(\widehat{s}^{k+} - s^{k+1}\right). \end{cases} \tag{B.32}$$

As highlighted by the presence of the question mark $s^?$ in (B.31–B.32), a problem that one faces when implementing (B.31–B.32) is that of choosing which image s to use for the calculation of f_s and Ω (both depend, in fact, on the gradient of s). If the latest measurement $s = s^k = Y^k$ is used and the integration time step for (B.31) is smaller than the camera time step, the risk is that, during the "inner" integration steps of (B.31), s is not aligned with the other quantities[3]. One possible solution would be to propagate the last measurement Y^k together with the observer state during the propagation part of the estimation. However this implies a considerable computational burden. Another possibility would be, instead, to use the estimated image $\widehat{s} = \widehat{Y}$ in the calculation of f_s and Ω. To show the effect of this choice more clearly, we prefer to consider again the fully continuous observer dynamics (B.14) and to write the explicit expression of the different quantities. The

[3]Note that, in principle, a similar issue existed for the other SfM problems considered in this thesis, whenever the integration step of the observer was chosen smaller than the camera frame rate. However, while in the other cases its effects where negligible, as demonstrated by the experimental results, initial simulation tests seem to indicate a higher sensitivity w.r.t. this problem in the dense case.

observer dynamics, for the planar case, would hence be written as

$$\begin{cases} \dfrac{\partial \widehat{Y}}{\partial t} = -\nabla_{\boldsymbol{\pi}} \widehat{Y}^T \Theta - \nabla_{\boldsymbol{\pi}} \widehat{Y}^T \psi \widehat{\zeta} - h \widetilde{Y} \\ \dfrac{\partial \widehat{\zeta}}{\partial t} = -\nabla_{\boldsymbol{\pi}} \widehat{\zeta}^T \Theta - \nabla_{\boldsymbol{\pi}} \widehat{\zeta}^T \psi \widehat{\zeta} + \alpha \nabla_{\boldsymbol{\pi}} \widehat{Y}^T \psi \widetilde{Y} \\ \qquad + \widehat{\zeta} \boldsymbol{e}_3^T \left(\widehat{\zeta} \boldsymbol{v} - [\boldsymbol{\pi}]_\times \boldsymbol{\omega} \right) + q \nabla_{\boldsymbol{\pi}}^2 \widehat{\zeta}, \end{cases}$$

and the error dynamics would become

$$\begin{cases} \dfrac{\partial \widetilde{Y}}{\partial t} = -\nabla_{\boldsymbol{\pi}} \widetilde{Y}^T \Phi - h \widetilde{Y} - \nabla_{\boldsymbol{\pi}} \widehat{Y}^T \psi \widetilde{\zeta} & \text{(B.34a)} \\ \dfrac{\partial \widetilde{\zeta}}{\partial t} = -\nabla_{\boldsymbol{\pi}} \widetilde{\zeta}^T \Phi + \alpha \nabla_{\boldsymbol{\pi}} \widehat{Y}^T \psi \widetilde{Y} + q \nabla_{\boldsymbol{\pi}}^2 \widehat{\zeta} & \text{(B.34b)} \\ \qquad - \nabla_{\boldsymbol{\pi}} \widehat{\zeta}^T \psi \widetilde{\zeta} - \widetilde{\zeta} \, [\boldsymbol{\pi}]_\times \boldsymbol{\omega} + \left(\widehat{\zeta}^2 - \zeta^2 \right) \boldsymbol{e}_3^T \boldsymbol{v}. \end{cases}$$

The expression (B.34) presents some intuitive interpretations whose actual implications are still to be investigated. First of all we can still recognize a skew-symmetric structure with dissipation term h as in the other cases, but now with the coupling matrix being $\Omega(\widehat{s}, \boldsymbol{v}) = \nabla_{\boldsymbol{\pi}} \widehat{Y}^T \psi$ instead of $\Omega(s, \boldsymbol{v})$. Secondly we can notice the presence of an additional term in the form

$$\begin{cases} \dfrac{\partial \widetilde{Y}}{\partial t} = -\nabla_{\boldsymbol{\pi}} \widetilde{Y}^T \Phi \\ \dfrac{\partial \widetilde{\zeta}}{\partial t} = -\nabla_{\boldsymbol{\pi}} \widetilde{\zeta}^T \Phi \end{cases}.$$

This term is a linear (Φ depends on ζ and not $\widehat{\zeta}$) variable-coefficient convection equation (see [Tho13b]) for the image and the disparity map that should not either reduce or increase the error but just "move it around" with a velocity determined by the optical flow Φ.

Another important effect of the use of a numerical resolution scheme is the, already mentioned, introduction of undesired numerical dissipation/smoothing effects. We have already seen this phenomenon in the propagation of the depth map in Fig. B.6. More important, however, are the effects on the propagation of the image $\widehat{s}$. As a matter of fact, due to numerical damping, the prediction error $\widetilde{s}$ will not only reflect the depth estimation error, but it will also be due to numerical effects. This clearly affects the effectiveness of $\widetilde{s}$ in the update phase. We believe that, since only the low frequency components of $\widehat{s}$ are preserved during the propagation (which tends to damp the high frequency ones), these are the only ones that should be relied on during the innovation step. One possible solution that we envision, then, is to apply a low-pass filtering action to the prediction error $\widetilde{s}$ before calculating the update term (B.32) of the filter. The optimal shape and pass-band of such

filter, however, are still to be determined. Finally it might be worth investigating the use of alternative metrics for the discrepancy between $\widehat{s}$ and s such as, e.g. the mutual information successfully exploited in [TM12] for visual control purposes.

B.7 Conclusions

In this appendix we reported some preliminary results in the context of dense photometric structure estimation from motion. We modeled the system dynamics with a system of PDEs and we proposed an observer (also in the form of a system of PDEs) characterized by a structure reminiscent of the one used for the other geometric primitives presented in this thesis. We also underlined, however, some important differences between this case and the other sparse estimation problems addressed in this work. Due to the infinite dimensionality of the problem, for instance, one cannot simply extend the finite dimensional case stability proof to this infinite dimensional case. We also discussed some intuitive observability properties of the system that are similar to those characterizing the other SfM problems. After this, we proposed a physically inspired regularization strategy that can improve the estimation in areas that are scarcely observable, especially if the environment is planar in these areas. We then pointed out the limitations of this modeling and estimation strategy when dealing with occlusions and consequent depth discontinuities. Finally we briefly discussed some practical considerations that might suggest the use of some alternative observer structure (use of the estimated image in the propagation phase and low-pass filtering of the prediction error).

At the moment, we have obtained promising results for simple simulated conditions, but we still lack a proper validation in more realistic scenarios. However, we believe that our approach has some potential and we are working towards the resolution of the current issues.

Appendix C

A primer on port-Hamiltonian systems

In this appendix we wish to provide a short and informal overview on the topic of port-Hamiltonian (pH) systems. For a more complete and formal survey on the topic we suggest to refer to [van06], from which most of this material is inspired, or to the many classical works referenced therein. We also give, at the end of the chapter, a brief introduction to bond-graphs, often used for a graphical representation of pH systems.

C.1 Introduction to port-Hamiltonian systems

Port-Hamiltonian systems theory historically originates from the combination of two very classical frameworks: on one hand the Hamiltonian approach which was initially developed in the context of analytic mechanics and, on the other hand, the network approach commonly used in many electrical engineering applications. The pH framework, in fact, provides a powerful tool to analyze (and, more importantly, control) complex systems that can be thought of as an interconnection of simpler Hamiltonian systems. The "appeal" of the pH framework also comes from the fact that it can often generate intuitive physical interpretations for very complex control systems.

Let us consider a very simple example: a mass-spring system. As well known, the dynamics of this system are governed by the second-order differential equation

$$m\ddot{x} + kx = e \tag{C.1}$$

where m is the mass, k is the spring elastic coefficient, e is an external force and x is the system configuration variable, i.e. the position of the mass. Equation (C.1)

can also be rewritten in terms of the linear momentum $p = m\dot{x}$ as

$$\dot{p} + kx = e.$$

Defining the state vector $\mathbf{x} = [x, p]^T$ and the system *Hamiltonian*

$$\mathcal{H}(\mathbf{x}) = \frac{p^2}{2m} + \frac{1}{2}kx^2 \geq 0, \tag{C.2}$$

the dynamics (C.1) can then be rewritten as

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{H}(x,p)}{\partial x} \\ \frac{\partial \mathcal{H}(x,p)}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e = \boldsymbol{S}\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}) + \boldsymbol{B}e, \tag{C.3}$$

which is the *canonical Hamiltonian equation* for a mechanical system, see e.g. [FM06]. Note that matrix

$$\boldsymbol{S} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = -\boldsymbol{S}^T,$$

also called the system *structure matrix*, is skew-symmetric. Matrix $\boldsymbol{B}$ is, instead called the *input matrix*. This structure has important consequences on the system energy balance as it will be explained in the following.

