Singularity and Stability Analysis of vision-based controllers

Jorge García Fontán

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

Specialité

Informatique

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Jorge García Fontán

Pour obtenir le grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Singularity and Stability Analysis of vision-based controllers

Thèse dirigée par Mohab Safey El Din et Sébastien Briot

soutenue le mardi 24 janvier 2023

après avis des **rapporteurs** :

M. Andreff	Professeur, Université de Franche-Comté, FEMTO-ST
M. Schicho	Professeur, Johannes Kepler University, RISC

devant le **jury** composé de :

M. Briot	Directeur de recherche, CNRS, LS2N
M. Busé	Directeur de recherche, INRIA Université Côte d'Azur
M. Chaumette	Directeur de recherche, INRIA Rennes-Bretagne Atlantique, IRISA
M. Graillat	Professeur, Sorbonne Université, LIP6
Mme. Fantoni	Directrice de recherche, CNRS, LS2N
M. Safey El Din	Professeur, Sorbonne Université, LIP6

avec **membre invité** :

M. Martinet	Directeur de recherche, INRIA Sophia Antipolis

Résumé

L'**asservissement visuel**, ou *visual servoing* (VS), fait référence aux méthodes de commande en Robotique basées sur des données de vision par ordinateur. Il s'agit d'un vaste domaine de recherche qui trouve des applications dans de nombreux domaines de la robotique : spatiale, aérienne, industrielle, médicale, dans la navigation des voitures autonomes, ou pour la réalité augmentée, entre autres [ZGHL16, AKNP14, MUS15]. Cette thèse porte sur l'étude des **cas d'échec** de l'asservissement visuel référencée image, ou *Image-Based Visual Servoing* (IBVS), où la loi de contrôle est définie sur les l'espace des informations visuelles $\mathbf{s} \in \mathbb{R}^k$ calculées directament dans l'image. Les cas d'échec découlent de deux situations :

- Les **singularités de la matrice d'interaction L** gouvernant le modèle cinématique de l'IBVS, qui peuvent conduire à une perte de contrôlabilité du système [ECR92, MR93], et qui pesent également sur la précision des méthodes d'estimation de la situation de la camèra [ZH06].
- L'existence de **multiples équilibres stables** du système, à cause de quoi la stabilité globale des systèmes IBVS n'est pas garantie [CH06].

Ces questions et leurs effets sur les performances des contrôleurs IBVS sont bien connus de la communauté [CH06]. ; cependant, l'étude des cas d'échec a, jusqu'à récemment, reçu peu d'attention, à cause de la grande complexité des systèmes algébriques concernés. Les singularités découlent des conditions de **perte de rang** de la matrice $\mathbf{L}$, de taille $(k \times 6)$, et ont été décrites pour les cas des informations visuelles très simples : N points [MR93, PENB+21] et 3 droites [BMC16]. D'autre part, les points d'équilibre sont les **minimaux locaux** d'une fonction de l'erreur, qui est **polynomiale** en les variables du système, et n'ont jamais été identifiés précisément, à notre connaissance.

Parce que les systèmes d'équations découlant de ces problèmes sont des polynomiaux avec degrés élevés et en de plusieurs variables, et parce que nous exigeons des résultats *exacts* et *certifiés*, nous privilégions l'utilisation de **méthodes symboliques** plutôt que de méthodes numériques. Nous utilisons donc des outils de la **géométrie algébrique** et du **calcul formel**, en particulier le calcul des **bases de Gröbner** [CLO13] en utilisant des algorithmes et des logiciels de pointe [BES21].

Les deux contributions principales de cette thèse sont :

1. Une analyse complète des **conditions de singularité** liées à l'observation de **quatre et cinq droites**. Ceci étend les résultats précédents concernant les conditions de singularité pour d'autres informations visuelles.
2. Le calcul des **configurations d'équilibre** des controlleurs IBVS basés sur l'observation de $N = 4$ **points**. Il s'agit du premier résultat concernant le calcul des minimaux locaux de l'IBVS.

Cette thèse doctorale a été financée par le projet ANR Sesame (funding ID : ANR-18-CE33-0011), axé sur l'étude des singularités et des propriétés de stabilité de l'asservissement visuel.

Mots clés Asservissement visuel · Singularités · Stabilité globale · Points critiques · Calcul formel · Bases de Gröbner

Abstract

Visual Servoing [CH06, HHC96] refers to the use of computer vision information for robot motion control. It is a large area of research with applications in many fields of Robotics: spatial, aerial, industrial, medical, in the autonomous navigation of vehicles, or for augmented reality among others [ZGHL16, AKNP14, MUS15]. This PhD thesis deals with the study of the **failure cases** of Image-Based Visual Servoing (IBVS), a class of vision-based controllers where the control law is defined on the space of the visual features $\mathbf{s} \in \mathbb{R}^k$ computed from the image measurements. The failure cases arise from two situations:

- The **singularities of the interaction matrix L** that governs the kinematics of IBVS, which can lead to a loss of controllability of the system [ECR92, MR93], and which also impact the **accuracy** of **pose computation** methods [ZH06].
- The existence of **multiple stable equilibria** of the system, as a consequence of which the global stability of IBVS systems is not guaranteed [CH06].

These issues and their effects on the performance of IBVS controllers are well-known in the community [CH06]; however the identification of the failure cases has received little attention until recently, due largely to the complexity of the algebraic systems involved. The singularities arise from the conditions of **rank-deficiency** of the matrix $\mathbf{L}$, of size $(k \times 6)$, and have been described for the cases of very basic image features: N image points [MR93, PENB+21] and 3 straight lines [BMC16]. On the other hand, the points of equilibrium are the **local minima** of a potential error function, which is **polynomial** on the system variables, and have never been identified accurately, to our knowledge.

Because the systems of equations arising from these problems are polynomial, with high degrees and in many variables, and because we require *exact*, *certified* results, we privilege the use of **symbolic methods** over numerical methods. We therefore use tools from **algebraic geometry** and **computer algebra**, in particular **Gröbner bases** [CLO13] computations using state-of-the-art algorithms and software [BES21].

The two main contributions of the thesis are:

1. We performed a complete analysis of the **singularity conditions** in the observation of **four and five image lines.** This extends previous results concerning the singularity conditions for other image features.
2. We obtained the **critical points** of IBVS from the observation of $N = 4$ **point features**. This is the first result regarding the computation of the local minima of IBVS.

This PhD thesis was funded by the French ANR project Sesame (funding ID: ANR-18-CE33-0011), which aims to further our knowledge of the singularities and stability properties of visual servo controllers.

Keywords Visual servoing · Singularities · Global stability · Critical points · Computer algebra · Gröbner bases application

Acknowledgements

The completion of this PhD thesis would not have been possible without the help and support of many people around me.

First, I am deeply grateful to my PhD advisors. To Mohab Safey El Din, for his scientific and personal advice, his optimism and constant encouragement, and for looking after his students during a world epidemic. To Sébastien Briot, who welcomed me warmly in his team in Nantes, and was patient and supportive during these four years. I learned a lot from both of them.

I wish to also thank Nicolas Andreff and Josef Schicho, for their time and effort spent in reviewing this thesis, as well as Laurent Busé, François Chaumette, Isabelle Fantoni and Stef Graillat, for participating in the committee of my defense.

I thank my colleagues from the Sesame project, for the enriching discussions, interesting problems and fruitful collaborations: Abhilash, Alessandro, Alexandre, Ezio, François, Philippe, Swann and Olivier. It was a pleasure working with all of them.

Very warm thanks to everyone in the PolSys team: to Jérémy and Vincent, for their help both with my scientific research and teaching duties; to Andrew, Dimitri, Hieu, Lorenzo, Phuoc, Pierre, Rafael, Sriram, for their company and the time spent together at conferences, and very especially to Georgy and Rémi, for what I hope will be a long-lasting friendship.

To the friends that have accompanied, supported and cared for me during these last four years: especially to Marta, Yago, Fernando, Maider, Valentina, and the many people I have met and lived with at the Cité Universitaire, at Collège d'Espagne and Maison de l'Argentine, who made Paris feel like a home.

Lastly, I am grateful to my parents and my siblings Julia and Andrés, for their tireless and unconditional love and support, which can travel across long distances. I do not know where I am going, but I am here thanks to them.

Contents

Chapter 1

Introduction

1.1 Motivation and problem statement

The objective of this PhD thesis is to explore the **failure cases of vision-based controllers** in Robotics, from a rigorous mathematical standpoint and with the use of **exact computational tools** from **algebraic geometry** and **computer algebra**. The failure cases originate from two sources: the **singularities** of the governing kinematic equations, and the existence of multiple **equilibrium points**, which affects the global asymptotic convergence of the controller.

Vision-based control, or **visual servoing** (VS) [CH06, HHC96], refers to the use of feedback data from **computer vision** for the closed-loop control of robotic tasks, motivated by the interest in robots capable of interacting accurately with their environment, in the presence of uncertainties, or even in unknown surroundings. A standard classification differentiates between **Position-Based Visual Servoing** (**PBVS**) [WSN87], where the visual data is used to estimate the relative camera-object configuration, via some pose determination method, and where the control inputs are defined in the configuration space of the camera, and **Image-Based Visual Servoing** (**IBVS**) [ECR92], in which the current pose of the camera is not recovered, and instead the control law is defined directly in the space of the image measurements. This thesis is focused on IBVS methods, although some of the problems we will treat are of relevance more generally in the fields of computer vision and visual servoing.

In Image-Based Visual Servoing, we define a vector $\mathbf{s} \in \mathbb{R}^k$ of **visual features**, or image parameters, computed from the projection of 3D elements of the scene on the camera image. The visual features can be anything from the coordinates of the projections of 3D points, to straight lines, segments, image moments or, in general, to any object that can be digitally segmented in an image [Cha90, Cha04]. A robot task can then be executed by specifying a target value $\mathbf{s}^\star$ and minimizing an error function $\mathbf{e} = (\mathbf{s} - \mathbf{s}^\star)$ defined as the difference between the current and the reference value of the features.

The kinematics of visual servo schemes are governed by the equations of the interaction model, which relates the velocities of the visual features on the camera image $\dot{\mathbf{s}}$ and the relative camera-object velocity $\mathbf{v}_c \in \mathbb{R}^6$, through the **interaction matrix** $\mathbf{L} \in \mathbb{R}^{k\times 6}$, the Jacobian of the features with respect to the parameters describing the pose of the camera:

$$\dot{\mathbf{s}} = \mathbf{L}\ \mathbf{v}_c.$$

The interaction matrix is therefore key to the design of the possible control laws. The control is effectuated at a kinematic level, by specifying an **input camera velocity** proportional to the error vector, according to a **control matrix** $\mathbf{C} \in \mathbb{R}^{6\times k}$:

$$\mathbf{v}_c = -\lambda\, \mathbf{C}\, (\mathbf{s} - \mathbf{s}^\star) \tag{1.1}$$

with $\lambda > 0$ a scalar constant. An outer control-loop is then used in general to convert the end-effector velocity into the required robot joint inputs. The control matrix $\mathbf{C}$ is typically

based on an estimation of the interaction matrix $\mathbf{L}$, which requires an approximation of some of the parameters of the 3D features, such as the depth distribution of the points in the scene [CH06].

The interaction matrix also plays a central role in the so-called **3D pose localisation problem** [RBPD81, HCLL89, DD95, MC02], closely related to IBVS. A classical problem in computer vision, with applications in robot control and augmented reality among others [MUS15], the pose localisation problem consists of determining the parameters describing the camera location and orientation in its workspace from the correspondences of a set of 3D features and their projections on the image.

Since the introduction of vision-based control in the 80s [Cha90], it has found an extensive application in many fields of Robotics: industrial [ZGHL16], medical [AKNP14, LNBT00], spatial [IOH03], aerial [BMG+09, OC11], in augmented reality [MC02, MUS15], for navigation of autonomous vehicles [DSRC11], etc. The performance of these controllers has been exhaustively demonstrated in practice. IBVS systems are generally stable, and robust with respect to errors in the camera or the robot calibration, or to image measurement errors [DJSW02, CH06, MC08]. However, this performance is by no means guaranteed in general, in particular due to two main reasons:

- **Singularities of the interaction matrix**, which can occur due to the degeneracy of the projections of the features on the image, or for specific configurations of the camera relative to the observed object. When the matrix $\mathbf{L}$ becomes rank-deficient, the system can become **locally uncontrollable**, due to a destabilizing control input [ECR92]. The singularities of the interaction matrix are also known to impact the **accuracy of pose estimation** algorithms and, in particular, they are in some cases related to changes in the number of possible solutions of the problem [Rie14, ZH06].

- The existence of multiple **local minima** of the controlled system other than the desired final configuration. The local minima are **stable points of equilibrium** for the camera, where the control input becomes zero for a non-zero value of the error [CH06, PENB+21]. These points may be located anywhere in the workspace, and arbitrarily close to the desired configuration. Then, the **steady-state error** may be critical for applications with high-precision requirements, such as medical or industrial Robotics.

The existence of regions of the configuration space for which the rank of the interaction matrix drops was an issue identified early on in the visual servoing community [ECR92, MR93]. A common strategy to avoid the singularities is to use a *redundant* number k of visual features (i.e. $k > 6$ for a camera with six spatial degrees of freedom). However, this is not enough to guarantee that the matrix is always of full rank. Furthermore, by using additional features, the system becomes overconstrained, allowing the appearance of local minima [CH06]. The problem of identifying the singularities consists of determining the relative camera-object configurations, if any, for which the matrix $\mathbf{L}$ is rank-deficient. It is a computationally hard problem, as it requires studying the vanishing conditions of the determinant, or the maximal minors, of $\mathbf{L}$, leading to a polynomial equation, or system of equations, of high degree and with a large number of variables: those describing the spatial pose of the camera, as well as the parameters defining the configuration of the 3D object. It is nevertheless a crucial problem

for understanding the performance and the limitations of IBVS and of pose determination methods.

Regarding the stability of IBVS, local asymptotic convergence towards the desired configuration $\mathbf{s}^\star$ is typically guaranteed in a sufficiently small region around $\mathbf{s}^\star$, and if the approximations used to compute the control matrix $\mathbf{C}$ are not too rough [CH06]. In order to ensure global asymptotic stability, the matrix product $\mathbf{L}\,\mathbf{C} \in \mathbb{R}^{k\times k}$ should be positive definite everywhere in the configuration space. However, when using a redundant number $k > 6$ of features, this product is of maximum rank 6, and positive definiteness cannot be ensured. In this case, there may exist multiple local minima of the system, which are configurations such that the error $\mathbf{e} \in \ker \mathbf{C}$. An assessment of the global stability behaviour of visual servoing requires identifying the local minima, and delimiting the regions of attraction, in the configuration space, where the controlled system will converge to each of these points.

The study of the singularities of the interaction matrix, and of the global stability properties, in particular the exact computation of the local minima of IBVS controllers, constitute the core of this PhD thesis. Specifically we address the two following problems, which are developed later in detail:

- The analysis of the **singularity** conditions related to the observation of **image lines**.
- The computation of the **critical points of IBVS** from the observation of **3D points**.

Both of these problems boil down to the resolution of systems of polynomial equations in multiple variables.

The algorithms used to solve polynomial systems of equations fall into two large categories: **numerical** (e.g. Newton's method, numerical homotopy continuation) [AY78, LW93, BT18] and **symbolic methods** (e.g. multivariate resultant, triangular sets, Gröbner bases) [CKY89, ALM99, Buc70, BW98]. While numerical methods can provide efficient approximations of the solutions, they rely on floating point arithmetic and are thus liable to round-off errors; furthermore, guaranteeing the convergence of the methods, or providing a certificate of their outputs is tricky, due to the nonlinearity of the equations. On the contrary, symbolic algorithms deal with **exact representations** of the mathematical objects and the algebraic sets, and their outputs are **certified**. Symbolic methods are thus privileged in applications that require exact or high-precision results, or where numerical methods fail to be reliable due to the complexity of the equations.

In this thesis we rely extensively on the use of algebraic geometry and exact (symbolic) computer algebra tools, in particular **Gröbner bases** computations [CLO13, EF17]. In algebraic geometry, a system of polynomials is said to define a **polynomial ideal**, which consists of all the algebraic combinations of the generating polynomials. The geometric counterpart of an ideal is called an **algebraic variety**, consisting of the set of all the common

(complex) roots of the polynomials in the ideal.

A Gröbner basis is a particular family of generators of a polynomial ideal, which can be computed algorithmically, and that can be used to answer important queries about algebraic sets. They can be used, for instance, to determine whether a polynomial is contained in a given ideal, to eliminate variables from a set of equations, to compute the projections of algebraic sets, or to compute a rational parametrization of the set of solutions.

The works on Gröbner bases theory start off with Bruno Buchberger's PhD thesis in the 60s [Buc65], setting off a large area of research in computer algebra. The current state-of-the-art algorithms for Gröbner bases computations are based on the more modern F4 [Fau99] and F5 [Fau02] algorithms by Faugère. Efficient implementations of these algorithms for solving systems of polynomials exist in computer algebra systems like Magma or Maple, or in libraries such as FGb [Fau10] or `msolve` [BES21].

1.2 Previous works

Regarding the kinematic singularities of visual servoing, a well-known result is the singularity conditions related to the observation of **three image points** [Tho66, MR93]. In this case, a singularity occurs if the three points are aligned, or when the camera centre is contained in the cylinder which contains the three points and is perpendicular to the plane they define, as illustrated in Fig. 1.1. This result was recently extended in [PENB+21] to the observation of **four points**. In the case of four non-coplanar points, there are always between two and six singular configurations for the camera; if the four points lie on the same plane, a singularity occurs if all the points and the camera share the same circle.

In another recent result [BMC16], the singularity conditions related to the observation of **three lines** in space where exposed. For this problem, a singularity occurs if the camera lies on one of two surfaces in space containing the three lines, and described respectively by a quadratic and a cubic polynomial. We will be coming back to this result when we present the analysis of the singularities in the observation of more than three image lines.

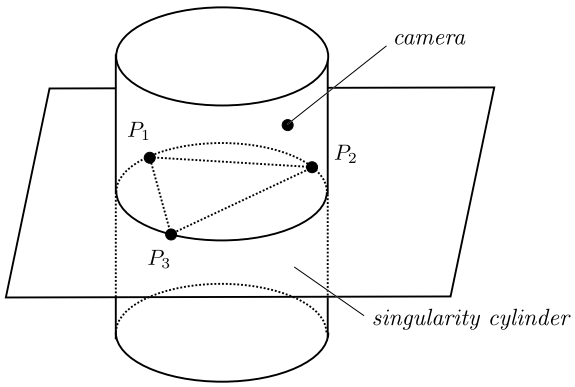


Figure 1.1: Singularity cylinder in the observation of three 3D points.

Regarding the stability properties of IBVS controllers, the local stability around the global minimum of the system was studied in [CH06] by linearizing the kinematic equations of the controlled system. In a small neighbourhood around the desired configuration, **local asymptotic stability** is guaranteed if the approximations used to design the control matrix, such as estimations of the depth distribution of the 3D points in the camera frame, are good enough. A small number of works have dealt with assessing the robustness of these methods with respect to uncertainties in the approximations, but only for simple cases, such as errors in the camera calibration model [Esp94, DJSW02], or for planar objects where the orientation of the plane with respect to the camera frame is not known [MMR09].

To our knowledge, a study of the **global stability** behaviour of visual servoing, taking into account the nonlinearity of the governing equations, has never been addressed so far; neither has the **computation of the local minima** of IBVS from a given set of features. The work presented in this thesis is intended to fill in this gap, developing a strategy to compute the critical points of visual servoing from the observation of image points, using exact computing methods, and presenting the first results of this kind.

1.3 Contributions

In this thesis we achieved two main objectives. The first is to compute the singularity conditions for the interaction model when observing more than three straight lines in space, extending the results of [BMC16] for three lines. The second is the computation of the critical points in IBVS from four reference points, as a first step for a global stability analysis of visual servoing methods.

1.3.1 Singularity analysis in the observation of lines

We performed a complete **singularity analysis** for the **Perspective-4-Line** (P4L) and **Perspective-5-Line** (P5L) problems; that is, a study of the conditions of rank-deficiency of the interaction matrix in the observation of four and five lines in space. It constitutes an extension of the results of [BMC16], in which the singularities of P3L were computed. Both in that work and in ours, the **Plücker representation** is used to describe 3D lines, and their 2D projections: each line in the projective space $\mathbb{P}^3$ is described by six coordinates, subject to a quadratic constraint. The representation of lines by a Plücker vector is redundant, but complete and singularity free. A single image line can be used to control two degrees of freedom of the camera, such that at least three lines are needed to fully constrain the system.

In [BMC16], the authors derive a new basis for the vector space spanned by the rows of the interaction matrix for an image line, and show that this basis describes itself a system of Plücker lines, such that a singularity of the interaction matrix is equivalent to the degeneracy of the system of lines. They use this result to show that a **singularity of P3L** occurs if the camera centre lies on either a **quadratic**, or a **cubic surface**, which depend on the configuration of the observed lines.

In our work, we first provide a new geometric insight into the singularity conditions of P3L exposed in [BMC16]. In fact, the quadratic surface corresponds to the *complementary regulus* of the three feature lines; in other words, it is the locus of all lines which intersect the three

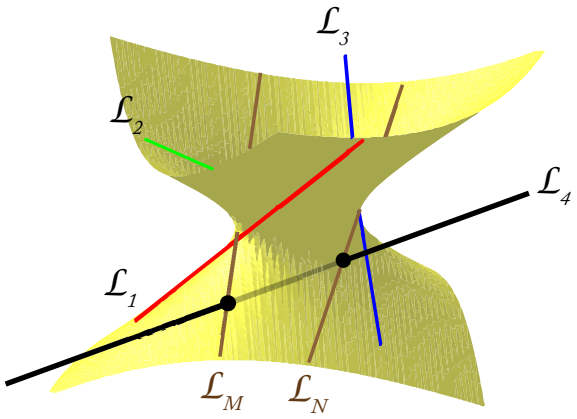


Figure 1.2: Given four spatial lines $\mathcal{L}_1, \ldots, \mathcal{L}_4$, there are in general two (complex) transversal lines $\mathcal{L}_N$ and $\mathcal{L}_M$, which are at the intersection of four hyperboloids.

observed lines, and which span a one-sheeted hyperboloid. This result is further exploited to show that, in the observation of any three or more lines, a singularity will occur if the camera centre lies on any line **transversal** to the observed lines (see Fig. 1.2).

In the case of P4L (and P5L), the number of features used is redundant (i.e. more than the degrees of freedom needed to control). The interaction matrix is thus rectangular, and the conditions for rank-deficiency are described by a system of polynomial equations arising from its maximal minors. We use **Gröbner bases** computations, along with **algebraic geometry** tools (theory of algebraic elimination, saturation of ideals, decomposition and intersection of algebraic varieties) to study the ideal generated by these minors. We find that, for P4L, two types of singularities can occur:

1. If the camera centre lies on one of the **transversal lines**. In general, given four lines in space, there can be 0, 1 or 2 such lines, the solutions of a system of quadratic equations, as illustrated in Fig. 1.2. We provide a condition, as an inequality in terms of the parameters of the feature lines, such that this system does not have any real solutions, and such that this type of singularity can be avoided.

2. In general there can be up to 10 additional camera poses for which the interaction matrix loses rank, and which cannot be avoided. These are the solutions of a zero-dimensional system of equations of higher degree.

In the case of five feature lines, we show that, for a generic configuration, there exist no transversal lines, so there are in general no positive dimensional components of the singularity locus and, if any, the singularities must consist of, at most, a finite number of camera positions. Further, since a singularity of P5L must be a singularity for P4L for any subset of four out of the five lines, we conjecture that there are no singularities at all, outside of the zero-set of some polynomial depending on the system parameters. Our observations for specific configurations of the five lines support this claim.

We support our findings with results from numerical simulations that illustrate the negative effects when performing visual servoing or pose determination near a singular configuration. We find that the ill-conditioning of the interaction matrix results in high magnitude, destabilizing control inputs for IBVS, and in poor, sometimes abhorrent, pose estimations. In particular we observe that the one-dimensional components of the singularity locus, or *line singularities*, have a more acute destabilizing impact, leading to divergence and errors that blow up in magnitude, while the *point singularities* typically only have a local effect.

These results were published as a paper in the International Journal of Computer Vision [GFNBSED22], in collaboration with Abhilash Nayak, Sébastien Briot and Mohab Safey El Din.

1.3.2 Computation of the critical points of IBVS from 4 points

In an effort to better understand the global stability properties of IBVS controllers, we succeeded, for the first time to our knowledge, to compute the points of local equilibrium in the observation of four points in space, using exact polynomial system solving methods.

We focus on the case of controllers that are *gradient-like* with respect to the magnitude of the error vector $V(\mathbf{s}) = \frac{1}{2}||\mathbf{s} - \mathbf{s}^\star||^2$; that is, control strategies such that the system evolves always in the direction of decreasing $V(\mathbf{s})$. For these systems, $V(\mathbf{s})$ is a Lyapunov-like function, and the points of equilibrium correspond to **critical points** of this function with respect to the parameters describing the camera pose. We show that the critical points can be computed as the solution of the following system of equations, where the system parameters, describing the arrangement of the observed points and the desired final camera configuration $\mathbf{s}^\star$, are assumed fixed:

$$\mathbf{L}^T (\mathbf{s} - \mathbf{s}^\star) = \mathbf{0}, \tag{1.2}$$

plus a set of constraints coming from the geometry of the problem. The matrix $\mathbf{L}$ is the usual interaction matrix.

When using N image points as features, the resulting polynomial system is **zero-dimensional** (it has a finite number of solutions), and is tractable using state-of-the-art software for polynomial system solving, even though it is very heavy computationally. We succeeded to solve this system for the cases of $N = 4$ generic points, as well as for special configurations of interest, such as four coplanar points, using `msolve` [BES21], albeit with computing times reaching several weeks over 12 computer cores. An illustration is given in Fig. 1.3.

We then present two improvements of the modeling that allow us to simplify the computations significantly. One of them is based on **exploiting the symmetries** of the solution set by defining a change of coordinates invariant to these symmetries and deriving a new system of equations in the new variables by means of Gröbner bases with an elimination ordering. The other is applicable only in the case of **planar objects** (all the points lying on the same plane), and relies on applying the **Jacobian criterion** to reduce the multiplicity of the solutions. The new system of polynomials obtained has the same solutions as the original system, but the ideal it generates is of lower degree. This allows us to compute the critical points in the case of $N = 4$ generic points in a matter of 2-3 days and, in the case of planar objects, in just a few hours. We present a collection of examples of different configurations with their critical points and describe how to retrieve the camera spatial pose corresponding to each solution in

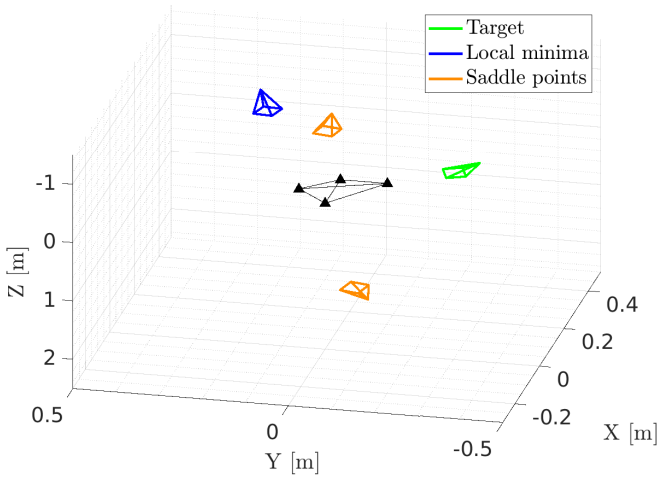


Figure 1.3: Critical points in the observation of a generic planar object.

the space of the system variables, and how to classify them in local minima, maxima or saddle points.

A paper containing the results on the computation of the critical points of IBVS is under preparation and will be submitted to a journal soon.

1.3.3 Structure of the thesis

This thesis is structured in two blocks. In the first part we present the theoretical background necessary to introduce the problem and the methods used in our work. In Chapter 2 we present the preliminaries from Robotics and vision-based control. Starting from a review of rigid-body mechanics, we move onto an introduction of sensor-based control and specifically Image-Based Visual Servoing, leading to the two problems that constitute the topic of the thesis: the singularities of the interaction matrix and the local points of equilibrium of the controlled system. Chapter 3 deals with the mathematical tools from algebraic geometry and computer algebra that we will use to address these problems. We introduce polynomial ideals, as a mathematical structure that generalizes systems of polynomial equations, and their geometric counterparts, algebraic varieties, which are the sets defined by the common solutions of these systems, as well as the different properties and operations that can be performed with these objects. We then give an overview of Gröbner bases, a powerful tool from computer algebra to solve problems with polynomial ideals. We present a summary of their properties and applications, and a historical review of the algorithms to compute them.

The second part is devoted to the contributions of the PhD thesis on the study of IBVS failure cases. Chapter 4 concerns the singularity analysis related to the observation of feature lines. First, we give a new insight into the singularity conditions [BMC16] for the case of three lines, and then we present the study of the singularities for four and five image lines, in generic configurations as well as for lines bounds by orthogonality and parallelism constraints, which are of interest for applications of IBVS in structured environments. A series of experimental

simulations is presented to substantiate these results and evaluate the impact of the singularities on visual servo control and on pose determination algorithms. We also expose the singularity conditions related to minimal combinations of image lines and points features, which are reported in Appendix A. The Chapter 5 is dedicated to the computation of the critical points of IBVS from the observation of N reference points. The problem is modeled as a system of polynomial equations in the space of the image parameters. We then present an algorithmic improvement of this modeling, allowing to simplify the computations of the equilibrium points, and a collection of results for different configurations of the four points using exact polynomial based system solving methods. The results are compared to computations using real homotopy continuation, that illustrate that numerical methods are not reliable for this problem, due to the high nonlinearity of the equations. Finally in Chapter 6 we briefly summarize the main results of the PhD thesis and propose some directions of research for future works.

Part I

Preliminaries

Chapter 2

Image-Based Visual Servoing

Abstract. In this chapter we review the fundamentals of vision-based control and introduce the two main classes of problems that affect the performance of the classical Image-Based Visual Servoing methods, and that constitute the main topic of this thesis: 1) the study of the singularities of the interaction model related to the projection of the 3D features, and 2) the identification of the local minima and the estimation of the regions of convergence. We give an overview of the previous works concerned with the study of these issues and present some of the open problems that are studied in this PhD thesis.

2.1 Preliminaries from Robotics

We start with a brief overview of some mathematical background and common terminology from Robotics and Mechanical Engineering that are necessary to introduce vision-based control. In particular we review rigid-body motions and the velocities of rigid bodies, and define the algebraic mappings involved in the geometric and kinematic description of robots. For most of this section, we follow the book by Park and Lynch, [LP17], which is a comprehensive introduction to the mathematical description of robot architectures, although many other classical references are available [SKK08, MLS17].

2.1.1 Coordinate transformations

In this work we will typically deal with objects that are free to move and rotate in the Cartesian space. We refer to such an object as a **rigid-body**, and we say it has a total of six degrees of freedom, since its configuration can be described by a set of six independent parameters.

Consider a rigid-body $\mathcal{B}$ moving in space. The **pose** of $\mathcal{B}$ represents its position and its orientation relative to a particular frame of reference. Let us define a **spatial frame** of reference $\mathcal{F}_o$ relative to the surrounding scene, and a **body frame** of reference $\mathcal{F}_b$, instantaneously attached to the rigid-body. Then, the pose $\mathbf{g}$ of $\mathcal{B}$ relative to $\mathcal{F}_o$ is described by a couple $(\mathbf{p}, \mathbf{R})$, where $\mathbf{p} \in \mathbb{R}^3$ is the **position vector** of the origin of $\mathcal{F}_b$ expressed in $\mathcal{F}_o$, and $\mathbf{R} \in SO(3)$ is the (3×3) **rotation matrix** describing the orientation of $\mathcal{F}_b$ with respect to $\mathcal{F}_o$. The group of all the rotation matrices in 3D is the Special Orthogonal Group $SO(3)$, or the group of all orthogonal matrices ($\mathbf{R}^{-1} = \mathbf{R}^T$) with $\det(\mathbf{R}) = 1$. With these constraints, a rotation matrix is fully determined by only three independent parameters, as it is required. There exist multiple other representations of a spatial rotation, which we will discuss later.

Note: All the frames of reference considered here are **stationary**. For simplicity we will usually refer to a "frame attached to body $\mathcal{B}$", but we will always mean a reference frame which is *instantaneously* coincident with a coordinate system moving with $\mathcal{B}$. This is not particularly relevant for a geometric and kinematic characterization of rigid-body mechanics, but should be taken carefully for a dynamics study, which we do not deal with in this work.

Often we represent simultaneously the position and orientation of an object using a single (4×4) matrix, denoted a **homogeneous transformation matrix**.

Definition 2.1. *(**Homogeneous transformation matrix**) The pose* $\mathbf{g} = (\mathbf{p}, \mathbf{R})$ *of a rigid-body relative to a particular coordinate system is described by a* (4×4) ***homogeneous transformation matrix**, defined as*

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{2.1}$$

Definition 2.2. *(**Special Euclidean Group**) The set of all homogeneous transformation matrices (equivalently the set of all possible translations and rotations in 3D space), is called the **Special Euclidean Group** $SE(3) = \mathbb{R}^3 \times SO(3)$, or the group of **rigid-body transformations**.*

Using a slight abuse of notation, we will often say $\mathbf{g} = \mathbf{M} = (\mathbf{p}, \mathbf{R}) \in SE(3)$ for the different representations of a spatial pose, even if they correspond to objects with different dimensions.

The group of rigid-body transformations $SE(3)$ is *not* a vector space, but rather it is a *Lie group*, in other words it is also a differentiable manifold. We will later see what the differential of a spatial pose represents. The elements of $SE(3)$ can in fact be used to represent several things:

1. The spatial 3D pose of a rigid-body $\mathcal{B}$ relative to a frame of reference $\mathcal{F}_o$.
2. A spatial transformation (a rotation plus a translation) of a 3D vector $\mathbf{x}$.
3. A change of coordinate system for a vector $\mathbf{x}$.

We defined the homogeneous transformation matrix (2.1) as representing the pose of the body-frame $\mathcal{F}_b$ relative to $\mathcal{F}_o$. In Robotics, this is typically made explicit by writing ${}^o\mathbf{M}_b$, where the subscript indicates the "target" frame and the superscript the "base" frame (see Fig. 2.1). The same transformation matrix is used to convert a vector expressed in $\mathcal{F}_b$-coordinates to its coordinates in $\mathcal{F}_o$. Let $\mathbf{x} \in \mathbb{R}^3$ be a vector, and ${}^b\mathbf{x} = [x_1 \; x_2 \; x_3]^T$ be its coordinates in the frame $\mathcal{F}_b$. Then in the space frame we have

$$\begin{bmatrix} {}^o\mathbf{x} \\ 1 \end{bmatrix} = {}^o\mathbf{M}_b \cdot \begin{bmatrix} {}^b\mathbf{x} \\ 1 \end{bmatrix} \tag{2.2}$$

which is equivalent to

$${}^o\mathbf{x} = \mathbf{R} \cdot {}^b\mathbf{x} + \mathbf{p}. \tag{2.3}$$

Here, $\mathbf{R}$ and $\mathbf{p}$ could also be written ${}^o\mathbf{R}_b$ and ${}^o\mathbf{p}_b$. Note that with this notational convention, the subscript of a transformation matrix that pre-multiplies a vector coincides with the superscript of that vector.