If the system also contains some viscous friction, then one has

$$m\ddot{x} + d\dot{x} + kx = e$$

where $d \geq 0$ is the damping coefficient and (C.3) becomes

$$\dot{\mathbf{x}} = \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & d \end{bmatrix} \right) \begin{bmatrix} \frac{\partial \mathcal{H}(x,p)}{\partial x} \\ \frac{\partial \mathcal{H}(x,p)}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e = (\boldsymbol{S} - \boldsymbol{R})\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}) + \boldsymbol{B}e \tag{C.4}$$

where

$$\boldsymbol{R} = \begin{bmatrix} 0 & 0 \\ 0 & d \end{bmatrix} \succeq 0$$

is also called the system *resistive matrix* and $\mathcal{H}$ is still given by (C.2). Equation (C.4) can be further generalized by making the structure matrix $\boldsymbol{S}$, the resistive matrix $\boldsymbol{R}$ and the input matrix $\boldsymbol{B}$ state dependent. By doing so, one obtains the generic form of *input-state-output port-Hamiltonian systems* that represent a large class of physical processes. The dynamics of these systems have the from

$$\begin{cases} \dot{\mathbf{x}} = [\boldsymbol{S}(\mathbf{x}) - \boldsymbol{R}(\mathbf{x})]\,\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}) + \boldsymbol{B}(\mathbf{x})\mathbf{u} \\ \mathbf{y} = \boldsymbol{B}(\mathbf{x})^T\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}) \end{cases} \tag{C.5}$$

where $\mathbf{x}$ is an element of a q-dimensional manifold $\mathcal{X}$, $\boldsymbol{S}(\mathbf{x}) = -\boldsymbol{S}(\mathbf{x})^T$, $\boldsymbol{R}(\mathbf{x}) = \boldsymbol{R}(\mathbf{x})^T \succeq 0$, $\mathbf{u} \in \mathbb{R}^v$ is a control input and $\mathbf{y} \in \mathbb{R}^v$ is the system output.

As already mentioned, these systems have very interesting energy balance properties. To show this, let us compute the dynamics of $\mathcal{H}(\mathbf{x})$ for a system in the form (C.5). One easily finds that

$$\dot{\mathcal{H}}(\mathbf{x}) = \nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x})^T\dot{\mathbf{x}} = -\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x})^T\boldsymbol{R}(\mathbf{x})\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}) + \mathbf{y}^T\mathbf{u}$$

which, integrated back, gives

$$\mathcal{H}(\mathbf{x}(t)) = \mathcal{H}(\mathbf{x}(t_0)) - \int_{t_0}^{t} \nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}(\tau))^T\boldsymbol{R}(\mathbf{x}(\tau))\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}(\tau))\mathrm{d}\tau + \int_{t_0}^{t}\mathbf{y}(\tau)^T\mathbf{u}(\tau)\mathrm{d}\tau.$$

Now let us assume that $\mathcal{H}$ is lower bounded and that, w.l.o.g., $\mathcal{H} \geq 0$ as it is the case for (C.2). Under this assumption, one can conclude that

$$\begin{aligned}\mathcal{H}(\mathbf{x}(t_0)) &= \mathcal{H}(\mathbf{x}(t)) + \int_{t_0}^{t} \nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}(\tau))^T\boldsymbol{R}(\mathbf{x}(\tau))\nabla_{\mathbf{x}}\mathcal{H}(\mathbf{x}(\tau))\mathrm{d}\tau - \int_{t_0}^{t}\mathbf{y}(\tau)^T\mathbf{u}(\tau)\mathrm{d}\tau \\ &\leq -\int_{t_0}^{t}\mathbf{y}(\tau)^T\mathbf{u}(\tau)\mathrm{d}\tau\end{aligned}$$

i.e. the amount of energy that can be extracted from the system in the interval $[t_0, t]$ (represented by the last term) cannot be larger than the initial amount of energy, $\mathcal{H}(\mathbf{x}(t_0))$, contained in the system at time t_0. In other words, if $\mathcal{H}$ is lower bounded, the pH system (C.5) does not internally generate energy and, therefore, it is said to be *passive*. The couple $(\mathbf{u}, \mathbf{y})$ can be used to inject or extract energy from the system and, for this reason, it is also called a *power port*.

Passivity is a very important concept in control theory because it guarantees some stability properties: intuitively, if a system cannot generate energy on its own, then, if the external input $\mathbf{u}$ is set to zero, the system will either start oscillating or (if $\boldsymbol{R} \succ 0$) it will converge to an equilibrium, but the system trajectories will never diverge. This intuitive fact can be formally demonstrated using Lyapunov's analysis.

While passivity gives an overall energy characterization of a system from an input-output point of view, the pH framework provides more insights also on the internal structure of a system and on how the energy is actually exchanged between its components. To show this, let us consider again the simple mass-spring system. This latter can also be thought of as an interconnection between two different subsystems (the mass and the spring) connected in a "feedback" configuration (see Fig. C.1) with

$$\mathcal{U}: \begin{cases} \dot{x} = u_{\mathcal{U}} \\ y_{\mathcal{U}} = \dfrac{\partial \mathcal{H}_{\mathcal{U}}}{\partial x} = kx \end{cases}, \quad \mathcal{K}: \begin{cases} \dot{p} = u_{\mathcal{K}} \\ y_{\mathcal{K}} = \dfrac{\partial \mathcal{H}_{\mathcal{K}}}{\partial p} = \dfrac{p}{m} \end{cases},$$

and

$$\mathcal{H}_U(x) = \frac{1}{2}kx^2, \quad \mathcal{H}_K(p) = \frac{1}{2m}p^2, \quad \mathcal{H} = \mathcal{H}_U(x) + \mathcal{H}_K(p).$$

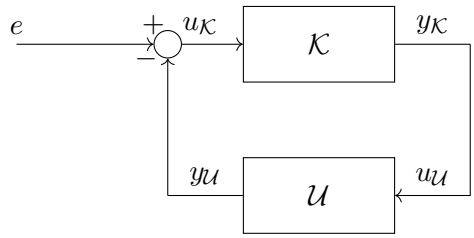


Figure C.1 – **Representation of the mass-spring system as an interconnection between two subsystems**. $\mathcal{K}$ is associated to the mass and stores the kynetic energy of the system. $\mathcal{U}$ is associated to the spring and stores the potential energy of the system.

Note that both $\mathcal{K}$ and $\mathcal{U}$ have a pH structure (C.5) with $\mathbf{x}_{\mathcal{U}} = x$ and $\mathbf{x}_{\mathcal{K}} = p$ respectively, and with $\boldsymbol{S}_{\mathcal{U}} = \boldsymbol{S}_{\mathcal{K}} = 0$, $\boldsymbol{R}_{\mathcal{U}} = \boldsymbol{R}_{\mathcal{K}} = 0$ and $\boldsymbol{B}_{\mathcal{U}} = \boldsymbol{B}_{\mathcal{K}} = 1$. These two systems represent two *energy storage* elements associated with the potential and kinetic energies of the mass-spring system given by $\mathcal{H}_U$ and $\mathcal{H}_K$ respectively. The full system energy is given by the sum of these two contributions and its dynamics is

$$\dot{\mathcal{H}}(x,p) = \frac{\partial \mathcal{H}}{\partial x}\dot{x} + \frac{\partial \mathcal{H}}{\partial p}\dot{p} = \frac{\partial \mathcal{H}}{\partial p}e = \dot{x}e$$

and thus the mass-spring system is passive, as expected, and it can exchange energy with the environment through the *power variables* $(e, \dot{x})$. The two subsystems also have their own power ports, in fact:

$$\dot{\mathcal{H}}_{\mathcal{U}} = \frac{\partial \mathcal{H}_{\mathcal{U}}}{\partial x}\dot{x} = y_{\mathcal{U}}u_{\mathcal{U}}, \quad \dot{\mathcal{H}}_{\mathcal{K}} = \frac{\partial \mathcal{H}_{\mathcal{K}}}{\partial p}\dot{p} = y_{\mathcal{K}}u_{\mathcal{K}}$$

and, through them, they can exchange energy between each other in a conservative way (due to passivity, no energy can either be created or destroyed "inside" the mass-spring system). This internal energy exchange is regulated, in fact, by the power-preserving feedback connection

$$\begin{bmatrix} u_{\mathcal{U}} \\ u_{\mathcal{K}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} y_{\mathcal{U}} \\ y_{\mathcal{K}} \end{bmatrix} + \begin{bmatrix} 0 \\ e \end{bmatrix},$$

between the two systems, that results in the skew-symmetric structure matrix $\boldsymbol{S}$ in (C.3).

When modeling complex physical systems one often ends up with a system of both differential and algebraic equations (DAEs). These can be modeled in the pH framework by replacing the structure matrix $\boldsymbol{S}(\mathbf{x})$ with a more general *Dirac structure* defined as follows. Let $\mathcal{F}$ be a finite-dimensional vector space, called the *flow space* and let $\mathcal{F}^*$ be its dual space (i.e. the space of linear functions on $\mathcal{F}$), called the *effort space*. In general the state space of the pH system $\mathcal{X}$ is a manifold and the flow space is its tangent space $T_{\mathbf{x}}\mathcal{X}$ at $\mathbf{x}$ where as the effort space is the co-tangent space $T^*_{\mathbf{x}}\mathcal{X}$. The space of power variables is given by the Cartesian product

Physical domain	Flow $\boldsymbol{f}$	Effort $\boldsymbol{e}$
Electric	Current	Voltage
Magnetic	Voltage	Current
Potential (mechanics)	Velocity	Force
Kinetic (mechanics)	Force	Velocity
Potential (hydraulic)	Volume flow	Pressure
Kinetic (hydraulics)	Pressure	Volume flow
Chemical	Molar flow	Chemical potential
Thermal	Entropy flow	Temperature

Table C.1 – **Power variables for some physical domains**

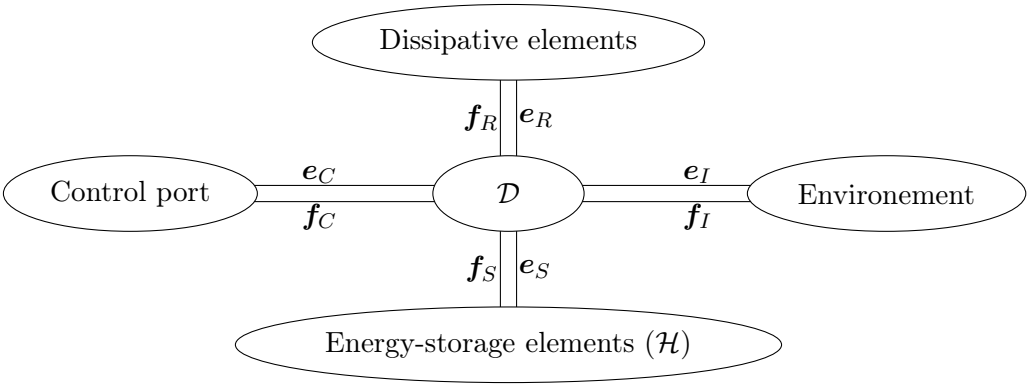


Figure C.2 – **General structure of a port-Hamiltonian system** with the Dirac structure $\mathcal{D}$ in the center and the four ports representing internal energy-storage ($\mathcal{S}$) and dissipative ($\mathcal{R}$) elements and the interconnection with the control action ($\mathcal{C}$) and the environment ($\mathcal{I}$).

between the two $\mathcal{F} \times \mathcal{F}^*$ where the power is defined by the *duality product* between efforts and flows:

$$W = \langle \boldsymbol{f} | \boldsymbol{e} \rangle \,, \quad \boldsymbol{f} \in \mathcal{F}, \boldsymbol{e} \in \mathcal{F}^*$$

In table C.1 we report a list of power variables for some typical modeling domains. On $\mathcal{F} \times \mathcal{F}^*$ one can define the *canonical symmetric bilinear form* $\langle \, , \, \rangle_{\mathcal{F} \times \mathcal{F}^*}$ as

$$\langle (\boldsymbol{f}_1, \boldsymbol{e}_1), (\boldsymbol{f}_2, \boldsymbol{e}_2) \rangle_{\mathcal{F} \times \mathcal{F}^*} = \langle \boldsymbol{f}_1 | \boldsymbol{e}_1 \rangle + \langle \boldsymbol{f}_2 | \boldsymbol{e}_2 \rangle \,, \quad \boldsymbol{f}_i \in \mathcal{F}, \boldsymbol{e}_i \in \mathcal{F}^*$$

Given a subspace $\mathcal{D} \in \mathcal{F} \times \mathcal{F}^*$, its orthogonal complement w.r.t. the bilinear form is defined as:

$$\mathcal{D}^\perp = \left\{ (\boldsymbol{f}_1, \boldsymbol{e}_1) \in \mathcal{F} \times \mathcal{F}^* | \langle (\boldsymbol{f}_1, \boldsymbol{e}_1), (\boldsymbol{f}_2, \boldsymbol{e}_2) \rangle_{\mathcal{F} \times \mathcal{F}^*} = 0, \, \forall (\boldsymbol{f}_2, \boldsymbol{e}_2) \in \mathcal{D} \right\}$$

If $\mathcal{D} = \mathcal{D}^\perp$ then $\mathcal{D}$ is a *constant Dirac structure*. Note that one has

$$\dim \mathcal{D} = \dim \mathcal{D}^\perp = \dim(\mathcal{F} \times \mathcal{F}^*) - \dim \mathcal{D} = 2 \dim \mathcal{F} - \dim \mathcal{D} \iff \dim \mathcal{D} = \dim \mathcal{F}.$$

The Dirac structure defines a power-preserving or power-continuous relation be-

tween the power variables, in fact if $(\boldsymbol{f}, \boldsymbol{e}) \in \mathcal{D}$, then $(\boldsymbol{f}, \boldsymbol{e}) \in \mathcal{D}^{\perp}$ and the bilinear form returns:

$$\langle (\boldsymbol{f}, \boldsymbol{e}), (\boldsymbol{f}, \boldsymbol{e}) \rangle_{\mathcal{F} \times \mathcal{F}^*} = 2 \langle \boldsymbol{f} | \boldsymbol{e} \rangle = 0.$$

In the most general case, then, a pH system can be represented formally, as a Dirac structure with four power ports (see Fig. C.2):

- $\mathcal{S}$ is interconnected to internal energy-storage elements. The energy is defined by the Hamiltonian function $\mathcal{H} : \mathcal{X} \mapsto \mathbb{R}$ and the power balance can be written as $\dot{\mathcal{H}} = - \langle \boldsymbol{f}_S | \boldsymbol{e}_S \rangle$;
- $\mathcal{R}$ is interconnected to internal energy-dissipation elements. The power variables at this port satisfy the constraint $\langle \boldsymbol{f}_R | \boldsymbol{e}_R \rangle \leq 0$;
- $\mathcal{C}$ is interconnected to the control action;
- $\mathcal{I}$ is the port associated with the interaction of the system with its environment.

The interest in pH systems comes from the fact that they have some modularity properties: as shown for the mass-spring example, an interconnection between two pH systems through a Dirac structure is, again, a pH system with total energy (the Hamiltonian function) given by the sum of the energies of the two subsystems. Furthermore, if its Hamiltonian function $\mathcal{H}$ is lower bounded, then a pH system is passive. Conversely, most passive systems can be written in the pH form.

Further generalizations of the pH systems are also possible. In particular, the concept can be extended to infinite dimensional systems governed by algebraic and partial differential equations. We believe that this could be useful to analyze the stability of dense structure estimation schemes as those described in Appendix B, but, at the moment, this is more of a conjecture. For more information about pH systems we suggest, again, to refer to [van06].

C.2 Bond-graphs

A convenient graphical representation of pH systems, that highlights their network properties, is that of bond-graphs, see Fig. C.3. In these graphs each system element is represented by a node and the links between different nodes (called bonds) represent, in general, power connections. The power variables associated to each bond are usually written in the middle of the bond with the effort variable conventionally on the top (or left) and the flow variable on the bottom (or right). The direction in which the power flows positively through the bond is indicated by a half-arrow pointing in the direction of the flow variable. Finally a vertical trait indicates the *causality* of the bond, i.e. the side of the bond on which the effort is imposed.

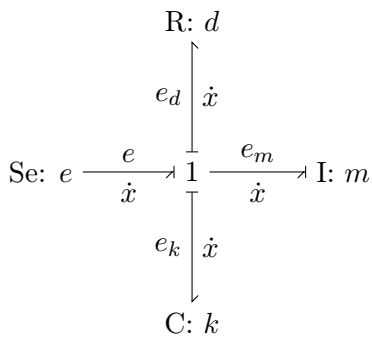


Figure C.3 – **Bond graph representation of the mass-spring-damper system**. The constitive equations for each element are R: $e_d = d\dot{x}$; I: $e_k = k\int \dot{x}(\tau)\mathrm{d}\tau$; C: $\dot{x} = \frac{1}{m}\int e_m(\tau)\mathrm{d}\tau$, and finally, for the 1-junction, $e - e_k - e_d - e_m = 0$ where the signs in the summation were chosen according to the direction of positive power represented by the half-arrows.

Bond graphs can also contain "signal bonds", i.e. connections in which the amount of power exchanged is negligible. These are represented by full arrows and they are used to indicate measurements or parameters. The basic elements that can be represented on the bond graph are:

C elements describe integral relations between flow and effort, e.g. $e(t) = \frac{1}{C}\int f(\tau)\mathrm{d}\tau$. They are used to represent energy storage elements such as electric capacitors;

I elements describe integral relations between effort and flow , e.g. $f(t) = \frac{1}{L}\int e(\tau)\mathrm{d}\tau$. They are used to represent dual energy storage elements such as electric inductors;

R elements describe algebraic relations between flow and effort, e.g. $e = Rf$. They are used to represent dissipative elements such as electric resistors or viscous friction;

Se sources of effort represent constraints on the effort variable, i.e. they fix the effort independently on the flow. They are used to represent effort generators such as electric voltage supplies;

Sf sources of flow represent constraints on the flow variable, i.e. they fix the flow independently on the effort. They are used to represent flow generators such as electric current supplies.