We denote the vector $\begin{bmatrix} {}^b\mathbf{x}^T & 1 \end{bmatrix}^T \in \mathbb{P}^3$ in (2.2) as the vector of **homogeneous coordinates** of ${}^b\mathbf{x}$. Throughout this thesis we will indistinctly use the same name for vectors expressed in projective (homogeneous) or affine (regular) coordinates. Abusing the notation, we will

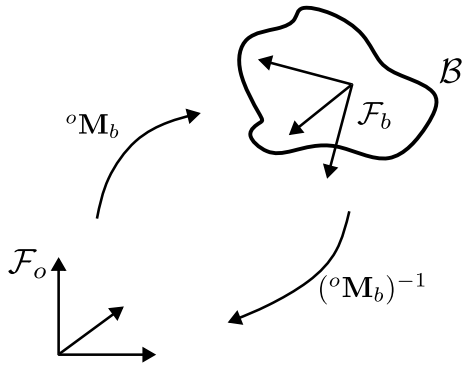


Figure 2.1: A *homogeneous transformation matrix* describes the relative pose of two reference frames.

use ${}^o\mathbf{x} = {}^o\mathbf{M}_b \cdot {}^b\mathbf{x}$, where it should be clear from a dimensional criterion that homogeneous coordinates are being used.

We can perform a change of reference frame in the other sense, from the spatial coordinates to body coordinates, by inverting the homogeneous matrix: ${}^b\mathbf{x} = {}^b\mathbf{M}_o \cdot {}^o\mathbf{x}$, where ${}^b\mathbf{M}_o = ({}^o\mathbf{M}_b)^{-1}$ (Fig. 2.1).

Proposition 2.1. *The **inverse** of a homogeneous matrix* $\mathbf{M}$ *is another homogeneous matrix* $\mathbf{M}^{-1} \in SE(3)$ *defined as*

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.4}$$

Hence, if $(\mathbf{p}, \mathbf{R})$ represents the pose of a frame $\mathcal{F}_b$ relative to $\mathcal{F}_o$, then ${}^b\mathbf{p}_o = -\mathbf{R}^T\mathbf{p}$ is the position vector of the origin of $\mathcal{F}_o$, expressed in $\mathcal{F}_b$, and ${}^b\mathbf{R}_o = \mathbf{R}^T = \mathbf{R}^{-1}$ is the rotation matrix required to transform frame $\mathcal{F}_o$ into $\mathcal{F}_b$.

Just like for spatial rotations, the composition of two rigid-body transformations (or two coordinate system transformations) is performed by multiplying the corresponding homogeneous matrices

Proposition 2.2. *The **product** of two homogeneous matrices* $\mathbf{M}_1$, $\mathbf{M}_2$ *is another homogeneous matrix* $\mathbf{M}_3 = \mathbf{M}_1 \cdot \mathbf{M}_2 \in SE(3)$.

Let $\mathcal{F}_o$ and $\mathcal{F}_b$ be two frames of reference with a relative pose given by ${}^o\mathbf{M}_b$, and consider a new frame $\mathcal{F}_d$ with a pose relative to $\mathcal{F}_o$ given by ${}^o\mathbf{M}_d$. Then, the transformation between frames $\mathcal{F}_b$ and $\mathcal{F}_d$ is described by the following product

$${}^b\mathbf{M}_d = {}^b\mathbf{M}_o \cdot {}^o\mathbf{M}_d = ({}^o\mathbf{M}_b)^{-1} \cdot {}^o\mathbf{M}_d. \tag{2.5}$$

Proposition 2.3. *The multiplication of homogeneous matrices is **associative**:* $\mathbf{M}_1(\mathbf{M}_2\mathbf{M}_3) = (\mathbf{M}_1\mathbf{M}_2)\mathbf{M}_3$*, but in general it is **not commutative**:* $\mathbf{M}_1\mathbf{M}_2 \neq \mathbf{M}_2\mathbf{M}_1$.

It is worth mentioning that Propositions 2.1, 2.2, and 2.3 confirm that set $SE(3)$ of homogeneous transformation matrices is a group.

Note: The Special Euclidean Group $SE(3)$ is a subset of the Euclidean Group $E(3)$: the set of all **isometries** in Euclidean space, or all the transformations in 3D that preserve the distance between any two points. Aside from translations and rotations, these include **reflections about a plane, an axis or through a point**, which preserve distances and angles, but they change the **handedness** of the object. These transformations are sometimes known as *indirect isometries*, to distinguish them from the *direct isometries* (the elements of $SE(3)$). The indirect isometries form a *coset* of $SE(3)$, denoted $E^-(3)$, such that $E^-(3) = E(3) \backslash SE(3)$. Any element of $E(3)$ can also be represented by a vector $\mathbf{p} \in \mathbb{R}^3$ and an orthogonal matrix $\mathbf{R} \in O(3)$, if we drop the constraint $\det(\mathbf{R}) = 1$; therefore we have $E(3) = \mathbb{R}^3 \times O(3)$.

2.1.2 Other representations of rotations

The representation of rotations using (3×3) orthogonal matrices is **redundant**, since there are 9 entries in the matrix and we need six constraints (coming from $\mathbf{R}^T\mathbf{R} = \mathbf{R}\mathbf{R}^T = \mathbf{I}$ and $\det(\mathbf{R}) = 1$) to return to the three-dimensional space of rotations. Likewise, for homogeneous matrices, we need to impose ten constraints on the space of all (4×4) matrices to describe the six-dimensional space $SE(3)$. Other common parametrizations of rotations in the 3D space include: an axis $\mathbf{u}$ and an angle of rotation θ, unit quaternions, or Euler angles:

- **$\theta\mathbf{u}$-vector.** In the $\theta\mathbf{u}$ representation, the vector $\mathbf{u} = [u_x\, u_y\, u_z]^T$ is the normalized direction of the axis of rotation, and it is multiplied by a scalar θ which is the angle of rotation about that axis. This is also a redundant representation, since we assign four parameters to represent the three-dimensional space of rotations, and a constraint of the form $||\mathbf{u}||= 1$ is required. For a fixed $\theta\mathbf{u}$ vector, the rotation matrix $\mathbf{R}$ representing the same rotation can be computed using *Rodrigues' Formula* [LP17]:

$$\mathbf{R} = \mathbf{I} + (\sin\theta)[\mathbf{u}] + (1 - \cos\theta)[\mathbf{u}]^2, \tag{2.6}$$

  where $\mathbf{I}$ is the identity matrix and $[\mathbf{u}]$ is the usual skew-symmetric matrix associated to $\mathbf{u}$ such that its product with any vector $\mathbf{a} \in \mathbb{R}^3$ equals the cross-product $[\mathbf{u}]\,\mathbf{a} = \mathbf{u} \times \mathbf{a}$:

$$[\mathbf{u}] = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}. \tag{2.7}$$

  Inversely, given a rotation matrix $\mathbf{R}$, the axis $\mathbf{u}$ can be computed as the eigenvector of $\mathbf{R}$ that corresponds to an eigenvalue equal to 1 (since $(\mathbf{R} - \mathbf{I})\mathbf{u} = \mathbf{0}$), and the angle of rotation is equal to

$$\theta = \operatorname{acos}\left(\frac{\operatorname{tr}(\mathbf{R}) - 1}{2}\right) \tag{2.8}$$

  where $\operatorname{tr}(\mathbf{R})$ is the trace of the matrix $\mathbf{R}$.

- **Unit quaternions.** Quaternions were introduced by Hamilton in 1843 as a generalization of the complex numbers, and are composed of four independent parameters

$\mathbf{q} = (q_0, q_1, q_2, q_3)$; the group of all quaternions is typically denoted $\mathbb{H}$. Like rotation matrices and the $\theta\mathbf{u}$-vectors, quaternions are a redundant parametrization for rotations. In order to describe a rotation of angle θ about an axis $\mathbf{u}$, we set $q_0 = \cos(\theta/2)$ and $[q_1\ q_2\ q_3]^T = \mathbf{u} \cdot \sin(\theta/2)$. This quaternion satisfies $||\mathbf{q}||^2 = q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, thus leaving three independent parameters as required. Given the quaternion $\mathbf{q} = (q_0, q_1, q_2, q_3)$, the corresponding rotation matrix is

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & -2(q_0 q_3 - q_1 q_2) & 2(q_0 q_2 + q_1 q_3) \\ 2(q_0 q_3 + q_1 q_2) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & -2(q_0 q_1 - q_2 q_3) \\ -2(q_0 q_2 - q_1 q_3) & 2(q_0 q_1 + q_2 q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{2.9}$$

The operations defined on $\mathbb{H}$ are: quaternion multiplication, inversion and composition; the rotation of a vector $\mathbf{x} \in \mathbb{R}^3$ by a quaternion $\mathbf{q}$ corresponds to its composition, in homogeneous coordinates, with $\mathbf{q}$. We refer to [Gra08] for the description of these operations and for a comprehensive introduction to quaternions and their use in rigid-body dynamics.

- **Euler angles.** The relative orientation of two frames is described by three independent parameters that represent successive rotations around the body frame axes to transform one frame to the other. They provide an intuitive parametrization of rotations in 3D space, but not a complete one: three parameters are not enough to fully parametrize the $SO(3)$ manifold. As a consequence, there exist **representation singularities**: there are some spatial orientations which cannot be described by the three angles; this is a well-known problem of the Euler representation, known as a *gimbal lock*. We will therefore avoid using Euler angles in this work, although we refer to [MLS17] for a description of this very common parametrization.

2.1.3 Velocities of rigid bodies

Consider a rigid-body $\mathcal{B}$ moving relative to a fixed spatial frame $\mathcal{F}_o$. As before, we define the body-frame $\mathcal{F}_b$ as the frame which is instantaneously coincident with a frame that moves with the body, and we represent its pose with respect to $\mathcal{F}_o$ by $\mathbf{g} = (\mathbf{p}, \mathbf{R}) = {}^o\mathbf{M}_b$. The motion of $\mathcal{B}$ is parametrized by a **linear** and an **angular velocity** components, which must be defined relative to a point. Unless otherwise specified, we will always consider this point as the origin of the current frame of reference.

Let ${}^o\mathbf{v}_b$ and ${}^o\boldsymbol{\omega}_b$ be the linear and angular velocities of $\mathcal{B}$ relative to $\mathcal{F}_o$, expressed in the spatial frame $\mathcal{F}_o$, and let $\mathcal{P}$ be a point moving with the rigid-body, with spatial coordinates ${}^o\mathbf{x}_p$. The velocity of point $\mathcal{P}$ expressed in $\mathcal{F}_o$ is given by

$${}^o\dot{\mathbf{x}}_p = {}^o\boldsymbol{\omega}_b \times {}^o\mathbf{x}_p + {}^o\mathbf{v}_b \tag{2.10}$$

Note that the linear velocity component ${}^o\mathbf{v}_b$ is *not* the linear velocity of the origin of $\mathcal{F}_b$ in $\mathcal{F}_o$-coordinates (this would be $\dot{\mathbf{p}} = {}^o\boldsymbol{\omega}_b \times \mathbf{p} + {}^o\mathbf{v}_b$), but rather it is the linear velocity of a point attached to the rigid-body which is instantaneously coincident with the origin of the current frame $\mathcal{F}_o$.

The two vectors ${}^o\mathbf{v}_b$ and ${}^o\boldsymbol{\omega}_b$ can be assembled into a single six-dimensional vector, denoted a **velocity twist**.

Definition 2.3. *(**Velocity twist**) The spatial velocity of a rigid-body $\mathcal{B}$ is described by a **velocity twist**: a six-dimensional vector composed of the **linear** and **angular velocity** components $\mathbf{v}$ and $\boldsymbol{\omega}$ of $\mathcal{B}$, expressed about a point:*

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^6 \tag{2.11}$$

Definition 2.4. *Given a rigid-body $\mathcal{B}$ at a given pose, the set of all possible velocity twists it can take generates a vector space called $se(3)$: the **Lie Algebra** of $SE(3)$. It is defined as the tangent space to the $SE(3)$ manifold at the origin.*

We define the **spatial twist** of the frame $\mathcal{F}_b$ as the vector ${}^o\boldsymbol{\tau}_b = \begin{bmatrix} {}^o\mathbf{v}_b^T & {}^o\boldsymbol{\omega}_b^T \end{bmatrix}^T \in se(3)$, expressed relative to $\mathcal{F}_o$. Similarly, we define the **body twist** to be the vector ${}^b\boldsymbol{\tau}_b = \begin{bmatrix} {}^b\mathbf{v}_b^T & {}^b\boldsymbol{\omega}_b^T \end{bmatrix}^T \in se(3)$, expressed in body-frame coordinates. The angular velocity ${}^b\boldsymbol{\omega}_b$ represents the same object in $\mathbb{R}^3$ as ${}^o\boldsymbol{\omega}_b$, but expressed relative to $\mathcal{F}_b$, i.e. ${}^b\boldsymbol{\omega}_b = \mathbf{R}^T \cdot {}^o\boldsymbol{\omega}_b$. The linear velocity component ${}^b\mathbf{v}_b$, on the other hand, is *not* the same object as ${}^o\mathbf{v}_b$, but rather it represents the linear velocity of the origin of $\mathcal{F}_b$, in $\mathcal{F}_b$-coordinates, i.e. ${}^b\mathbf{v}_b = \mathbf{R}^T\dot{\mathbf{p}}$.

In a similar way to how the skew-symmetric matrix $[\boldsymbol{\omega}]$ in (2.7) is a (3×3) matrix representation of an angular velocity $\boldsymbol{\omega} \in \mathbb{R}^3$, we can represent a velocity twist vector $\boldsymbol{\tau} \in \mathbb{R}^6$ by a (4×4) matrix that we call $[\boldsymbol{\tau}]$.

Definition 2.5. *(**Matrix representation of a twist**) The (4×4) matrix representation $[\boldsymbol{\tau}]$ of a velocity twist $\boldsymbol{\tau}$ is*

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^6 \qquad \Longleftrightarrow \qquad [\boldsymbol{\tau}] = \begin{bmatrix} [\boldsymbol{\omega}] & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} \in se(3). \tag{2.12}$$

With this representation, and using homogeneous coordinates for the vector ${}^o\mathbf{x}_p$, equation (2.10) becomes

$${}^o\dot{\mathbf{x}}_p = [{}^o\boldsymbol{\tau}_b]\ {}^o\mathbf{x}_p \tag{2.13}$$

Note that (2.13) is the *instantaneous* velocity of a point that is rigidly attached to a body that moves with a spatial velocity ${}^o\boldsymbol{\tau}_b$ with respect to the fixed frame. If we express the location of that point relative to the body frame of reference ${}^b\mathbf{x}_p = {}^b\mathbf{M}_o\,{}^o\mathbf{x}_p$, then obviously ${}^b\dot{\mathbf{x}}_p = 0$.

We can now introduce the time-derivative of a pose represented as a homogeneous matrix.

Lemma 2.4. *(**Differentiation of homogeneous transformation matrices**) Let ${}^o\mathbf{M}_b = (\mathbf{p}, \mathbf{R})$ be the current pose of a moving frame of reference $\mathcal{F}_b$ relative to a fixed frame $\mathcal{F}_o$, and let ${}^o\boldsymbol{\tau}_b \in se(3)$ be its spatial twist, and ${}^b\boldsymbol{\tau}_b \in se(3)$ its body twist. Then, the time derivative of ${}^o\mathbf{M}_b$ is*

$${}^o\dot{\mathbf{M}}_b = \begin{bmatrix} \dot{\mathbf{R}} & \dot{\mathbf{p}} \\ \mathbf{0} & 0 \end{bmatrix} = [{}^o\boldsymbol{\tau}_b]\ {}^o\mathbf{M}_b = {}^o\mathbf{M}_b\ [{}^b\boldsymbol{\tau}_b]. \tag{2.14}$$

The last two expressions in (2.14) are equivalent. Thus, given the *spatial twist* ${}^o\boldsymbol{\tau}_b$ of a frame $\mathbf{F}_b$, we obtain the derivative of the homogeneous matrix ${}^o\mathbf{M}_b$ by post-multiplying the velocity twist (in its matrix representation) by ${}^o\mathbf{M}_b$ while, if given the body twist $[{}^b\boldsymbol{\tau}_b]$ instead, we pre-multiply the twist by ${}^o\mathbf{M}_b$.

Lemma 2.4 leads to the following proposition, that allows us to change the coordinate system with respect to which a velocity twist is defined.

Proposition 2.5. *Given ${}^o\boldsymbol{\tau} \in se(3)$ a velocity twist expressed in $\mathcal{F}_o$, we can obtain ${}^s\boldsymbol{\tau}$ expressed in any other frame $\mathcal{F}_s$ as*

$$[{}^s\boldsymbol{\tau}] = ({}^o\mathbf{M}_s)^{-1}[{}^o\boldsymbol{\tau}]{}^o\mathbf{M}_s \tag{2.15}$$

and going in the other sense

$$[{}^o\boldsymbol{\tau}] = {}^o\mathbf{M}_s[{}^s\boldsymbol{\tau}]({}^o\mathbf{M}_s)^{-1} \tag{2.16}$$

The expressions above can be rearranged as a matrix-vector product, to relate it to the vector representation of the velocity twist.

Definition 2.6. *(**Adjoint map matrix**) Let $\mathbf{g} = (\mathbf{p}, \mathbf{R})$ be a spatial pose. The **adjoint representation** of $\mathbf{g}$ is*

$$\mathrm{Ad}(\mathbf{g}) = \begin{bmatrix} \mathbf{R} & [\mathbf{p}]\,\mathbf{R} \\ 0 & \mathbf{R} \end{bmatrix} \tag{2.17}$$

With Definition 2.6, (2.16) can be expressed as

$${}^o\boldsymbol{\tau} = \mathrm{Ad}({}^o\mathbf{M}_s){}^s\boldsymbol{\tau} = \begin{bmatrix} \mathbf{R} & [\mathbf{p}]\,\mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \begin{bmatrix} {}^s\mathbf{v} \\ {}^s\boldsymbol{\omega} \end{bmatrix} \tag{2.18}$$

and equivalently, going in the other direction we have ${}^s\boldsymbol{\tau} = \mathrm{Ad}({}^s\mathbf{M}_o){}^o\boldsymbol{\tau}$ where ${}^s\mathbf{M}_o = ({}^o\mathbf{M}_s)^{-1}$.

2.1.4 Some definitions from Robotics

A **robot** is an actuated mechanism composed of a series of (generally rigid) **links** connected by **joints** that allow the relative motion between the different parts. Actuation of some or all of the joints, normally by means of electric motors, is what causes the robot to move. The **end-effector** is the part of the robot where the tool is attached, designed to interact with the environment, such as the gripper at the end of a robotic arm.

When all the links of the robot are assembled in series, we refer to a **serial manipulator**. This is the case of the familiar robotic arms with six-degrees of freedom such as the open chain robot by Kuka in Fig. 2.2a, used extensively in industrial assembly chains, or of the SCARA robot in Fig. 2.2b, which has three translational and one rotational degrees-of-freedom and is used for pick-and-place tasks.

On the other hand, if the robot manipulator consists of two or more kinematic chains (i.e. open chains of links connected by joints) that connect the base to the end-effector in a **closed loop**, we speak of a **parallel robot**. Classical examples of parallel robots include the Gough-Stewart platform or *hexapod* manipulator (Fig. 2.3), with six degrees-of-freedom and widely used for high-payload applications, from testing rigs for vehicle tyres, to flight simulators, as well as in large telescope mounts; or the DELTA robot (Fig. 2.4), one of the most commercially successful robot designs, with only three translational degrees-of-freedom, and with applications in food, medical, electronic and manufacturing industries.

Parallel robots have some advantages over their serial counterparts, such as large payload capacity, high precision, rigidity and speed [Mer05]. However, they also present several drawbacks, notably a smaller workspace, a high coupling in the kinematic relationships and

(a) A robotic arm with six degrees-of-freedom by Kuka. The robot tool can move and rotate in any direction in its workspace.

(b) A SCARA robot by Omicron with four degrees of freedom: the end-effector can move in space in three directions and rotate about the vertical axis.

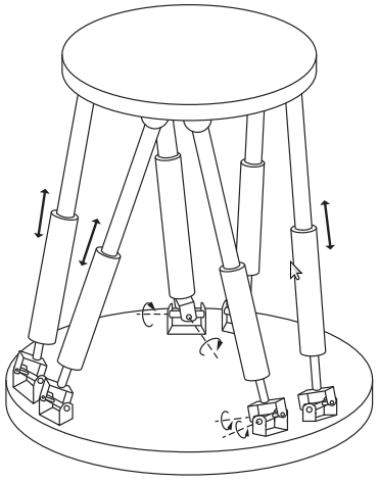
Figure 2.2: Examples of serial or open-chain manipulators.

the presence of singularities in their operational space, which we will discuss later in relation with the singularities characteristic to vision-based control.

In a serial robot, all the joints are actuated, while for parallel manipulators, only a subset of them are. The joints that can be controlled are called the **active joints**, while the **passive joints** are allowed to move freely. The configuration of the joints at each time, that is, the *posture* of the robot, is described by the active joint variables $\mathbf{q}$. The set of all possible values of $\mathbf{q}$ defines the the **configuration space** $\mathcal{Q}$ of the robot.

We also define the **workspace** (or **task space**) $\mathcal{W}$, as the set of all possible configurations that the robot end-effector can attain. For instance, the workspace of the SCARA robot in Fig. 2.2b, which can translate along the three axes and rotate around the vertical direction, is $\mathcal{W} \subseteq \mathbb{R}^3 \times [0, 2\pi]$. For more general manipulators, such as the ones in Fig. 2.2a and Fig. 2.3, that can move and rotate in 3D, the workspace is $\mathcal{W} \subseteq SE(3)$. For these robots, the configuration of the end-effector is described by the relative pose of an end-effector frame $\mathcal{F}_e$ and a fixed base frame of reference $\mathcal{F}_o$, represented by $\mathbf{g} = {}^o\mathbf{M}_e$.

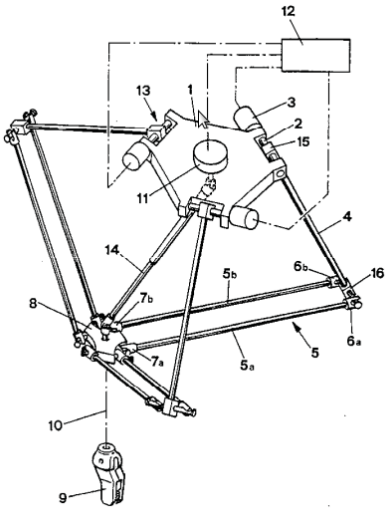
For a serial robot, the position and orientation of the end-effector is uniquely determined by the configuration of the joints [LP17]. The mapping that gives the end-effector pose as a function of the active joint variables $\mathbf{q}$ is called the **Direct Geometric Model** (DGM: $\mathcal{Q} \mapsto \mathcal{W}$). In order to compute the DGM of a robot, one defines a set of intermediate frames of reference $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3 \ldots$ attached to the successive robot links, whose relative configuration depend on the active variables $\mathbf{q}_i$ of each link. Then, the pose of the end-effector can be computed as the



(a) Diagram of a Gough-Stewart platform [LP17].

(b) The YTLA/AMiBA telescope on Mauna Loa in Hawai lies on an hexapod mount [HAC+09].

Figure 2.3: The 6-UPS robot, *Gough-Steward platform* or *hexapod* has six legs connecting the base to the end-effector and can move in six degrees-of-freedom through the actuation of the middle prismatic joints. It can support high loads and perform rapid, high precision motions.



(a) Schematic of the DELTA parallel robot [Cla90].

(b) The ABB FlexPicker DELTA robot.

Figure 2.4: The DELTA parallel robot has three translational degrees-of freedom. The robot platform is always parallel to the base and cannot rotate about the direction normal to that plane.

product of homogeneous matrices

$$ {}^{o}\mathbf{M}_e(\mathbf{q}) = {}^{o}\mathbf{M}_1(\mathbf{q}_1) \cdot {}^{1}\mathbf{M}_2(\mathbf{q}_2) \ldots {}^{k-1}\mathbf{M}_k(\mathbf{q}_k) \cdot {}^{k}\mathbf{M}_e. \tag{2.19} $$

Conversely, for parallel robots, a particular configuration of the active joints can map to several poses of the end-effector. For parallel manipulators it is typically easier to compute the **Inverse Geometric Model** (IGM: $\mathcal{W} \mapsto \mathcal{Q}$), which provides the joint coordinates as a function of the end-effector pose.

By taking the time derivative of the DGM (2.19) of a robot, we obtain the equations that map the velocities of the active joints $\dot{\mathbf{q}}$ to the spatial velocity of the end-effector, represented by a velocity twist $\dot{\mathbf{g}} = \boldsymbol{\tau}_e$, through the **Jacobian** matrix of the forward kinematics $\mathbf{J}$:

$$ \boldsymbol{\tau}_e = \mathbf{J}(\mathbf{q})\, \dot{\mathbf{q}}. \tag{2.20} $$

The relation (2.20) is known as the **Forward Kinematic Model** (FKM) of the robot.

For serial robots, singularities of the forward kinematic Jacobian correspond to configurations where the end-effector loses the ability to move or rotate in one or more directions. For instance, for a robotic arm such as the one in Fig. 2.2a, such a singularity occurs when the robot is in the fully stretched position; in this case the end-effector is unable to move along the direction of the arm. The singularities of the forward Jacobian $\mathbf{J}$ are called *Type I* or **kinematic singularities**.

The *inverse kinematic problem* consists of determining the actuation velocities required for a desired end-effector velocity, and the **Inverse Kinematic Model** (IKM) is defined as follows, where the matrix $\mathbf{J}_{inv}$ is the **inverse kinematic Jacobian**:

$$ \dot{\mathbf{q}} = \mathbf{J}_{inv}(\mathbf{q})\, \boldsymbol{\tau}_e. \tag{2.21} $$

Note that if $\dim(\mathbf{q}) = \dim(\mathbf{g})$ and if the Jacobian matrix $\mathbf{J}$ is never singular, then $\mathbf{J}_{inv} = \mathbf{J}^{-1}$, both for serial or parallel manipulators, although this is in general not the case.

For open chain manipulators, computing the inverse kinematics is usually more involved than computing the FKM, since an end-effector pose can often be reached from different postures, and it is typically done using a generalized inverse of the Jacobian matrix. For parallel robots, neither the Direct nor the Inverse Geometric Models are one-to-one mappings in general; usually we derive the Inverse Kinematics from the geometric constraints of the platform, and then we rely on numerical algorithms along with some information about the current configuration of the platform to compute the Forward Kinematic Model.

Because parallel robots have both active and passive joints, their kinematics are not as straight-forward as for serial ones. In particular parallel robots may present a different kind of singularities than open chain ones, which correspond to the rank-deficiencies of the inverse Jacobian, also known as *Type II* or **parallel singularities** [Mer05]. At a *Type II* singularity, the motion of the end-effector is not fully constrained by the action of the joints. As a consequence, it may gain an instantaneous degree-of-freedom and perform an uncontrollable motion. In Section 2.6, we will talk more about the singularities of the inverse Jacobian of parallel robots, and relate them to the singularities in the kinematic mappings used in vision-based control.

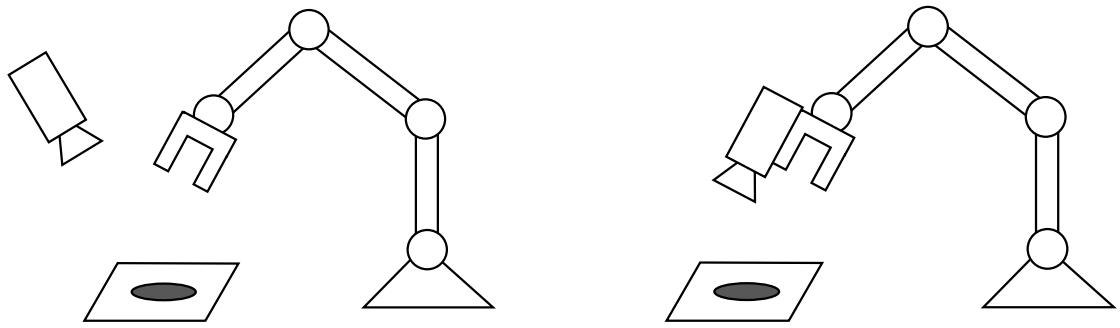


Figure 2.5: Eye-to-hand (left) / Eye-in-hand (right) configurations.

2.2 Sensor-based control

The use of *exteroceptive* sensors for robot control; that is, sensors that can provide information about the external state of the robot, is motivated by the demand for adaptable robots capable of interacting accurately with their environment. Exteroceptive sensors can be used to measure the position and orientation of the end-effector with respect to an object, proximity to obstacles, force, etc., in contrast with *proprioceptive* sensors, used to measure the internal state of a robot (such as its joints configurations, velocities...), and can be used to improve the accuracy of robotic tasks in the presence of uncertainties in their workspace, or to perform navigation tasks in unknown and dynamic surroundings.

Visual Servoing (VS), or vision-based control [CH06, HHC96], refers to a variety of methods for motion control in Robotics that use sensory feedback data from computer vision, that is, information extracted from the image of a camera. The mathematical foundations of VS were laid in the 80s [Cha90], and it has since found an application in a wide variety of Robotic fields: industrial, medical, spatial, aerial, in augmented reality, or for navigation of autonomous vehicles. Depending on the sensor integration mode, we speak of an *eye-in-hand* configuration, if the camera is mounted on the robot, such that the robot's motion induces a motion of the camera; *eye-to-hand*, if the camera is fixed in the workspace, and observes the robot's configuration (see Fig. 2.5), or more complex arrangements, such a stereovision system with multiple cameras observing from different viewpoints. The mathematical development for all these set-ups is similar, but in this PhD thesis we will always consider the classical, eye-in-hand configuration with a single camera.

In very general terms, the aim in vision-based control is to minimize an error function $\mathbf{e}$ which depends on the sensor input. Let the vector $\mathbf{m}(\mathbf{r})$ contain a set of measurements extracted from the visual sensor information (i.e. a set of image point coordinates, the directions of lines projected on the camera plane, the centroid of an object, etc.), which are a function of the configuration $\mathbf{r}$ of the camera. The image measurements are used to compute a set of k features $\mathbf{s}(\mathbf{m}(\mathbf{r}), \mathbf{a})$, where $\mathbf{a}$ is used to represent all the information needed to compute $\mathbf{s}$ that is not directly available from the camera image, such as the camera intrinsic parameters, or a 3D model of the object. The error is defined as the difference between the vector of

features $\mathbf{s}$ and a reference value $\mathbf{s}^\star$, which is specified:

$$\mathbf{e} = (\mathbf{s}(\mathbf{m}(\mathbf{r}), \mathbf{a}) - \mathbf{s}^\star)\,. \tag{2.22}$$

In order to control its motion, it is necessary to define the relation between the relative velocities of the features $\dot{\mathbf{s}}$ and the changes in the camera position $\dot{\mathbf{r}}$:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \dot{\mathbf{r}} \tag{2.23}$$

Definition 2.7. *When the vector $\dot{\mathbf{r}}$ represents the velocity twist of the camera in space (i.e. $\dot{\mathbf{r}} \in se(3)$) with respect to the reference frame of the camera, the relation (2.23) is the **interaction model** associated to the features $\mathbf{s}$, and the Jacobian $\mathbf{L} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}}$ is known as the **interaction matrix** [CH06].*

The standard approach to minimizing the error (2.22) is to define a velocity input that is proportional to the error according to a control matrix $\mathbf{C}$ [Cha90]:

$$\mathbf{v}_e = -\lambda \mathbf{C}\,(\mathbf{s} - \mathbf{s}^\star). \tag{2.24}$$

More elaborate control strategies could incorporate derivative or integral terms to affect performances such as convergence rates, oscillations or steady state errors.

In the current framework of vision-based control, we will always assume that we can define our control inputs as a velocity command for the robot's manipulator, and therefore we will not delve into the robot's Inverse Kinematic Model (IKM) describing the joint actuation inputs required to attain this motion, and which varies from one robot to another. If the inverse kinematic Jacobian $\mathbf{J}_{inv}(\mathbf{q})$ for a robot is known, the joint velocities can be computed from the wanted end-effector velocity as $\dot{\mathbf{q}} = \mathbf{J}_{inv}(\mathbf{q})\,\mathbf{v}_e$.

2.2.1 Classification of visual servo strategies

Visual servoing schemes mainly differ in the way that the vector $\mathbf{s}$ is defined, and whether it can be computed solely from the visual information extracted from the image, or it requires additional 3D information about the system, such as the pose of the camera relative to the scene or a blueprint of the object. According to this, a standard classification [HHC96, CH06] differentiates between

Image-Based Visual Servoing (IBVS). The vector $\mathbf{s}$ is computed directly from the camera input, and the control is effectuated in the 2D space of the image features, so it is a control method in 2D [CH06, HHC96]. No 3D information, such as the spatial location of the camera, is needed to compute the error function (2.22), although it plays a role in the system dynamics through the interaction model (2.23); for instance the interaction matrix related to image points involves the depth coordinates of the 3D points along the focal axis, which is not available from the camera image. Many control strategies rely on this information, which is either obtained from additional sensors [KMM+96] or estimated (heuristically, or from partial pose estimation strategies [CH07]), in order to compute the control matrix in (2.24).

IBVS methods are known to be quite robust and stable [DJSW02], and they present several advantages over other strategies: a full 3D model of the object is not required, hence they are suitable for tasks where the environment is not known, such as robot navigation in unexplored surroundings. In addition, by controlling the robot in the space of image parameters, one can avoid problems common to other approaches, such as features leaving the field of view of the camera. On the other hand, IBVS methods also present some drawbacks, in particular the potential control and stability issues that arise as a consequence of the appearance of singularities and local minima [BCM16, PENB+21]. Furthermore, because the sensor space where trajectories are defined is not the Euclidean space, IBVS methods lead in some cases to unintuitive and inefficient camera motions, where the rotational and translational degrees of freedom are highly coupled [CH06].

Position-Based Visual Servoing (PBVS). Here, the image measurements $\mathbf{m}(t)$ are used to compute the relative position and orientation of the camera with respect to the fixed-frame, and the control input is defined on the task space of the camera [WSN87, CH06], hence, it is a type of 3D servoing. The computation of the pose relies on runtime algorithms and requires 3D information from the object that is not available from the image; such as a blueprint model. Since the robot trajectories are defined in the Euclidean space, PBVS controllers are always globally stable, and usually lead to more efficient trajectories in the 3D space and a nice decoupling between the translational and rotational motions. On the contrary, PBVS methods are very sensitive to modeling errors. Because the visual information is only used to compute the current pose prior to defining the control input, and the feedback loop in terms of the sensor state is removed, errors in the 3D model of the object or in the camera calibration that affect the pose estimation step will propagate onto the final state. Hence, PBVS methods tend to have a worse accuracy than IBVS. Additionally, because the trajectories are not defined in the sensor space, it is a well-known problem that during PBVS the object may leave the field of view of the camera, leading to problems in control and accuracy [TMCG02, CH06].

Hybrid modes. Some approaches try to exploit the advantages of both Image and Position-Based Visual Servoing by integrating 2D and 3D feedback information. For instance in the $2\frac{1}{2}$D Visual Servoing method [MCB99], the authors use a combination of features extracted from the image, and position-based features to separately control the translation and rotational degrees of freedom of the camera, leading to a nice decoupled motion while ensuring that the object never leaves the image frame. It offers good stability properties and produces nice trajectories in Cartesian space, while not requiring a complete 3D model of the object; however it does rely on real-time partial pose estimation to compute the 3D features [CM00].