The interconnection between different elements can be done via power-continuous nodes in which energy can only flow and can never be created/accumulated/dissipated:

0 junctions represent interconnections in which all power ports have the same effort and the flows sum to zero. They are equivalent to parallel electric connections and represent the Kirchoff's current law;

1 junctions represent interconnections in which all power ports have the same flow and the efforts sum to zero. They are equivalent to serial electric connections and represent the Kirchhoff's voltage law;

TF transformers describe relations between efforts and flows at the input and output ports, e.g. $\boldsymbol{e}_2 = a\boldsymbol{e}_1$ and $a^T\boldsymbol{f}_2 = \boldsymbol{f}_1$. Note that the power conservation imposes $\boldsymbol{e}_2^T\boldsymbol{f}_2 = \boldsymbol{e}_1^T\boldsymbol{f}_1$. Transformers can also be *modulated* (MTF) in the sense that the transformation can depend on some external signal. The velocity transformation described by the robot Jacobian is an example in this sense where one has $\boldsymbol{f}_2 = \dot{\boldsymbol{r}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{f}_1$, and, because of the power conservation condition, $\boldsymbol{e}_1 = \boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{\epsilon} = \boldsymbol{J}(\boldsymbol{q})^T\boldsymbol{e}_2$.

GY gyrators describe crossed relations between efforts and flows at the input and output ports, e.g. $\boldsymbol{e}_2 = a\boldsymbol{f}_1$ and $a^T\boldsymbol{f}_2 = \boldsymbol{e}_1$. Again note that $\boldsymbol{e}_2^T\boldsymbol{f}_2 = \boldsymbol{e}_1^T\boldsymbol{f}_1$. As for transformers, also gyrators can be modulated (MGY).

To give a simple example, the mass-spring-damper system could be represented, using bond-graphs, as in Fig. C.3. For more information about bond graphs and their use for modeling pH systems one can refer to, e.g., [DMSB09].

References

Thesis related publications

[1] R. Spica and P. Robuffo Giordano, "A Framework for Active Estimation: Application to Structure from Motion," in *52nd IEEE Conf. on Decision and Control*, Florence, Italy, Dec. 2013, pp. 7647–7653.

[2] P. Robuffo Giordano, R. Spica, and F. Chaumette, "An Active Strategy for Plane Detection and Estimation for a Monocular Camera," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014, pp. 4755–4761.

[3] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion for Spherical and Cylindrical Targets," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, Jun. 2014, pp. 5434–5440.

[4] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Plane Estimation by Active Vision from Point Features and Image Moments," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 6003–6010.

[5] P. Robuffo Giordano, R. Spica, and F. Chaumette, "Learning the Shape of Image Moments for Optimal 3D Structure Estimation," in *2015 IEEE Int. Conf. on Robotics and Automation*, Seattle, WA, May 2015, pp. 5990–5996.

[6] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion: Application to Point, Sphere and Cylinder," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1499–1513, 2014.

[7] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Coupling Visual Servoing with Active Structure from Motion," in *2014 IEEE Int. Conf. on Robotics and Automation*, Hong Kong, China, May 2014, pp. 3090–3095.

[8] R. Spica, P. Robuffo Giordano, and F. Chaumette, "A Framework for Coupling Visual Control and Active Structure from Motion," in *2015 IEEE Int. Conf. on Robotics and Automation Workshop: Scaling Up Active Perception*, Seattle, WA, May 2015.

[9] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Bridging Visual Control and Active Perception via a Large Projection Operator," *IEEE Trans. on Robotics*, under consideration for publication.

[10] R. Spica, G. Claudio, F. Spindler, and P. R. Giordano, "Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms," in *2014 IEEE Int. Conf. on Robotics and Automation workshop: MATLAB/Simulink for Robotics Education and Research*, Hong Kong, China, Jun. 2014.

Bibliography

[AABM14] G. Allibert, D. Abeywardena, M. Bangura, and R. Mahony, "Estimating body-fixed frame velocity and attitude from inertial measurements for a quadrotor vehicle," in *2014 IEEE Conf. on Control Applications*, Antibes-Juan Les Pins, France, Oct. 2014, pp. 978–983.

[AAM14] J. D. Adarve, D. J. Austin, and R. Mahony, "A Filtering Approach for Computation of Real-Time Dense Optical-flow for Robotic Applications," in *2014 Australasian Conf. on Robotics and Automation*, Melbourne, Australia, Dec. 2014.

[ABT71] G. F. Amelio, W. J. J. Bertram, and M. F. Tompsett, "Charge-coupled imaging devices: Design considerations," *IEEE Trans. on Electron Devices*, vol. 18, no. 11, pp. 986–992, Nov. 1971.

[ACGM94] Y. Aoustin, C. Chevallereau, A. Glumineau, and C. H. Moog, "Experimental results for the end-effector control of a single flexible robotic arm," *IEEE Trans. on Control Systems Technology*, vol. 2, no. 4, pp. 371–381, 1994.

[Adi85] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 7, no. 4, pp. 384–401, 1985.

[AEK+04] H. Antti, T. M. Ebbels, H. C. Keun, M. E. Bollard, O. Beckonert, J. C. Lindon, J. K. Nicholson, and E. Holmes, "Statistical experimental design and partial least squares regression analysis of biofluid metabonomic NMR and clinical chemistry data for screening of adverse drug effects," *Chemometrics and intelligent laboratory systems*, vol. 73, no. 1, pp. 139–149, 2004.

[AK71] S. Arimoto and H. Kimura, "Optimum input test signals for system identification–an information-theoretical approach," *Int. Journ. of Systems Science*, vol. 1, no. 3, pp. 279–290, 1971.

[AOB14] H. J. Asl, G. Oriolo, and H. Bolandi, "An adaptive scheme for image-based visual servoing of an underactuated UAV," *Int. Journ. of Robotics and Automation*, vol. 29, no. 1, 2014.

[AOM02] E. Altuğ, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *2002 IEEE Int. Conf. on Robotics and Automation*, vol. 1, Washington, DC, May 2002, pp. 72–77.

[APC13] D. Agravante, J. Pages, and F. Chaumette, "Visual servoing for the reem humanoid robot's upper body," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 5233–5238.

[Arm89] B. Armstrong, "On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics," *Int. Journ. of Robotics Research*, vol. 8, no. 6, pp. 28–48, 1989.

[ASNM10] J. Arróspide, L. Salgado, M. Nieto, and R. Mohedano, "Homography-based ground plane detection using a single on-board camera," *IET Intell. Transp. Syst.*, vol. 4, no. 2, pp. 149–160, 2010.

[AT67] M. Athans and E. Tse, "A Direct Derivation of the Optimal Linear Filter Using the Maximum Principle," *IEEE Trans. on Automatic Control*, vol. 12, no. 6, pp. 690–698, Dec. 1967.

[AWB88] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. Journ. on Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.

[AWCS13] M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "Path planning for motion dependent state estimation on micro aerial vehicles," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 3926–3932.

[Baj88] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[BCM13] M. Bakthavatchalam, F. Chaumette, and E. Marchand, "Photometric moments: New promising candidates for visual servoing," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 5241–5246.

[BDRDS+13] D. Borrmann, P. J. De Rezende, C. C. De Souza, S. P. Fekete, S. Friedrichs, A. Kröller, A. Nüchter, C. Schmidt, and D. C. Tozoni, "Point guards and point clouds: Solving general art gallery problems," in *29th annual Symp. on Computational Geometry*, New York, NY, USA, Jun. 2013, pp. 347–348.

[BDW06] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[Bea72] A. E. Beach, *The Science record; a compendium of scientific progress and discovery.* New York, Munn, 1872.

[BELN11] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a new Accumulator Design," *3D Research*, vol. 2, no. 2, pp. 1–13, 2011.

[Ber09] D. S. Bernstein, *Matrix Mathematics: Theory, Facts, and Formulas*, 2nd ed. Princeton University Press, 2009.

[BH96] G. Besançon and H. Hammouri, "On uniform observation of nonuniformly observable systems," *Systems & control letters*, vol. 29, no. 1, pp. 9–19, 1996.

[BJ03] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[BJB94] J. L. Barron, D. J. F. J, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. Journ. on Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[BK12] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library.* O'Reilly & Associates, 2012.

[BMH03] J. P. Barreto, F. Martin, and R. Horaud, "Visual servoing/tracking using central catadioptric images," in *Experimental Robotics VIII*. Springer, 2003, pp. 245–254.

[BMW+02] F. Bourgaul, A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, Lausanne, Switzerland, Oct. 2002, pp. 540–545.