IBVS methods and their failure cases are the central topic of this thesis and will be dealt with in a lot more detail in what follows. We will hence always assume that the features $\mathbf{s}$ are extracted from the camera image and are a function of the pose of the camera, and the control strategies are defined in the space of the image parameters. Nevertheless, some of the problems studied here, in particular the appearance of *singularities of the interaction model* related to the image features, are also of importance more generally in the domain of computer vision, in particular for *pose determination*. The problem of pose determination consists of

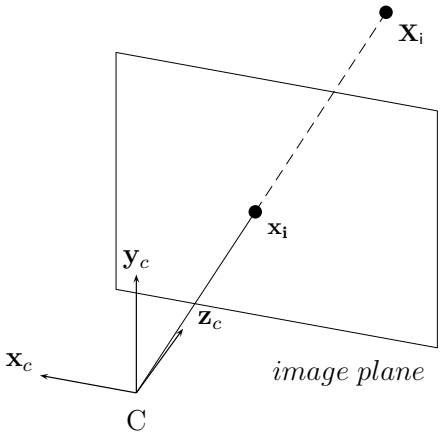


Figure 2.6: Observation of a single point in a pin-hole camera model. The 3D point $\mathbf{X}_i = [X_i\ Y_i\ Z_i]^T$ projects on the image with coordinates $\mathbf{x}_i = \left[\frac{X_i}{Z_i}\ \frac{Y_i}{Z_i}\right]^T$.

computing the position and orientation of the camera relative to a particular object-frame from the information projected on the image (see Section 2.5), and is a central part for defining strategies for Position-Based Visual Servoing, in which this information must be computed prior to determining the control input.

2.3 Interaction model

Here we set out the equations that govern the interaction model (2.23) in the classical IBVS, *eye-in-hand* control of a camera with six degree of freedom in space. The value of the image parameters $\mathbf{s}(\mathbf{m}, \mathbf{a})$ can be obtained from the measurements $\mathbf{m}$ (i.e. the raw pixel coordinates of the image), and the additional information $\mathbf{a}$ is nothing but the intrinsic camera parameters: focal length, pixel ratio, coordinates of the principal point... Typically we express $\mathbf{s}$ as the projection of some some 3D features $\boldsymbol{\Theta}(\mathbf{p})$ on the image plane of the camera, via the projective mapping $\boldsymbol{\Pi} : \mathbb{R}^3 \to \mathbb{R}^2$.

$$\mathbf{s} = \boldsymbol{\Pi}(\boldsymbol{\Theta}(\mathbf{p})), \tag{2.25}$$

where $\mathbf{p}$ represents the spatial configuration of the camera, that is, its position and orientation, relative to a fixed world-frame.

In the following, we will use an idealized *pin-hole* camera model with focal length equal to 1, and assume that the intrinsic camera parameters can be normalized. This formulation is quite general, since any other camera model based on projective geometry (i.e. fish-eye, catadioptric lenses) can be represented by a pin-hole model up to a continuous transformation [MR93]. Let us define the camera reference frame $\mathcal{F}_c$ with its origin at the focal point C and the Z-axis aligned with the focal axis. In the pin-hole camera model, a point with coordinates

$\mathbf{X} = [X\ Y\ Z]^T$ in this frame is projected on the image on a 2D point $\mathbf{x} = \mathbf{\Pi}(\mathbf{X})$ with coordinates (see Fig. 2.6)

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix} \tag{2.26}$$

The coordinates (2.26) can be used as the visual features corresponding to a single feature point $\mathbf{s} = [x\ y]^T$. Computing the interaction model (2.23) requires deriving the variation of $\mathbf{s}$ induced by a motion of the camera. Differentiating (2.26) in time, we obtain

$$\begin{aligned} \dot{x} &= \dot{X}/Z - X\dot{Z}/Z^2 = (\dot{X} - x\dot{Z})/Z \\ \dot{y} &= \dot{Y}/Z - Y\dot{Z}/Z^2 = (\dot{Y} - y\dot{Z})/Z \end{aligned} \tag{2.27}$$

An instantaneous camera velocity is represented by a velocity twist $\boldsymbol{\tau}_c = \begin{bmatrix} \mathbf{v}_c^T & \boldsymbol{\omega}_c^T \end{bmatrix}^T$, a six-dimensional vector containing its linear and angular velocity components: $\mathbf{v} = [v_x\ v_y\ v_z]^T$ and $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]^T$, expressed relative to the camera frame of reference $\mathcal{F}_c$. The vector $\boldsymbol{\tau}$ belongs to $se(3)$, the vector space that is locally tangent to the space of rigid-body transformations (see Section 2.1). Since we assume that the surrounding scene is fixed, we can regard it as a rigid-body moving relative to a "fixed" camera frame with an instantaneous velocity twist $\begin{bmatrix} -\mathbf{v}_c^T & -\boldsymbol{\omega}_c^T \end{bmatrix}^T$ in the frame $\mathcal{F}_c$. The velocity of the 3D point $\mathbf{X}$ is thus

$$\dot{\mathbf{X}} = -\mathbf{v}_c - \boldsymbol{\omega}_c \times \mathbf{X} \tag{2.28}$$

The expressions (2.28) are known as the *motion-field equations* [LHP80]. By inserting (2.28) in (2.27) we get

$$\begin{aligned} \dot{x} &= -v_x/Z + xv_z/Z + xy\omega_x - (1+x^2)\omega_y + y\omega_z, \\ \dot{y} &= -v_y/Z + yv_z/Z + (1+y^2)\omega_x - xy\omega_y - x\omega_z, \end{aligned} \tag{2.29}$$

which can be rearranged into a matrix-vector product to arrive at the **interaction model** (2.23).

Proposition 2.6. *Let $\mathbf{X} = [X\,Y\,Z]^T$ be the coordinates of a 3D point in the camera frame, and $\mathbf{s} = [x\ y]^T$ be the coordinates of its projection on the image. The velocities of $\mathbf{s}$ are related to a velocity twist of the camera $\boldsymbol{\tau}_c$ by*

$$\dot{\mathbf{s}} = \mathbf{L}_\mathbf{x} \boldsymbol{\tau}_c, \tag{2.30}$$

*where $\mathbf{L}_\mathbf{x}$ is the **interaction matrix** for the 3D point $\mathbf{X}$:*

$$\mathbf{L}_\mathbf{x}(\mathbf{s}, Z) = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix}. \tag{2.31}$$

The matrix $\mathbf{L}_\mathbf{x}$ depends not only on the image coordinates $\mathbf{s}$, but also on the depth Z of the point along the focal axis. The value of Z cannot be obtained directly from the camera input, and rather it must be estimated through another method, or measured using a different kind of sensor, such as a laser beam [KMM+96] or an acoustic sensor [LBS18].

Proposition 2.7. *Let* $\mathbf{s} = [s_1 \ \dots \ s_k]^T$ *be a vector of* k *visual features. The interaction matrix* $\mathbf{L}$ *associated to the vector* $\mathbf{s}$ *is obtained by stacking the matrices* $\mathbf{L}_i$ *corresponding to each individual feature:*

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 \\ \vdots \\ \mathbf{L}_k \end{bmatrix} \in \mathbb{R}^{k \times 6} \tag{2.32}$$

For instance, the interaction matrix associated to N points is formed by concatenating the matrices (2.31) for each point. In order to control the six degrees of freedom of the camera, we require at least three 3D points, that is $k \geq 6$. However it is known that, when using only three image points, there are regions of the parameter space where the rank of matrix $\mathbf{L}$ drops. Furthermore, there exist in general four distinct camera poses from which the image of the three points is the same; in other words, there are four different solutions for the pose localisation problem (see Section 2.5). Therefore, at least four points are almost always used in practice.

The image coordinates of points are by no means the only features that can be considered for IBVS. In his PhD thesis [Cha90], F. Chaumette derives the interaction model for many different geometric primitives: line segments, circles, spheres, cylinders..., and his approach can be extended for any object whose projection $f(\mathbf{x}, \mathbf{y})$ is parametrized by a set of image coordinates $(\mathbf{x}, \mathbf{y})$. These features can be combined arbitrarily by concatenating the interaction matrices as in (2.32). For instance, straight lines in space and their 2D counterparts can be described by six-dimensional vectors of Plücker coordinates. We will go back extensively to the Plücker representation of lines in Chapter 4, where we will study the singularities of $\mathbf{L}$ in the observation of lines.

Another type of visual feature of interest for IBVS are image moments [Cha04], which are widely used in image processing and pattern recognition. For an object which projects on the image over a region Ω, the moment m_{ij}, of order $(i + j)$, is defined as

$$m_{ij} = \int \int_{\Omega} x^i y^j \, dx \, dy. \tag{2.33}$$

Image moments are useful as they provide a generic representation of any object that can be segmented in an image, and can be used to derive intuitive geometric information about the observed body, such as the centroid (the geometric centre), or its first and second moments of area, from which the orientation of the object principal axes is obtained. The use of image moments for visual servoing applications is of increasing interest because the feature extraction step, that is, the computation of the moments from the raw pixel information of the camera, is efficient and robust, and not very sensitive to possible distortions of the image (e.g. when the object partially leaves the field of view, or when there are changes in the lighting of the environment). Using visual features based on moments has demonstrated a good performance in domains such as aerial [BMHC06, BMG+09, OC11] or underwater robotics [KPD13, ZZGA21].

2.4 Defining the velocity input

We assume that the camera can be controlled by specifying an input spatial velocity defined by a twist $\mathbf{v}_c$ in the reference frame of the camera. For simplicity, the desired feature values $\mathbf{s}^\star$ are assumed fixed. Then, the error evolves with the visual features as $\dot{\mathbf{e}} = \dot{\mathbf{s}} = \mathbf{L}\boldsymbol{\tau}_c$. If we wished to approximate an exponential decoupled decrease of the error vector ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$), then we could specify a camera velocity

$$\mathbf{v}_c = -\lambda\,\mathbf{L}^+\,(\mathbf{s} - \mathbf{s}^\star) \tag{2.34}$$

resulting in the error evolving like

$$\mathbf{e} = -\lambda\,\mathbf{L}\mathbf{L}^+\,(\mathbf{s} - \mathbf{s}^\star) \tag{2.35}$$

where the control matrix $\mathbf{L}^+$ is a generalized inverse of the interaction matrix $\mathbf{L}$, such as the Moore-Penrose pseudoinverse.

Definition 2.8. *Let $\mathbf{A} \in \mathbb{C}^{m\times n}$ be a rectangular matrix with complex entries. The **Moore-Penrose pseudoinverse** of $\mathbf{A}$ is always defined, and is another matrix $\mathbf{A}^+$ satisfying the following four criteria, where $(\cdot)^\star$ is the complex conjugate of a matrix*

$$\begin{array}{llll} 1) & \mathbf{A}\mathbf{A}^+\mathbf{A} = \mathbf{A}, & 2) & \mathbf{A}^+\mathbf{A}\mathbf{A}^+ = \mathbf{A}^+, \\ 3) & (\mathbf{A}\mathbf{A}^+)^\star = \mathbf{A}\mathbf{A}^+, & 4) & (\mathbf{A}^+\mathbf{A})^\star = \mathbf{A}^+\mathbf{A}. \end{array} \tag{2.36}$$

If $\mathbf{A}$ has real entries and its columns are linearly independent ($\mathrm{rank}(\mathbf{A}) = n$), the Moore-Penrose pseudoinverse is equal to

$$\mathbf{A}^+ = \left(\mathbf{A}^T\,\mathbf{A}\right)\,\mathbf{A}^T \tag{2.37}$$

Using the pseudoinverse as the control matrix satisfies that the product $\mathbf{L}\mathbf{L}^+ \succeq 0$ always, so that the error never increases. However, in general the matrix $\mathbf{L} = \mathbf{L}(\mathbf{s}, \mathbf{Z})$ depends not only on the features $\mathbf{s}$, but also on some 3D parameters of the object, such as the depth distribution $\mathbf{Z} = [Z_1\ \ldots\ Z_N]^T$ of the 3D points, that are not readily available from the camera image. Therefore we cannot use $\mathbf{L}$ directly for the computation of $\mathbf{L}^+$, and the control law used in practice is

$$\mathbf{v}_c = -\lambda\,\widehat{\mathbf{L}^+}\,(\mathbf{s} - \mathbf{s}^\star) \tag{2.38}$$

where $\widehat{\mathbf{L}}$ is an approximation of the interaction matrix, and $\widehat{\mathbf{L}^+}$ the corresponding pseudoinverse.

Some strategies commonly used to compute the control matrix are [CH06, ECR92]:

1. Estimating the current interaction matrix at each iteration of the control law: $\widehat{\mathbf{L}^+} = \mathbf{L}^+(\mathbf{s}, \widehat{\mathbf{Z}})$; in this case we rely on additional sensor information, or otherwise on an algorithmic or heuristic approximation $\widehat{\mathbf{Z}}$ of the depth distribution $\mathbf{Z}$.

2. Evaluating the interaction matrix at the final desired configuration $\widehat{\mathbf{L}^+} = \mathbf{L}|^+_{\mathbf{s}^\star}$; this choice is a sufficiently good approximation in a vicinity of the target $\mathbf{s}^\star$, with the additional advantage that the control matrix need not be recomputed at each timestep.

3. Using a weighted average of the last two, such as $\widehat{\mathbf{L}^+} = \frac{1}{2}(\mathbf{L}^+ + \mathbf{L}^+|_{\mathbf{s}^\star})$ which is a choice that shows good stability and convergence properties in practice [Mal04].

2.5 Relation to pose estimation

A problem dual to vision-based control, with many applications in augmented reality and robotics [MUS15], is the so-called **3D localisation problem**, also known as **pose** (or viewpoint) **determination**. It consists of computing the parameters that represent the position and the orientation of a camera, relative to a particular world or object frame, from the correspondences between a set of 3D features and their 2D projections on the camera image.

When the image measurements used are n image points matched with their 3D counterparts, the problem is known as the **Perspective-From-n-Points** (**PnP**) problem, and is a classical problem in computer vision [HCLL89]. There are analytical solutions for the cases $n = 3$ [RBPD81] (for P3P there are in general up to four distinct solutions) and $n = 4$ [HCLL89]. For $n \geq 6$ non-degenerate points, the solution is always unique, but the system is overdetermined and is typically solved using a least-squares approach. Efficient algorithms for solving the PnP problem for an arbitrary number of points have existed for a long time [L$^+$91, DD95, KLS14]. By analogy with PnP, when the features used are straight lines, matched with their 2D projections, we refer to the **Perspective-From-n-Lines** (**PnL**) problem [DRLR89, XZCK16, WXC20]. Other kind of simple geometric primitives have also been considered in the past [MC02], such as cylindrical objects, conics, contours...

The general approach to solving the localisation problem consists of minimizing the reprojection error

$$\widehat{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{n} (\mathbf{m} - \mathbf{\Pi}(\mathbf{\Theta}(\mathbf{p}))) \tag{2.39}$$

where $\mathbf{p}$ is the vector of parameters representing the pose of the observer, $\mathbf{m}$ is the vector of image measurements, and $\mathbf{\Theta}$ is the set of parameters representing the 3D features in the camera frame, which are projected onto the image plane through the projective mapping $\mathbf{\Pi}$.

The pose estimation algorithms can be classified in *direct* or *iterative* methods. Direct methods, such as [L$^+$91, LXX12] for the PnP problem, or [ZXLK12, XZCK16] for PnL, usually rely on a linearization of the polynomial equations involved to solve the optimization problem (2.39). Iterative approaches, such as the classical Newton descent or Levenberg-Marquadt, try to recursively improve the estimation of the pose parameters $\mathbf{p}$, and are typically faster and more accurate than direct ones, but they rely on a sufficiently good initial estimation of the camera pose to converge. They involve the Jacobian matrix $\frac{\partial \mathbf{m}}{\partial \mathbf{p}}$ of the image measurements with respect to the pose parameters, sometimes denoted **image Jacobian**. If the pose $\mathbf{p}$ is an element of $SE(3)$, the group of rigid-body transformations, and the differential of $\mathbf{p}$ is expressed in its tangent space $se(3)$, then the image Jacobian is equivalent to the well-known interaction matrix from IBVS [MC02].

An iterative method worth mentioning here is **Virtual Visual Servoing** (**VVS**), developed in [MC02] for applications in augmented reality, where visual servoing is used to control the motion of a *virtual* camera such that the final image matches the image measurements of the real camera. The formulation of VVS is equivalent to that of vision-based control: the error function to minimize is

$$\mathbf{e}(\widehat{\mathbf{p}}) = (\mathbf{s}(\widehat{\mathbf{p}}) - \mathbf{s}^\star) \tag{2.40}$$

where, in this case, $\mathbf{s}^\star$ is the vector of image measurements, which is assumed fixed, and $\widehat{\mathbf{s}} = \mathbf{s}(\widehat{\mathbf{p}})$ are the virtual visual features, which depend on the current estimation $\widehat{\mathbf{p}}$ of the camera pose. The advantage of VVS over other pose determination algorithms is that it can be adapted to integrate any kind of image feature, or combinations of them, for which the interaction model (2.23) can be computed. However, VVS typically requires that the initial guess $\widehat{\mathbf{p}}_0$ is not too far away from the true pose, and is susceptible to the same issues as classical IBVS, namely, the appearance of singularities of the interaction matrix, and the existence of multiple local minima.

In particular, it is known that the singularities of the interaction matrix may lead to losses in the accuracy, not only for Virtual Visual Servoing, but for general pose computation methods, even for direct algorithms that do not explicitly involve the Jacobian matrix $\frac{\partial \mathbf{m}}{\partial \mathbf{p}}$ [PENB+21, GFNBSED22]. However, most of the existent literature is concerned with computing the solutions of the 3D localisation, and improving the numerical accuracy and efficiency of the algorithms, while the study of the failure cases, such as the singularities, or the existence of multiple solutions, has received little attention [MR93]. The issues concerning the performances of pose computation algorithms and vision-based control, and the previous research that has been done to study them, are treated in the next section.

2.6 Issues in Visual Servoing

The performance of Image-Based Visual Servo control is hindered by two main classes of problems:

1. The singularities of the interaction model of the 2D features, which lead to issues in controllability and stability of IBVS [HHC96], and affect the accuracy of 3D pose localisation algorithms [PENB+21].
2. The existence of local minima of the closed-loop system [PENB+21], as a consequence of which the global asymptotic convergence of IBVS towards the desired configuration is not guaranteed [CH06].

2.6.1 Singularities of the interaction matrix

In order to perform a visual servo task with n degrees of freedom, at least $k \geq n$ independent visual features need to be considered, to construct an interaction matrix $\mathbf{L} \in \mathbb{R}^{k \times n}$ with $\text{rank}(\mathbf{L}) = n$. However, it is possible that, for specific configurations between the camera and the observed object, the interaction matrix becomes locally rank-deficient. In this case, the robot gains an uncontrollable degree of freedom and may become instantaneously unstable. Specifically, when $\text{rank}(\mathbf{L}) < n$, then $\ker(\mathbf{L})$ has dimension at least one; and there exists a non-null velocity twist $\boldsymbol{\sigma} \in se(3)$ in this nullspace for which the value of the image parameters does not change

$$\boldsymbol{\sigma} \in \ker(\mathbf{L}) \quad \Leftrightarrow \quad \dot{\mathbf{s}} = \mathbf{L}\,\boldsymbol{\sigma} = \mathbf{0}. \tag{2.41}$$

As a consequence, the velocity control input (2.38) can become infinitely high in magnitude, leading to a loss in controllability. Furthermore, singularities of the interaction matrix are also singularities in the kinematic maps involved in the pose determination algorithms from 3D-2D correspondences, and can also lead to losses in the accuracy and stability of these methods. In particular, it is known that, at a singularity, the number of solutions (the number of different camera poses associated to the current image) of the 3D pose localisation problem may change, as shown for P3P in [Rie14] and [ZH06].

A common strategy to avoid the appearance of singularities is to use a larger number of image features than the degrees of freedom of the task $k > n$; then, the system becomes overconstrained, and it is less likely that the interaction matrix suffers a loss of rank. However, as additional features are considered, another issue occurs: the appearance of local minima; that is, configurations where the camera has converged to a non-zero error; this is explored in the following section. Furthermore, even introducing redundancies, singularities cannot be ruled out without a rigorous mathematical analysis. It is thus crucial, for evaluating the performances of IBVS controllers and pose computation algorithms, to identify the singularity conditions and how to avoid them.

The study of the singularities of IBVS has been hindered by the complexity of the algebraic systems that determine when the interaction matrix becomes rank-deficient. These are polynomial equations in many variables and of very high degrees, that arise from the determinant of $\mathbf{L}$, if it is square, or of its maximal submatrices, if $k > n$. Even with the increasing capacity of processors and with state-of-the-art software for polynomial system solving, computing the solutions of these systems and providing a geometrical interpretation is both a computational and theoretical challenge.

A classical result [Tho66] is the determination of the singularity conditions of the Perspective-From-3-Points (P3P) problem; that is, the singularities in the interaction model related to three image points. In this case, the interaction matrix becomes rank-deficient if the three 3D points are aligned, or if the camera centre lies in the cylinder that contains the three points and is perpendicular to the plane that they define (see Fig. 2.7). This result has been revisited several times using different approaches [MR93, Pap95, FTC11], and most recently in [BCM16] by using a concept labeled the **hidden robot**.

The hidden robot was introduced in [BM13], originally as a tool to evaluate the singularities of a class of vision-based controllers used to improve the accuracy of parallel robots, by observing the configurations of the robot legs, and later was generalized to the study of the singularities of larger classes of controllers, including IBVS, where it was used in [PENB+21] and [BMC16] to compute for the first time the singularities in the P4P (Perspective-From-4-Points) and P3L (Perspective-From-3-Lines) problems. The approach consists of constructing a **virtual parallel mechanism** whose links join the camera centre to each of the 3D features, and apply the same geometric constraints on the position of the camera.

The hidden robot is thus a **visual representation of the geometric mapping** between the observation space and the Cartesian space (see Fig. 2.8), and shares the same **geometric** and **kinematic properties** as the pose localisation problem [BM13], i.e.

1. The solutions of the Forward Geometric Model (FGM) of the hidden robot are also solutions of the 3D localisation problem of the perspective camera.

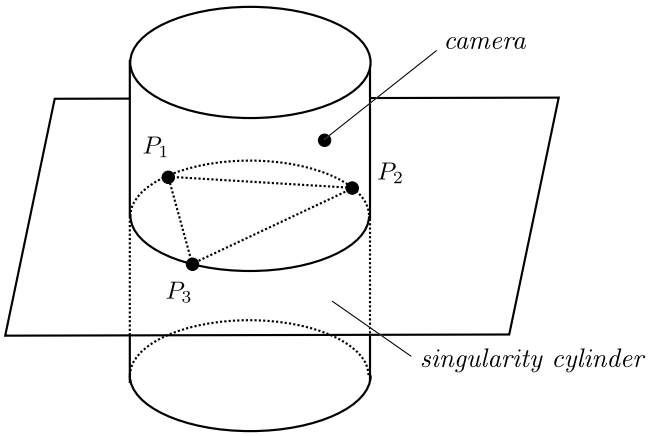


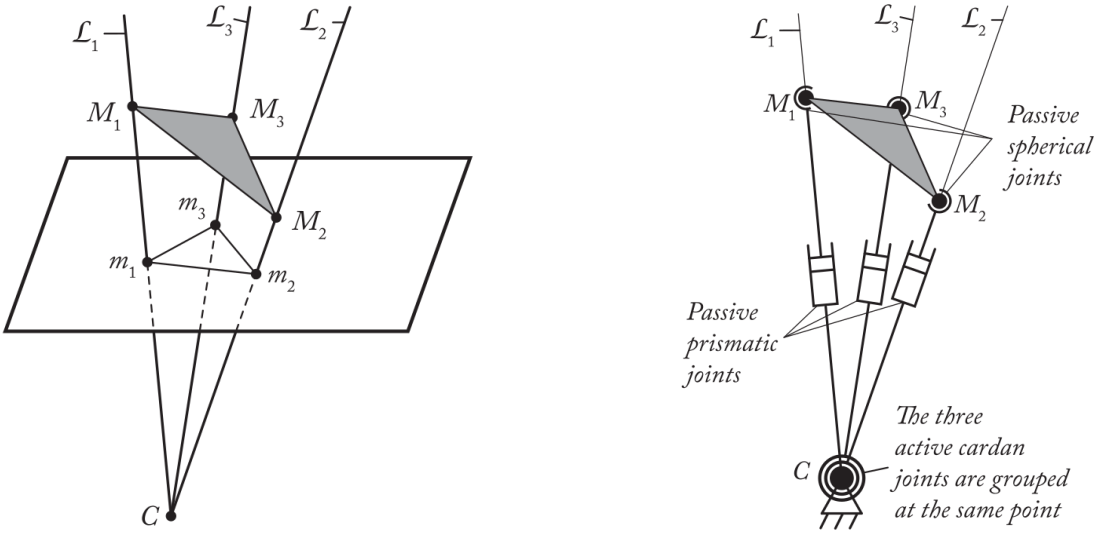
Figure 2.7: A singularity of the P3P problem occurs if the camera centre C lies on the cylinder that contains the three points and is perpendicular to their plane.

2. The singularities of the inverse kinematic Jacobian matrix of the hidden robot (also called *Type II* or *parallel* singularities) are the same as the singularities of the interaction matrix.

As a result, the singularity conditions of the interaction matrix can be studied from the singularities of the inverse kinematic Jacobian $\mathbf{J}_{inv}$ of the virtual robot.

It is well-known in Robotics that the rows of the inverse Jacobian of a parallel robot represent the system of wrenches applied by each of the kinematic chains on the robot's platform and constraining its motion [Mer05]. In screw theory, a wrench is a six-dimensional vector used to represent a torque and a force through a point applied to a body. At a singularity of the inverse Jacobian, the robot gains an uncontrollable motion; that is, it is not fully constrained by the system of wrenches acting on it. As a consequence, the singularities are the configurations for which the system of wrenches defined by the rows of the inverse Jacobian become degenerate. Therefore, the system of wrenches that acts on the platform of the hidden robot constitutes a basis for the row vectors of the interaction matrix, in the sense that their conditions of degeneracy are equivalent. The degeneracy of a system of wrenches is a familiar topic in Robotics and Mechanical Engineering, and there exist a variety of tools to compute them and give a geometric characterization, such as Grassmann geometry or the Grassmann-Cayley Algebra (GCA), which provide a coordinate-free system used to describe the incidence of certain geometric elements.

For the P3P problem, the authors of [BCM16] propose a hidden robot with a 3-$\underline{\text{U}}$PS architecture (Fig. 2.8); that is, a three-legged parallel robot where each kinematic chain is formed by a universal ($\underline{\text{U}}$) joint, followed by a prismatic (P), and a spherical (S) joint. The three legs are attached by the $\underline{\text{U}}$-joint to the camera centre C, and by the S-joint to each of the observed 3D points. The underlining of the letter $\underline{\text{U}}$ indicates that the universal joint is actuated, and the others are passive. To derive this mechanism, they show that the passive motions allowed by each robotic leg are the same as those allowed by the geometric constraints



(a) Observation of three points in space. From each image point m_i, the location of the 3D point M_i is only known up to the line $\mathcal{L}_i$.

(b) Hidden robot model: a 3-UPS architecture with all the active cardan (U) joints attached to the camera centre C, and the spherical (S) joints connected to each 3D point.

Figure 2.8: The hidden robot model is a representation of the kinematic mapping involved in the interaction model related to a set of visual features. The figures are taken from the analysis of the singularities of P3P in [BCM16].

of each image point on the motion of the camera, and that there exists a global diffeomorphism between the observation features $\mathbf{s}$ and the active joint variables.

By examining the system of wrenches that constrain the platform of the 3-UPS robot, and using existent results from Grassmann-Cayley Algebra [BHS06, KWCC09], they derive the well-known singularity conditions of the P3P problem. This result was extended in [PENB+21] to the general PnP with $n > 3$. For this case, the interaction matrix (or the corresponding Jacobian matrix of the hidden robot) is not square, and the conditions of singularity are determined by a system of equations arising from the vanishing of its maximal minors. The authors use Gröbner bases computations (see Chapter 3.2) to show that, when using at least four non-degenerate points, there can only be a finite number of camera configurations that result in a singularity of the interaction matrix.

The hidden robot approach was also used in [BMC16] to compute the singularity conditions of the P3L problem. In this case, the singularities occur when the camera centre is contained in a region defined by the union of two surfaces, described respectively by polynomials of degree 2 and 3, and which depend on the relative configurations of the 3D lines. We review this particular result in Section 4.3 of this work.

Summary

The rank-deficiencies of the interaction matrix $\mathbf{L}$ affect both the stability of IBVS and the accuracy of pose determination algorithms. The singularity conditions are described by large systems of parametric polynomial equations in many variables, computationally very expensive to solve.
Recently, a concept labeled the *hidden robot* [BCM16, BMC16] was used to simplify the equations in the Perspective-n-Points and the Perspective-n-Lines problems, by relating the solutions to the singularities in the kinematic mapping of a virtual robot architecture.

Previous works

- 3 *image points*: A singularity when the points are aligned or when the camera lies on a cylinder containing the points [Tho66, MR93].
- $n > 3$ *coplanar points*: A singularity when all the points and the camera centre lie in the same circle [BCM16].
- $n > 3$ *non-coplanar points*: A finite number of singular camera locations [PENB+21].
- 3 *image lines*: A singularity when the camera is on a cuadric or on a cubic surface defined by the 3D lines [BMC16].

Open problems

- $n > 3$ image lines. In Chapter 4 of this thesis we study the cases $n = 4$ and 5.
- Other 2D features (moments, combinations of different features...). In Appendix A we analyse the singularities related to combinations of point and line features.

2.6.2 Stability analysis of IBVS

Here we review the current state of the art of the stability properties of IBVS controllers specifically; in Section 2.7 we will review some methods, based on Lyapunov's theory, that can be used to assess the global stability of dynamic systems in general.

When the camera velocity input is defined proportional to the error vector $\mathbf{e} = (\mathbf{s} - \mathbf{s}^\star) \in \mathbb{R}^k$ as $\mathbf{v}_c = -\lambda \widehat{\mathbf{L}^+}\, \mathbf{e}$, the error will evolve according to the relation

$$\frac{d\mathbf{e}}{dt} = -\lambda \mathbf{L}\widehat{\mathbf{L}^+}\, \mathbf{e}. \tag{2.42}$$

To evaluate the stability of the system (2.42), let us consider as a candidate Lyapunov function the half norm of the error vector $\mathcal{L} = \frac{1}{2}(\mathbf{e}^T\mathbf{e})$. Then, differentiating $\mathcal{L}$ in time:

$$\dot{\mathcal{L}} = \mathbf{e}^T\dot{\mathbf{e}} = -\lambda\ \mathbf{e}^T\, \mathbf{L}\widehat{\mathbf{L}^+}\, \mathbf{e}. \tag{2.43}$$

The global asymptotic stability of this system around the point $\mathbf{e} = 0$ is guaranteed if the following condition is ensured everywhere in the parameter space except for $\mathbf{e} = 0$:

$$\mathbf{L}\widehat{\mathbf{L}^+} \succ 0. \tag{2.44}$$

If the number of visual features equals the number of degrees of freedom of the robot task ($k = 6$ for a camera free to move and rotate in $\mathbb{R}^3$), and if the interaction matrix $\mathbf{L}$ and the control matrix $\widehat{\mathbf{L}^+}$ are always of full-rank, then the condition (2.44) is ensured if the approximations used to compute $\widehat{\mathbf{L}^+}$ are not too coarse. However, in IBVS, a redundant number of visual features is often used in order to avoid the appearance of singularities of the matrix $\mathbf{L}$. Then, the condition (2.44) can never be ensured, since $\mathbf{L}\widehat{\mathbf{L}}^+ \in \mathbb{R}^{k\times k}$ is of maximum rank 6. For this case, configurations such that $\mathbf{e} \in \ker(\widehat{\mathbf{L}}^+)$ correspond to *fixed points* of the system: points for which the camera velocity (2.38) is null for a non-zero error. If some of these configurations are local minima, there exists a region of the parameter space around each of these where the camera will converge to a stable configuration with a fixed-state error. Such cases are often observed to occur in practice [CM00, CH06, PENB+21].

Determining *a priori* whether an IBVS scheme will converge to the desired configuration or to a local minima is a hard but important problem, particularly because it is possible that these minima are located quite near each other, and near the global minimum. If this is the case, it may not be easy to evaluate whether the camera has converged to the correct position, but we may be in the presence of non-negligible errors. This may be crucial for applications with high-precision requirements, such as industrial or medical robotics.

The local stability of these controllers around the global minimum has been studied by linearizing the equations (2.42) in ([CH06], pages 8-9). In a small neighbourhood around the point $\mathbf{e} = 0$, local asymptotic stability is in general guaranteed if the estimations involved in $\widehat{\mathbf{L}^+}$ are sufficiently good. In a few works, the authors have tried precisely to quantify a margin of how good these estimations should be, but only for some very simple cases, such as the presence of uncertainties in the camera calibration model [Esp94, DJSW02], or for the observation of planar objects where the orientation of the plane is estimated [MMR09].

However, the analysis of the global stability of IBVS methods remains largely an open problem, as does the identification of the local minima of the system. A direct application of Lyapunov's theory to compute the regions of convergence would imply evaluating the eigenvalues of the product $\mathbf{L}\widehat{\mathbf{L}}^+ \in \mathbb{R}^{k\times k}$ throughout the whole parameter space, and is outside of our computational reach. As for the local minima, they are the solutions of large algebraic systems of very high degrees and, to our knowledge, no one has ever been successful in computing them, even for the simplest possible configurations, such as the observation of image points.

In the second part of this thesis (Chapter 5), we develop a strategy and a set of tools to answer the latter question, the computation of the points of equilibrium, and present the first results of their kind: the computation of the local minima of IBVS from the observation of four image points.

Summary

IBVS methods are in general locally asymptotically stable in a region around the desired camera pose $\mathbf{s}^\star$; however due to the existence of other stable equilibria other than the global minimum, the convergence and the global stability properties are not guaranteed.

State of the art

- Local asymptotic stability guaranteed if the approximations in $\widehat{\mathbf{L}^+}$ are not too coarse [CH06].
- Robustness of IBVS with respect to camera calibration errors [Esp94, DJSW02] or to uncertainties in the depth distribution of planar objects [MMR09].

Open problems

- Computing the local minima in the observation of image points (and other features, i.e. image lines, moments, etc.).
- Estimating the size of the region of attraction around the global minimum $\mathbf{s}^\star$.

2.7 Lyapunov Stability

In this section we introduce some basic notions from **Lyapunov's stability theory**, a widely used tool to assess the stability of non-linear dynamical systems [KB60, Sas13]. With its origins in Aleksandr Lyapunov's *second method* (or *direct method*) for stability analysis [Lya92], Lyapunov's stability theory encompasses an array of theorems, and variants that generalize the theory to larger classes of dynamical systems [Bla99]. We summarily present here the concepts that we make use of in this thesis, and refer to [Sas13, SL+91] for a more complete introduction to the non-linear stability of dynamical systems.

Consider a *time-invariant* **dynamical system** (i.e. one whose governing equations do not explicitly depend on time) with state variables $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$, described by a general differential equation of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)), \quad \mathbf{x}_0 = \mathbf{x}(t_0) \tag{2.45}$$

where $t \geq 0$ represents time. A point $\mathbf{x}^\star$ in $\mathbb{R}^n$ is an **equilibrium point** of the system (2.45) if $\mathbf{f}(\mathbf{x}^\star) = \mathbf{0}$.