[Bow12] M. D. Bowan, "Integrating vision with the other senses," 2012.

[BP73] R. Bolles and R. Paul, "The use of sensory feedback in a programmable assembly system," DTIC Document, Tech. Rep., 1973.

[BRG13] G. Besançon, I. Rubio Scola, and D. Georges, "Input selection in observer design for non-uniformly observable systems," in *9th IFAC Symp. on Nonlinear Control Systems*, Toulouse, France, Sep. 2013, pp. 664–669.

[BS70] W. S. Boyle and G. E. Smith, "Charge coupled semiconductor devices," *Bell System Technical Journal*, vol. 49, no. 4, pp. 587–593, 1970.

[BS83] S. Boyd and S. Sastry, "On parameter convergence in adaptive control," *Systems & control letters*, vol. 3, no. 6, pp. 311–319, 1983.

[BSL+11] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, and M. J. B. R. Szeliski, "A database and evaluation methodology for optical flow," *Int. Journ. on Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[BSLK04] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[Bur74] J. M. Burgers, *The nonlinear diffusion equation : asymptotic solutions and statistical problems*. Dordrecht-Holland; Boston : D. Reidel Pub. Co, 1974.

[BW04] L. Beyer and J. Wulfsberg, "Practical robot calibration with ROSY," *Robotica*, vol. 22, no. 05, pp. 505–512, 2004.

[BWDA00] J. E. Banta, L. M. Wong, C. Dumont, and M. A. Abidi, "A next-best-view system for autonomous 3-d object reconstruction," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 30, no. 5, pp. 589–598, 2000.

[CAK99] F. Conticelli, B. Allotta, and P. K. Khosla, "Image-based visual servoing of nonholonomic mobile robots," in *38th IEEE Conf. on Decision and Control*, vol. 4, Phoenix, AZ, Dec. 1999, pp. 3496–3501.

[Cam08] A. A. Campbell-Swinton, "Distant electric vision," *Nature*, vol. 78, p. 151, 1908.

[CBBJ96] F. Chaumette, S. Boukir, P. Bouthemy, and D. Juvin, "Structure from controlled motion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, no. 5, pp. 492–504, 1996.

[CCSS91] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. Journ. of Robotics Research*, vol. 10, no. 4, pp. 410–425, 1991.

[CDW+14] N. Cazy, C. Dune, P.-B. Wieber, P. Robuffo Giordano, and F. Chaumette, "Pose error correction for visual features prediction," in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Chicago, IL, Sep. 2014, pp. 791–796.

[CFJS02] A. Chiuso, P. Favaro, H. Jin, and S. Soatto, "Structure from Motion Causally Integrated Over Time," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 523–535, 2002.

[CH66] R. Courant and D. Hilbert, *Methods of mathematical physics.* CUP Archive, 1966, vol. 1.

[CH06] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[CH07] F. Chaumette and S. Hutchinson, "Visual servo control, Part II: Advanced approaches," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[CH10] S. Candido and S. Hutchinson, "Minimum uncertainty robot path planning using a pomdp approach," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 1408–1413.

[Cha94] F. Chaumette, "Visual servoing using image features defined upon geometrical primitives," in *1994 IEEE Conf. on Decision and Control*, Lake Buena Vista, FL, Dec. 1994, pp. 3782–3787.

[Cha04] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.

[Che53] H. Chernoff, "Locally optimal designs for estimating parameters," *Annals of Mathematical Statistics*, vol. 24, pp. 586–602, 1953.

[CK88] C. Chevallereau and W. Khalil, "A new method for the solution of the inverse kinematics of redundant robots," in *1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, Apr. 1988, pp. 37–42.

[CK08] R. Cohen and L. Katzir, "The generalized maximum coverage problem," *Information Processing Letters*, vol. 108, no. 1, pp. 15–22, 2008.

[CM10] A. Cristofaro and A. Martinelli, "Optimal Trajectories for Multi Robot Localization," in *49th IEEE Conf. on Decision and Control*, Atlanta, GA, Dec. 2010, pp. 6358–6364.

[CM11] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 828–834, 2011.

[CMKB02] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," in *2002 IEEE Int. Conf. on Robotics and Automation*, vol. 2, Washington, DC, May 2002, pp. 1327–1332.

[CMS11] A. I. Comport, R. Mahony, and F. Spindler, "A visual servoing model for generalised cameras: Case study of non-overlapping cameras," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 5683–5688.

[Cor93] P. I. Corke, *Visual Control of Robot Manipulators – A Review.* World Scientific Press, 1993, ch. 1, pp. 1–31.

[Cor10] P. Corke, "Spherical Image-Based Visual Servo and Structure Estimation," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 5550–5555.

[Cor11] P. I. Corke, *Robotics, Vision & Control: Fundamental Algorithms in Matlab.* Springer, 2011.

[CRE91] F. Chaumette, P. Rives, and B. Espiau, "Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing," in *1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, Apr. 1991, pp. 2248–2253.

[CSKW03] K.-H. Cho, S.-Y. Shin, W. Kolch, and O. Wolkenhauer, "Experimental design in systems biology, based on parameter sensitivity analysis using a monte carlo method: A case study for the tnfα-mediated nf-κ b signal transduction pathway," *Simulation*, vol. 79, no. 12, pp. 726–739, 2003.

[Dav03] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *2003 IEEE Int. Conf. on Computer Vision*, Nice, France, Oct. 2003, pp. 1403–1410.

[DBSM07] F. Dias, F. Berry, J. Serot, and F. Marmoiton, "Hardware, Design and Implementation Issues on a Fpga-Based Smart Camera," in *1st ACM/IEEE Int. Conf. on Distributed Smart Cameras*, Vienna, Austria, Sep. 2007, pp. 20–26.

[DK06] F. Dellaert and M. Kaess, "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *Int. Journ. of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[DLMO00] A. De Luca, R. Mattone, and G. Oriolo, "Stabilization of an underactuated planar 2r manipulator," *Int. Jour. of Robust and Nonlinear Control*, vol. 10, no. 4, pp. 181–198, 2000.

[DM02] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[DMSB09] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach.* Springer, 2009.

[DOR07] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "On-Line Estimation of Feature Depth for Image-Based Visual Servoing Schemes," in *2007 IEEE Int. Conf. on Robotics and Automation*, Rome, Italy, Apr. 2007, pp. 2823–2828.

[DOR08] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature depth observation for image-based visual servoing: Theory and experiments," *Int. Journ. of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.

[DRMS07] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[DS96] J. E. Dennis Jr and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations.* Siam, 1996.

[Dua71] C. B. Duane, "Close-range camera calibration," *Photogrammetric Engineering & Remote Sensing*, vol. 37, no. 8, pp. 855–866, 1971.

[DWB06] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[ECR92] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[Ehr55] E. Ehrenfeld, "On the efficiency of experimental design," *Annals of Mathematical Statistics*, vol. 26, pp. 247–255, 1955.

[EMMH13] A. Eudes, P. Morin, R. Mahony, and T. Hamel, "Visuo-inertial fusion for homography-based filtering and estimation," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 5186–5192.

[FC05] D. Folio and V. Cadenat, "A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment," in *44th IEEE Conf. on Decision and Control and European Control Conf.*, Seville, Spain, Dec. 2005, pp. 3898–3903.

[FC09] R. T. Fomena and F. Chaumette, "Improvements on Visual Servoing From Spherical Targets Using a Spherical Projection Model," *IEEE Trans. on Robotics*, vol. 25, no. 4, pp. 874–886, 2009.

[Fix57] R. S. Fixot, *American Journal of Ophthalmology*, vol. Aug., 1957.

[FKS07] M. Fujita, H. Kawai, and M. W. Spong, "Passivity-Based Dynamic Visual Feedback Control for Three-Dimensional Target Tracking: Stability and L_2-Gain Performance Analysis," *IEEE Trans. on Control Systems Technology*, vol. 15, no. 1, pp. 40–52, 2007.

[FM05] J. Fung and S. Mann, "Openvidia: parallel gpu computer vision," in *13th annual ACM Int. Conf. on Multimedia*, Singapore, Nov. 2005, pp. 849–852.

[FM06] A. Fasano and S. Marmi, *Analytical mechanics: an introduction.* Oxford University Press, 2006.

[FMS06] M. Fruchard, P. Morin, and C. Samson, "A framework for the control of nonholonomic mobile manipulators," *Int. Journ. of Robotics Research*, vol. 25, no. 8, pp. 745–780, 2006.

[FPS14] C. Forster, M. Pizzoli, and D. Scaramuzza, "Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicle," in *2014 Robotics: Science and Systems Conf.*, Berkeley, CA, Jul. 2014.

[Fri96] M. I. Friswell, "The derivatives of repeated eigenvalues and their associated eigenvectors," *Journal of Vibration and Acoustics*, vol. 118, no. 3, pp. 390–397, 1996.