Definition 2.9. *Suppose that the system (2.45) has an equilibrium point at $\mathbf{x}^\star$, i.e. $\mathbf{f}(\mathbf{x}^\star) = \mathbf{0}$. We say that the point $\mathbf{x}^\star$ is*

- ***Locally stable** if, for every $\epsilon > 0$, there exists $\delta > 0$ such that, if $||\mathbf{x}_0|| < \delta$, then $||\mathbf{x}(t)|| < \epsilon$ for all $t > t_0$. In other words, if trajectories starting close to the equilibrium point remain close to it.*
- ***Locally asymptotically stable** if it is stable and, futhermore, it is **attractive**, i.e. $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^\star$.*

- ***Globally asymptotically stable**, if any trajectory starting anywhere in $\mathcal{X}$ converges to $\mathbf{x}^\star$, i.e. $\lim_{t\to\infty}\mathbf{x}(t)=\mathbf{x}^\star$ for all $\mathbf{x}_0\in\mathcal{X}$.*

Definition 2.10. *If the equilibrium point $\mathbf{x}^\star$ is **locally asymptotically stable** in a subdomain $\mathcal{S}\subset\mathcal{X}$ containing $\mathbf{x}^\star$, then $\mathcal{S}$ is said to be a **region of attraction** (**RoA**) of the system around $\mathbf{x}^\star$.*

The *second method* of stability of Lyapunov allows evaluating the stability of an equilibrium point without the necessity to integrate the differential equation (2.45) along the trajectories of the system. It relies on the construction of a so-called **Lyapunov function**, satisfying certain conditions.

Theorem 2.8. *(**Lyapunov's Stability Theorem**) Let $\mathcal{S}\subset\mathcal{X}$ be a bounded domain containing $\mathbf{x}^\star$. If there exists a continuous differentiable function $V:\mathcal{S}\mapsto\mathbb{R}$ that is **positive-definite** on this domain i.e. $V(\mathbf{x})>0$ for $\mathbf{x}\in\mathcal{S}$, except for $V(\mathbf{x}^\star)=0$, and such that*

$$\dot{V}(\mathbf{x})=\frac{dV(\mathbf{x})}{dt}\leq 0 \quad \textit{for all } \mathbf{x}\in\mathcal{S} \quad \textit{and} \quad \dot{V}(\mathbf{x}^\star)=0, \tag{2.46}$$

*then the equilibrium $\mathbf{x}^\star$ is **locally stable**, and we say that $V(\mathbf{x})$ is a **Lyapunov function** of the system. Further, if $\dot{V}(\mathbf{x})<0$ for all $\mathcal{S}\backslash\{\mathbf{x}^\star\}$, then $\mathbf{x}^\star$ is **asymptotically stable** in the region $\mathcal{S}$.*

Theorem (2.8) is equally valid if the equilibrium point $\mathbf{x}^\star$ is a **local minimum** of the function $V(\mathbf{x})$ but $V(\mathbf{x}^\star)=V_{min}>0$, since one can always define another Lyapunov function $V_2(\mathbf{x})=V(\mathbf{x})-V_{min}$. In can also be extended to include **global stability** if (2.46) holds for the whole domain $\mathcal{X}$ and if we require that the Lyapunov function $V(\mathbf{x})$ is **radially unbounded**, that is $V(\mathbf{x})\to\infty$ when $||\mathbf{x}||\to\infty$.

A stronger version of Lyapunov's stability theorem is **Lasalle's invariance principle** [LL61], that allows making a statement about asymptotic stability while relaxing the requirement of positive definiteness on $-\dot{V}(\mathbf{x})$, and is also applicable to dynamical systems that do not admit a Lyapunov function, for instance, systems that lack a point of equilibrium but have stable orbits. In particular, Lasalle's principle uses the definition of a positively invariant set.

Definition 2.11. *A **positively invariant set** of the system (2.45) is a bounded domain $\mathcal{S}\subset\mathcal{X}$ such that, if $\mathbf{x}_0\in\mathcal{S}$, then $\mathbf{x}(t)\in\mathcal{S}$ for all time $t\geq t_0$.*

A dynamical system may have a positively invariant set even if it does not have any stable equilibrium points. On the other hand, a positively invariant set may contain multiple equilibrium points.

Theorem 2.9. *(**Lasalle's Invariance Principle Theorem**) Let $\mathcal{S}\subset\mathcal{X}$ be a bounded domain and let $V:\mathcal{S}\mapsto\mathbb{R}$ be a continuously differentiable **positive-definite** function on $\mathcal{S}$ satisfying*

$$\dot{V}(\mathbf{x})=\frac{dV(\mathbf{x})}{dt}\leq 0 \quad \textit{for all } \mathbf{x}\in\mathcal{S}. \tag{2.47}$$

We define the points of $\mathcal{S}$ *where* $\dot{V}(\mathbf{x})$ *vanishes as*

$$\mathcal{M} = \mathcal{S} \ \cap \ \{\mathbf{x} : \ \dot{V}(\mathbf{x}) = 0\} \tag{2.48}$$

Then, all trajectories $\mathbf{x}(t)$ *starting in* $\mathcal{S}$ *converge to the* ***largest positively invariant set*** *in* $\mathcal{M}$.

Corollary 2.9.1. *If* $\mathcal{S}$ *contains an equilibrium point* $\mathbf{x}^\star$*, and if* $\mathcal{M}$ *contains only* $\mathbf{x}^\star$*, then the equilibrium point is* ***asymptotically stable*** *in* $\mathcal{S}$.

Corollary (2.9.1) allows establishing the asymptotic stability of a point of equilibrium in a domain $\mathcal{S}$ even if the derivative (2.47) is only negative semi-definite. Another useful corollary is the following.

Corollary 2.9.2. *Let* $V(\mathbf{x})$ *be a Lyapunov function for the system around* $\mathbf{x}^\star$*. The* ***sub-level sets*** *of* $V(\mathbf{x})$

$$\mathcal{V}_c = \{\mathbf{x} \in \mathcal{X} : \ V(\mathbf{x}) \leq c\} \tag{2.49}$$

for some $c > 0$*, are* ***positively invariant sets*** *of the system.*

Lyapunov's stability theory can be applied to the analysis of the global stability behaviour of IBVS methods, in particular with the aim of determining, or otherwise estimating, the **region of attraction around the global minimum**.

Let us express the time-derivative of the Lyapunov function as $\dot{V}(\mathbf{x}) = \nabla_{\mathbf{x}} V \cdot \mathbf{f}(\mathbf{x})$. Then, Theorem (2.9) implies that, if a trajectory $\mathbf{x}(t)$ converges to a point, it must converge to a critical point of $V(\mathbf{x})$, satisfying $\nabla_{\mathbf{x}} V = \mathbf{0}$. Therefore, if given a Lyapunov function $V(\mathbf{x})$ for a region $\mathcal{S} \subseteq \mathcal{X}$, and if $\mathbf{x}^\star$ is the only point of equilibrium inside $\mathcal{S}$ for which $\dot{V}(\mathbf{x}^\star) = 0$, then $\mathcal{S}$ is a region of attraction around $\mathbf{x}^\star$.

For the case of IBVS, we propose to use the *potential-like* function equal to the norm of the error vector $V(\mathbf{s}) = \frac{1}{2}(\mathbf{s} - \mathbf{s}^\star)^T(\mathbf{s} - \mathbf{s}^\star)$ as a candidate Lyapunov function. If we can compute all the critical points $\mathbf{s}_{crit}$ of $V(\mathbf{s})$, and determine a region $\mathcal{S}$ in the parameter space such that $\mathbf{s}^\star$ is the only critical point in $\mathcal{S}$, then convergence towards $\mathbf{s}^\star$ is guaranteed from any initial point inside $\mathcal{S}$.

Corollary (2.9.2) can be useful for obtaining an estimate of such a region $\mathcal{S}$ by comparing the value of the potential $V(\mathbf{s})$ at the different critical points $\mathbf{s}_{crit}$ and studying the sub-level sets of the function $V(\mathbf{s})$. For instance, if we can determine a value c such that there are no critical points with $V(\mathbf{s}_{crit}) \leq c$ other than $\mathbf{s}^\star$ (for which we have $V(\mathbf{s}^\star) = 0$), then the level set (2.49) is a region of attraction around the global minimum $\mathbf{s}^\star$. In this thesis, we addressed the problem of the computation of the critical points of IBVS when using N point features (see Section 5).

Chapter 3

Some notions from Algebraic Geometry

3.1 Polynomial ideals and algebraic varieties

In this section we introduce the basic mathematical structures that we will use when dealing with systems of polynomial equations: **polynomial ideals** and **algebraic varieties**, and some fundamental definitions and properties that will be useful to us. Throughout this section we follow the classical reference by Cox, Little and O'Shea [CLO13]. The proofs for all the fundamental statements presented below can be found there and we will not repeat them here.

Without going into a lot more detail, we define

- A **field** is a set on which the addition ($+$) and multiplication ($\times$) operations are defined, and behave just like they do for the fields of real $\mathbb{R}$, complex $\mathbb{C}$ or rational $\mathbb{Q}$ numbers: they satisfy commutativity and associativity, the inverse of the addition is defined and, for non-zero elements, so is the multiplicative inverse. Here we will use $\mathbb{K}$ to refer to an arbitrary field, although in this work we will always work in one of $\mathbb{R}$, $\mathbb{C}$ or $\mathbb{Q}$.

- A **ring** is a set on which multiplication and addition are defined, but where not all non-zero elements are invertible under multiplication. The set of all polynomials involving some variables $(x_1, \ldots, x_n)$ and with coefficients in a field $\mathbb{K}$ defines the **polynomial ring** $\mathbb{K}[x_1, \ldots, x_n]$.

- A field $\mathbb{K}$ is **algebraically closed** if every non-constant polynomial in the univariate polynomial ring $\mathbb{K}[x]$ has a root in $\mathbb{K}$. For instance, the field of real numbers $\mathbb{R}$ is not algebraically closed (e.g. $f(x) = x^2 + 1 \in \mathbb{R}[x]$ has no real root), but the field of complex numbers $\mathbb{C}$ is. We say that $\mathbb{C}$ is the **algebraic closure** of $\mathbb{R}$: the smallest algebraically closed field that contains $\mathbb{R}$.

- A subset $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$ is an **ideal** if it satisfies the following properties

  1. $0 \in I$.
  2. If $f, g \in I$, then $f + g \in I$.
  3. If $f \in I$ and $h \in k[x_1, \ldots, x_n]$, then $hf \in I$.

Definition 3.1. *Let $f_1, \ldots, f_s \in \mathbb{K}[\mathbf{x}]$ be a set of polynomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$ with coefficients in $\mathbb{K}$. Then, the **ideal** defined by $f_1, \ldots, f_s$ is*

$$\langle f_1, \ldots, f_s \rangle = \left\{ \sum_{i=1}^{s} h_i f_i \mid h_i \in k[x_1, \ldots, x_n] \right\}. \tag{3.1}$$

Definition 3.1 satisfies the properties stated above for an ideal $I \subseteq \mathbb{K}[\mathbf{x}]$. Just like vector subspaces are generated by all the possible linear combinations of a set of vectors with linear

coefficients, ideals are generated by all the possible algebraic combinations (with polynomial coefficients) of their polynomial generators. Note that, if a point $\mathbf{a} \in \mathbb{K}^n$ is a solution of the polynomial system $f_1(\mathbf{x}) = \dots f_s(\mathbf{x}) = 0$, then, for any polynomial $g \in \langle f_1, \dots, f_s \rangle$, then $g(\mathbf{a}) = 0$. This observation shows that the ideal $\langle f_1, \dots, f_s \rangle$ is an algebraic object to study in order to get information about the solutions of the system of equations. Note also that the ideal generated by a constant element is the whole polynomial ring $\langle 1 \rangle = \mathbb{K}[\mathbf{x}]$, since any other element of $\mathbb{K}[\mathbf{x}]$ can be expressed as itself times the identity, and that the smallest possible ideal is $\langle 0 \rangle$, which contains only the zero element. Let us summarize some of the operations that are defined on the sets of ideals.

Proposition 3.1. *Let I, J be two ideals in $R = \mathbb{K}[\mathbf{x}]$. The following operations are defined and produce another ideal in R:*

- *Sum $I + J = \{f + g \mid f \in I, g \in J\}$.*
- *Product of ideals $I \times J = \{fg \mid f \in I, g \in J\}$.*
- *Intersection $I \cap J$.*

Definition 3.2. *An ideal $I \subset \mathbb{K}[\mathbf{x}]$ is said to be **radical** if, for a polynomial $f \in \mathbb{K}[\mathbf{x}]$, and for any $m \in \mathbb{Z}_+$ a positive integer,*

$$f^m \in I \quad \textit{implies} \quad f \in I. \tag{3.2}$$

Definition 3.3. *The **radical** of an ideal I, denoted $\sqrt{I}$, is the set*

$$\sqrt{I} = \{f \mid f^k \in I \text{ for some } k \in \mathbb{Z}_+\} \tag{3.3}$$

Note that $I \subseteq \sqrt{I}$ and that $\sqrt{I}$ is a radical ideal by definition 3.2.

Example 3.1. *The ideal $I = \langle x^2, y^2 \rangle \in \mathbb{R}[\mathbf{x}]$ is not radical, because $x, y \notin I$, but $x, y \in \sqrt{I}$.*

Definition 3.4. *Some other common definitions about ideals with particular properties are*

- *An ideal is $I \subseteq R$ is said to be **proper** if $I \neq R$.*
- *A proper ideal $I \subset R$ is **maximal** if there is no other ideal $J \subset R$ such that $I \subsetneq J \subsetneq R$.*
- *$I \subset R$ is a **prime ideal** if $fg \in I$ implies that either $f \in I$ or $g \in I$.*

In the case of univariate polynomials, every ideal $I \in \mathbb{K}[x]$ is always generated by a single polynomial $I = \langle f \rangle$. Given any other basis $f_1, \dots, f_s$ of I, the polynomial f can be computed as the **greatest common denominator** of the generators $f = \gcd(f_1, \dots, f_s)$. In the multivariate polynomial ring $\mathbb{K}[x_1, \dots, x_n]$, ideals in general are not defined by a single polynomial, but they are always generated by a finite polynomial basis.

Theorem 3.2. ***(Hilbert's Basis Theorem)*** *Every ideal I in $\mathbb{K}[x_1, \dots, x_n]$ is **finitely generated**, that is, there exist a finite set of polynomials $f_1, \dots, f_s$ such that $I = \langle f_1, \dots, f_s \rangle$.*

Ideals can have many different bases: different families of polynomials that generate the same ideal. In Section 3.2 we review a particular kind of basis for an ideal with useful computational properties: Gröbner bases.

Having defined the basic algebraic structure arising from systems of polynomial equations, let us move onto the description of the geometric sets defined by the solutions of such a system.

Definition 3.5. *Let $f_1, \ldots, f_s$ in $\mathbb{K}[x_1, \ldots, x_n]$ be a set of polynomials, and let $\mathbb{F}$ be an algebraic closure of the field $\mathbb{K}$. The **algebraic variety** defined by $f_1, \ldots, f_s$ is the set of points of $\mathbb{F}^n$, where all the polynomials f_i vanish:*

$$\mathbf{V}(f_1, \ldots, f_s) := \{(a_1, \ldots, a_n) \in \mathbb{F}^n \mid f_i(a_1, \ldots, a_n) = 0,\ i = 1, \ldots, s\}. \tag{3.4}$$

The variety $\mathbf{V}(f_1, \ldots, f_s)$ is nothing but the set of solutions in $\mathbb{F}^n$ of the system of equations

$$\begin{aligned} f_1(x_1, \ldots, x_n) &= 0 \\ &\vdots \\ f_s(x_1, \ldots, x_n) &= 0. \end{aligned} \tag{3.5}$$

Just as we can take algebraic combinations of the polynomials (3.5) to find a different family of equations with the same solution set, any other basis that generates the same ideal produces the same variety.

Lemma 3.3. *The variety $\mathbf{V}(f_1, \ldots, f_s)$ depends only on the ideal $I = \langle f_1, \ldots, f_s \rangle$, and not on any particular basis of polynomials. The set $\mathbf{V}(I) = \mathbf{V}(f_1, \ldots, f_s)$ is the variety defined by the ideal I.*

Example 3.2. *The variety defined by $I = \langle xz, yz \rangle$ is the union of the z-axis with the plane defined by $z = 0$: $\mathbf{V}(I) = \{z = 0, x = y = 0\}$. Another basis for the same ideal is $I = \langle (x - y)z, (x + y)z \rangle$, which generates the same variety.*

A variety $\mathbf{V}(f_1, \ldots, f_s) \subseteq \mathbb{F}^n$ can be the empty set $\emptyset$, if the polynomials $f_1, \ldots, f_s$ in $\mathbb{K}[\mathbf{x}]$ do not have any common roots in $\mathbb{F}$. If the field $\mathbb{K}$ is algebraically closed (i.e. $\mathbb{K} = \mathbb{F}$), then there is a powerful result about the ideal generated by $f_1, \ldots, f_s$.

Theorem 3.4. *(**The Weak Nullstellensatz**) Let $\mathbb{K}$ be an algebraically closed field, and $I \in \mathbb{K}[\mathbf{x}]$ be a polynomial ideal. Then*

$$\mathbf{V}(I) = \emptyset \quad \Leftrightarrow \quad I = \mathbb{K}[\mathbf{x}]. \tag{3.6}$$

Given a system of polynomial equations $f_1 = \cdots = f_s = 0$, one can verify if the solution set is empty over the complex numbers $\mathbb{C}$, by checking whether if $1 \in \langle f_1, \ldots, f_s \rangle$, in which case $\langle f_1, \ldots, f_s \rangle = \mathbb{C}[\mathbf{x}]$. This is only true because $\mathbb{C}$ is algebraically closed; that is, every non-constant univariate polynomial $f \in \mathbb{C}[x]$ has a root in $\mathbb{C}$. For example, the polynomial $x^2 + 1$ has no roots over the real numbers $\mathbb{R}$, so $\mathbf{V}_{\mathbb{R}}(x^2 + 1) = \emptyset$, but it does over $\mathbb{C}$, where $\mathbf{V}_{\mathbb{C}}(x^2 + 1) = \{\pm\sqrt{-1}\}$.

Lemma 3.5. *The transformation* $\mathbf{V} : \mathbb{K}[\mathbf{x}] \mapsto \mathbb{F}^n$ *that maps a polynomial ideal* I *to the algebraic variety* $\mathbf{V}(I)$ *is* ***inclusion-reversing****, that is* $I \subset J$ *implies* $\mathbf{V}(J) \subseteq \mathbf{V}(J)$*, but it is* ***not one-to-one****: Different ideals can generate the same variety.*

Example 3.3. *Consider* $\langle x, y\rangle \subset \langle x, y, z\rangle \subseteq \mathbb{K}[x, y, z]$*. The corresponding varieties are* $\mathbf{V}(x, y, z) = (0, 0, 0) \subset \mathbf{V}(x, y) = \{x = y = 0\}$*.*

Example 3.4. *The two ideals* $\langle x^2, y^2\rangle \subsetneq \langle x, y\rangle \subset \mathbb{K}[x, y]$ *define the same algebraic variety, namely the origin:* $\mathbf{V}(x, y) = \mathbf{V}(x^2, y^2) = \{(0, 0)\}$*.*

We have established that a polynomial ideal I defines an algebraic variety $V = \mathbf{V}(I)$, which is the set of common roots of all the polynomials in I. We will now see that a variety V also defines itself a polynomial ideal: the set of all polynomials that vanish on V.

Definition 3.6. *Let* $V = \mathbf{V}(f_1, \ldots, f_n) \in \mathbb{K}^n$ *be an algebraic variety defined by polynomials* $f_i \in \mathbb{K}[\mathbf{x}]$*. The ideal defined by the variety* V *is*

$$\mathbf{I}(V) = \{f \in \mathbb{K}[\mathbf{x}] \mid f(\mathbf{a}) = 0,\ \forall \mathbf{a} \in V\} \tag{3.7}$$

The mapping $\mathbf{I} : \mathbb{K}^n \mapsto \mathbb{K}[\mathbf{x}]$ is also **inclusion-reversing**: $V \subset W$ implies $\mathbf{I}(\mathbf{V}) \subset \mathbf{I}(W)$, and it is **injective** (one-to-one): different algebraic varieties define different ideals. Hilbert's *Strong Nullstellensatz* tells us exactly what is the ideal defined by an algebraic variety.

Theorem 3.6. ***(The Strong Nullstellensatz).*** *Let* $\mathbb{K}$ *be an algebraically closed field and* $I \subseteq \mathbb{K}[\mathbf{x}]$ *a polynomial ideal. Then*

$$\mathbf{I}(\mathbf{V}(I)) = \sqrt{I}. \tag{3.8}$$

Therefore, for $\mathbb{K}$ algebraically closed, if the ideal I is radical ($I = \sqrt{I}$), it equals the ideal generated by its variety: $I = \mathbf{I}(\mathbf{V}(I))$. There is a *one-to-one* correspondence between algebraic varieties and radical ideals.

Example 3.5. *Consider again the ideal* $I = \langle x^2, y^2\rangle \subset \mathbb{K}[x, y]$*. The variety defined by* I *is the origin:* $\mathbf{V}(x^2, y^2) = \{(0, 0)\}$*, and the corresponding ideal is* $\mathbf{I}(\mathbf{V}(I)) = \langle x, y\rangle$*. From Definition (3.1), it is clear that this is equal to the radical ideal* $\sqrt{I}$*.*

In the following box we summarize the fundamental relations between algebraic varieties and the defining polynomial ideals.

Theorem 3.7. *(**The Ideal-Variety correspondence**) Let $\mathbb{K}$ be any field. Then*

- *The maps*

$$\begin{array}{rcl} \textit{algebraic varieties} & \xrightarrow{\mathbf{I}} & \textit{polynomial ideals} \\ \textit{polynomial ideals} & \xrightarrow{\mathbf{V}} & \textit{algebraic varieties} \end{array} \tag{3.9}$$

  *are **inclusion reversing**, i.e.*

$$I_1 \subseteq I_2 \Rightarrow \mathbf{V}(I_1) \supseteq \mathbf{V}(I_2) \quad \textit{and} \quad V_1 \subseteq V_2 \Rightarrow \mathbf{I}(V_1) \supseteq \mathbf{I}(V_2)$$

- *The map $\mathbf{I}$ is always **injective**, so for every variety V:*

$$\mathbf{V}(\mathbf{I}(V)) = V. \tag{3.10}$$

  For every ideal I the map $\mathbf{V}$ satisfies

$$\mathbf{V}(\sqrt{I}) = \mathbf{V}(I). \tag{3.11}$$

- *If $\mathbb{K}$ is **algebraically closed**, and we restrict to **radical** ideals, the maps (3.9) have a **one-to-one** correspondence and are inverses of each other.*

$$\textit{radical ideals} \quad \underset{\mathbf{I}}{\overset{\mathbf{V}}{\rightleftarrows}} \quad \textit{algebraic varieties} \tag{3.12}$$

Next we sum up some of the operations defined on algebraic varieties and some useful definitions.

Lemma 3.8. *(**Sums and Intersections of Varieties**). Let I, J be two ideals in $R = \mathbb{K}[\mathbf{x}]$, and let $V = \mathbf{V}(I)$, $W = \mathbf{V}(J)$ be the varieties they define. The union and intersection of V and W are also algebraic varieties and are defined as:*

$$\begin{aligned} V \cap W &= \mathbf{V}(I + J), \\ V \cup W &= \mathbf{V}(I \times J). \end{aligned} \tag{3.13}$$

Example 3.6. *Let $I = \langle x^2 + y^2 - z \rangle$, and $J = \langle z - 4 \rangle$. The variety $\mathbf{V}(I)$ defines the cylindrical paraboloid with axis along the z coordinate axis, and $\mathbf{V}(J)$ is the plane $z = 4$. The intersection of the two is the circle of radius 2 at plane $z = 4$: $\mathbf{V}(I + J) = \mathbf{V}(x^2 + y^2 - 4, z - 4)$, while their union is $\mathbf{V}(I \times J) = \mathbf{V}((x^2 + y^2 - z)(z - 4))$.*

The **set difference** $V \backslash W$ of two varieties V and W is in general *not* an algebraic variety; that is, it cannot be described as the roots of a system of polynomials. The smallest affine variety that contains it is the **Zariski closure** of $V \backslash W$, which amounts to "patching up the holes" left by removing the points of W from V, and is computed from the **saturation** of the ideals defining V and W.

Definition 3.7. *The **Zariski closure** of a subset $S \subseteq \mathbb{K}^n$, denoted $\overline{S}$, is the smallest algebraic variety containing the set: $S \subseteq \overline{S}$. If V is a variety and $S \subseteq V$ is a subset, then S is said to be **Zariski dense** in V if $V = \overline{S}$.*

Definition 3.8. *(**Ideal Saturation**) If I and J are ideals in $\mathbb{K}[\mathbf{x}]$, the **saturation** of I over J is defined as*

$$I : J^\infty = \{f \in \mathbb{K}[\mathbf{x}] \mid \text{ for all } g \in J, \text{ there is } k \in \mathbb{Z}_+ \text{ such that } fg^k \in I\} \tag{3.14}$$

Lemma 3.9. *(**Set difference**) Let $\mathbb{K}$ be algebraically closed and let $I, J \in \mathbb{K}[\mathbf{x}]$ be ideals. Then*

$$\overline{\mathbf{V}(I) \backslash \mathbf{V}(J)} = \mathbf{V}(I : J^\infty) \tag{3.15}$$

Geometrically, taking the saturation of a I over J amounts to removing the roots of the polynomials in J from the variety $\mathbf{V}(I)$. Taking the Zariski closure guarantees that the set $\overline{\mathbf{V}(I) \backslash \mathbf{V}(J)}$ is an algebraic variety. Expression (3.15) implies that, if any component of $\mathbf{V}(J)$ is contained in $\mathbf{V}(I : J^\infty)$, then it is a set of strictly smaller dimension than $\mathbf{V}(I)$.

Example 3.7. *Consider the variety $V = \mathbf{V}((x^2 + y^2 - z)(z - 4))$ from Example 3.6. This set is formed by the union of a cylindrical paraboloid and the plane $z = 4$. Removing the points of $W = \mathbf{V}(z - 4)$ from V would yield the paraboloid minus the curve of intersection of the two sets: $V \backslash W = \{z = x^2 + y^2\} \backslash \{x^2 + y^2 = 4, z = 4\}$. This is not an algebraic variety, since there are no polynomials that vanish everywhere on this surface but not on the curve. The corresponding ideal saturation:*

$$\langle (x^2 + y^2 - z)(z - 4) \rangle : \langle (z - 4) \rangle^\infty = \langle x^2 + y^2 - z \rangle$$

gives the polynomial defining the hyperboloid. Therefore $\mathbf{V}(x^2 + y^2 - z) = \overline{V \backslash W}$.

By Lemma 3.8, the union of two algebraic varieties is another variety. This opens up the path for expressing algebraic varieties as the union of more fundamental sets.

Definition 3.9. *A variety V is said to be **irreducible** if, whenever it is written as the union of two subsets $V = V_1 \cup V_2$, it means that either $V = V_1$ or $V = V_2$.*

Lemma 3.10. *Every algebraic variety V can be decomposed as the union of its **irreducible components***

$$V = V_1 \cup \ldots \cup V_r \tag{3.16}$$

Lemma 3.11. *When $\mathbb{K}$ is an algebraically closed field, there is a correspondence between **irreducible** varieties and **prime** ideals*

$$V \subseteq \mathbb{K}^n \text{ is an irreducible variety} \iff \mathbf{I}(V) \subseteq \mathbb{K}[\mathbf{x}] \text{ is a prime ideal.} \tag{3.17}$$

Recall from Definition 3.4 that a prime ideal is any ideal I such that if a product of two polynomials $fg \in I$, then either $f \in I$ or $g \in I$.

A fundamental concept in the study of algebraic varieties is that of **dimension**. For our purposes we will use the following definition, although there exist multiple others (see Chapter

9 of [CLO13], and Chapters 8 and 13 of [Eis13]), some of which relate the geometric concept to algebraic properties of the corresponding ideals.

Let $V \subseteq \mathbb{F}^n$ be an affine variety in the space of variables $(x_1, \ldots, x_n)$, with $\mathbb{F}$ being an algebraically closed field, and let $H_r \subseteq \mathbb{F}^n$ be the coordinate subspace of dimension $n-r$ obtained by setting the r first variables $x_1, \ldots, x_r$ to zero. We define the **projection map** onto the space H_r

$$\pi_r : \mathbb{F}^n \mapsto \mathbb{F}^r \tag{3.18}$$

as the map that sends a point $(a_1, \ldots a_n) \in \mathbb{F}^n$ to $(a_{r+1}, \ldots, a_n) \in H_r$. The projection of V onto H_r is denoted $\pi_r(V)$.

Definition 3.10. *(**Dimension**) Let $V \subseteq \mathbb{F}^n$ be an algebraic variety. The **dimension** $\dim(V)$ of V is the largest dimension $d = n - r$ of a coordinate subspace $H_r \subseteq \mathbb{F}^n$ for which the projection of V onto H_r is Zariski dense.*

The notion of dimension of an algebraic variety is analogous to the dimension of a vector space, although more subtle. Typically it coincides with the intuition of dimension as the minimal number of parameters needed to describe the points of a region of the set, although this is not always the case.

The linear space $\mathbb{K}^n$ is obviously of dimension n. If $\mathbb{K}$ is algebraically closed, adding one polynomial constraint will in general decrease the dimension by 1 (e.g. in $\mathbb{R}^3$ a surface defined by one polynomial should be two-dimensional, a curve defined two polynomials should be of dimension one...). However, more complex cases can occur. For instance, a variety may be the union of components of different dimension.

Example 3.8. *The variety $\mathbf{V}(x^2 + y^2 - z)$ defines a 2D surface in $\mathbb{R}^3$. When intersected with the plane $z = 4$, it defines a one dimensional curve $\mathbf{V}(x^2 + y^2 - 4, z - 4)$. However the variety $\mathbf{V}((x^2 + y^2 - z)y, (x^2 + y^2 - z)z)$, although defined by two equations, is the union of the 2D surface with a one-dimensional component, the axis $y = z = 0$.*

Lemma 3.12. *Let $V = V_1 \cup \ldots \cup V_r$ be an algebraic variety. The **dimension** of V is the largest of the dimensions of its irreducible components: $dim\, V = sup_i\ dim\,(V_i)$.*

The dimension can also be understood as a **local** property. For almost every point $\mathbf{p}$ of a variety V (except for the points contained in a strictly smaller subvariety, if any), the local dimension of V at $\mathbf{p}$, denoted $\dim_{\mathbf{p}}(V)$, is equal to the **dimension of the tangent space** of V at $\mathbf{p}$. Without going into further detail, we will simply mention that the points where this is not true are by definition the singular points of V.

Another useful definition is that of the **degree of an algebraic variety**.

Definition 3.11. *(**Degree of an algebraic variety**) Let $V \subseteq \mathbb{F}^n$ be a variety of dimension d in a space of dimension n. The **degree** of V is the number of points of its **intersection** with d **generic linear varieties** (generic hyperplanes) of dimension $n - 1$.*

For a variety $V = \mathbf{V}(f)$ generated by a single, *square-free* polynomial $f \in \mathbb{K}[\mathbf{x}]$, the degree of the variety is simply the degree of f. For a zero-dimensional variety, the degree is the total

number of points of the variety. For more complex cases, the degree can be computed applying Definition 3.11.

The intrinsically geometrical definition of the degree of a variety can be translated into the algebraic setting to define the degree of a polynomial ideal. However the degree of an ideal is a more subtle notion: it is defined in terms of the degree of the *homogenization* of the ideal, and relies on the definition of the *Krull dimension* and the *Hilbert Series*, which we will not delve into. We refer to [Laz21] for more information, and will limit ourselves to defining the degree of an ideal of dimension zero in relation with the degree of its algebraic variety.

Definition 3.12. *Let $\mathbb{K}$ be an algebraically closed field, and let $I = \langle f_1, \ldots, f_s \rangle \subset \mathbb{K}[\mathbf{x}]$ be a **zero-dimensional** polynomial ideal. The **degree** of I is the total number of solutions of $f_1 = \cdots = f_s = 0$ **counted with multiplicity**. If I is a **radical** ideal, then all the solutions have multiplicity 1, and the degree of the ideal equals the number of common roots, i.e. $deg(I) = deg(\mathbf{V}(I))$.*

Next we present the basic notions of what occurs when we eliminate a subset of all the variables from a given set of polynomial equations.

Definition 3.13. *(**Elimination Ideal**) Let $I = \langle f_1, \ldots, f_s \rangle \subseteq R$ be an ideal in $R = \mathbb{K}[x_1, \ldots, x_n]$. The $\ell - th$ **elimination ideal** I_ℓ is the ideal of $\mathbb{K}[x_{\ell+1}, \ldots, x_n]$ defined by*

$$I_\ell = I \cap \mathbb{K}[x_{\ell+1}, \ldots, x_n]. \tag{3.19}$$

The ideal I_ℓ consists of all the algebraic combinations of the polynomial equations $f_1 = \cdots = f_s = 0$ that only involve the variables $x_{\ell+1}$ to x_n. Thus, computing the $\ell - th$ elimination ideal for a system of polynomials amounts to *eliminating* the first ℓ variables from the equations. Note that different orderings of the variables lead to different elimination ideals. The zero-th elimination ideal is defined as $I_0 = I$, and every other elimination ideal forms a descending chain:

$$I \supseteq I_1 \supseteq \ldots \supseteq I_{n-1} \supseteq I_n = \langle 0 \rangle$$

Elimination ideals are related to the projections of algebraic varieties on coordinate subspaces of lower dimension.

Theorem 3.13. *Let $\pi_\ell : \mathbb{C}^n \mapsto \mathbb{C}^{n-\ell}$ be the projection map (3.18) onto the last $n-\ell$ coordinates. For an ideal $I \subseteq \mathbb{C}[\mathbf{x}]$, the elimination ideal I_ℓ is related to the projection $\pi_\ell(V)$ of the variety $V = \mathbf{V}(I)$ as follows*

$$\overline{\pi_\ell(V)} = \mathbf{V}(I_\ell). \tag{3.20}$$

In other words, $\mathbf{V}(I_\ell)$ is the Zariski closure of $\pi_\ell(V)$.

In the next section, we will see that we can compute elimination ideals, and therefore the projections of algebraic varieties, from the Gröbner basis, when using the *lexicographical* monomial ordering.

To conclude, we summarize here some typical problems one faces when working with polynomial ideals, in order of increasing difficulty. Let $I = \langle f_1, \ldots, f_s \rangle \subseteq \mathbb{K}[\mathbf{x}]$ be an ideal.

1. (***Emptiness of the solution set***) Determine if $1 \in I$. This is equivalent to saying that $I = \mathbb{K}$, or to saying that the system $f_1 = \cdots = f_s = 0$ does not have a solution over an algebraic closure of $\mathbb{K}$.

2. (***Ideal membership***) Given a polynomial $f \in \mathbb{K}[\mathbf{x}]$, determine if $f \in I$.

3. (***Dimension of the solution set***) Compute the dimension of the solution set $\mathbf{V}(I)$.

4. (***Compute the solutions***) If $\dim(\mathbf{V}(I)) = 0$, count the number of points of $\mathbf{V}(I)$ and isolate them (over the real or complex numbers).

3.2 Gröbner Bases

In this section we present the fundamental definitions and properties of Gröbner bases, an essential tool in computer algebra for solving problems with polynomial ideals. Gröbner bases can be used, for instance, to determine whether a polynomial is contained in a given ideal, to eliminate a subset of variables from a system of equations, to compute the projection of an algebraic set, or to obtain a rational parametrization of the solutions. We refer to the classical reference [CLO13] for an introduction to Gröbner bases and for the proofs of the theorems stated below.

We start by reviewing some preliminary definitions necessary to introduce the theory of Gröbner bases in Section 3.2.1; then in Section 3.2.2 we give the definition of Gröbner bases and some of their uses and applications. Finally in Section 3.2.3 we give an overview of the current state-of-the-art algorithms for working with polynomial systems of equations based on Gröbner bases.

3.2.1 Preliminaries on Gröbner bases

In the ring of univariate polynomials, the classical *Euclidean algorithm* solves the ideal membership problem straight away. Any ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ is generated by a single polynomial $I = \langle g \rangle$: the *greatest-common-denominator* of any set producing I. Any other polynomial $f \in \mathbb{K}[\mathbf{x}]$ is in the ideal if and only if it is divisible by g, which is verified if the remainder of the Euclidean division of f by g is the zero polynomial.

The division algorithm is not directly extensible to the case of **multivariate** systems of polynomials. One first issue is that the division algorithm in one variable relies on dividing the monomials of the dividend polynomial, by the polynomial divisor, in order of descending degree: the degree imposes an ordering on the terms of f. For monomials containing more than one variable, the degree alone is not sufficient to define a hierarchy over the terms of $\mathbb{K}[\mathbf{x}]$, and one needs to precise some kind of ordering of the terms to perform division by multivariate polynomials. A first ingredient to define Gröbner bases is to define a **monomial ordering**.

Consider the ring $\mathbb{K}[\mathbf{x}]$ of polynomials in n variables $\mathbf{x} = (x_1, \ldots, x_n)$. Any monomial in the ring can be expressed as

$$x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} = \mathbf{x}^{\boldsymbol{\alpha}} \tag{3.21}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}_{\geq 0}^n$ is a vector of exponents, and the degree of the monomial is $|\boldsymbol{\alpha}| = \sum_{i=1}^n \alpha_i$. We can define a ordering of the monomials $\mathbf{x}^{\boldsymbol{\alpha}}$ in terms of an ordering of the

elements of $\mathbb{Z}^n_{\geq 0}$, such that, for $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{Z}^n_{\geq 0}$, $\boldsymbol{\alpha} \succ \boldsymbol{\beta}$ implies $\mathbf{x}^{\boldsymbol{\alpha}} \succ \mathbf{x}^{\boldsymbol{\beta}}$ and viceversa. However, not any such hierarchy is valid in order to define a multivariate division algorithm; a valid monomial ordering is one that satisfies the following properties.

Definition 3.14. *A **monomial ordering** on $\mathbb{K}[x_1, \ldots, x_n]$, denoted "$\succ$", is a relation the elements of $\mathbb{Z}^n_{\geq 0}$, or, equivalently, a relation on the set of monomials $\mathbf{x}^{\boldsymbol{\alpha}}$, where $\boldsymbol{\alpha} \in \mathbb{Z}^n_{\geq 0}$, that satisfies:*

- *"$\succ$" is a **total** and **transitive** ordering on $\mathbb{Z}^n_{\geq 0}$, i.e. for every $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, only one of the following is true*
$$\boldsymbol{\alpha} \succ \boldsymbol{\beta}, \qquad \boldsymbol{\alpha} = \boldsymbol{\beta} \quad or \quad \boldsymbol{\alpha} \prec \boldsymbol{\beta}, \tag{3.22}$$
and for $\boldsymbol{\lambda} \in \mathbb{Z}^n_{\geq 0}$, $\boldsymbol{\alpha} \succ \boldsymbol{\beta}$, $\boldsymbol{\beta} \succ \boldsymbol{\lambda}$ implies $\boldsymbol{\alpha} \succ \boldsymbol{\lambda}$.
- *For $\boldsymbol{\lambda} \in \mathbb{Z}^n_{\geq 0}$, if $\boldsymbol{\alpha} \succ \boldsymbol{\beta}$ then $\boldsymbol{\alpha} + \boldsymbol{\lambda} \succ \boldsymbol{\beta} + \boldsymbol{\lambda}$.*
- *Every non-empty subset of monomials in $\mathbb{K}[\mathbf{x}]$ has a smallest element. The smallest element of the ordering is always 1, corresponding to the vector $\boldsymbol{\alpha} = \mathbf{0}$.*

Definition 3.15. *The following are commonly used monomial orderings in $\mathbb{K}[\mathbf{x}]$. Let $\mathbf{x} = (x_1, \ldots, x_n)$ be the list of variables and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)$, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}^n_{\geq 0}$ be vectors of exponents.*

- ***Lexicographic ordering** $lex(x_1 \succ \ldots \succ x_n)$: We say that $\boldsymbol{\alpha} \succ_{lex} \boldsymbol{\beta}$ if the leftmost non-zero entry of $(\boldsymbol{\alpha} - \boldsymbol{\beta}) \in \mathbb{Z}^n_{\geq 0}$ is positive. In the lex ordering, any monomial containing x_i is always larger than any monomial in the variables $\mathbf{x}' = (x_{i+1}, \ldots, x_n)$.*
Examples: Consider $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{Z}^3_{\geq 0}$ and the $\text{lex}(x \succ y \succ z)$ ordering,

  (i) $(1,0,0) \succ_{lex} (0,1,0) \succ_{lex} (0,0,1)$ *i.e.* $x \succ_{lex} y \succ_{lex} z$.

  (ii) $\boldsymbol{\alpha} = (1,1,0) \succ_{lex} \boldsymbol{\beta} = (0,3,1)$ *i.e.* $\mathbf{x}^{\boldsymbol{\alpha}} = xy \succ_{lex} \mathbf{x}^{\boldsymbol{\beta}} = y^3 z$,
  because $(\boldsymbol{\alpha} - \boldsymbol{\beta}) = (1, -2, -1)$.

  (iii) $\boldsymbol{\alpha} = (2,3,4) \succ_{lex} \boldsymbol{\beta} = (2,3,2)$ *i.e.* $x^2y^3z^4 \succ_{lex} x^2y^3z^2$,
  because $(\boldsymbol{\alpha} - \boldsymbol{\beta}) = (0,0,2)$.

- ***Graded reverse lexicographic ordering** $grevlex(x_1 \succ \ldots \succ x_n)$: We say $\boldsymbol{\alpha} \succ_{grevlex} \boldsymbol{\beta}$ if $|\boldsymbol{\alpha}| > |\boldsymbol{\beta}|$, or if $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}|$ and the rightmost, non-zero entry of $(\boldsymbol{\alpha} - \boldsymbol{\beta}) \in \mathbb{Z}^n_{\geq 0}$ is negative. Monomials in the grevlex ordering are ordered by total degree. For monomials with equal total degree, grevlex prioritizes lowest degree in the last variable(s).*
Examples: Consider $\mathbb{K}[x,y,z]$ and $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{Z}^3_{\geq 0}$,

  (i) $\boldsymbol{\alpha} = (2,1,1) \succ_{grevlex} \boldsymbol{\beta} = (3,0,0)$ *i.e.* $x^2yz \succ_{grevlex} x^3$,
  because $|\boldsymbol{\alpha}| = 4 > |\boldsymbol{\beta}| = 3$.

  (ii) $\boldsymbol{\alpha} = (1,4,1) \succ_{grevlex} \boldsymbol{\beta} = (2,1,3)$ *i.e.* $xy^4z \succ_{grevlex} x^2yz^3$,
  because $(\boldsymbol{\alpha} - \boldsymbol{\beta}) = (-1, 3, -2)$.

- ***Elimination order**: A monomial order is said to eliminate a subset of the variables $\mathbf{x}' \subseteq \mathbf{x}$ if any monomial in the variables $\mathbf{x} \backslash \mathbf{x}'$ is larger than any monomial containing any variables in $\mathbf{x}'$.*

Different choices of monomial orderings have different consequences on the Gröbner bases computations and on the properties of the final basis. For instance, the lexicographical ordering, which is a particular instance of an elimination order, can be used to compute the elimination ideal with respect to subset of the variables. We will discuss this in Section 3.2.3.

Let $f = \sum a_i \mathbf{x}^{\boldsymbol{\alpha}_i} \in \mathbb{K}[\mathbf{x}]$ be a polynomial in variables $\mathbf{x} = (x_1, \ldots, x_n)$, with its terms ordered following an ordering "$\succ$". The following are useful definitions.

Definition 3.16. *Let $\overline{a}\mathbf{x}^{\overline{\boldsymbol{\alpha}}}$ be the largest term of f with a non-zero coefficient $\overline{a}$ with respect to the monomial ordering "$\succ$".*

- *The vector exponent $\overline{\boldsymbol{\alpha}}$ is called the **multidegree** of f. The total degree of f is $deg(f) = |\overline{\boldsymbol{\alpha}}|$.*
- *The coefficient $\overline{a}$ is the **leading coefficient** of f, denoted $LC(f)$.*
- *The **leading monomial** $LM(f)$ of f is $\mathbf{x}^{\overline{\boldsymbol{\alpha}}}$.*
- *The **leading term** $LT(f)$ of f is $\overline{a}\mathbf{x}^{\overline{\boldsymbol{\alpha}}}$: $LT(f) = LC(f)LM(f)$.*

Having specified an ordering "$\succ$" for the terms in $\mathbb{K}[\mathbf{x}]$, division of multivariate polynomials can be performed just like in the classical way. However, this is not enough to solve the *ideal membership problem*. Let $f \in \mathbb{K}[\mathbf{x}]$ be a polynomial and $I \subseteq \mathbb{K}[\mathbf{x}]$ an ideal generated by polynomials $(q_1, \ldots, q_r)$. One could divide f by each of the q_i one by one; if the remainder is the zero polynomial, then it is clear that $f \in I$. However, in general the result of this algorithm is not unique: it depends on the order on which the division is performed by each of the q_i.

Example 3.9. *Let $f = x^4y + x^3y - x^2y + 3\,y - 3 \in \mathbb{R}[x, y]$ and $I = \langle q_1, q_2 \rangle \subseteq \mathbb{R}[x, y]$, with $q_1 = x^3 + 3$ and $q_2 = x^2y - y + x$, and consider the $\mathrm{lex}(x \succ y)$ ordering. Dividing f by (q_1, q_2) in that order, we obtain*

$$f = (xy + y) \cdot q_1 - q_2 - 3xy - y + x - 3$$

where the remainder is non-zero. However, dividing f by (q_2, q_1), we get

$$f = x^2 \cdot q_2 + (y - 1) \cdot q_1$$

thus showing that $f \in \langle q_1, q_2 \rangle$.

The example shows that a multipolynomial division algorithm alone is not enough to solve the ideal membership problem. We will see in the next section that if the polynomial basis $G = (g_1, \ldots, g_s)$ for the ideal I is a Gröbner basis, then the remainder of the polynomial division of any polynomial f by G is unique, and does not depend on the order of the divisors.

Definition 3.17. *A **monomial ideal** $I \subseteq \mathbb{K}[\mathbf{x}]$ is an ideal generated by a set of monomials of $\mathbb{K}[\mathbf{x}]$. By **Dickson's Lemma**, I is generated by a finite monomial basis of $\mathbb{K}[\mathbf{x}]$.*

Let $I = \langle F \rangle \subseteq \mathbb{K}[\mathbf{x}]$ be the ideal generated by a set of polynomials $F = (f_1, \ldots, f_s)$. Given a monomial ordering "$\succ$", we define $LM(F) = (LM(f_1), \ldots, LM(f_s))$ as the set of leading monomials of the polynomials f_i, and $\langle LM(F) \rangle$, the monomial ideal defined by $LM(F)$. Similarly, we define $LM(I)$ as the set of leading monomials of *all* the polynomials in I, and $\langle LM(I) \rangle$ the corresponding ideal. In general we have $\langle LM(I) \rangle \neq \langle LM(F) \rangle$, namely because there may be polynomials, which are consequences of $f_1, \ldots, f_s$, whose leading monomials cannot be generated from the leading monomials of F. Gröbner bases are defined using this condition.

Example 3.10. *Let $F = (f_1, f_2)$ be two polynomials in two variables $f_1 = x^3 + x + 3y$ and $f_2 = x^2y - y$ and let $I = \langle F \rangle$ be the ideal they define, and consider the lex$(x \succ y)$ ordering. The ideal defined by the leading monomials of F is $\langle LM(F) \rangle = \langle x^3, x^2y \rangle$. Now consider*

$$yf_1 - xf_2 = 2xy + 3y^2.$$

This polynomial is in I, so $LM(yf_1 - xf_2) = xy \in \langle LM(I) \rangle$ by definition. However $xy \notin \langle x^3, x^2y \rangle$, so $\langle LM(I) \rangle \neq \langle LM(F) \rangle$.

3.2.2 Properties of Gröbner bases

A Gröbner basis is a particular set of generators for an ideal, which depends on a monomial ordering that must be specified *a priori*, and for which the division of any polynomial by the elements of the basis yields a unique remainder, regardless of the order of the divisors. The formal definition of Gröbner bases is as follows.

Definition 3.18. *(**Gröbner basis**) Given an ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ and a monomial ordering "$\succ$", a set $G_\succ = (g_1, \ldots, g_s)$ of polynomial generators of I (i.e. $I = \langle G_\succ \rangle$) is a **Gröbner basis (GB)** for I if $\langle LM(G_\succ) \rangle = \langle LM(I) \rangle$.*

In other words, a set of polynomials generating I is a Gröbner basis if and only if the **leading term** of any polynomial in I is divisible by the leading term of one of the polynomials in the basis. In Section 3.2.3 we will present some of the algorithms used to compute a Gröbner basis from a given set of polynomials describing I.

Lemma 3.14. *Every ideal $I \subseteq \mathbb{K}[\mathbf{x}]$ has a Gröbner basis. Furthermore, any Gröbner basis for an ideal I is a set of generators of I.*

In general, for a given ideal and a monomial ordering, there can be many sets of generators that constitute a Gröbner basis. However, these are all equivalent up to certain algorithmic reductions.

Proposition 3.15. *A **reduced Gröbner basis** for a polynomial ideal I is a Gröbner basis G such that $LC(g_i) = 1$ for all $g \in G$, and for all $g \in G$, no monomial of g lies in $\langle LT(G \backslash \{g\}) \rangle$. Given an ideal I and a monomial ordering, the reduced Gröbner basis of I is **unique**.*

Proposition 3.16. *(**Normal Form**) Let $I \subseteq \mathbb{K}[\mathbf{x}]$ be an ideal, and let $G_\succ = (g_1, \ldots, g_s)$ be a Gröbner basis for I for some monomial ordering "$\succ$". Then, for any $f \in \mathbb{K}[\mathbf{x}]$, there exist a **unique** polynomial $\overline{f}^{G_\succ} \in \mathbb{K}[\mathbf{x}]$, such that f can be written*

$$f = q_1 \cdot g_1 + \cdots + q_s \cdot g_s + \overline{f}^{G_\succ}. \tag{3.23}$$

*for some $q_1, \ldots, q_s \in \mathbb{K}[\mathbf{x}]$. The polynomial remainder $\overline{f}^{G_\succ}$ is called the **normal form** of f with respect to $G_\succ$, and is also denoted by $NF(f, G, \succ)$.*

The normal form $NF(f, G, \succ)$ is nothing but the remainder of the multivariate polynomial division of f by $g_1, \ldots, g_s$. We say that *reducing* any polynomial f by $G_\succ$ means computing $NF(f, G, \succ)$. When $G_\succ = (g_1, \ldots, g_s)$ is a Gröbner basis for the ordering "$\succ$", Proposition 3.16 states that this remainder is unique, and does not depend on the order in which the division by each of the polynomials g_i is performed. Sometimes, Proposition 3.16 is used as the definition of Gröbner bases, i.e. a basis G of I is a GB if and only if the remainder of the division of *any* polynomial $f \in \mathbb{K}[\mathbf{x}]$ by G is unique.

Note: The **normal form** will in general depend on the chosen monomial order, like the basis $G_\succ$ does. However, if the polynomial f belongs in $\langle G_\succ \rangle$, then the normal form is also zero for any other monomial order. To show this, consider $\overline{f}^{G_\succ} \neq 0$; then, since the division algorithm has terminated, $LM(\overline{f}^{G_\succ}) \notin \langle LM(G_\succ) \rangle = \langle I \rangle$, therefore $\overline{f}^{G_\succ} \notin I$. This fundamental property of Gröbner bases allows us to solve the *Ideal Membership Problem*.

Corollary 3.16.1. *(**Ideal Membership Problem**) Fix a monomial ordering on $\mathbb{K}[\mathbf{x}]$. Let $G = (g_1, \ldots, g_s)$ be a **Gröbner basis** for an ideal $I \subseteq \mathbb{K}[\mathbf{x}]$. Then, for any polynomial $f \in \mathbb{K}[\mathbf{x}]$, we have that $f \in I$ if and only if $NF(f, G) = 0$.*

Therefore, given an ideal $I = \langle f_1, \ldots, f_t \rangle$ and another polynomial $f \in \mathbb{K}[\mathbf{x}]$, if we compute a Gröbner basis G for I, we can determine univocally if $f = 0$ is a consequence of $f_1 = \cdots = f_t = 0$, by dividing f by the elements of G.

We mentioned before that Gröbner bases with respect to some monomial orders can be used to **eliminate** a subset of the variables from a system of polynomials. In particular we have the following theorem.

Theorem 3.17. *(**Elimination theorem**) Let $I \subseteq \mathbb{K}[\mathbf{x}]$ be an ideal and let G be a Gröbner basis for I with respect to lex$(x_1 \succ \ldots \succ x_n)$ ordering. Then, for every $1 \leq \ell \leq n$, the set*

$$G_\ell = G \cap \mathbb{K}[x_{\ell+1}, \ldots, x_n] \tag{3.24}$$

*is a Gröbner basis for the $\ell - th$ **elimination ideal** $I_\ell = I \cap \mathbb{K}[x_{\ell+1}, \ldots, x_n]$.*

Hence, the elements of the basis G that do not involve the first ℓ variables describe the elimination ideal in variables $(x_{\ell+1}, \ldots, x_n)$. Further, they form a Gröbner basis for this ideal. We saw that the $\ell - th$ elimination ideal of I is related to the **projection** $\pi_\ell(V)$ of its algebraic variety $V = \mathbf{V}(I)$ on the coordinate subspace of its last $n - \ell$ coordinates, by $\mathbf{V}(I_\ell) = \overline{\pi_\ell(V)}$; that is, the variety of the ideal I_ℓ is the **Zariski closure** of the projection of V.

In the lex($x_1 \succ \ldots \succ x_n$) order, every monomial containing a variable x_i is larger than any other monomial containing only variables $(x_{i+1}, \ldots, x_n)$. As a consequence the Gröbner basis computed with this order describes not only the first elimination ideal I_1 (all the polynomials in I not involving x_1), but also all the successive elimination ideals $I_2, I_3 \ldots$, if they are not empty. In an analogous way to how Gaussian elimination provides a triangularization of linear systems of equations, Gröbner bases with respect to the *lexicographical* ordering provide a **triangular description** of a polynomial system. If one wished instead to only eliminate a subset $\mathbf{x}' \subseteq \mathbf{x}$ of the variables, then there are other, more efficient, elimination orderings than the lexicographical one, as defined in Definition 3.15.

An approach to solving systems of polynomial equations consists of eliminating some of the variables, computing the solutions of the corresponding elimination ideal I_ℓ, and then extending these solutions to solutions of the full ideal I. However, we said that the variety of the $\ell - th$ elimination ideal $\mathbf{V}(I_\ell)$ is the Zariski closure of the projection $\pi_\ell(\mathbf{V}(I))$. This means that $\pi_\ell(\mathbf{V}(I)) \subseteq \mathbf{V}(I_\ell)$, but that $\mathbf{V}(I_\ell)$ may contain points that are not in $\pi_\ell(\mathbf{V}(I))$. A solution $(a_{l+1}, \ldots, a_n) \in \mathbf{V}(I_\ell)$ is called a *partial solution*; the Extension Theorem tells us under what conditions this solution is the projection of a point $(a_1, \ldots, a_n) \in \mathbf{V}(I)$ or, equivalently, when this partial solution can be extended to a solution of the full ideal I.

Theorem 3.18. *(**The Extension Theorem**) Let $\mathbb{K}$ be an algebraically closed field and let $I = \langle f_1, \ldots, f_s \rangle \subset \mathbb{K}[x_1, \ldots, x_n]$ be an ideal, and I_1 be the first elimination ideal. Let us write the polynomials f_i, $1 \leq i \leq s$ as follows:*

$$f_i = c_i(x_2, \ldots, x_n) \cdot x_1^{N_i} + \text{ terms with degree} < N_i \text{ in } x_1 \tag{3.25}$$

where $N_i \geq 0$ and $c_i \in \mathbb{K}[x_2, \ldots, x_n]$ is a non-zero polynomial that does not involve x_1. Suppose that a partial solution is given by $(a_2, \ldots, a_n) \in \mathbf{V}(I_1)$. Then, if $(a_2, \ldots, a_n) \notin \mathbf{V}(c_1, \ldots, c_s)$, there exists an $a_1 \in K$ such that $(a_1, \ldots, a_n) \in \mathbf{V}(I)$.

That is, we express each f_i as a polynomial in its first variable alone, with coefficients that are polynomials in $(x_2, \ldots, x_n)$. When using an elimination order that eliminates x_1, the term $c_i(x_2, \ldots, x_n) \cdot x_1^{N_i}$ is the leading term of f. The points in $\mathbf{V}(I_1)$ that are outside $\pi_1(\mathbf{V}(I))$ are precisely the points for which all the coefficients c_i vanish.

Theorem 3.18 precises in which cases a partial solution from the first elimination elimination ideal I_1 extends to a full solution of $I = I_0$. However, note that the $\ell - th$ elimination ideal I_ℓ for I is the first elimination ideal for the ideal $I_{\ell-1}$. Therefore the Extension Theorem imposes a condition for extending a partial solution $(a_{\ell+1}, \ldots, a_n) \in \mathbf{V}(I_\ell)$ to a solution $(a_\ell, \ldots, a_n) \in \mathbf{V}(I_{\ell-1})$ in one higher dimension, and eventually to a full solution in $\mathbf{V}(I)$ by going one variable at a time.

We will now highlight the particular cases of **zero-dimensional** ideals and their solutions. For $\mathbb{K}$ an algebraically closed field, we say that $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$ is of dimension zero if its variety $\mathbf{V}(I) \subseteq \mathbb{K}^n$ is **finite** and **non-empty**. Such an ideal has the following property.

Lemma 3.19. *Let $\mathbb{K}$ be an **algebraically closed** field and $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$ an ideal of **dimension zero**. Then, for every x_i, there exists a **univariate** polynomial $p_i \in \mathbb{K}[x_i]$ such that $p_i \in I$.*

Definition 3.19. *(**Shape position**) Let the field $\mathbb{K}$ be algebraically closed. A **radical, zero-dimensional** ideal $I \subseteq \mathbb{K}[x_1, \ldots, x_n]$ is said to be in **shape position** if its **reduced Gröbner basis** $G = (g_1, \ldots, g_s)$ with respect to $\text{lex}(x_1 \succ \ldots \succ x_n)$ has the following form:*

$$\begin{aligned} g_1 &= x_1 + u_1(x_n) \\ g_2 &= x_2 + u_2(x_n) \\ &\vdots \\ g_{n-1} &= x_{n-1} + u_{n-1}(x_n) \\ g_n &= w(x_n) \end{aligned} \tag{3.26}$$

A useful proposition, known as the **Shape lemma**, claims that any radical, zero-dimensional ideal can be brought into shape position by a *generic linear change of variables* [BMMT94].

For a system with a structure like (3.26), one can recover all the solutions by evaluating the roots of a univariate polynomial $w(x_n) = 0$, and then substituting the solution back in all the other polynomials g_i.

The Shape lemma, and the structure of the *lex* basis in Definition 3.19, presume that the input ideal is radical. For non-radical ideals, i.e. when some solutions appear with a multiplicity larger than 1, we can always represent a finite algebraic set using a so-called **zero-dimensional rational parametrization** [BES21, Kro82].

Definition 3.20. *(**Zero-dimensional rational parametrization**) Let $f_1, \ldots, f_s \in \mathbb{K}[x_1, \ldots, x_n]$ be a system of polynomials defining an ideal I of dimension zero. A rational parametrization of the roots of the system is a couple $(\mathcal{P}, \ell)$, where ℓ is a linear form $\ell = \lambda_1 x_1, \ldots, \lambda_n x_n$, with $\lambda_i \in \mathbb{K}$, and $\mathcal{P} = (w, w', v_1, \ldots, v_n) \in \mathbb{K}[t]$ is a set of univariate polynomials in a new variable t, that satisfy*

- *The polynomial w is square-free, and $w' = \frac{dw}{dt}$.*
- $\deg(v_i) < deg(w)$ *for all* $1 \leq i \leq n$.
- $\lambda_1 v_1 + \cdots + \lambda_n v_n = tw' \mod w$.
- *The solution set of $f_1 = \cdots = f_s = 0$ is the same as the set*

$$\left\{ \left(-\frac{v_1(\theta)}{w'(\theta)}, \ldots, -\frac{v_n(\theta)}{w'(\theta)} \right) \mid w(\theta) = 0 \right\} \tag{3.27}$$

Each solution $(a_1, \ldots, a_n)$ of the system of polynomials $f_1 = \cdots = f_s = 0$ is obtained from the evaluation of the vector of rational functions $\left(-\frac{v_1}{w'}, \ldots, -\frac{v_n}{w'}\right)$ at each of the roots of a univariate polynomial $w(t)$, which is called the **elimination polynomial**. The roots of $w(t) = 0$ can be computed with arbitrary precision using root isolation algorithms [RZ04, Tsi16].

So far we have described some of the uses and properties of Gröbner bases, but not a method to compute them. In the next section we present the original algorithm by Buchberger, and briefly review the state-of-the-art methods and software for solving systems of polynomial equations using GBs.

3.2.3 Algorithms and state-of-the-art

The first algorithm to compute a Gröbner basis from a set of generators of an ideal is due to Bruno Buchberger [Buc65, Buc70], who gave a description of such a basis and study the fundamental results of Gröbner bases theory.

Given a basis $F = (f_1, \ldots, f_s)$ for an ideal $I \subseteq \mathbb{K}[\mathbf{x}]$, Buchberger's algorithm proceeds iteratively by constructing new polynomials whose leading terms are *not* generated by the leading terms of F. These new terms are called the **S-polynomials** and are defined as follows. Define a monomial ordering "$\succ$", and let $\text{lcm}(f, g)$ denote the **least-common-multiple** of two polynomials f, g with respect to "$\succ$". The leading monomial $LM(\cdot)$ and leading term $LT(\cdot)$ of a polynomial are defined as in Definition 3.16. Then, the S-polynomial of a pair $f, g \in \mathbb{K}[\mathbf{x}]$ is:

$$S(f, g) = \text{lcm}(LM(f), LM(g)) \left(\frac{f}{LT(f)} - \frac{g}{LT(g)} \right) \tag{3.28}$$

The S-polynomials are specifically constructed to eliminate the leading terms of the generating pair f and g.

Theorem 3.20. ***(Buchberger's Criterion)*** *Let $I \subseteq \mathbb{K}[\mathbf{x}]$ be an ideal and fix a monomial ordering "$\succ$". A basis $G = (g_1, \ldots, g_s)$ of I is a Gröbner basis if and only if, for every pair g_i, g_j, $i \neq j$, the remainder of the division of $S(g_i, g_j)$ by G is zero.*

Unlike Definition 3.18, Buchberger's Criterion provides a constructive definition of a Gröbner basis. Given a set of generators $G = (g_1, \ldots, g_s)$ for an ideal I and a monomial ordering, one can compute all the S-polynomials $S(g_i, g_j)$ for every pair of polynomials in G using (3.28). Trivially, the polynomials $S(g_i, g_j)$ lie in $\langle g_1, \ldots, g_s \rangle$. If, for some pair g_i, g_j, the *reduction* of $S(g_i, g_j)$ on division by G is *not* zero, it is then added to the basis; otherwise it is discarded. If any polynomials are added to the basis, then they need to be considered in the construction of new S-polynomials, until Buchberger's Criterion is met. The algorithm is guaranteed to always finish and to produce a Gröbner basis for the ideal I.

The cost of computing a Gröbner basis can be quite high, and increases rapidly with the number of variables and equations and with the degrees of the input polynomials; namely because the size of the coefficients in the intermediate polynomial computations can grow enormously, even for outputs of moderate size. In particular, Buchberger's algorithm presents two main drawbacks that limit its performance:

1. Many of the S-polynomials reduce to zero, providing no new information about the final basis. These *trivial* reductions can amount to most of the computing time.

2. The pairs used to construct the S-polynomials are chosen freely from the elements of the basis.

More recent results have allowed to improve on these two aspects. Several criteria [Buc79, KB78] have been proposed to discard *a priori* some of the useless polynomial pairs that lead to trivial computations. In 1999, Faugère introduced the F4 algorithm [Fau99], a variant of Buchberger's that reduces the Gröbner basis computations to a linear algebra problem. Instead of selecting a polynomial pair at each step, the algorithm can carry out multiple S-polynomial

reductions at a time by performing Gaussian elimination on a matrix, called a **multiplication matrix**, constructed from the coefficients of the basis elements by indexing the columns by the monomials of the system. Nevertheless, in practice, for large systems, F4 still produces many zero reductions. The F4 algorithm is implemented in polynomial solving libraries such as FGb [Fau10], msolve [BES21] or computer algebra systems like Maple and Magma.

In another algorithm from 2002, called F5 [Fau02], also by Faugère, the goal is to remove beforehand the critical pairs that lead to zero reductions, and this is done by introducing **signatures**, an element which keeps track of all the previous computations that reduce to zero at a small computational overhead. In particular, for generic systems, no zero reductions are performed. Many *signature-based* variants of the F5 algorithm have been proposed since, a comprehensive review of which can be found in [EF17].

We mentioned before that different choices of monomial orderings affect the computation of the Gröbner basis, as well as the final output. In particular, the ***grevlex*** ordering generally yields bases of smallest total degree and coefficient sizes, and for most systems, it is typically the easiest to compute. On the other hand, the ***lexicographical*** ordering provides an explicit triangular description of the polynomial ideal, which conveys more geometrical information about the solutions. A logical strategy to solve a polynomial system is to first compute a *grevlex* basis, and then use a **change of ordering algorithm**, to convert that basis to the *lexicographical* ordering. The FGLM algorithm [FGLM93], named after Faugère, Gianni, Lazard and Mora, does this for **zero-dimensional** ideals.

Throughout this thesis, we rely particularly on implementations of the F4 algorithm, in the Maple computer algebra system, and in the FGb [Fau10] library, when dealing with polynomial systems of positive dimension. For systems with finitely many solutions, we use the `msolve` library [BES21], which is based on an efficient F4 implementation relying on fast linear algebra methods, on the FGLM algorithm for changing the monomial ordering, and on a dedicated univariate real root isolation algorithm. The `msolve` library is currently the state-of-the-art for the computation of the real solutions of multivariate polynomial systems.

Summary

Given an ideal $I \subseteq \mathbb{K}[\mathbf{x}]$, a **Gröbner basis** $G = (g_1, \ldots, g_s)$ is a set of generators of I such that every polynomial $f \in \mathbb{K}[\mathbf{x}]$ can be written as

$$f = h_1 \cdot g_1 + \ldots h_s \cdot g_s \ + \ NF(f, G)$$

with a **unique** remainder $NF(f, G)$, i.e. the **Normal Form** of f w.r.t. G.

Given a Gröbner basis G for an ideal I, we have $\quad f \in \langle I \rangle \ \Leftrightarrow \ NF(f, G) = 0$.

A Gröbner basis depends only on any set of generators of I and a **monomial ordering**.

Monomial orderings

- **Lexicographical (lex)**: Provides a triangular description of the polynomial system.
- **Degree reverse lexicographical (grevlex)**: (Generally) yields lowest total degree and smallest coefficient sizes.
- **Elimination orderings**: Used to eliminate a subset of the variables.

State of the art

- Gröbner bases computations:
  - Buchberger's algorithm [Buc65, Buc70]: Involves the computation of **S-polynomials** from pairs of generators, and the reduction of the S-polynomials by the elements of the basis.
  - F4 [Fau99] Reduces the Gröbner basis computation to Gaussian elimination on a **multiplication matrix**.
  - F5 [Fau02, EF17]: Introduces **signatures** to track previous computations and remove pairs that lead to trivial reductions.
- Change of ordering:
  - FGLM [FGLM93]: Produces a Gröbner basis with **lexicographic** ordering from an input **grevlex** basis.
- Software:
  - Maple, FGb [Fau10]: Gröbner bases computations for general systems.
  - `msolve` [BES21]: Real root computation of zero-dimensional systems.

Part II

Contributions

Chapter 4

Singularity analysis in the observation of lines

Abstract. This chapter deals with Image-Based Visual Servoing and pose estimation from the observation of four and five lines in space. Our main interest is to determine the relative configurations of the camera and the observed lines that lead to issues in the stability of the control scheme and losses in the accuracy of the pose computation due to singularities of the Jacobian matrix that governs the kinematics of the problem. These conditions can be obtained as the real solutions of a system of polynomials arising from the maximal minors of this matrix; therefore we use tools from computational algebraic geometry, in particular Gröbner bases, to compute the configurations that make all of its minors vanish simultaneously. By choosing a suitable basis for the matrix, we revisit previous results for the problem in the case of three lines to show that one of the known singularity conditions corresponds to when the camera centre is on the surface of the hyperboloid of one sheet uniquely defined by the lines. This result is then further exploited to prove that, for $n \geq 3$ lines, a singularity will occur if there exist one or more lines which are transversal to all of the observed lines and when the camera lies on one such line. Therefore, this type of line singularity can be avoided if one can choose a configuration such that no real lines intersect all the observed lines. However, in the case of $n = 4$, it is shown that there can always be up to ten inevitable camera locations for which the interaction matrix is singular. For $n = 5$ lines, we find that there are no singularities in the generic case, although they may appear for specific configurations, e.g. when there is a line transversal to all five lines. The singularities are also characterized for four and five lines bound by orthogonality and parallelism constraints. Furthermore, a visual servoing library is used to conduct some simulated experiments to substantiate the theoretical results. As expected, we observe problems in control in the vicinity of a singularity as well as increased errors in pose estimation.

The work presented in this chapter was done in collaboration with Abhilash Nayak, and was published in the International Journal of Computer Vision [GFNBSED22].

Keywords Pose estimation · Visual servoing · Singularity · PnL

4.1 Introduction

4.1.1 Motivation and problem statement

A standard problem in computer vision, which has many applications in augmented reality [MUS15] and robotics (especially in visual servoing [HHC96]) is the estimation of the pose of a camera based on the features projected in the camera image. When the 2D image is a set of n points that are projections of their 3D counterparts on the image plane, the problem is known as PnP (Perspective-n-Point) and has been dealt with exten-

sively in the literature [GHTC03, HCLL89, KSS11, WH06]. Similarly, when the features observed by the camera are n straight lines, the problem is referred to as PnL (Perspective-n-Line) [DRLR89, XZCK16, WXC20].

In particular, the PnP and PnL problems involve computing the parameters that define the pose of a camera (the six degrees of freedom that determine its position and orientation), which is an element of SE(3), the Special Euclidean group, from the 3D-2D correspondences of the n observed points or lines. By taking the time derivatives of the parameters involved in PnP or PnL, we obtain the so-called motion-field equations [LHP80] that are crucial to visual servoing in robotics. They involve the mapping between the time derivative of the camera pose parameters belonging to se(3), being the Lie algebra of SE(3) (3D vector space of translational and orientational velocities of the camera) and the relative velocities of the projected features on the image plane, through the *image Jacobian* or *interaction matrix* [CH06, CH07, CH08].

The problem of determining the singularities of the interaction matrix is crucial, especially for the following reasons:

- In visual servoing tasks, we face potential accuracy and controllability issues of the robot when the camera is in the vicinity of a singularity [HHC96].
- The singularities are known to considerably worsen the pose reconstruction accuracy [PENB+21]. Moreover, they are are known to influence the number of solutions of the pose localisation problem as shown by [Rie14] and [ZH06] in the case of P3P.

Determining those singularities is nevertheless a computationally (and analytically) hard problem since it requires solving the complex algebraic systems arising due to the loss of rank of the interaction matrix. As a result, the singularity analysis in the past has been limited to simple image features, such as the observation of three points in space (P3P). For this problem, a well-known result is that a singularity occurs if the three points are aligned or if the camera lies on the cylinder that contains the three points and is perpendicular to the plane they define [MR93]. This result and tools from algebraic geometry were recently used by [PENB+21] to show that, in P4P, there are always two to six camera configurations where the corresponding interaction matrix becomes rank-deficient.