[FS12] F. Fraundorfer and D. Scaramuzza, "Visual odometry, part ii: Matching, robustness, optimization, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[GA10] M. S. Grewal and A. P. Andrews, "Applications of Kalman Filtering in Aerospace 1960 to the Present," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 69–78, 2010.

[GBC+15] V. Gibert, L. Burlion, A. Chriette, J. Boada-Bauxell, and F. Plestan, "A new observer for range identification in perspective vision systems," in *Advances in Aerospace Guidance, Navigation and Control*, 2015, pp. 401–412.

[GBR12] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano, "Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Algarve, Oct. 2012, pp. 2153–2159.

[GBR13] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano, "A Comparison of Scale Estimation Schemes for a Quadrotor UAV based on Optical Flow and IMU Measurements," in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013, pp. 5193–5200.

[GBSO13] B. Guerreiro, P. Batista, C. Silvestre, and P. Oliveira, "Globally Asymptotically Stable Sensor-Based Simultaneous Localization and Mapping," *IEEE Trans. on Robotics*, vol. 29, no. 6, pp. 1380–1395, 2013.

[GBSR15] V. Grabe, H. H. Bülthoff, D. Scaramuzza, and P. Robuffo Giordano, "Nonlinear ego-motion estimation from optical flow for online control

of a quadrotor uav," *Int. Journ. of Robotics Research*, vol. 34, no. 8, pp. 1114–1135, 2015.

[GH07] N. Gans and S. A. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans. on Robotics*, vol. 3, no. 23, pp. 530–540, 2007.

[Gib62] J. J. Gibson, "Observations on active touch." *Psychological review*, vol. 69, no. 6, p. 477, 1962.

[Gib79] J. J. Gibson, *The ecological approach to visual perception.* Houghton, Mifflin and Company, 1979.

[Gib88] E. J. Gibson, "Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge," *Annual review of psychology*, vol. 39, no. 1, pp. 1–42, 1988.

[GK92] M. Gautier and W. Khalil, "Exciting trajectories for the identification of base inertial parameters of robots," *Int. Journ. of Robotics Research*, vol. 11, no. 4, pp. 362–375, 1992.

[GMW81] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization.* Academic Press, 1981.

[GPM89] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice – a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[Har94] R. Hartley, "Euclidean reconstruction from uncalibrated views," in *Applications of Invariance in Computer Vision*, ser. Lecture Notes in Computer Science, J. L. Mundy, A. Zisserman, and D. Forsyth, Eds. Springer, 1994, vol. 825, pp. 235–256.

[HCM94] G. D. Hager, W.-C. Chang, and A. S. Morse, "Robot feedback control based on stereo vision: Towards calibration-free hand-eye coordination," in *1994 IEEE Int. Conf. on Robotics and Automation*, San Diego, CA, May 1994, pp. 2850–2856.

[HDE98] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 4, pp. 525–532, 1998.

[HHC+96] S. Hutchinson, G. D. Hager, P. Corke *et al.*, "A tutorial on visual servo control," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[HK77] R. Hermann and A. J. Krener, "Nonlinear controllability and observability," *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 728–740, 1977.

[HK+89] S. Hutchinson, A. C. Kak *et al.*, "Planning sensing strategies in a robot work cell with multi-sensor capabilities," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 6, pp. 765–783, 1989.

[HKM08] S. Holmes, G. Klein, and D. W. Murray, "A square root unscented Kalman filter for visual monoSLAM," in *2008 IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 3710–3716.

[HM02] T. Hamel and R. Mahony, "Visual servoing of an under-actuated rigid body system: An image based approach," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 2, pp. 187–198, 2002.

[HMT+11] T. Hamel, R. Mahony, J. Trumpf, P. Morin, and M.-D. Hua, "Homography estimation on the special linear group based on direct point correspondence," in *50th IEEE Conf. on Decision and Control and 2011 European Control Conf.*, Orlando, FL, Dec. 2011, pp. 7902–7908.

[HMTP13] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 1736–1741.

[HP79] J. Hill and W. T. Park, "Real time control of a robot with a mobile camera," in *9th Int. Symp. on Industrial Robots*, Washington, D.C., Mar. 1979, pp. 233–246.

[HS81] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[JU97] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, Orlando, FL, Apr. 1997, pp. 182–193.

[JUD95] S. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *1995 American Control Conf.*, vol. 3, Seattle, WA, Jun. 1995, pp. 1628–1632.

[Kai98] T. Kailath, *Linear Systems.* Prentice Hall International, 1998.

[Kal60] R. E. Kalman, "Contributions to the theory of optimal control," in *1st IFAC Congr. on Automatic Control*, Moscow, Russia, Jun. 1960, pp. 481–492.

[KB61] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Journal of Fluids Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[KDS+07] V. Kallem, M. Dewan, J. P. Swensen, G. D. Hager, and N. J. Cowan, "Kernel-based visual servoing," in *2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, Oct. 2007, pp. 1975–1980.

[KGS+11] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 3607–3613.

[Kie74] J. Kiefer, "General equivalence theory for optimum designs (approximate theory)," *The annals of Statistics*, pp. 849–879, 1974.

[KJR+11] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *2011 IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, May 2011, pp. 3281–3288.

[KM85] C. S. Kubrusly and H. Malebranche, "Sensors and controllers location in distributed systems – a survey," *Automatica*, vol. 21, no. 2, pp. 117–128, 1985.

[KM07] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *6th IEEE/ACM Int. Symp. on Mixed and Augmented Reality*, Nara, Japan, nov 2007, pp. 225–234.

[KMM+96] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette, "Visual servoing in robotics scheme using a camera/laser-stripe sensor," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 743–750, 1996.

[KRD08] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Trans. on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[Kre03] A. J. Krener, "The convergence of the extended kalman filter," in *Directions in mathematical systems theory and optimization.* Springer, 2003, pp. 173–182.

[Kru13] E. Kruppa, "Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung," *Sitzungsberichte der Mathematisch Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften*, vol. 122, pp. 1939–1948, 1913.

[KT09] C. Kreutz and J. Timmer, "Systems biology: experimental design," *The FEBS journal*, vol. 276, no. 4, pp. 923–942, 2009.

[KWHT10] O. Koch, M. R. Walter, A. S. Huang, and S. Teller, "Ground robot navigation using uncalibrated cameras," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 2423–2430.

[Lam60] J. H. Lambert, *Photometria Sive de Mensura et Gradibus Luminus, Colorum et Umbrae.* sumptibus viduae Eberhardi Klett, 1760.

[Lan05] M. F. Land, "The optical structures of animal eyes," *Current Biology*, vol. 15, no. 9, pp. R319 – R323, 2005.

[LCG+13] V. Lebastard, C. Chevallereau, A. Girin, N. Servagent, P.-B. Gossiaux, and F. Boyer, "Environment reconstruction and navigation with electric sense based on a kalman filter," *Int. Journ. of Robotics Research*, vol. 32, no. 2, pp. 172–188, 2013.

[Lea09] R. Leardi, "Experimental design in chemistry: a tutorial," *Analytica chimica acta*, vol. 652, no. 1, pp. 161–172, 2009.

[Lev60] M. J. Levin, "Optimum estimation of impulse response in the presence of noise," *IRE Trans. on Circuit Theory*, vol. 7, no. 1, pp. 50–56, 1960.

[LeV02] R. J. LeVeque, *Finite volume methods for hyperbolic problems.* Cambridge University Press, 2002, vol. 31.

[LF92] M. F. Land and R. D. Fernald, "The evolution of eyes," *Annual review of neuroscience*, vol. 15, no. 1, pp. 1–29, 1992.

[LH88] H. C. Longuet-Higgins, "Multiple Interpretations of a Pair of Images of a Surface," *Proc. of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 418, no. 1854, pp. 1–15, 1988.

[LH06] P. F. O. Lindner and B. Hitzmann, "Experimental design for optimal parameter estimation of an enzyme kinetic process based on the analysis of the fisher information matrix," *Journ. of theoretical biology*, vol. 238, no. 1, pp. 111–123, 2006.

[LHD06] C. Leung, S. Huang, and G. Dissanayake, "Active slam using model predictive control and attractor based exploration," in *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Beijing, China, Oct. 2006, pp. 5026–5031.

[Lie77] A. Liegeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 7, no. 12, pp. 868–871, 1977.

[LJPS10] C.-H. Lin, S.-Y. Jiang, Y.-J. Pu, and K.-T. Song, "Robust Ground Plane Detection for Obstacle Avoidance of Mobile Robots Using a Monocular Camera," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 3706–3711.

[LK81] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *7th Int. Joint Conf. on Artificial Intelligence*, vol. 81, Vancouver, BC, aug 1981, pp. 674–679.

[LL86] D.-T. Lee and A. K. Lin, "Computational complexity of art gallery problems," *IEEE Trans. on Information Theory*, vol. 32, no. 2, pp. 276–282, 1986.

[LO91] A. D. Luca and G. Oriolo, "The reduced gradient method for solving redundancy in robot arms," *Robotersysteme*, vol. 7, no. 2, pp. 117–122, 1991.