In the case of PnL, most of the research has been focused on finding solutions to P3L [DRLR89, XZCK16, WXC20] without looking at the singularity problem, to the best of our knowledge. However, recently, the singularities in P3L were determined using a tool called the *hidden robot* which was introduced in [BM13]. It proved to be efficient in determining the singularities of vision-based controllers applied to parallel robots and broader classes of visual servo controllers [BMR15, RBM13, BM13, BCM16, BMC16]. With this method, it is possible to compute a change of basis for the rows of the interaction matrix, leading to a simplified system of equations. For the problem of visual servoing using three image lines, the *hidden robot* concept was used in [BMC16] to show that the singularities appear when the camera lies on a quadric or a cubic surface.

4.1.2 Main results

In the present chapter we first provide a geometrical insight on the results in [BMC16] for the singularities of the P3L problem. The authors show that the singularity condition can

be expressed as the vanishing of a determinant that factors as the product of a quadratic and a cubic polynomial. Using line geometry, we demonstrate that the surface described by the quadratic polynomial is the hyperboloid of one sheet that is uniquely described by the three observed lines. This hyperboloid is also the ruled surface described by the set of all lines which are transversal to the three spatial lines. We exploit this result to obtain the following proposition which applies to the general PnL problem for the observation of $n \geq 3$ lines:

Proposition 4.1. *Consider the interaction model related to the observation of n lines in space. If there exist one or more straight lines that intersect the n feature lines, then the interaction matrix is singular when the camera focal point lies on any such line. These lines are described by the roots of a single polynomial (for $n = 3$) or a system of polynomials (for $n > 3$) of degree two, each of which describes the surface of a hyperboloid of one-sheet.*

We then present a complete analysis of the singularities in P4L and P5L, for which we make use of the techniques presented in [BMC16] to obtain a new basis for the rows of the interaction matrix that lead to a simplified system of equations. The P4L and P5L problems are similar to the simpler P3L, but they present a significant increase in analytical and computational complexity. While the singularities for P3L are described by the roots of a single polynomial, for $n > 3$ it is a system of polynomials - the maximal minors of the matrix that describes the new basis. We use tools from algebraic geometry and computer algebra, namely decomposition and saturation of ideals, Gröbner bases and elimination theory (refer to Chapter 3), to characterize the solutions of these systems. We show that, in the case of P4L, two types of singularities can occur:

1. When the camera centre lies on a line which is transversal to the four observed lines, a condition which stems from Proposition 4.1. In the case of four lines, the transversal lines are described by the intersection of the four one-sheeted hyperboloids generated by each triplet of lines. In general there can be either 0, 1, or 2 such lines in the real space, depending on the sign of a discriminant, and the four lines are thus said to be in an elliptic, parabolic, or hyperbolic congruence respectively. We make use of Gröbner bases to derive a condition, as an inequality in terms of the parameters, such that these lines are purely complex, and such that the one-dimensional singularities of the system can be avoided.

2. In general there can also be up to 10 isolated singularity point locations for the camera, the solutions of a zero-dimensional system of polynomials of higher degree, which can be easily computed using Gröbner bases.

In the case of $n = 5$ image lines, there exist no singularities in general, except for specific configurations, which consist of the zero-set of a system of polynomials in the space of the parameters. We also make a special focus on the singularities of the P4L and P5L problems when the lines are bound by orthogonality and parallelism constraints; these are configurations which often appear in practical applications in structured environments, e.g. edge tracking, navigation in urban areas, in corridors or any buildings, and are as a consequence commonly used as visual features for robot control tasks.

To support our results, we present a number of numerical simulations of robot control and pose estimation using VISP [MSC05], a C++ library dedicated to visual-servoing applications. The experiments confirm the singularity conditions we have described, and illustrate the undesirable impacts these singularities have on the controllability and stability of IBVS and on the accuracy of pose determination methods.

These results lead to a paper published in the International Journal of Computer Vision [GFNBSED22] (joint work with Abhilash Nayak, Sébastien Briot and Mohab Safey El Din).

In the Appendix A, we also present an analysis of the singularity conditions of the interaction matrix when the observation features are minimal combinations of points and straight lines.

Structure of the chapter. In Section 4.2 we first recall the form of the interaction matrix related to image line features, and describe the computation of a simplified basis for its rows. Section 4.3 revisits the singularities in the P3L [BMC16] problem and puts forth their geometric interpretation. Sections 4.4 and 4.5 give a complete analysis of the singularities in P4L and P5L, respectively, with a focus on the special cases where the observed lines are bound by orthogonality and parallelism constraints. Section 4.6 presents experimental results from simulations based on the singularities computed for P4L and P5L. Finally, Section 4.7 draws conclusions.

4.2 Row basis of the interaction matrix

In this section, we review the interaction model related to the observation of n lines in space, and we describe the derivation of a new basis for the rows of the interaction matrix, that results in a simplified system of equations describing the singularity conditions. This leads to a new, geometrical interpretation of the singularity results obtained in [BMC16] for the P3L problem, and it allows the analysis of the more general PnL problem.

4.2.1 Recalls on the interaction model related to image lines

In what follows, and without loss of generality, we will use the standard pin-hole camera model with focal length equal to 1, and the **z**-axis oriented along the optical axis. However, any other camera model based on projective geometry could be used [MR93].

The set of all possible straight lines in the three dimensional projective space $\mathbb{P}^3$ is a four-dimensional manifold; in other words, a line can be defined using four independent parameters. There are several possible representations for a line: two points in space, a point and a direction vector, the angles formed with the reference axes and its distance from the origin, etc. Here we will use the **Plücker representation**, which is complete and free of representation singularities; that is, it maps all the points on the four dimensional manifold. Further, the Plücker representation will be useful later for computing a new basis of the interaction matrix.

We start by reviewing the geometric description of lines by Plücker coordinates. A 3D line $\mathcal{L}_i$ can be characterized in the camera frame by a six-dimensional Plücker vector $[\mathbf{U}_i^T \ \mathbf{L}_i^T]^T \in \mathbb{R}^6$,

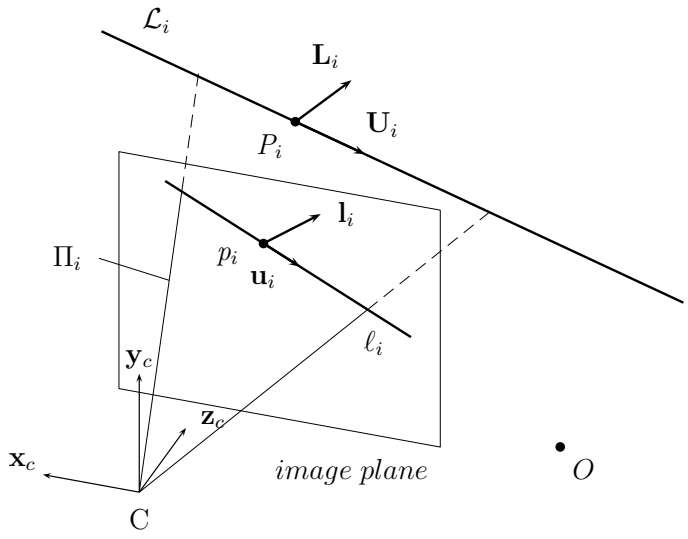


Figure 4.1: Perspective projection of a 3D line.

where $\mathbf{U}_i$ is the direction of the line, and $\mathbf{L}_i$ is defined as the cross product $\mathbf{L}_i = \mathbf{X}_i \times \mathbf{U}_i$, with $\mathbf{X}_i = \overrightarrow{CP_i}$ the position vector from the focal point C to any point P_i on the line. The Plücker coordinates of a line are bound by the constraint

$$\mathbf{U}_i \cdot \mathbf{L}_i = 0 \tag{4.1}$$

In some places in the literature, it is required that the vector $\mathbf{U}_i$ is normalized, i.e. $||\mathbf{U}_i||= 1$. In this work we do not impose this constraint, and instead allow a degree of freedom in the magnitude of $\mathbf{U}_i$, although this does not affect any of the following results.

The line $\mathcal{L}_i$ is projected on the image plane of the camera on a 2D line ℓ_i (see Fig. 4.1) with Plücker coordinates $[\mathbf{u}_i{}^T \ \mathbf{l}_i{}^T]$, where $\mathbf{u}_i$ is the image line direction, and $\mathbf{l}_i = \mathbf{x}_i \times \mathbf{u}_i$ for any point on ℓ_i with position vector $\mathbf{x}_i$. The coordinates of $\mathcal{L}_i$ and of its projection are related by the perspective equations [Cha90]:

$$\mathbf{u}_i = \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{zi} \end{bmatrix} = \begin{bmatrix} L_{yi}/\Delta_i \\ -L_{xi}/\Delta_i \\ 0 \end{bmatrix}; \quad \mathbf{l}_i = \begin{bmatrix} l_{xi} \\ l_{yi} \\ l_{zi} \end{bmatrix} = \begin{bmatrix} L_{xi}/\Delta_i \\ L_{yi}/\Delta_i \\ L_{zi}/\Delta_i \end{bmatrix} \tag{4.2}$$

where $\mathbf{L}_i = [L_{xi} \ L_{yi} \ L_{zi}]^T$ and $\Delta_i = \sqrt{L_{xi}^2 + L_{yi}^2}$ is a depth factor. The image line ℓ_i is fully determined from the three coordinates l_{xi}, l_{yi} and l_{zi}, so it suffices to use $\mathbf{l}_i$ as the vector of features for the line in the visual servo scheme.

The **interaction model** (2.23) for the line $\mathcal{L}_i$ relates the velocities of the coordinates $\mathbf{l}_i$ on the image to the spatial velocity of the camera frame. A relative camera-object velocity is represented by a **velocity twist** $\boldsymbol{\tau}_c$, a six-dimensional vector composed of its linear and

angular velocity components, $\mathbf{v}_c$ and $\boldsymbol{\omega}_c$, expressed in the camera reference frame:

$$\boldsymbol{\tau}_c = \begin{bmatrix} \mathbf{v}_c \\ \boldsymbol{\omega}_c \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4.3}$$

The camera twist is an element of $se(3)$, the Lie algebra of $SE(3)$, the group of rigid-body transformations. In other words, it is a vector tangent to the manifold describing the pose of the camera frame (see Chapter 2, Section 2.1.3).

Differentiating the vector $\mathbf{l}_i$ in (4.2) with respect to time we obtain:

$$\frac{d\mathbf{l}_i}{dt} = \frac{1}{\Delta_i^3} \begin{bmatrix} L_{yi}^2 & -L_{xi}L_{yi} & 0 \\ -L_{xi}L_{yi} & L_{xi}^2 & 0 \\ -L_{xi}L_{zi} & -L_{yi}L_{zi} & \Delta_i^2 \end{bmatrix} \dot{\mathbf{L}}_i. \tag{4.4}$$

Meanwhile, the variation of the 3D line coordinates $\mathbf{L}_i$ is given by $\dot{\mathbf{L}}_i = \dot{\mathbf{X}}_i \times \mathbf{U}_i + \mathbf{X}_i \times \dot{\mathbf{U}}_i$. Using the *motion-field equations* (2.28) and $\dot{\mathbf{U}}_i = -\boldsymbol{\omega}_c \times \mathbf{U}_i$, we get [Cha90, RE87]

$$\dot{\mathbf{L}}_i = -\mathbf{v}_c \times \mathbf{U}_i - \boldsymbol{\omega}_c \times \mathbf{L}_i. \tag{4.5}$$

Finally, inserting (4.5) in (4.4) we can express the interaction model for the features $\mathbf{l}_i$ as

$$\frac{d\mathbf{l}_i}{dt} = \mathbf{M}_i \boldsymbol{\tau}_c, \tag{4.6}$$

where $\mathbf{M}_i$ is the **interaction matrix** corresponding to a single image line [Cha90]:

$$\mathbf{M}_i = \begin{bmatrix} -\frac{l_{xi}l_{yi}U_{zi}}{\Delta_i} & -\frac{l_{yi}^2U_{zi}}{\Delta_i} & \frac{l_{yi}(l_{xi}U_{xi}+l_{yi}U_{yi})}{\Delta_i} & -l_{xi}l_{yi}l_{zi} & -l_{yi}^2l_{zi} & l_{yi} \\ \frac{l_{xi}^2U_{zi}}{\Delta_i} & \frac{l_{xi}l_{yi}U_{zi}}{\Delta_i} & -\frac{l_{xi}(l_{xi}U_{xi}+l_{yi}U_{yi})}{\Delta_i} & l_{xi}^2l_{zi} & l_{xi}l_{yi}l_{zi} & -l_{xi} \\ -\frac{(U_{yi}+l_{yi}l_{zi}U_{zi})}{\Delta_i} & \frac{(U_{xi}+l_{xi}l_{zi}U_{zi})}{\Delta_i} & -\frac{l_{zi}(l_{xi}U_{yi}-l_{yi}U_{xi})}{\Delta_i} & -l_{yi}(l_{zi}^2+1) & l_{xi}(l_{zi}^2+1) & 0 \end{bmatrix} \tag{4.7}$$

with $\mathbf{U}_i = [U_{xi}\ U_{yi}\ U_{zi}]^T$, and with $\mathbf{l}_i$, Δ_i as defined above.

The matrix $\mathbf{M}_i$ has dimension (3×6), but maximum rank 2, because the first and second rows are related by $l_{yi}\,\mathbf{M}_{i1} + l_{xi}\,\mathbf{M}_{i2} = 0$. Hence, we can control at maximum two degrees of freedom of the camera with each image line, and at least three lines are necessary to fully constrain the system [AEH02].

When $n > 1$ lines are considered, the interaction matrix $\mathbf{M}_{(n)}$, relating a change in the full vector of features $\dot{\mathbf{s}} = [\dot{\mathbf{l}}_1^T, \ldots, \dot{\mathbf{l}}_n^T]^T$ to the camera velocity, is obtained by stacking the three rows of $\mathbf{M}_i$ corresponding to each line $\mathcal{L}_i$:

$$\mathbf{M}_{(n)} = [\mathbf{M}_1^T,\ \mathbf{M}_2^T, \ldots, \mathbf{M}_n^T]^T. \tag{4.8}$$

4.2.2 Revisiting the interaction matrix as a system of Plücker lines

In [BMC16], Briot, Martinet and Chaumette describe how to compute a new basis for the space spanned by the rows of the interaction matrix (4.7) by interpreting them as vectors of Plücker coordinates spanning a system of lines. They rely on a mechanical engineering approach to construct a *hidden robot* (see Chapter 2, Section 2.6.1): a virtual parallel robot architecture that shares the same geometric and kinematic properties as the observation of a set of spatial lines, i.e.

1. The solutions of the Forward Geometric model of the hidden robot are also solutions of the 3D localisation problem of the perspective camera.

2. The singularities of the inverse kinematic Jacobian matrix of the hidden robot are the same as the singularities of the interaction matrix (4.7).

The rows of the inverse kinematic Jacobian of a parallel robot represent the wrenches (a wrench is a six-dimensional vector or a *screw*, that represents a force and a torque) that constrain the robot platform [Mer05]. Further, it is generally accepted that for all parallel robots, these wrenches can actually be simplified as a system of Plücker lines if the points at which the forces are expressed are cleverly chosen [Mer05]. At a singularity of the inverse Jacobian (also called a *Type 2* or *parallel* singularity), the robot gains an uncontrollable motion; that is, it is not fully constrained by the system of wrenches acting on it. As a consequence, the singularities are the configurations for which the system of Plücker lines defined by the rows of the inverse Jacobian become degenerate. The authors then proceed by relying on arguments and previous results from Grassmann-Cayley algebra to derive the degeneracy conditions for the system of Plücker lines.

Here we use a purely algebraic argument to illustrate why the rows of the interaction matrix can be understood as a system of Plücker lines. Let us rewrite equation (4.4) as

$$\frac{d\mathbf{l}_i}{dt} = \frac{1}{\Delta^3} \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix}^T \dot{\mathbf{L}}_i, \tag{4.9}$$

with

$$\begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 \end{bmatrix} = \begin{bmatrix} L_{yi}^2 & -L_{xi}L_{yi} & 0 \\ -L_{xi}L_{yi} & L_{xi}^2 & 0 \\ -L_{xi}L_{zi} & -L_{yi}L_{zi} & \Delta_i^2 \end{bmatrix}^T. \tag{4.10}$$

Let us also express the variation of $\mathbf{L}_i$ as

$$\dot{\mathbf{L}}_i = \begin{bmatrix} [\mathbf{U}_i]_\times & [\mathbf{L}_i]_\times \end{bmatrix} \boldsymbol{\tau}_c, \tag{4.11}$$

where $[\mathbf{U}_i]_\times$ and $[\mathbf{L}_i]_\times$ are the skew-symmetric matrices associated to vectors $\mathbf{U}_i$ and $\mathbf{L}_i$. Then, the interaction matrix (4.7) can be written as

$$\mathbf{M}_i = \begin{bmatrix} (\mathbf{p}_1 \times \mathbf{U}_i)^T & (\mathbf{p}_1 \times \mathbf{L}_i)^T \\ (\mathbf{p}_2 \times \mathbf{U}_i)^T & (\mathbf{p}_2 \times \mathbf{L}_i)^T \\ (\mathbf{p}_3 \times \mathbf{U}_i)^T & (\mathbf{p}_3 \times \mathbf{L}_i)^T \end{bmatrix}. \tag{4.12}$$

By analysing the matrix in (4.10), we point out that the first and second rows $\mathbf{p}_1$ and $\mathbf{p}_2$ are related by $L_{xi}\mathbf{p}_1 + L_{yi}\mathbf{p}_2 = 0$. Note also that $\mathbf{p}_j \cdot \mathbf{L}_i = 0$ for $i, j = 1, 2, 3$, i.e. that the row vectors $\mathbf{p}_j$, are all orthogonal to $\mathbf{L}_i$; and that the vector $\mathbf{p}_3$ is linearly independent from $\mathbf{p}_1$, $\mathbf{p}_2$ as long as $\Delta_i \neq 0$. Since $\mathbf{L}_i$ is, by definition, also orthogonal to the vectors $\mathbf{X}_i$ and $\mathbf{U}_i$, the vectors $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ span the same subspace as $\{\mathbf{X}_i, \mathbf{U}_i\}$, namely, the plane Π_i containing the line and the focal point C, and whose normal has direction $\mathbf{L}_i$.

As a consequence, the vectors $\mathbf{U}_i \times \mathbf{p}_j$ and $\mathbf{L}_i \times \mathbf{p}_j$ in (4.12) are always mutually orthogonal (or zero) for each j (i.e. they satisfy (4.1)) and, therefore, the rows of the interaction matrix $\mathbf{M}_i$ can also be regarded as the coordinates of a system of Plücker lines. Note that the case where $\mathbf{U}_i \times \mathbf{p}_j = 0$ is not a degenerate case, but instead corresponds to a projective line at infinity; likewise $\mathbf{L}_i \times \mathbf{p}_j = 0$ corresponds to a line passing through the origin.

4.2.3 Change of basis for the rows of the interaction matrix

We have just shown that the vectors $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$ in (4.4) span the same vector subspace as $\{\mathbf{X}_i, \mathbf{U}_i\}$. In other words, $\mathbf{X}_i$ and $\mathbf{U}_i$ can be expressed as linear combinations of the vectors $\mathbf{p}_j$. We have

$$\mathbf{X}_i = \sum_{j=1}^{3} a_j \cdot \mathbf{p}_j \quad \text{and} \quad \mathbf{U}_i = \sum_{j=1}^{3} b_j \cdot \mathbf{p}_j. \tag{4.13}$$

for some integers a_j and b_j, $j = 1, 2, 3$. Assuming that the factor Δ_i in (4.4) is non-zero, the matrix

$$\mathbf{H}_i = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3, \end{bmatrix} \tag{4.14}$$

is always of rank 2, such that the product $\mathbf{H}_i \cdot \mathbf{M}_i$ is

$$\mathbf{H}_i \cdot \mathbf{M}_i = \begin{bmatrix} (\mathbf{X}_i \times \mathbf{U}_i)^T & (\mathbf{X}_i \times \mathbf{L}_i)^T \\ \mathbf{0}_{1\times 3} & (\mathbf{U}_i \times \mathbf{L}_i)^T \end{bmatrix}. \tag{4.15}$$

and is also always of rank 2.

The matrix (4.15) is proportional to the basis $\boldsymbol{\xi}_i$ for the rows of $\mathbf{M}_i$, obtained differently in [BMC16], and which can be expressed as

$$\boldsymbol{\xi}_i = \begin{bmatrix} \boldsymbol{\xi}_{i1} \\ \boldsymbol{\xi}_{i2} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{i1}^T & (\overrightarrow{QP_i} \times \mathbf{f}_{i1})^T \\ \mathbf{0}_{1\times 3} & \mathbf{m}_{i2}^T \end{bmatrix}, \tag{4.16}$$

where $\mathbf{f}_{i1}$ and $\mathbf{m}_{i2}$ are any two vectors colinear respectively to $\mathbf{L}_i$ and $\mathbf{U}_i \times \mathbf{L}_i$:

$$\mathbf{f}_{i1} \;\propto\; \mathbf{X}_i \times \mathbf{U}_i = \mathbf{L}_i, \quad \mathbf{m}_{i2} \;\propto\; \mathbf{U}_i \times \mathbf{f}_{i1}, \tag{4.17}$$

while Q is any point in space. Note that, by definition, $\mathbf{f}_{i1}$ is any vector normal to the plane Π_i which contains the line and the focal point C, and $\mathbf{m}_{i2}$ is any vector orthogonal to both $\mathbf{f}_{i1}$ and the line direction $\mathbf{U}_i$.

Let us remark that, in (4.16), the Plücker vector $\boldsymbol{\xi}_{i1}$ represents a straight line passing through the point P_i with direction $\mathbf{f}_{i1}$, while $\boldsymbol{\xi}_{i2}$ is a line at infinity (alias an *ideal line*) in the projective space, with direction $\mathbf{m}_{i2}$.

Some observations arise from the fact that the basis $\boldsymbol{\xi}_i = \left[\boldsymbol{\xi}_{i1}^T \ \boldsymbol{\xi}_{i2}^T\right]^T$ is spanned by a system of lines:

- Degeneracy of a system of lines is independent of the choice of point Q (appearing in (4.16)) at which the lines are expressed [Mer05]. Nevertheless, when computing the analytical expressions, all lines must be given in reference to the same point. Note however that the vector $\mathbf{f}_{i1}$ is still directly dependent on the location of C.

- The conditions for degeneracy of any system of lines depend only on the relative configuration of the lines [BMC16, KWCC09]. Specifically, they are independent of the frame where the Plücker vectors are expressed, and therefore of the relative orientation of the object and camera frames.

In particular the second remark will be useful to simplify the computations in the following sections by assuming a constant zero orientation for the camera.

The new basis $\boldsymbol{\xi}_i$ is a valid representation so long as the depth factor $\Delta_i = \sqrt{L_{xi}^2 + L_{yi}^2}$ appearing in (4.2) is non-zero. This excludes only 2 camera configurations: *1)* when line $\mathcal{L}_i$ is fully contained in the plane $Z = 0$ *of the camera frame* and *2)* when the focal point C lies on the line $\mathcal{L}_i$; in both situations the coordinates L_{xi} and L_{yi} vanish. These are degenerate cases for which the projection mapping in (4.2) is ill-defined, so we will not consider them in the sections that follow.

Finally, based on the previous results, a basis $\boldsymbol{\xi}_{(n)}$ for the full interaction matrix $\mathbf{M}_{(n)}$ is obtained by stacking the rows of (4.16) for each line i:

$$\boldsymbol{\xi}_{(n)} = [\boldsymbol{\xi}_1^T \ \boldsymbol{\xi}_2^T \dots \boldsymbol{\xi}_n^T]^T \in \mathbb{R}^{2n\times 6}. \tag{4.18}$$

For instance, a basis for the interaction matrix $\mathbf{M}_{(3)}$ corresponding to P3L is given by $\boldsymbol{\xi}_{(3)} = [\boldsymbol{\xi}_{11}^T \ \boldsymbol{\xi}_{12}^T \ \boldsymbol{\xi}_{21}^T \ \boldsymbol{\xi}_{22}^T \ \boldsymbol{\xi}_{31}^T \ \boldsymbol{\xi}_{32}^T]^T$. The singularities of this matrix were analysed algebraically in [BMC16]. These results are revisited in the next section from a geometric point of view, which will then be used to analyse the singularities both algebraically and geometrically in the observation of more than three lines.

4.3 Revisiting the singularities in P3L

4.3.1 Parametrization

As mentioned in Section 4.2, the set of lines in the three dimensional projective space $\mathbb{P}^3$ is a four-dimensional manifold, so a line can be defined using four independent parameters. In the Plücker representation of lines, only four out of the six Plücker coordinates are independent. To compute the Plücker coordinates, we will describe each line using the position of a point and a direction vector relative to a fixed object frame $\mathcal{F}_o : (O, \mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$, with its axes defining an orthonormal, right-handed basis. By carefully choosing the object frame $\mathcal{F}_o$, we can reduce the number of parameters needed to define the system. The first line can be placed on the $\mathbf{x}_o$ axis, and the second line parallel to the plane $z_o = 0$ and intersecting the $\mathbf{z}_o$ axis. Any other line can be defined using its points of intersection with any two of the three planes $x_o = 0$,

$y_o = 0$ or $z_o = 0$. So, the third line is defined using its points of intersection with the plane $x_o = 0$ and $z_o = 0$. This leaves us 7 parameters to define three lines using the following two points M_i and N_i on each line.

$$\begin{aligned} \overrightarrow{OM_1} &= [0\ 0\ 0]^T, & \overrightarrow{ON_1} &= [1\ 0\ 0]^T, \\ \overrightarrow{OM_2} &= [0\ 0\ d_1]^T, & \overrightarrow{ON_2} &= [r_1\ r_2\ d_1]^T, \\ \overrightarrow{OM_3} &= [d_2\ d_3\ 0]^T, & \overrightarrow{ON_3} &= [0\ r_3\ r_4]^T. \end{aligned} \tag{4.19}$$

Since the rows of the interaction matrix defined in Section 4.2.3 consist of some affine and ideal lines, defined using the direction vector of the lines, the parametrization can be simplified by considering the direction vector $\mathbf{U}_i = \overrightarrow{OM_i} - \overrightarrow{ON_i}$, $i = 1, 2, 3$ and changing the parameters $-r_1 = s_1$, $-r_2 = s_2$, $-r_4 = s_4$, $d_3 - r_3 = s_3$ as follows:

$$\begin{aligned} \overrightarrow{OM_1} &= [0\ 0\ 0]^T, & \mathbf{U}_1 &= [1\ 0\ 0]^T, \\ \overrightarrow{OM_2} &= [0\ 0\ d_1]^T, & \mathbf{U}_2 &= [s_1\ s_2\ 0]^T, \\ \overrightarrow{OM_3} &= [d_2\ d_3\ 0]^T, & \mathbf{U}_3 &= [d_2\ s_3\ s_4]^T. \end{aligned} \tag{4.20}$$

Next, we parametrize the position of the camera focal point C relative to $\mathcal{F}_o$ by the vector ${}^o\overrightarrow{OC} = [X\ Y\ Z]^T$, where the superscript ${}^o(\cdot)$ indicates that the coordinates are expressed in the frame $\mathcal{F}_o$, and we define the camera frame $\mathcal{F}_c : \{C, \mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c\}$ centred at C, with $\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c$ also an orthonormal basis. It was noted in Section 4.2.3 that the singularity conditions of the problem are independent of the relative orientation of the object and camera frames. Hence, for the computations we will assume that $\mathcal{F}_c$ can be obtained from $\mathcal{F}_o$ by a direct translation by the vector ${}^o\overrightarrow{OC}$.

The singularity loci will be given in terms of the location of the focal point C relative to the fixed object frame $\mathcal{F}_o$, that is as a set of expressions involving variables X, Y and Z. In practice it is sometimes more convenient to express the solutions in terms of the position of the origin O relative to the camera frame $\mathcal{F}_c$. If the relative orientation of frames $\mathcal{F}_o$ and $\mathcal{F}_c$ is represented by a rotation matrix ${}^c\mathbf{R}_o$, we can retrieve the position of the origin O in $\mathcal{F}_c$ by introducing a new set of variables X', Y', Z' such that

$${}^c\overrightarrow{CO} = \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = -{}^c\mathbf{R}_o \cdot {}^o\overrightarrow{OC} = -{}^c\mathbf{R}_o \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \tag{4.21}$$

In the following computations, we can assume ${}^c\mathbf{R}_o$ to be the identity matrix.

4.3.2 Geometric interpretation of singularities in P3L

Singularities in the observation of three lines have already been determined algebraically [BMC16]. Here we provide a geometric interpretation of those results, which will aid us in determining the singularities in P4L and P5L. By choosing the point Q in (4.16) as the camera centre C, we have

$$\xi_i = \begin{bmatrix} \xi_{i1} \\ \xi_{i2} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{i1}^T & (\mathbf{X}_i \times \mathbf{f}_{i1})^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{i2}^T \end{bmatrix}, \tag{4.22}$$

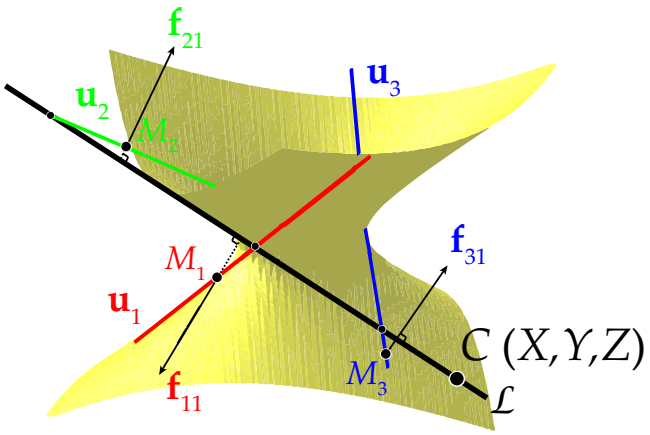


Figure 4.2: One of the singularities in P3L is when the camera centre C lies on the hyperboloid formed by the three observed lines.

where $\mathbf{X}_i = \overrightarrow{OP_i} - \overrightarrow{OC}$ with $\overrightarrow{OC} = [X, Y, Z]^T$ being the position vector of the camera centre, and $\mathbf{f}_{i1}$ and $\mathbf{m}_{i2}$ being vectors defined as (4.17). Consequently, a basis for the rows of the interaction matrix $\mathbf{M}_{(3)}$ can be obtained under the form $\boldsymbol{\xi}_{(3)} = [\boldsymbol{\xi}_{11}^T\ \boldsymbol{\xi}_{12}^T\ \boldsymbol{\xi}_{21}^T\ \boldsymbol{\xi}_{22}^T\ \boldsymbol{\xi}_{31}^T\ \boldsymbol{\xi}_{32}^T]^T$.

Under the assumption that the factor Δ_i defined in (4.2) is not zero for any i (in other words, that all the 3D lines project on a 2D line on the image plane), it was proven in [BMC16] that the degeneracy of the basis $\boldsymbol{\xi}_{(3)}$ and thus of the interaction matrix $\mathbf{M}_{(3)}$ occurs if and only if one of the following two conditions is satisfied:

$$G = \mathbf{f}_{11} \cdot (\mathbf{f}_{21} \times \mathbf{f}_{31}) = 0 \tag{4.23}$$

or

$$H = \mathbf{m}_{12} \cdot (\mathbf{m}_{22} \times \mathbf{m}_{32}) = 0, \tag{4.24}$$

that is, when the three vectors $\mathbf{f}_{i1}$, $i = 1, 2, 3$ (respectively, $\mathbf{m}_{i2}$) defined in (4.17) are parallel to the same plane. The product $G \cdot H$ is in fact the determinant of the matrix $\boldsymbol{\xi}_{(3)}$. In the case of three general lines observed in space, it was shown in [BMC16] that a singularity appears when the camera lies either on a quadric or on a cubic surface, defined respectively by (4.23) and (4.24).

Using the parametrization (4.20), the quadratic factor (4.23) looks like:

$$\begin{aligned} G = {} & d_1 s_2 s_4 XY - s_2 (d_1 s_3 + d_3 s_4) XZ - d_1 s_1 s_4 Y^2 + (d_1 s_1 s_3 + d_2 s_2 s_4) YZ \\ & - d_1 s_4 (d_2 s_2 - d_3 s_1) Y + d_2 s_2 (d_3 - s_3) Z^2 - d_1 d_2 s_2 (d_3 - s_3) Z. \end{aligned} \tag{4.25}$$

The solutions of $G = 0$ define a one-sheeted hyperboloid in $\mathbb{R}^3$, leading to the following proposition.

Proposition 4.2. *A singularity of the interaction matrix in the observation of three lines $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ occurs when the camera centre C lies on the hyperboloid of one sheet uniquely described by the three lines, with equation $G(X, Y, Z) = 0$. This hyperboloid is the ruled surface spanned by the locus of all lines which are transversal to all $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$.*

Proof. The quadratic factor is the determinant of the upper left (3×3) matrix of the basis $\boldsymbol{\xi}_{(3)}$. Let the kernel of the matrix $\begin{bmatrix} \mathbf{f}_{11} & \mathbf{f}_{21} & \mathbf{f}_{31} \end{bmatrix}$ be the direction vector of a line $\mathcal{L}$. Since the affine lines are represented according to (4.17), the line $\mathcal{L}$ must lie in a plane containing $\mathbf{X}_i$ and $\mathbf{U}_i$, say Π_i. Consequently, $\mathcal{L}$ has to intersect all $\mathbf{u}_i$ and therefore the observed three lines (see Fig. 4.2). As a result, it belongs to the complementary regulus of a hyperboloid of one sheet defined by the regulus of the observed lines [OSG20, Chapter 2]. $\mathcal{L}$ is called the *transversal* line. Moreover, $\mathcal{L}$ must be the intersection of planes Π_1, Π_2 and Π_3. Since C belongs to Π_i for all i, it should lie in their intersection too and hence $\mathcal{L}$ has to contain C. □

Corollary 4.2.1. *When the camera centre C lies on the hyperboloid defined by the three observed lines, the kernel of the interaction matrix $\boldsymbol{\xi}_{(3)}$ is an ideal line whose moment vector is the same as the direction vector of the line passing through C and intersecting the observed lines. As a result, we face problems in control for infinitesimal translations of the camera along the transversal line. Furthermore, the finite lines constituting the rows of the interaction matrix are all parallel to the same plane whose normal vector is along the transversal. Then, the six lines $\boldsymbol{\xi}_{i1}$ and $\boldsymbol{\xi}_{i2}$, $i = 1, 2, 3$ are said to be in a* singular linear line complex *[PW01, Chapter 3].*

The latter remark assures that the kernel of the interaction matrix represents the Plücker coordinates of a line. In terms of screw theory [Hun87], it is always a screw of infinite pitch.

Unfortunately, the geometric interpretation is not as straightforward when C lies on the cubic surface $H = 0$ of (4.24) leading to singularities. In this case, the lines $\boldsymbol{\xi}_{i1}$ and $\boldsymbol{\xi}_{i2}$ belong to a *regular linear line complex* [PW01, Chapter 3]. In terms of screw theory [Hun87], the kernel is no longer a line but a screw, meaning that the controllability issues arise when the camera performs this instantaneous screw motion. Additionally, unlike the hyperboloid, the cubic surface is not uniquely defined by the three observed lines. This is due to a classic result from Geometry by Arthur Cayley and George Salmon who showed in 1849 that there are 27 lines on a cubic surface (refer to [Laz14] for a proof from an algebraic geometry point of view). Therefore, computational algebraic techniques will be employed to deal with this singularity in the case of P4L.