[Lon81] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.

[Lue64] D. G. Luenberger, "Observing the state of a linear system," *IEEE Trans. on Military Electronics*, vol. 8, no. 2, pp. 74–80, 1964.

[Lue66] D. G. Luenberger, "Observers for multivariable systems," *IEEE Trans. on Automatic Control*, vol. 11, no. 2, pp. 190–197, 1966.

[LXP07] F. L. Lewis, L. Xie, and D. Popa, *Optimal and robust estimation: with an introduction to stochastic control theory.* CRC press, 2007, vol. 29.

[Mar12] A. Martinelli, "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Trans. on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.

[MC07] N. Mansard and F. Chaumette, "Task sequencing for high-level sensor-based control," *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 60–72, 2007.

[MC10] M. Marey and F. Chaumette, "A new large projection operator for the redundancy framework," in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 3727–3732.

[Meh74] R. K. Mehra, "Optimal input signals for parameter estimation in dynamic systems–survey and new results," *IEEE Trans. on Automatic Control*, vol. 19, no. 6, pp. 753–768, 1974.

[MHMM09] E. Malis, T. Hamel, R. Mahony, and P. Morin, "Dynamic estimation of homography transformations on the special linear group for visual servo control," in *2009 IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 1498–1503.

[MK89] A. A. Maciejewski and C. A. Klein, "The singular value decomposition: Computation and applications to robotics," *Int. Journ. of Robotics Research*, vol. 8, no. 6, pp. 63–79, 1989.

[MKC08] R. Mebarki, A. Krupa, and F. Chaumette, "Image moments-based ultrasound visual servoing," in *2008 IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 113–119.

[MKK09] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Trans. on Robotics*, vol. 25, no. 3, pp. 670–685, 2009.

[MKS89] L. Matthies, T. Kanade, and R. Szeliski, "Kalman filter-based algorithms for estimating depth from image sequences," *Int. Journ. on Computer Vision*, vol. 3, no. 3, pp. 209–238, 1989.

[MM13] F. Morbidi and G. L. Mariottini, "Active target tracking and cooperative localization for teams of aerial vehicles," *IEEE Trans. on Control Systems Technology*, vol. 21, no. 5, pp. 1694–1707, 2013.

[MMR10] E. Malis, Y. Mezouar, and P. Rives, "Robustness of Image-Based Visual Servoing With a Calibrated Camera in the Presence of Uncertainties in the Three-Dimensional Structure," *IEEE Trans. on Robotics*, vol. 26, no. 1, pp. 112–120, Feb. 2010.

[MOP07] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Trans. on Robotics*, vol. 23, no. 1, pp. 87–100, 2007.

[Mor80] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Ph.D. dissertation, Stanford University, 1980.

[MS85] L. A. McGee and S. F. Schmidt, "Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry," National Aeronautics and Space Administration, Tech. Rep. 86847, 1985.

[MS05] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[MS12] R. Mahony and S. Stramigioli, "A port-hamiltonian approach to image-based visual servo control for dynamic systems," *Int. Journ. of Robotics Research*, vol. 31, no. 11, pp. 1303–1319, 2012.

[MSC05] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics & Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

[MSK88] L. Matthies, R. Szeliski, and T. Kanade, "Incremental Estimation of Dense Depth Maps from Image Sequences," in *1988 IEEE Conf. on Computer Vision and Pattern Recognition*, Ann Arbor, MI, Jun. 1988, pp. 366–374.

[MSKS03] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An invitation to 3D vision.* Springer, 2003.

[MT95] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive and Robust.* Prentice Hall, 1995.

[Nee62] J. Needham, *Science and Civilisation in China: Volume 4, Physics and Physical Technology, Part 1, Physics.* Cambridge University Press, 1962.

[NHY87] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-Priority Based Redundancy Control of Robot Manipulators," *Int. Journ. of Robotics Research*, vol. 6, no. 2, pp. 3–15, 1987.

[Nis04] D. Nistér, "An efficient solution to the five-point relative pose problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, 2004.

[NNB06] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journ. of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[NRM09] E. D. Nerurkar, S. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," in *2009 IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, May 2009, pp. 1402–1409.

[O'r87] J. O'rourke, *Art gallery theorems and algorithms.* Oxford University Press, 1987.

[Par06] K.-J. Park, "Fourier-based optimal excitation trajectories for the dynamic identification of robots," *Robotica*, vol. 24, no. 05, pp. 625–633, 2006.

[PBGM62] L. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mishchenko, *The Mathematical Theory of Optimal Processes.* Interscience, 1962.

[PCCB10] A. Petiteville, M. Courdesses, V. Cadenat, and P. Baillion, "On-line estimation of the reference visual features application to a vision based long range navigation task," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 3925–3930.

[Pit99] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016–1030, 1999.

[PL00] T. Papadopoulo and M. I. A. Lourakis, "Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications," in *European Conf. on Computer Vision'00.* Dublin: Springer, Jun. 2000, pp. 554–570.

[PNF+08] M. Pollefeys, D. Nistér, J.-M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed Real-Time Urban 3D Reconstruction from Video," *Int. Journ. on Computer Vision*, vol. 78, no. 2-3, pp. 143–167, 2008.

[PPTN08] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Trans. on Robotics*, vol. 24, no. 5, pp. 946–957, 2008.

[PSI+13] L. Polok, M. Solony, V. Ila, P. Smrz, and P. Zemcik, "Efficient implementation for block matrix operations for nonlinear least squares problems in robotic applications," in *2013 IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, May 2013, pp. 2263–2269.

[RAS09] E. A. Rady, M. M. E. Adb El-Monsef, and M. M. Seyam, "Relationships among several optimality criteria," *InterStat*, vol. 247, pp. 1–11, 2009.

[RBG13] I. Rubio Scola, G. Besançon, and D. Georges, "Input optimization for observability of state affine systems," in *5th IFAC Symp. on System Structure and Control*, vol. 5, no. 1, Grenoble, France, Feb. 2013, pp. 737–742.

[RCB04] S. D. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: a survey," *Pattern Recognition*, vol. 37, no. 3, pp. 429–446, 2004.

[RDO08] P. Robuffo Giordano, A. De Luca, and G. Oriolo, "3D structure identification from image moments," in *2008 IEEE Int. Conf. on Robotics and Automation*, Pasadena, CA, May 2008, pp. 93–100.

[RFSB13] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *Int. Journ. of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[RH97] M. Riordan and L. Hoddeson, "The origins of the pn junction," *IEEE Spectrum*, vol. 34, no. 6, pp. 46–51, 1997.

[RK98] F. Reyes and R. Kelly, "Experimental evaluation of fixed-camera direct visual controllers on a direct-drive robot," in *1998 IEEE Int. Conf. on Robotics and Automation*, vol. 3, Leuven, Belgium, May 1998, pp. 2327–2332.

[RLH12] W. Rackl, R. Lampariello, and G. Hirzinger, "Robot excitation trajectories for dynamic parameter estimation using optimized b-splines," in *2012 IEEE Int. Conf. on Robotics and Automation*, Saint Paul, MN, May 2012, pp. 2042–2047.

[Rus11] J. Russ, *The Image Processing Handbook, Sixth Edition.* CRC Press, 2011.

[RVA+06] P. Renaud, A. Vivas, N. Andreff, P. Poignet, P. Martinet, F. Pierrot, and O. Company, "Kinematic and dynamic identification of parallel mechanisms," *Control engineering practice*, vol. 14, no. 9, pp. 1099–1109, 2006.

[SB03] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Int. Joint Conf. on Artificial Intelligence*, 2003, pp. 1127–1134.

[SB11] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness.* Courier Corporation, 2011.

[SBO+10] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and E. López, "Real-time hierarchical stereo visual SLAM in large-scale environments," *Robotics and Autonomous Systems*, vol. 58, no. 8, pp. 991–1002, 2010.

[Ser89] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 4, pp. 472–490, 1989.

[SF11] D. Scaramuzza and F. Fraundorfer, "Visual odometry, part 1: The first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[SGL+13] S. J. Skates, M. A. Gillette, J. LaBaer, S. A. Carr, L. Anderson, D. C. Liebler, D. Ransohoff, N. Rifai, M. Kondratovich, Ž. Težak, E. Mansfield, A. L. Oberg, I. Wright, G. Barnes, M. Gail, M. Mesri, C. R. Kinsinger, H. Rodriguez, and E. S. Boja, "Statistical design for biospecimen cohort size in proteomics-based biomarker discovery and verification studies," *Journ. of proteome research*, vol. 12, no. 12, pp. 5383–5394, 2013.

[SGT+97] J. Swevers, C. Ganseman, D. B. Tükel, J. De Schutter, and H. Van Brussel, "Optimal robot excitation and identification," *IEEE Trans. on Robotics and Automation*, vol. 13, no. 5, pp. 730–740, 1997.

[SI73] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks," *IEEE Trans. on Pattern Recognition*, vol. 5, no. 2, pp. 99–108, 1973.

[SK08] B. Siciliano and O. Khatib, *Springer handbook of robotics.* Springer, 2008.

[SMD10] H. Strasdat, J. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?" in *2010 IEEE Int. Conf. on Robotics and Automation*, Anchorage, AK, May 2010, pp. 2657–2664.

[SP94] S. Soatto and P. Perona, "On the Exact Linearization of Structure From Motion," California Institute of Technology, Tech. Rep. CIT-CDS 94-018, 1994.

[SR05] R. Sim and N. Roy, "Global A-Optimal Robot Exploration in SLAM," in *2005 IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 661–666.

[SRR03] W. Scott, G. Roth, and J.-F. Rivest, "View planning for automated 3d object reconstruction inspection," *ACM Computing Surveys*, vol. 35, no. 1, 2003.

[SSM62] G. L. Smith, S. F. Schmidt, and L. A. McGee, "Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle," National Aeronautics and Space Administration, Tech. Rep. 135, 1962.

[SSN11] Y. Suttasupa, A. Sudsang, and N. Niparnan, "Plane Detection for Kinect Image Sequences," in *2011 IEEE Int. Conf. on Robotics and Biomimetics*, Phuket, Thailand, Dec. 2011, pp. 970–975.

[SSS06] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," *ACM Trans. on Graphics*, vol. 25, no. 3, pp. 835–846, 2006.

[SSVO09] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control.* Springer, 2009.

[SVL10] M. T. Spaan, T. S. Veiga, and P. U. Lima, "Active cooperative perception in network robot systems using POMDPs," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 4800–4805.

[Sze10] R. Szeliski, *Computer Vision: Algorithms and Applications.* Springer London, 2010.

[TAB+71] M. F. Tompsett, G. F. Amelio, W. J. J. Bertram, R. R. Buckley, W. J. McNamara, J. C. J. Mikkelsen, and D. A. Sealer, "Charge-coupled imaging devices: Experimental results," *IEEE Trans. on Electron Devices*, vol. 18, no. 11, pp. 992–996, Nov. 1971.

[Tay79] J. H. Taylor, "The cramer-rao estimation error lower bound computation for deterministic nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 24, no. 2, pp. 343–344, 1979.

[TBF05] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* MIT Press, 2005.

[TC05] O. Tahri and F. Chaumette, "Point-Based and Region-Based Image Moments for Visual Servoing of Planar Objects," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.

[TEC12] P. Taddei, F. Espuny, and V. Caglioti, "Planar motion estimation and linear ground plane rectification using an uncalibrated generic camera," *Int. Journ. on Computer Vision*, vol. 96, no. 2, pp. 162–174, 2012.

[Tho13a] J. W. Thomas, *Numerical partial differential equations: conservation laws and elliptic equations.* Springer Science & Business Media, 2013, vol. 33.

[Tho13b] J. W. Thomas, *Numerical partial differential equations: finite difference methods.* Springer Science & Business Media, 2013, vol. 22.

[TK01] G. Taylor and L. Kleeman, "Flexible self-calibrated visual servoing for a humanoid robot," in *Australasian Conf. on Robotics and Automation 2001*, Sydney, Australia, Nov. 2001, pp. 79–84.

[TL89] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.

[TL05] S. Thrun and Y. Liu, "Multi-robot SLAM with sparse extended information filers," in *11th Int. Symp. of Robotics Research*, Siena, Italy, Oct. 2005, pp. 254–266.

[TLK+04] S. Thrun, Y. Liu, D. Koller, A. Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. Journ. of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[TM12] C. Teuliere and E. Marchand, "Direct 3D servoing using dense depth maps," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 1741–1746.

[TMCG02] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *2002 IEEE Int. Conf. on Robotics and Automation*, vol. 2, Washington, D.C., May 2002, pp. 1624–1629.

[TMHF00] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, *Bundle adjustment – a modern synthesis.* Springer, 2000, ch. 5, pp. 298–372.

[Tsa87] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journ. of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[TTAE00] V. A. Tucker, A. E. Tucker, K. Akers, and J. H. Enderson, "Curved flight paths and sideways vision in peregrine falcons (Falco peregrinus)," *The Journ. of Experimental Biology*, vol. 203, no. 24, pp. 3755–63, 2000.

[Tuc00] V. A. Tucker, "The deep fovea, sideways vision and spiral flight paths in raptors," *The Journ. of Experimental Biology*, vol. 203, no. 24, pp. 3745–54, 2000.

[UC05] D. Ucinski and Y. Chen, "Time-optimal path planning of moving sensors for parameter estimation of distributed systems," in *44th IEEE Conf. on Decision and Control and 2005European Control Conf.*, Seville, Spain, Dec. 2005, pp. 5257–5262.

[van06] A. van der Schaft, "Port-hamiltonian systems: an introductory survey," in *2006 Int. Congr. of Mathematicians*, Madrid, Spain, Aug. 2006, pp. 1339–1365.

[VBP08] N. Vaskevicius, A. Birk, and K. Pathak, "Fast plane detection and polygonalization in noisy 3D range images," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, France, Sep. 2008, pp. 3378–3383.

[VM07] J. Vogel and K. Murphy, "A non-myopic approach to visual search," in *4th Canadian Conf. on Computer and Robot Vision*, Montreal, Que, May 2007, pp. 227–234.

[VTS12] L. Valente, R. Y.-H. Tsai, and S. Soatto, "Information Gathering Control via Exploratory Path Planning," in *46th IEEE Annual Conf. on Information Sciences and Systems*, Princeton, NJ, Mar. 2012, pp. 1–6.

[Wal43] A. Wald, "On the efficient design of statistical investigations," *Annals of Mathematical Statistics*, vol. 14, pp. 134–140, 1943.

[WHB96] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[Wid76] D. V. Widder, *The heat equation.* Academic Press, 1976, vol. 67.

[WM13] A. D. Wilson and T. D. Murphey, "Optimal trajectory design for well-conditioned parameter estimation," in *2013 IEEE Int. Conf. on Automation Science and Engineering*, Madison, WI, Aug. 2013, pp. 13–19.

[WSM14] A. D. Wilson, J. A. Schultz, and T. D. Murphey, "Trajectory synthesis for fisher information maximization," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1358–1370, 2014.

[WSN85] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic visual servo control of robots: An adaptive image-based approach," in *1985 IEEE Int. Conf. on Robotics and Automation*, vol. 2, St. Louis, MO, Mar. 1985, pp. 662–668.

[WWR93] S. Wijesoma, D. Wolfe, and R. Richards, "Eye-to-hand coordination for vision-guided robot control applications," *Int. Journ. of Robotics Research*, vol. 12, no. 1, pp. 65–78, 1993.

[XMX+10] L. Xin, L. Maoliu, J. Xuesong, Z. Zhijie, and Y. Shanshan, "An adaptive motion estimation algorithm based on mutual information for depth information estimation," in *2010 Int. Conf. on Information Networking and Automation*, vol. 1, Kunming, China, Oct. 2010, pp. V1–285–V1–288.

[XS12] J. Xue and X. Su, "A new approach for the bundle adjustment problem with fixed constraints in stereo vision," *Optik - International Journal for Light and Electron Optics*, vol. 123, no. 21, pp. 1923 – 1927, 2012.

[Yat64] F. Yates, "Sir ronald fisher and the design of experiments," *Biometrics*, pp. 307–321, 1964.

[YFG+10] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, no. 2, pp. 390–396, 2010.

[ZAR12a] N. Zarrouati, E. Aldea, and P. Rouchon, "Robust depth regularization explicitly constrained by camera motion," in *21st IEEE Int. Conf. on Pattern Recognition*, Tsukuba, Japan, Nov. 2012, pp. 3606–3609.

[ZAR12b] N. Zarrouati, E. Aldea, and P. Rouchon, "SO(3)-invariant asymptotic observers for dense depth field estimation based on visual data and known camera motion," in *American Control Conf. 2012*, Montreal, QC, Jun. 2012, pp. 4116–4123.

[ZFBR14] D. Zelazo, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," *Int. Journ. of Robotics Research*, vol. 64, no. 1, pp. 105–128, 2014.

[ZFR14] D. Zelazo, A. Franchi, and P. Robuffo Giordano, "Rigidity theory in se(2) for unscaled relative position estimation using only bearing measurements," in *2014 European Control Conf.*, Strasbourg, France, Jun. 2014, pp. 2703–2708.

[Zha92] H. Zhang, "Optimal sensor placement," in *1992 IEEE Int. Conf. on Robotics and Automation*, Nice, France, May 1992, pp. 1825–1830.

[Zhe00] Z. Zhengyou, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[ZKA+09] T. Zhang, Y. Kang, M. Achtelik, K. Kühnlenz, and M. Buss, "Autonomous hovering of a vision/imu guided quadrotor," in *2009 IEEE Int. Conf. on Mechatronics and Automation*, Changchun, China, Aug. 2009, pp. 2870–2875.