4.4 Singularities in P4L

4.4.1 Parametrization

Following Section 4.3.1, the first three lines are defined according to (4.20). The fourth line is defined using its two points of intersection P_4 and N_4 with the planes $x_o = 0$ and $y_o = 0$, respectively:

$$\overrightarrow{OP_4} = [0, d_4, d_5]^T, \ \overrightarrow{ON_4} = [r_5, 0, r_6]^T. \tag{4.26}$$

Thus, the direction vector of the fourth line is given by $\mathbf{U}_4 = \overrightarrow{OP_4} - \overrightarrow{ON_4}$. After replacing $-r_5$ and $d_5 - r_6$ by s_5 and s_6, respectively, we have

$$\overrightarrow{OP_4} = [0, d_4, d_5]^T,\ \mathbf{U}_4 = [s_5, d_4, s_6]^T. \tag{4.27}$$

As mentioned in Section 4.3.1, the assumption that the relative orientation between the camera frame $\mathcal{F}_c$ and the object frame $\mathcal{F}_o$ is zero remains valid in the following analysis.

4.4.2 Singularity analysis

From the remarks made in Section 4.2.3, a basis $\boldsymbol{\xi}_i = [\xi_{i1}^T\ \xi_{i2}^T]^T$ for the rows of the interaction matrix for each line is computed as in (4.16), with point Q taken as the camera centre C, and with vectors $\mathbf{f}_{i1}$ and $\mathbf{m}_{i2}$ given by (4.17).

Singularities of the interaction matrix $\mathbf{M}_{(4)}$ of the four lines appear when the (8×6) matrix

$$\boldsymbol{\xi}_{(4)} = \begin{bmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \boldsymbol{\xi}_3 \\ \boldsymbol{\xi}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{11}^T & (\overrightarrow{QP_1} \times \mathbf{f}_{11})^T \\ \mathbf{0} & \mathbf{m}_{12}^T \\ \mathbf{f}_{21}^T & (\overrightarrow{QP_2} \times \mathbf{f}_{21})^T \\ \mathbf{0} & \mathbf{m}_{22}^T \\ \mathbf{f}_{31}^T & (\overrightarrow{QP_3} \times \mathbf{f}_{31})^T \\ \mathbf{0} & \mathbf{m}_{32}^T \\ \mathbf{f}_{41}^T & (\overrightarrow{QP_4} \times \mathbf{f}_{41})^T \\ \mathbf{0} & \mathbf{m}_{42}^T \end{bmatrix} \tag{4.28}$$

formed by stacking the rows in (4.16) for all lines, becomes rank-deficient. This is the case if and only if all the 28 maximal minors of (4.28) vanish simultaneously.

All the entries of the matrix $\boldsymbol{\xi}_{(4)}$ are polynomials in the variables $\{X, Y, Z\}$ representing the camera location, with coefficients which are polynomials in the parameters $\boldsymbol{\eta} = \{s_1, s_2, s_3, s_4, s_5, s_6, d_1, d_2, d_3, d_4, d_5\}$. The maximal minors of $\boldsymbol{\xi}_{(4)}$ then form a system of 28 polynomials, which we call p_i. We denote $\mathcal{I}_{28} = \langle p_1, p_2, \ldots, p_{28} \rangle$ the **polynomial ideal** generated by these minors. We refer to Chapter 3 for the theoretical background about polynomial ideals necessary to follow this section.

In geometric terms, the locus of complex solutions of a system of polynomial equations is the **algebraic variety** of their ideal. The (complex) solutions of all the polynomials in $\mathcal{I}_{28}$ define a variety $\mathbf{V}(\mathcal{I}_{28}) \subset \mathbb{C}^3$, which consists of all points where the matrix $\boldsymbol{\xi}_{(4)}$ becomes rank-deficient, and which depends on the parameters $\boldsymbol{\eta}$.

To get a better insight of $\mathbf{V}(\mathcal{I}_{28})$, we describe below how it can be split into subsets, i.e. written as the union of the solution sets of simpler systems of polynomials.

Consider the 28 maximal submatrices of size (6×6) of $\boldsymbol{\xi}_{(4)}$. Up to a reordering of their rows, they can be classified in three groups:

$$\boldsymbol{\xi}_{1234}^{ij} = \begin{bmatrix} \mathbf{f}_{i1}^T & (\mathbf{X}_i \times \mathbf{f}_{i1})^T \\ \mathbf{f}_{j1}^T & (\mathbf{X}_j \times \mathbf{f}_{j1})^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{12}^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{22}^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{32}^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{42}^T \end{bmatrix}, \tag{4.29}$$

$$\boldsymbol{\xi}^{ijk}_{\ell mn} = \begin{bmatrix} \mathbf{f}_{i1}^T & (\mathbf{X}_i \times \mathbf{f}_{i1})^T \\ \mathbf{f}_{j1}^T & (\mathbf{X}_j \times \mathbf{f}_{j1})^T \\ \mathbf{f}_{k1}^T & (\mathbf{X}_k \times \mathbf{f}_{k1})^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{\ell 2}^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{m2}^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{n2}^T \end{bmatrix}, \tag{4.30}$$

$$\boldsymbol{\xi}^{1234}_{\ell m} = \begin{bmatrix} \mathbf{f}_{11}^T & (\mathbf{X}_i \times \mathbf{f}_{11})^T \\ \mathbf{f}_{21}^T & (\mathbf{X}_j \times \mathbf{f}_{21})^T \\ \mathbf{f}_{31}^T & (\mathbf{X}_k \times \mathbf{f}_{31})^T \\ \mathbf{f}_{41}^T & (\mathbf{X}_l \times \mathbf{f}_{41})^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{\ell 2}^T \\ \mathbf{0}_{(1\times 3)} & \mathbf{m}_{m2}^T \end{bmatrix}, \tag{4.31}$$

where i, j, k and ℓ, m, n range every triplet of numbers in $\{1, 2, 3, 4\}$.

There are six submatrices of the type (4.29), whose rows describe the Plücker vectors of two affine lines (lines in the affine space; in other words, lines which are *not* at infinity) and four lines at infinity. Since a line at infinity is described by only three parameters (the direction vector $\mathbf{m}_{i2}$), four such lines are always linearly dependent. Therefore the matrix $\boldsymbol{\xi}^{ij}_{1234}$, for any i and j, always has a rank of at most 5.

The submatrices of the second type (4.30) are block-triangular, and they are composed of row vectors that represent three affine lines and three lines at infinity. There are $\binom{4}{3} \times \binom{4}{3} = 16$ minors of this type. Their determinants are the products of two polynomials: $\det(\boldsymbol{\xi}^{ijk}_{\ell mn}) = G_{ijk} \cdot H_{\ell mn}$, with the following form:

$$G_{ijk} = \mathbf{f}_{i1} \cdot (\mathbf{f}_{j1} \times \mathbf{f}_{k1}), \quad H_{\ell mn} = \mathbf{m}_{\ell 2} \cdot (\mathbf{m}_{m2} \times \mathbf{m}_{n2}). \tag{4.32}$$

The cases where the subindices $\{i, j, k\}$ and $\{\ell, m, n\}$ coincide correspond to the singularity conditions (4.23) and (4.24) in P3L for each triplet of lines taken individually. Thus, it is useful to note that, for a singularity of the four lines, a necessary, but not sufficient condition is that each triplet of lines is in turn in a singular configuration. Let us consider all the polynomials arising from the determinants of the matrices of type $\boldsymbol{\xi}^{ijk}_{\ell mn}$, and let us denote $\mathcal{I}_{16} = \langle G_{123}H_{123}, \ldots, G_{234}H_{234} \rangle$ the ideal they generate. Since these polynomials are a subset of all of the maximal minors of the (8×6) matrix (4.28), the ideal $\mathcal{I}_{16}$ is contained in the larger ideal $\mathcal{I}_{28}$.

Finally, there are six submatrices $\boldsymbol{\xi}^{1234}_{\ell m}$ of the third category. Their determinants are of degree 5 in the ring $\mathbb{Q}[X, Y, Z]$. Let them generate an ideal $\mathcal{K}$.

It follows that the union of ideals $\mathcal{I}_{16}$ and $\mathcal{K}$ yields $\mathcal{I}_{28}$. Dually, the intersection of their varieties yields $\mathbf{V}(\mathcal{I}_{28})$ (see Chapter 3 or the more complete reference [CLO13]):

$$\mathcal{I}_{28} = \mathcal{I}_{16} \cup \mathcal{K} \quad \Rightarrow \quad \mathbf{V}(\mathcal{I}_{28}) = \mathbf{V}(\mathcal{I}_{16}) \cap \mathbf{V}(\mathcal{K}), \tag{4.33}$$

We first thoroughly analyse the ideal $\mathcal{I}_{16}$ to show how it can be further decomposed into two sub-ideals and then incorporate the analysis of $\mathcal{K}$. All the mathematical derivations shown below can be followed in Maple file contained in the github repository jorge-gf/thesis-archive [GF22]. Since a solution of the polynomials in $\mathcal{I}_{28}$ must also be a solution for the

polynomials in $\mathcal{I}_{16}$, we say that the variety $\mathbf{V}(\mathcal{I}_{28})$ is contained in $\mathbf{V}(\mathcal{I}_{16})$:

$$\mathcal{I}_{16} \subseteq \mathcal{I}_{28} \Longrightarrow \mathbf{V}(\mathcal{I}_{28}) \subseteq \mathbf{V}(\mathcal{I}_{16}), \tag{4.34}$$

although $\mathbf{V}(\mathcal{I}_{16})$ may contain points outside $\mathbf{V}(\mathcal{I}_{28})$.

The ideal $\mathcal{I}_{16}$ can be factorized as the product of two simpler ideals:

$$\mathcal{I}_{16} = \mathcal{G} \times \mathcal{H}, \tag{4.35}$$

where

$$\mathcal{G} = \langle \mathcal{G}_{123}, \mathcal{G}_{124}, \mathcal{G}_{134}, \mathcal{G}_{234} \rangle, \tag{4.36}$$
$$\mathcal{H} = \langle \mathcal{H}_{123}, \mathcal{H}_{124}, \mathcal{H}_{134}, \mathcal{H}_{234} \rangle. \tag{4.37}$$

It implies that the variety $\mathbf{V}(\mathcal{I}_{16})$ is the union of two smaller sets: $\mathbf{V}(\mathcal{I}_{16}) = \mathbf{V}(\mathcal{G}) \cup \mathbf{V}(\mathcal{H})$. That is, the polynomials in $\mathcal{I}_{16}$ vanish whenever $G_{ijk} = 0$ for all i, j, k; or when $H_{\ell mn} = 0$ for all ℓ, m, n. As a consequence, we can rewrite (4.33) as

$$\begin{aligned} \mathbf{V}(\mathcal{I}_{28}) &= (\mathbf{V}(\mathcal{G}) \cup \mathbf{V}(\mathcal{H})) \cap \mathbf{V}(\mathcal{K}) \\ &= (\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K})) \cup (\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})) \,. \end{aligned} \tag{4.38}$$

The variety $\mathbf{V}(\mathcal{G})$ defined by the ideal in (4.36) in $\mathbb{C}[X, Y, Z]$ describes the intersection between four quadratic surfaces: each one a hyperboloid of one-sheet described by three out of the four lines. On the other hand, $\mathbf{V}(\mathcal{H})$, defined by the ideal (4.37), is the intersection of four cubic surfaces.

We can analyse the sub-varieties on the right hand side of (4.38) separately. First, we can check if $\mathbf{V}(\mathcal{K})$ or a component of it lies in $\mathbf{V}(\mathcal{G})$ or in $\mathbf{V}(\mathcal{H})$. For instance, if $\mathbf{V}(\mathcal{G}) \subset \mathbf{V}(\mathcal{K})$ then the analysis is much simpler since the intersection between those varieties would just yield $\mathbf{V}(\mathcal{G})$.

Gröbner bases, which we introduced in Chapter 3, Section 3.2, as a tool to solve computationally problems with polynomial ideals, can be used for this purpose. In particular, given a polynomial ideal I and another polynomial in the same ring f, one can determine if $f \in I$ by computing a Gröbner basis $G_I = \{g_1, \ldots, g_s\}$ for I and using a division algorithm on the polynomial f by each of the elements of this basis. The remainder of the division is called the **normal form** of f by G_I, and is written $\overline{f}^{G_I}$, or $NF(f, G_I)$. The polynomial f is contained in the ideal I if and only if $\overline{f}^{G_I} = 0$. For instance, by obtaining a Gröbner basis for the ideal $\mathcal{G}$, and computing the normal form of the polynomials that define $\mathcal{K}$ by this basis, we can determine whether the ideal $\mathcal{K}$ is contained in $\mathcal{G}$.

In what follows, we use Gröbner bases computations, and the result from Proposition 4.2, to analyse separately each of the two components of the set (4.38): $\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K})$ and $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$.

Analysis of the variety $\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K})$

Let $gb_G = \{g_1, \ldots, g_s\}$ be a Gröbner basis of the ideal $\mathcal{G}$ with respect to *pure lexicographical* monomial ordering, denoted by $\succ_{lex}$, with $X \succ_{lex} Y \succ_{lex} Z \succ_{lex} s_1 \succ_{lex} \ldots \succ_{lex} s_6 \succ_{lex}$

$d_1 \succ_{lex} \ldots \succ_{lex} d_5$ (refer to Chapter 3, Section 3.2 for the definition of the different monomial orderings). Here, we are assuming that our polynomials lie in the ring $\mathbb{Q}[X, Y, Z, \boldsymbol{\eta}]$ of polynomials in variables X, Y, Z and $\boldsymbol{\eta}$ with rational coefficients, where $\boldsymbol{\eta} = \{d_1, \ldots, d_5, s_1, \ldots, s_6\}$. By considering $s_1, \ldots, d_5$ as variables, our Gröbner bases computations will be valid under any specialization of the parameters.

Knowing gb_G, we can compute the normal form of the polynomials $p_i \in \mathcal{K}$, $i = 1, \ldots, 6$ to check if the varieties $\mathbf{V}(\mathcal{G})$ and $\mathbf{V}(\mathcal{K})$ share any component. It can be done using a multivariate polynomial division algorithm (the files are included in the repository jorge-gf/thesis-archive [GF22]), and we find that they vanish in every case: $\overline{p_i}^{\langle g_1, \ldots, g_s \rangle} = 0$ for all i. The consequence is that any common solution of the system $g_1 = \cdots = g_s = 0$ is also a solution of the system $p_1 = \cdots = p_6 = 0$ and is therefore a singular point of the matrix $\boldsymbol{\xi}_{(4)}$. In terms of algebraic varieties, this can be written as

$$\mathbf{V}(\mathcal{G}) \subseteq \mathbf{V}(\mathcal{K}) \text{ and hence } \mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K}) = \mathbf{V}(\mathcal{G}). \tag{4.39}$$

As a result, Equation (4.38) can be updated as

$$\mathbf{V}(\mathcal{I}_{28}) = \mathbf{V}(\mathcal{G}) \cup (\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})), \tag{4.40}$$

implying that one of the singularities in P4L is when the camera centre lies on the intersection of the four hyperboloids given by the variety $\mathbf{V}(\mathcal{G})$.

Additionally, this result can be geometrically interpreted as follows. The basis of the interaction matrix $\mathbf{M}_{(4)}$ in this case consists of four affine lines and four ideal lines. As we know from Section 4.3, one of the singularities in P3L is when the three affine lines belong to a singular linear line complex (meaning that they are parallel to the same plane; see Corollary 4.2.1). When this happens, the kernel of the interaction matrix is a line at infinity and the camera centre lies on a line that intersects all the three observed lines. Similarly, in the case of P4L, $\mathbf{V}(\mathcal{G})$ results in four affine lines of $\boldsymbol{\xi}_{(4)}$ being parallel to the same plane so that its kernel is a line at infinity. Hence, we can expect that a singularity occurs when the camera centre lies on a line that intersects the four observed lines. In fact, this is true for singularities in PnL for any $n \geq 3$.

Theorem 4.3. *Consider the observation of n lines $\mathcal{L}_1, \ldots, \mathcal{L}_n$ in space. If there exist one or more lines that intersect all of $\mathcal{L}_1, \ldots, \mathcal{L}_n$, then a singularity of the interaction matrix occurs if the camera centre is on one of these lines.*

Proof. Let the observed lines be $\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n$. Given a line $\mathcal{L}$ on which C lies, the distance between lines $\mathcal{L}$ and $\mathcal{L}_i$ is given by

$$d_{\mathcal{L}\mathcal{L}_i} = \left\| \frac{(\hat{\mathbf{l}} \times \hat{\mathbf{l}}_i)^T}{||\hat{\mathbf{l}} \times \hat{\mathbf{l}}_i||} (\overrightarrow{OC} - \overrightarrow{OP_i}) \right\|, \tag{4.41}$$

where P_i is a point on line $\mathcal{L}_i$, $\hat{\mathbf{l}}$ and $\hat{\mathbf{l}}_i$ are unit direction vectors of lines $\mathcal{L}$ and $\mathcal{L}_i$, respectively. $\mathcal{L}$ and $\mathcal{L}_i$ intersect when

$$(\mathbf{l} \times \mathbf{l}_i)^T (\overrightarrow{OC} - \overrightarrow{OP_i}) = 0, \tag{4.42}$$

$$(\mathbf{l}_i \times \mathbf{X}_i)^T \mathbf{l} = 0. \tag{4.43}$$

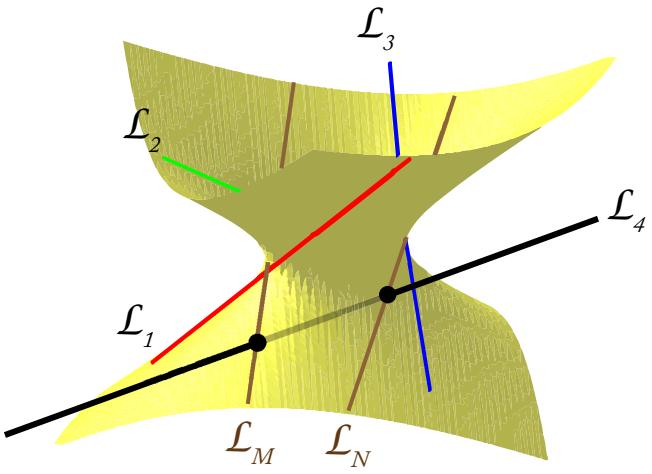


Figure 4.3: Four observed lines $\mathcal{L}_i$, $i = 1, 2, 3, 4$ in a hyperbolic congruence leading to two singular lines $\mathcal{L}_M$ and $\mathcal{L}_N$.

It follows from (4.17) that $\mathbf{f}_{i1}^T \mathbf{l} = 0$, $i = 1, \ldots, n$. Thus, for n lines, the matrix $\boldsymbol{\xi}_{(n)}$ in (4.18) has a kernel which represents a line at infinity with Plücker coordinates $(\mathbf{0}, \mathbf{l})$ and hence it is singular. □

For $n = 3$, the locus of lines intersecting the three observed lines is a hyperboloid of one sheet, which leads to Proposition 4.2. For $n = 4$, in the generic case, it is two lines and the observed lines belong to a linear line congruence [PW01]. For $n \geq 5$, there are in general no transversal lines [PW01].

The variety of the ideal $\mathcal{G}$ in (4.36) consists of the intersections of the four hyperboloids $\mathcal{G}_{123} = \mathcal{G}_{124} = \mathcal{G}_{134} = \mathcal{G}_{134} = 0$, which define precisely the lines transversal to $\mathcal{L}_1, \ldots, \mathcal{L}_4$. In general there are two such lines, which can be real or complex.

In the real domain $\mathbb{R}[X, Y, Z]$, the intersection of the four hyperboloids i.e. $\mathbf{V}(\mathcal{G})$ can be an empty set, a line or two lines. If we consider the hyperboloid defined by the first three observed lines, assuming the fourth line does not lie entirely on the hyperboloid, it can intersect the hyperboloid in $0, 1$ or 2 points. Then, the four lines are said to be in an elliptic, a parabolic or a hyperbolic line congruence, respectively [PW01]. A case of hyperbolic congruence is shown in Fig. 4.3. A line passing through the point of intersection and lying on the hyperboloid intersects all four lines. Thanks to the use of Gröbner bases, we can derive a condition such as to enforce the four observed lines to be in an elliptic congruence, thus avoiding the appearance of this type of *line singularities*.

By finding the Gröbner basis of the ideal $\mathcal{G}$ with respect to the monomial ordering $Y \succ_{\text{lex}} Z \succ_{\text{lex}} X$, we obtain four polynomials, the first of which is as follows:

$$a_2 Y^2 + a_1 Y + a_0 = 0, \tag{4.44}$$

where

$$
\begin{aligned}
a_2 &= d_1 s_4 \left(d_2 s_2 s_6 - d_3 s_1 s_6 - d_4 d_5 s_1 + d_4 s_1 s_6 + d_5 s_2 s_5\right), \\
a_1 &= (d_1 d_2 d_3 s_2 s_6 - d_1 d_2 d_4 s_2 s_4 - d_1 d_2 s_2 s_3 s_6 + d_1 d_3 d_4 s_1 s_4 \\
&\quad + d_1 d_4 d_5 s_1 s_3 - d_1 d_4 s_1 s_3 s_6 - d_1 d_4 s_2 s_4 s_5 - d_1 d_5 s_2 s_3 s_5 \\
&\quad + d_2 d_4 d_5 s_2 s_4 - d_2 d_4 s_2 s_4 s_6 - d_3 d_5 s_2 s_4 s_5) Z, \\
a_0 &= -d_4 s_2 (d_1 d_2 d_3 - d_1 d_2 s_3 - d_1 s_3 s_5 - d_2 d_3 d_5 + d_2 d_3 s_6 \\
&\quad + d_2 d_5 s_3 - d_2 s_3 s_6 - d_3 s_4 s_5) Z^2.
\end{aligned}
$$

Since the variety of $\mathcal{G}$ represents two real or complex lines, the quadratic element (4.44) must factorize into two linear polynomials which represent the planes containing the two lines that are transversal to the four observed lines (the remaining elements of the Gröbner basis can be used to deduce the equations of the two lines that constitute $\mathbf{V}(\mathcal{G})$; see Section 4.4.3 for an example). These planes and hence the transversals lying on them are either real or complex depending on the sign of the discriminant of (4.44). Assuming $a_2 \neq 0$, the equation (4.44) has no real solutions if the discriminant

$$\Delta = a_1^2 - 4 a_2 a_0 = Z^2 f(\boldsymbol{\eta}) < 0, \tag{4.45}$$

where $f(\boldsymbol{\eta})$ is a function of the parameters $\boldsymbol{\eta}$. Since $Z^2 \geq 0$, a necessary condition that there are no real transversals intersecting the four observed lines can be given by an inequality solely in terms of the parameters describing the four observed lines. This helps us to avoid the one dimensional singularities due to $\mathbf{V}(\mathcal{G})$. This is quite useful as we will see in the next section that the remaining singularities due to $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ are only of dimension zero, implying that they are isolated points.

Theorem 4.4. *For the Perspective-4-Line problem, a singularity occurs when C lies on any transversal line that intersects the four observed lines. The transversals appear as the intersection of the hyperboloids defined by four triplets of the observed lines. Moreover, by forcing the four observed lines to be in an elliptic congruence, we can make sure that the transversal lines are not real and therefore avoid the one dimensional singularities.*

Analysis of the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$

The analysis of the component $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ is slightly more involved. The files used to perform the computations are in jorge-gf/thesis-archive [GF22]. We obtain a Gröbner basis $gb_H = \{h_1, \ldots, h_t\}$ for the ideal $\mathcal{H}$ w.r.t. the ordering with $X \succ_{lex} Y \succ_{lex} Z$, and compute the normal form of the minors $p_i \in \mathcal{K}$ with respect to it. Note that we now consider polynomials in the ring $\mathbb{Q}(\boldsymbol{\eta})[X, Y, Z]$ of polynomials in X, Y and Z alone. This time, the residues are polynomials in X, Y and Z with coefficients that depend on the parameters:

$$\overline{p_i}^{\langle h_1, \ldots, h_t \rangle} = f_i \neq 0 \in \mathbb{Q}(\boldsymbol{\eta})[X, Y, Z], \ i = 1, \ldots, 6. \tag{4.46}$$

Since the residues f_i do not vanish, unlike (4.39), $\mathbf{V}(\mathcal{H}) \not\subseteq \mathbf{V}(\mathcal{K})$. It implies that any common solution of the system $h_1 = \cdots = h_t = 0$ is not a solution of the system $p_1 = \cdots = p_6 = 0$.

However, the analysis of $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ can be simplified by noting that $\mathbf{V}(\mathcal{K})$ contains the four observed lines and their two transversals. This is because the matrix $\boldsymbol{\xi}_{lm}^{1234}$ in (4.31) loses rank if C lies on the observed four lines or their transversals. As a consequence, $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ might contain points on the four observed lines and their two transversals. As we know from Theorem 4.3 that they are the singularity loci corresponding to $\mathbf{V}(\mathcal{G})$, we would like to remove them from the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$. We know that these six lines must lie in the union of four hyperboloids $\mathbf{V}(\mathcal{G}_{123}) \cup \mathbf{V}(\mathcal{G}_{124}) \cup \mathbf{V}(\mathcal{G}_{134}) \cup \mathbf{V}(\mathcal{G}_{234})$ that appear in (4.36). Therefore, we can remove each hyperboloid from $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ to obtain the remaining singularities. In algebraic geometry terms, removing one variety from the other amounts to computing the set difference of the varieties [CLO13].

The **set difference** of two affine varieties is generally not an affine variety but an open subset of a variety: it cannot in general be written as solutions of a system of polynomial equations. The smallest affine variety which contains it is called the *Zariski closure* of the difference, denoted with an overline (see Section 3.1). In this case, we need to find the following Zariski closure of the difference:

$$\mathbf{V}(\mathcal{F}) = \bigcap_{i=1}^{4} \overline{(\mathbf{V}(\mathcal{K}) \cap \mathbf{V}(\mathcal{H})) \setminus \mathbf{V}(S_i)}, \tag{4.47}$$

where S_i is an element of $\{\mathcal{G}_{123}, \mathcal{G}_{124}, \mathcal{G}_{134}, \mathcal{G}_{234}\}$.

From the correspondence between polynomial ideals and varieties, an ideal defining $\overline{V \setminus W}$ where V and W are affine varieties is obtained as the **saturation** of an ideal I defining V with an ideal J defining W, and is denoted by $I : J^{\infty}$ (see again Section 3.1). The ideal defining (4.47) is

$$\mathcal{F} = \bigcup_{i=1}^{4} (\mathcal{H} \cup \mathcal{K}) : S_i^{\infty}, \tag{4.48}$$

Due to a large number of variables leading to heavy computations, we did not succeed in determining $\mathcal{F}$ in (4.48) using the above approach for generic values of the parameters. Therefore, Section 4.4.3 shows an example where $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ is analysed for some specialization of the parameters $s_1, \ldots, s_6$, $d_1, \ldots, d_5$.

Since we are dealing here with polynomial systems, we know that for almost all values of the parameters, the specialized systems have all the same number of complex solutions [CLO13]. More precisely, there exists a polynomial B depending on the parameters, such that when specializing the parameters outside the zero set of B, the number of complex solutions to the system that we obtain remains invariant.

In our analysis, we have observed that, when specializing the parameters to random values and removing those solutions lying on the lines, one always obtains 10 complex solutions. This indicates that for generic values of the parameters (outside this zero-set of polynomial B), there are at most 10 isolated singularities in the case of P4L. Section 4.4.3 shows one such example where a random specialization of parameters yields 10 complex solutions of which 6 are real.

Proposition 4.5. *An additional singularity condition for the P4L problem is described by the variety (4.48). Generically, it consists of up to 10 isolated points in the configuration space of the camera.*

4.4.3 Singularities of P4L for example configuration

The singularities in P4L are determined for lines whose Plücker coordinates are arbitrarily chosen according to the following parameters:

$$\begin{aligned} &s_1 = 4,\ s_2 = -5,\ s_3 = 7,\ s_4 = 3,\ s_5 = -2,\ s_6 = 13, \\ &d_1 = 2,\ d_2 = 3,\ d_3 = 5,\ d_4 = 1,\ d_5 = 7. \end{aligned} \tag{4.49}$$

Then,

$$\begin{aligned} \mathcal{G} = \langle &-1765\,X^2 - 587\,XY - 878\,XZ + 660\,Y^2 + 232\,YZ - 122\,Z^2 + 2606\,X + 216\,Y + 598\,Z \\ &- 708,\ -177\,XY - 27\,XZ + 75\,Y^2 + 17\,YZ - 8\,Z^2 + 156\,Y + 6\,Z,\ -130\,XY + 40\,XZ \\ &- 104\,Y^2 - 62\,YZ + 10\,Z^2 + 188\,Y - 20\,Z,\ -30\,XY + 145\,XZ - 24\,Y^2 + 11\,YZ + 30\,Z^2 \\ &+ 210\,Y - 60\,Z\rangle. \end{aligned}$$

The Gröbner basis gb_G of $\mathcal{G}$ w.r.t. $Z \succ_{lex} Y \succ_{lex} X$ consists of four elements $\{g_1, g_2, g_3, g_4\}$. The first element factors (using the command `evala(AFactor)` in Maple) as follows:

$$\begin{aligned} g_1 = -2166Y^2 - 1166YZ + 50Z^2 & \\ = \frac{1}{2166}&\left(Z(\sqrt{448189} + 583) + 2166Y\right) \\ &\left(Z(\sqrt{448189} - 583) - 2166Y\right). \end{aligned}$$

By substituting the factors into the other elements of gb_G, $\mathcal{G}$ can be decomposed into two subideals whose varieties correspond to the two transversal lines intersecting all the four observed lines:

$$\begin{aligned} \mathcal{G} &= \mathcal{M} \cap \mathcal{N},\ \text{where} \\ \mathcal{M} &= \langle Z(\sqrt{448189} + 583) + 2166Y, \\ &\quad Z(329\sqrt{448189} - 587953) + 2166\sqrt{448189} \\ &\quad - 3822990X + 2822298\rangle, \\ \mathcal{N} &= \langle Z(\sqrt{448189} - 583) - 2166Y, \\ &\quad Z(329\sqrt{448189} + 587953) + 2166\sqrt{448189} \\ &\quad + 3822990X - 2822298\rangle. \end{aligned}$$

Following Section 4.4.2, it is straightforward to verify that $\overline{p_i}^{\langle g_1,\ldots,g_4\rangle} = 0\ \forall\ p_i \in \mathcal{K}$. So, the corresponding positive dimensional singularities are $\mathbf{V}(\mathcal{M})$ and $\mathbf{V}(\mathcal{N})$.

The ideals $\mathcal{H}$ and $\mathcal{K}$ are determined whereas only $\mathcal{H}$ is displayed here since $\mathcal{K}$ is too large:

$$\begin{aligned}\mathcal{H} = \langle &-73960\,X^3 - 46428\,YX^2 + 320426\,X^2Z + 88867\,XY^2 + 163934\,YXZ + 184389\,XZ^2 \\ &+ 62940\,Y^3 - 356381\,Y^2Z - 32282\,YZ^2 + 27183\,Z^3 + 210018\,X^2 - 721747\,XY \\ &- 416981\,XZ - 146898\,Y^2 - 118097\,YZ - 116973\,Z^2 + 111106\,X + 377082\,Y \\ &+ 153504\,Z - 56580, -3038\,YX^2 + 3686\,X^2Z - 2288\,XY^2 + 16544\,YXZ + 3344\,XZ^2 \\ &- 315\,Y^3 - 4111\,Y^2Z - 157\,YZ^2 + 663\,Z^3 + 27166\,XY - 4168\,XZ + 2769\,Y^2 \\ &- 13942\,YZ - 1527\,Z^2 - 25806\,Y + 690\,Z, -650\,YX^2 - 3350\,X^2Z - 195\,XY^2 \\ &+ 8450\,YXZ - 845\,XZ^2 + 260\,Y^3 + 3410\,Y^2Z + 390\,YZ^2 - 13390\,XY + 1630\,XZ \\ &+ 276\,Y^2 - 8372\,YZ + 15088\,Y, 225\,YX^2 - 1685\,X^2Z + 705\,XY^2 + 450\,YXZ - 345\,XZ^2 \\ &+ 420\,Y^3 - 1325\,Y^2Z - 645\,YZ^2 - 8056\,XY + 705\,XZ - 918\,Y^2 - 1527\,YZ + 5658\,Y\rangle \end{aligned} \tag{4.50}$$

The Gröbner basis gb_H of $\mathcal{H}$ w.r.t. $Z \succ_{lex} Y \succ_{lex} X$ contains 5 elements $\{h_1, h_2, h_3, h_4, h_5\}$. As proposed in Section 4.4.2, the normal forms can be calculated as $\overline{p_i}^{\langle h_1,\ldots,h_5\rangle} = f_i$. The residuals f_i do not vanish. Thus, the whole variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ is considered and it turns out to be of dimension 0 and degree 22 with 16 real solutions (see the attached Maple file):

Table 4.1: Elements of the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$.

	X	Y	Z
*1	-9.858	- 2.473	- 1.841
2	-0.720	0.0	0.0
*3	-0.320	0.010	0.220
4	-0.007	0.009	2.0
*5	0.054	0.009	1.842
6	0.328	0.0	0.0
7	0.918	0.141	- 2.082
*8	0.938	0.568	- 2.023
9	0.972	- 1.215	2.0
*10	1.011	0.794	- 0.885
11	1.016	0.371	- 1.984
12	1.218	0.390	- 0.918
13	3.231	0.0	0.0
14	3.880	7.054	0.880
*15	65.09	- 96.57	- 0.036
16	90.31	- 112.9	2.0

However, it can be verified that some of these points lie on the observed four lines or their transversals $\mathbf{V}(\mathcal{M})$ and $\mathbf{V}(\mathcal{N})$. Since any point incident with these lines leads to a singularity, we are interested in singular points that do not lie on them. They can be calculated by determining the ideal $\mathcal{F}$ using (4.48). The Gröbner basis gb_F of $\mathcal{F}$ w.r.t. $Z \succ_{lex} Y \succ_{lex} X$

has the following nice structure (called the *shape position*):

$$gb_F = \{f_a(Z),\ f_b(Z) + Y,\ f_c(Z) + X\},$$

where $f_a(Z) = \Sigma_{i=1}^{10} a_i Z^i$ is a degree 10 univariate polynomial in Z, $f_b(Z)$ and $f_c(Z)$ are also univariate polynomials in Z. It follows that $\mathbf{V}(\mathcal{F})$ is of degree 10. It consists of 6 real points marked with an asterisk each in Table 4.1. Thus, the singularity loci for this example include the four observed lines, their two transversals and 6 points.

Since the parameters were chosen randomly for this analysis, this indicates that, for values of the parameters outside the zero set of some polynomial depending only on the parameters (hence of measure zero), the real singular points in P4L can be up to 10.

Now that the generic case is treated, let us deal with a more specific case. Indeed, it is often the case that the observed lines in an environment are constrained with orthogonality and/or parallelism. This special case is considered in the next section and the singularities are determined with the proposed approach without specializing any parameters.

4.4.4 Singularities in P4L with orthogonality and parallelism

We consider three mutually orthogonal lines $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$, and a fourth one $\mathcal{L}_4$ with direction parallel to $\mathcal{L}_1$. The parametrization (4.20) and (4.27) cannot be used in this context since we need the lines to only intersect one of the planes $x_o = 0$, $y_o = 0$ or $z_o = 0$. The object frame $\mathcal{F}_o : (O, \mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$ is fixed relative to the four lines, with its axes defining an orthonormal, right-handed basis, and such that $\mathbf{x}_o$ is collinear to $\mathcal{L}_1$ and $\mathcal{L}_4$; $\mathbf{y}_o$ is collinear to $\mathcal{L}_2$, and $\mathbf{z}_o$ is collinear to $\mathcal{L}_3$. With this parametrization in the object frame, the direction vector $\mathbf{U}_i$ of the four lines and the coordinates of points P_i belonging to each of them are given by:

$$\begin{aligned}
\overrightarrow{OP_1} &= [0,0,0]^T, \quad & \mathbf{U}_1 &= [1,0,0]^T,\\
\overrightarrow{OP_2} &= [0,0,d_1]^T, \quad & \mathbf{U}_2 &= [0,1,0]^T,\\
\overrightarrow{OP_3} &= [d_2,d_3,0]^T, \quad & \mathbf{U}_3 &= [0,0,1]^T,\\
\overrightarrow{OP_4} &= [0,d_4,d_5]^T, \quad & \mathbf{U}_4 &= [1,0,0]^T.
\end{aligned}$$

Following the analysis done in the preceding section, the varieties $\mathbf{V}(\mathcal{G})$ and $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$ will be analysed separately.

In this context, the ideal $\mathcal{G}$ in (4.36) is calculated as follows:

$$\begin{aligned}
\mathcal{G} = \langle & XY(d_1-d_5) - XZ(d_3-d_4) - X(d_1d_4 - d_3d_5) + YZd_2 - Yd_1d_2 - Zd_2d_4 + d_1d_2d_4,\\
& (-d_3+Y)(Yd_5 - Zd_4),\ -(-d_1+Z)(Yd_5 - Zd_4),\ XYd_1 - XZd_3 + YZd_2 - Yd_1d_2 \rangle.
\end{aligned} \tag{4.51}$$

According to Theorem 4.3, we expect the positive dimensional singularities corresponding to $\mathbf{V}(\mathcal{G})$ to be the transversals that intersect the four observed lines. It can be verified by finding the Gröbner basis gb_G of $\mathcal{G}$ w.r.t. $Z \succ_{lex} Y \succ_{lex} X$. It consists of four elements $\{g_1, g_2, g_3, g_4\}$, where the first element factors as follows:

$$g_1 = (Z - d_1)(Yd_5 - Zd_4). \tag{4.52}$$

By substituting the factors into the other elements of gb_G, $\mathcal{G}_v$ can be decomposed as the product of two smaller ideals, $\mathcal{M}$ and $\mathcal{N}$, whose varieties correspond to the two transversal lines intersecting all the four observed lines:

$$\begin{aligned}\mathcal{G} &= \mathcal{M} \times \mathcal{N}, \quad \text{where} \\ \mathcal{M} &= \langle Z - d_1, Y - d_3 \rangle, \\ \mathcal{N} &= \langle Y d_5 - Z d_4, X(d_1 d_4 - d_3 d_5) + Y d_2 d_5 - d_1 d_2 d_4 \rangle. \end{aligned} \tag{4.53}$$

In the generic case of P4L, we showed that the one dimensional singularities can be avoided by choosing the lines such that they satisfy (4.44). It is a condition on the discriminant of the quadratic polynomial that appears as the first element of the Gröbner basis of $\mathcal{G}$. Similarly, here, the first polynomial of gb_G in (4.52) is quadratic in Z and its discriminant is $(Y d_5 - d_1 d_4)^2$, which is always non-negative. Hence the two transversal lines given by (4.53) are always real and the singularity cannot be avoided when the four observed lines adhere to the orthogonality and parallelism conditions of this section.

To analyse the remaining singularities, we need to determine the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K})$. To do so, the ideal $\langle \mathcal{H}, \mathcal{K} \rangle$ is considered (it is too large to be displayed here) and its Gröbner basis calculated:

$$\begin{aligned} gb_{HK} = \{ \; & Z\,(-d_5 + Z)\,(Z d_4 - d_3 d_5)\,(-d_1 + Z)\,, \\ & Y d_5 - Z d_4, \\ & -\,d_1 d_2 d_4\,(d_1 d_4 + d_3 d_5 - 2\,d_4 d_5)\,(d_1 d_4 - d_3 d_5) + \\ & (d_1 d_4 + d_3 d_5 - 2\,d_4 d_5)\,(d_1 d_4 + d_3 d_5)\,(d_1 d_4 - d_3 d_5)\,X - \\ & d_2 d_4\left(d_1{}^2 d_4{}^2 - 2\,d_1 d_3 d_4 d_5 - 2\,d_1 d_4{}^2 d_5 + d_3{}^2 d_5{}^2 - 2\,d_3 d_4 d_5{}^2\right) Z + \\ & 4\,d_2 d_4{}^3 Z^3 - 2\,d_2 d_4{}^2\,(d_1 d_4 + d_3 d_5 + 2\,d_4 d_5)\,Z^2 \; \}. \end{aligned} \tag{4.54}$$

The variety of gb_{HK} is zero dimensional with degree 4 (see the files in jorge-gf/thesis-archive [GF22]). It implies that it is made up of 4 points in $\mathbb{C}[X, Y, Z]$ as shown in Fig. 4.4 whose coordinates are as follows:

$$\begin{aligned} C_1 &= \left(0, \frac{d_1 d_4}{d_5}, d_1\right) \; ; \; C_2 = \left(\frac{(d_1 - d_5) d_2 d_4}{d_1 d_4 + d_3 d_5 - 2 d_4 d_5}, d_4, d_5\right); \\ C_3 &= \left(\frac{d_1 d_2 d_4}{d_1 d_4 + d_3 d_5}, 0, 0\right) \; ; \; C_4 = \left(d_2, d_3, \frac{d_3 d_5}{d_4}\right). \end{aligned} \tag{4.55}$$

As mentioned in Section 4.4.2, some or all of these points might lie on the four observed lines or their two singular transversals, which we know for sure belong to the singularity loci. To acknowledge that, the saturation ideal $\mathcal{F}$ of (4.48) can be determined to check if $C_i \in \mathbf{V}(\mathcal{F})$, because only then, C_i is a point outside any of these lines. However, the Gröbner basis of $\mathcal{F}$ yields $\{1\}$. By Hilbert's Nullstellensatz, when the Gröbner basis of an ideal is $\{1\}$, its generators do not have a common solution [CLO13] and hence, $\mathbf{V}(\mathcal{F}) = \emptyset$. Thus, $C_i \notin \mathbf{V}(\mathcal{F})$ for any $i = 1, 2, 3, 4$ implying that these points indeed lie on the four observed lines $(\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4)$ or their transversals $(\mathcal{L}_M, \mathcal{L}_N)$ as shown in Fig. 4.4 for some randomly chosen parameters $d_1, .., d_5$.

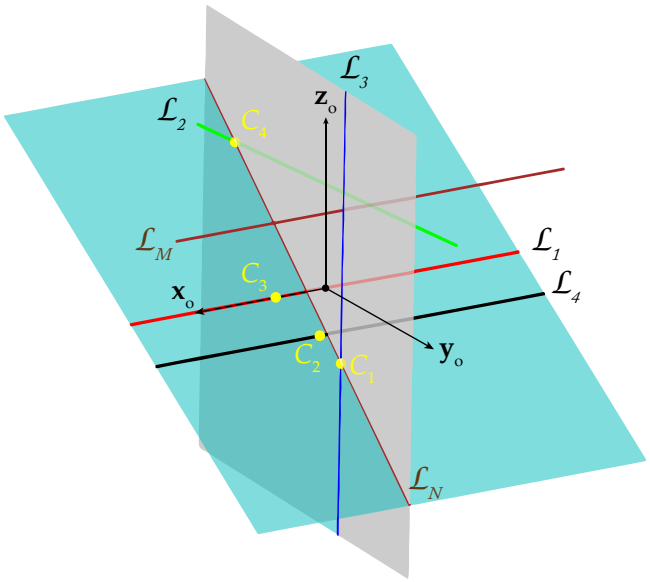


Figure 4.4: Singularities in P4L with orthogonality and parallelism constraints: Four observed lines $\mathcal{L}_i, i = 1, 2, 3, 4$ and their traversals $\mathcal{L}_M$ and $\mathcal{L}_N$.

Proposition 4.6. *For the special case of P4L considered in this section, the only singularities of the interaction matrix occur when C lies on one of the four observed lines or their two transversal lines.*

The results in this section remain valid under any permutation of the four observed lines and hence the analysis is valid for the case of four lines subject to any other similar orthogonality and parallelism constraints. In the following section we study the singularity cases when five lines are observed.

4.5 Singularities in P5L

4.5.1 Parametrization

Let us consider the first three lines defined by the parametrization (4.20) and the fourth one by (4.27). In the same vein, the fifth line is defined using its two points of intersections P_5 and N_5 with the planes $y_o = 0$ and $z_o = 0$, respectively:

$$\overrightarrow{OP_5} = [d_6, 0, d_7]^T, \ \overrightarrow{ON_5} = [r_7, r_8, 0]^T. \tag{4.56}$$

Thus, the direction vector of the fifth line is given by $\mathbf{U}_5 = \overrightarrow{OP_5} - \overrightarrow{ON_5}$. After changing the variables $-r_8 = s_8$ and $d_6 - r_7 = s_7$, we have

$$\overrightarrow{OP_5} = [d_6, 0, d_7]^T, \ \mathbf{U}_5 = [s_7, s_8, d_7]^T. \tag{4.57}$$

As mentioned in Section 4.3.1, the orientation of the camera frame $\mathcal{F}_c$ and of the object frame $\mathcal{F}_o$ is considered the same also for the following analysis.

4.5.2 Singularity analysis

From Section 4.2.3, we know that the rows of the interaction matrix $\mathbf{M}_{(5)}$ associated with the five lines represent a system of Plücker lines and that a basis for this system of lines can be expressed as $\boldsymbol{\xi}_{(5)} = [\boldsymbol{\xi}_{11}^T \ \boldsymbol{\xi}_{12}^T \ \dots \ \boldsymbol{\xi}_{51}^T \ \boldsymbol{\xi}_{52}^T]^T \in \mathbb{R}^{10\times 6}$ with $\boldsymbol{\xi}_{i1}$ and $\boldsymbol{\xi}_{i2}$ given by (4.16).

Let us then consider the ideal $\mathcal{I}_{210}$ generated by the maximal minors of $\boldsymbol{\xi}_{(5)}$, which in this case forms a system of 210 polynomials p_i in the variables X, Y, Z and the parameters $\boldsymbol{\eta} = \{s_1, \dots, s_8, d_1, \dots, d_7\}$: $\mathcal{I}_{210} = \langle p_1, \dots, p_{210} \rangle$.

Following a similar analysis of Section 4.4, the 210 minors can be divided into three categories. Of them, there are 55 that are equal to zero, because there are 55 submatrices containing four or five ideal lines (lines at infinity). These matrices always have a rank at most 5 since only three lines at infinity can be linearly independent at a time.

The second category are the determinants of block triangular matrices, composed of row vectors that represent three affine lines and three ideal lines. There are $\binom{5}{3} \times \binom{5}{3} = 100$ of them. Let these minors generate a subideal $\mathcal{I}_{100} \subseteq \mathcal{I}_{210}$. The generators of $\mathcal{I}_{100}$ are of the form $p_i = G_{ijk}H_{lmn}$, that are the products of the polynomials in (4.32), with the indices $\{i, j, k\}$ and $\{l, m, n\}$ ranging all triplets of numbers in $\{1, 2, 3, 4, 5\}$. Therefore, this ideal is the product of two smaller ideals: $\mathcal{I}_{100} = \mathcal{G} \times \mathcal{H}$, generated by 10 polynomials each:

$$\mathcal{G} = \langle G_{123}, \dots, G_{235} \rangle, \quad \mathcal{H} = \langle H_{123}, \dots, H_{235} \rangle. \tag{4.58}$$

The generators of $\mathcal{G}$ and $\mathcal{H}$ are too long to be given here, but $\mathcal{G}$ (resp. $\mathcal{H}$) is generated by polynomials of degree 2 (resp. 3) in $\mathbb{Q}[X, Y, Z]$, describing the singularity hyperboloids (resp. cubic surfaces) of P3L (see Section 4.3.2).

The remaining 55 minors with degree 5 each in $\{X, Y, Z\}$ constitute the last category. Let them generate a subideal $\mathcal{K}_{55} \subseteq \mathcal{I}_{210}$.

As before, we deduce that the solution set of the polynomials in $\mathcal{I}_{210}$ is contained in a larger variety which is the union of two varieties (see (4.38)):

$$\mathbf{V}(\mathcal{I}_{210}) = (\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K}_{55})) \cup (\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})) \,. \tag{4.59}$$

Our strategy will be to use the geometrical interpretation of previous sections wherever possible or else to use Gröbner bases computations and multivariate polynomial division to analyse the varieties $(\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K}_{55}))$ and $(\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55}))$ separately. Again, all mathematical derivations can be followed on the repository jorge-gf/thesis-archive [GF22].

Analysis of the variety $\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K}_{55})$

Let $gb_G = \langle g_1, \dots, g_s \rangle$ be a Gröbner basis for the ideal $\mathcal{G}$ with respect to the aforementioned lexicographical monomial ordering. We are again treating the parameters $\boldsymbol{\eta} = \{d_1, \dots, d_7, s_1, \dots, s_8\}$ as variables here; that is, we are considering the polynomials in the ring $\mathbb{Q}[X, Y, Z, \boldsymbol{\eta}]$. Then, we can compute the normal form of the polynomials $p_i \in \mathcal{K}_{55}$

with respect to this basis to find $\overline{p_i}^{\langle g_1,\ldots,g_s\rangle}$ for all i. We find $p_i = 0$ for all i, implying that $(\mathbf{V}(\mathcal{G}) \cap \mathbf{V}(\mathcal{K}_{55})) = \mathbf{V}(\mathcal{G})$. We already know from Theorem 4.3 that $\mathbf{V}(\mathcal{G})$ should be a line that intersects the five observed lines. We cannot always find a line that intersects the given five lines unless they belong to a singular linear line complex [PW01]. Thus, there must be a condition on the parameters such that $\mathbf{V}(\mathcal{G}) \neq \emptyset$. To find it, we consider a matrix with rows consisting of the Plücker coordinates of the observed lines $[\mathbf{U}_i,\ \overrightarrow{OP_i} \times \mathbf{U}_i]$ whose moment vectors are defined by considering the point Q in (4.16) as the origin of the object frame O:

$$\mathbf{L}_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ s_1 & s_2 & 0 & d_1 s_2 & -d_1 s_1 & 0 \\ d_2 & s_3 & s_4 & -s_4 d_3 & s_4 d_2 & d_2 d_3 - d_2 s_3 \\ s_5 & d_4 & s_6 & d_4 d_5 - d_4 s_6 & -s_5 d_5 & s_5 d_4 \\ s_7 & s_8 & d_7 & s_8 d_7 & d_6 d_7 - d_7 s_7 & -s_8 d_6 \end{bmatrix}. \tag{4.60}$$

A line intersects the observed five lines only if the kernel $\mathbf{k} = [k_1, k_2, k_3, k_4, k_5, k_6]^T$ of the matrix $\mathbf{L}_5$ satisifies the Plücker relation $k_1 k_4 + k_2 k_5 + k_3 k_6 = 0$. The first row of $\mathbf{L}_5$ imposes $k_1 = 0$. Eliminating k_i from the remaining four equations $\mathbf{L}_5 \mathbf{k} = 0$ and the Plücker relation leads to a polynomial h of degree 13 solely in terms of parameters $\boldsymbol{\eta}$. If this polynomial h is *not* zero, then there is no line transversal to $\mathcal{L}_1, \ldots, \mathcal{L}_5$, and therefore, that there are no one-dimensional singularities.

Proposition 4.7. *There are no one dimensional singularities for P5L when the five observed lines are generic. A polynomial condition on the parameters defining the configuration of the lines must be satisfied for a transversal to exist such that the incidence of the camera centre C on it leads to a singularity.*

The results of this section are substantiated through an example in Section 4.5.3.

Analysis of the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})$

The large number of variables and polynomials made it impossible to compute a Gröbner basis $gb_H = \{h_1, \ldots, h_t\}$ for the ideal generated by $\mathcal{H}$ in $\mathbb{Q}[\boldsymbol{\eta}, X, Y, Z]$, with the parameters as free-variables. Hence, we cannot evaluate generically (for arbitrary parameter values) the intersection of varieties $\mathcal{H}$ and $\mathcal{K}_{55}$, by reducing the minors $p_i \in \mathcal{K}_{55}$, $\overline{p_i}^{\langle h_1,\ldots,h_3\rangle} = f_i$ with respect to such a basis. However, we know that a singularity of P5L implies a singularity of each subset of four out the five lines. Therefore we know that the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})$ will constitute at most a finite number of points.

In Appendix B, we give an analysis of the $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})$ for some arbitrary values of the parameters $\boldsymbol{\eta}$. For a generic specialization (from several tests performed), we find that the intersection of both varieties is almost always empty. This lead us to the following statement, which still needs formal computational proof

Statement 4.1. *For P5L, there exist a polynomial, in the space of the parameters describing the configuration, such that specializing the parameters outside its zero-set, there exist no isolated (zero-dimensional) singularities.*

4.5.3 Singularities of P5L for an example configuration

The singularities in P5L are determined for lines whose Plücker coordinates are arbitrarily chosen as follows:

$$\begin{aligned} &s_1 = 4,\ s_2 = -5,\ s_3 = 7,\ s_4 = 3,\ s_5 = -2,\ s_6 = 13, s_7 = -11, s_8 = 6, \\ &d_1 = 2,\ d_2 = 3,\ d_3 = 5,\ d_4 = 1,\ d_5 = 7, d_6 = -4,\ d_7 = 11. \end{aligned} \tag{4.61}$$

The Gröbner basis gb_G of $\mathcal{G}$ w.r.t. $Z \succ_{lex} Y \succ_{lex} X$ is $\langle 1 \rangle$. By Hilbert's Nullstellensatz, $\mathbf{V}(\mathcal{G}) = \emptyset$.

If we choose the parameters according to (4.61) except d_7, which is chosen such that the parameters satisfy $h = 0$ in Section 4.5.2:

$$d_7 = \frac{128893236}{7630285} - \frac{24\sqrt{8508173023861}}{7630285},$$

then, calculating the Gröbner basis leads to:

$$\begin{aligned} gb_G = \{&2166Y + \left(\sqrt{448189} + 583\right) Z,\ -2166\sqrt{448189} - 2822298 \\ &+ 3822990X + \left(-329\sqrt{448189} + 587953\right) Z\}. \end{aligned}$$

It is the equation of a line intersecting the five observed lines.

The Gröbner basis gb_H of $\mathcal{H}$ w.r.t. $Z \succ_{lex} Y \succ_{lex} X$ contains 5 elements $\{h_1, h_2, h_3, h_4, h_5\}$. As proposed in Section 4.5.2, the normal forms can be calculated as $\overline{p_i}^{\langle h_1,\ldots,h_3\rangle} = f_i, i = 1, \ldots, 55$. The residues f_i do not vanish. Therefore, he Gröbner basis of the ideal $\langle \mathcal{H}, \mathcal{K}_{55} \rangle$ is calculated and it turns out to be $\langle 1 \rangle$. Hence, for a generic choice of parameters, there are no singularities in P5L.

Like P4L, it applies to P5L as well that, often, the observed lines in an environment are constrained with orthogonality and/or parallelism. One of these special cases is considered in the next section and the singularities are determined with the proposed approach without specializing any parameters.

4.5.4 Singularities in P5L with orthogonality and parallelism

As a continuation of Section 4.4.4, let us consider a fifth line $\mathcal{L}_5$, which is assumed to be collinear with axis $\mathbf{y_o}$. This way, we have three orthogonal lines $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$, a line $\mathcal{L}_4$ parallel to $\mathcal{L}_1$, and a line $\mathcal{L}_5$ parallel to $\mathcal{L}_2$. The location of lines $\mathcal{L}_1$ to $\mathcal{L}_4$ relative to frame $\mathcal{F}_o$ is still given by (4.51), and we parametrize $\mathcal{L}_5$ by

$$\overrightarrow{OP_5} = [d_6,\ 0,\ d_7]^T,\ \mathbf{U}_5 = [0,\ 1,\ 0]^T. \tag{4.62}$$

As before, if the line $\mathcal{L}_5$ was instead given parallel to line $\mathcal{L}_3$, these parametrization will still be valid upon a redefinition of the object frame $\mathcal{F}_o$ and the renaming of the lines.

Following the analysis done in the preceding section, the varieties $\mathbf{V}(\mathcal{G})$ and $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})$ will be analysed separately. In this context, the ideal $\mathcal{G}$ in (4.58) is calculated as follows:

$$\begin{aligned}\mathcal{G} = \langle &-XYd_5 + XYd_7 - XZd_3 + XZd_4 + Xd_3d_5 - Xd_4d_7 + YZd_2 - YZd_6 - Yd_2d_7 + Yd_5d_6 \\ &- Zd_2d_4 + Zd_3d_6 + d_2d_4d_7 - d_3d_5d_6, (-d_5 + Z)\,(Xd_1 - Xd_7 + Zd_6 - d_1d_6)\,, \\ &-(-d_2 + X)\,(Xd_1 - Xd_7 + Zd_6 - d_1d_6)\,, XYd_1 - XYd_5 - XZd_3 + XZd_4 - Xd_1d_4 \\ &+ Xd_3d_5 + YZd_2 - Yd_1d_2 - Zd_2d_4 + d_1d_2d_4, (-d_7 + Z)\,(Yd_5 - Zd_4)\,, -XYd_7 + XZd_3 \\ &- YZd_2 + YZd_6 + Yd_2d_7 - Zd_3d_6, (-d_3 + Y)\,(Yd_5 - Zd_4)\,, -Z\,(Xd_1 - Xd_7 + Zd_6 - d_1d_6)\,, \\ &-(-d_1 + Z)\,(Yd_5 - Zd_4)\,, XYd_1 - XZd_3 + YZd_2 - Yd_1d_2\rangle.\end{aligned} \tag{4.63}$$

We expect that the one dimensional singularity corresponding to $\mathbf{V}(\mathcal{G})$ must be the transversal that intersects the five observed lines according to Theorem 4.3. However, we know from Section 4.5.2 that the parameters used to define the five observed lines must satisfy a condition for this transversal line to exist. As before, we can determine it by imposing the Plücker relation on the kernel of the matrix whose rows are the Plücker coordinates of the observed lines $[\mathbf{U}_i,\ \overrightarrow{OP_i} \times \mathbf{U}_i]$ parametrized by (4.51) and (4.62). This condition leads to the following polynomial in terms of the parameters that should be zero.

$$h(\boldsymbol{\eta}) = d_5\,(d_1 - d_7)\,(d_1d_2d_4 - d_1d_4d_6 - d_2d_4d_7 + d_3d_5d_6)\,. \tag{4.64}$$

This polynomial can also be derived by finding the Gröbner basis gb_G of $\mathcal{G}$ in (4.63) w.r.t. the ordering $Z \succ_{lex} Y \succ_{lex} X \succ_{lex} d_1 \succ_{lex} \ldots \succ_{lex} d_7$. It consists of ten elements of which the first element is exactly $h(\boldsymbol{\eta})$. We look for conditions when $h = 0$ so that $\mathbf{V}(\mathcal{G}_v) \neq \emptyset$.

When the first factor of h vanishes, i.e. $d_5 = 0$, we get the line as the variety of the following ideal:

$$\mathcal{M}_1 = \langle Z, X - d_2\rangle. \tag{4.65}$$

When $d_1 - d_7 = 0$, we get

$$\mathcal{M}_2 = \langle Z - d_1, Y - d_3\rangle. \tag{4.66}$$

Finally, when the third factor of h vanishes, we have

$$\mathcal{M}_3 = \langle Yd_5 - Zd_4, X(d_1 - d_7) + Zd_6 - d_1d_6\rangle. \tag{4.67}$$

Figure 4.5 shows the third case where a line $\mathbf{V}(\mathcal{M}_3)$ intersects all five observed lines.

To analyse the remaining singularities, we need to determine $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})$. It amounts to analysing the ideal $\langle \mathcal{H}, \mathcal{K}_{55}\rangle$ (too large to be displayed here). Its Gröbner basis yields $\{1\}$ implying that the variety is empty and hence there are no isolated singularities in the generic case. However, it is possible that there are special relative configurations of the five lines for which $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55}) \neq \emptyset$. As mentioned in Section 4.5.2, it was not possible to find these configurations of the observed lines when they are generic, due to the computational complexity. Nonetheless, in this context, the constraints on the observed lines reduce the complexity and hence we are able to find the conditions on the parameters d_i, $i = 1, \ldots, 7$

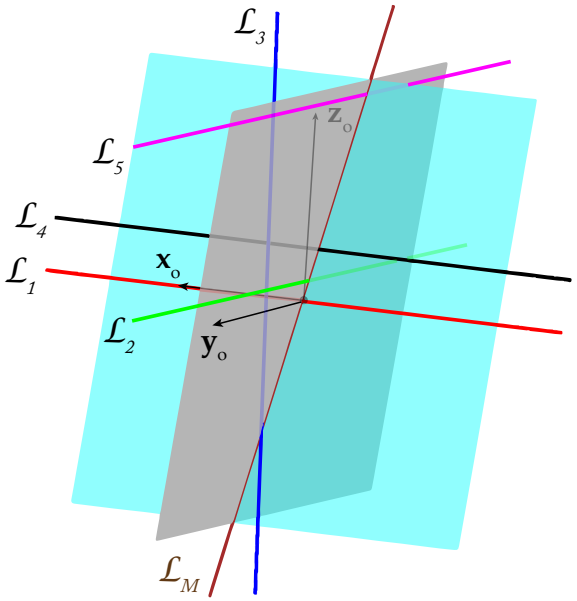


Figure 4.5: Singularities in P5L with orthogonality and parallelism constraints: Five observed lines $\mathcal{L}_i, i = 1, 2, 3, 4, 5$ and their traversal $\mathcal{L}_M$.

such that $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55}) \neq \emptyset$. To do so, $\mathbf{V}(\mathcal{H})$ is first calculated by finding the Gröbner basis gb_H of $\mathcal{H}$ w.r.t. the monomial order $Z \succ_{lex} Y \succ_{lex} X$. The basis gb_H contains only 3 linear terms and is of dimension 0; therefore, its variety is a point:

$$\begin{aligned} X &= \frac{d_1 d_2 d_4 + d_1 d_4 d_6 - d_2 d_4 d_7 - d_3 d_5 d_6}{2\,(d_1 - d_7)\,d_4}, \\ Y &= -\frac{d_1 d_2 d_4 - d_1 d_4 d_6 - d_2 d_4 d_7 - d_3 d_5 d_6}{2 d_5 d_6}, \\ Z &= -\frac{d_1 d_2 d_4 - d_1 d_4 d_6 - d_2 d_4 d_7 - d_3 d_5 d_6}{2 d_4 d_6}. \end{aligned} \tag{4.68}$$

For this point to be a singularity, it should also belong to the variety $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55})$ and hence it should absolutely lie in the variety $\mathbf{V}(\mathcal{K}_{55})$. Substituting the values of X, Y, Z in $\mathcal{K}_{55}$ leaves 36 non-zero polynomials solely in terms of parameters $d_1, \ldots, d_7$ (see the Maple file). They constitute the conditions for $\mathbf{V}(\mathcal{H}) \cap \mathbf{V}(\mathcal{K}_{55}) \neq \emptyset$.

Proposition 4.8. *For five lines bound by orthogonality and parallelism as in (4.62), the P5L problem has generically no singularities. The singularities may appear as a line and/or as a point for some special relative configurations of the five lines.*

Again, the results in this section remain valid under any permutation of the five observed lines and hence the analysis entails singularities in the case of five lines subject to any other orthogonality and parallelism constraints.

Table 4.2: Different cases of singularities in P4L and P5L.

Cases	Subcases	Singularity configurations
P3L	Three skew lines	C lies on the hyperboloid of one sheet uniquely defined by the observed lines or on a cubic surface that contains the three lines
P4L	Four lines in a hyperbolic congruence	C lies on two affine lines intersecting the four observed lines and up to 10 real points
	Four lines in a parabolic congruence	C lies on an affine line intersecting the four observed lines and up to 10 real points
	Four lines in an elliptic congruence	Up to 10 real points
	With orthogonality and parallelism constraints	C lies on the two affine lines intersecting the four observed lines
P5L	Five lines in a regular linear line complex	No singularities
	Five lines in a singular linear line complex	C lies on the line intersecting the five observed lines
	With orthogonality and parallelism constraints	No singularities; Special case: A line and/or a point

All cases corresponding to the singular configurations when observing three, four and five lines are summarized in Table 4.2.

4.6 Simulation results

This section illustrates the impact that the exposed singularities have on the behaviour of basic Image-Based Visual Servoing and pose determination algorithms.

From the results of Sections 4.4 and 4.5, we designed a series of simulated experiments where a free-flying camera is controlled in visual servoing from the observation of lines. First, we computed the singularity conditions for an example configuration of the lines. Then, we performed a numerical simulation of visual servoing in the vicinity of singular positions, using the visual servoing library ViSP [MSC05], to evaluate the impact that the loss of rank of the interaction matrix has in the controllability of the camera.

Another set of tests show the result of classical algorithms for pose estimation from lines when the camera is controlled in open-loop (the camera motion is specified beforehand) near any of the singularities. The files used to perform the simulations are available in the github repository jorge-gf/thesis-archive [GF22].

The results show that, at, or near a singularity of the interaction matrix, there is a significant loss of controllability in the visual servo scheme, as well as a poorer accuracy for the pose localisation algorithms. In particular, we observe that near point singularities,

Table 4.3: Initial and desired positions relative to a point on the singularity line $\mathcal{L}_M$.

	ΔX	ΔY	ΔZ	Note
Desired	0.30	−0.30	0.30	Target end position $\mathbf{s}^*$
Start 1	0.20	0.30	0	Near to singularity.
Start 2	−0.30	0.30	−0.30	Opposed to desired.
Start 3	0	0.30	0.40	Near to singularity.
Start 4	0.10	−0.60	0.10	Away from singularity.

i.e. isolated locations for the camera for which the interaction matrix is singular, this effect is relatively local and not destabilizing. On the other hand, close to the line singularities, we observe that the controlled system becomes highly unstable, and that the errors in the reconstructed pose grow several orders in magnitude.

4.6.1 Singularities in P4L

Let us consider four lines $\mathcal{L}_1, \ldots, \mathcal{L}_4$, defined by a point and a direction (4.20), and let us specialize the parameters arbitrarily as follows:

$$\begin{aligned} &s_1 = 4,\ s_2 = -5,\ s_3 = 7,\ s_4 = 3,\ s_5 = -2,\ s_6 = 13, \\ &d_1 = 2,\ d_2 = 3,\ d_3 = 5,\ d_4 = 1,\ d_5 = 7. \end{aligned} \tag{4.69}$$

We obtained the singularity conditions for this configuration as described in the previous sections. The computations are included in the Section 4.4.3. The singularities consist of 6 isolated points, deriving from Proposition (4.5), plus two lines $\mathcal{L}_M$, $\mathcal{L}_N$, transversal to $\mathcal{L}_1, \ldots, \mathcal{L}_4$, that come from Theorem (4.4), and which have the following Plücker coordinates:

$$\begin{aligned} \mathcal{L}_M &= [0.0830,\ 0.4989,\ -0.8627,\ 0,\ -0.9641,\ -0.5575], \\ \mathcal{L}_N &= [0.2067,\ -0.03902,\ -0.9776,\ 0,\ -0.3509,\ 0.01401]. \end{aligned} \tag{4.70}$$

VS towards a desired pose near singularity line $\mathcal{L}_M$

We selected four initial camera positions in the proximity of a point $\mathcal{P}_M$ with coordinates $[0.3962\ \ -4.337\ 7.50]^T$, which lies on the singularity line $\mathcal{L}_M$. From each starting point, we attempt to control the camera towards a desired position which is always the same. The coordinates of the start and target positions are defined relative to $\mathcal{P}_M$, and are shown in Table 4.3 along with a commentary on the choice of points. The camera orientations were chosen such that the four lines are clearly visible from all locations.

At each iteration, the controller tries to minimize an error function $\mathbf{s}(t) - \mathbf{s}^*$. The vector of features $\mathbf{s} = [\mathbf{l}_1'^T, \ldots, \mathbf{l}_4'^T]^T \in \mathbb{R}^{12}$ is composed of the coordinates l'_{xi}, l'_{yi} and l'_{zi} for each projected line $1 \leq i \leq 4$, while $\mathbf{s}^*$ contains the values of the features at the desired position. In order to achieve an exponential decrease of the error, the velocity input to the camera is

$$\boldsymbol{\tau}_c = -\lambda\, \mathbf{M}^+_{(4)}\, (\mathbf{s}(t) - \mathbf{s}^*)\,, \tag{4.71}$$

where $\mathbf{M}_{(4)}^{+}$ is the Moore-Penrose pseudoinverse of the interaction matrix (4.8) and λ is a gain factor which was set to 0.1. For these first simulations, we assume that all the parameters appearing in the matrix $\mathbf{M}_{(4)}$ are known and hence we can always obtain a perfect estimate of its pseudoinverse $\mathbf{M}_{(4)}^{+}$. Note that no noise was added to the visual data, and hence all the instabilities are due uniquely to the determinant of the interaction matrix vanishing at a singularity.

Figure 4.6 displays the camera trajectories starting from each initial position. A normal behaviour is achieved from Starts 1 and 3: the camera describes an almost straight line towards the desired pose, and the magnitude of the error vector decreases exponentially (see Fig. 4.7).

Start 2 is located opposite from the desired position relative to the point $\mathcal{P}_M$. The camera reaches the target point eventually, but it diverges along the singularity line as it approaches it (see Fig. 4.6). The inverse of the condition number κ of $\mathbf{M}_{(4)}$, shown in Fig. 4.7, reaches almost zero in the vicinity of the singularity, which means that the system in (4.71) is ill-conditioned and, as a result, the controller produces very high velocity commands causing instability. Figure 4.8 compares the velocity input profiles for Starts 1 and 2 throughout the simulation. The velocity inputs from Start 2 are two orders of magnitude higher than those produced in a stable situation. Note that although the distance to the desired point increases during the undesirable motion (Fig. 4.9), the magnitude of the error $||\mathbf{s} - \mathbf{s}^*||$ remains approximately constant (Fig. 4.7).

Start 4 is located slightly further away from the singularity line, and closer to the desired position. The trajectory converges but the camera is again subjected to considerably high velocities and does not approach the target monotonically (see Figs. 4.6 and 4.9). This illustrates that the impact of the singularities is not limited to the trajectories that directly cross a singular point, but instead, that there is an area of influence in the vicinity of a singularity where the behaviour can be affected by the high condition number of the interaction matrix.

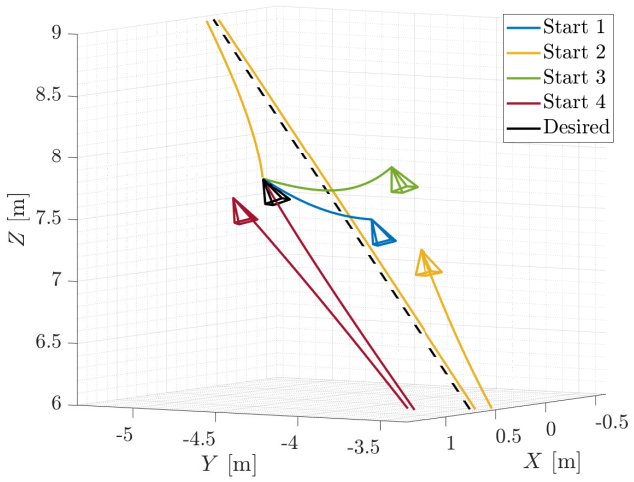


Figure 4.6: Visual servoing using four image lines, starting from four initial poses (coloured). The desired end pose is in black. The black dashed line is the singularity line $\mathcal{L}_M$ that intersects the four observed lines.

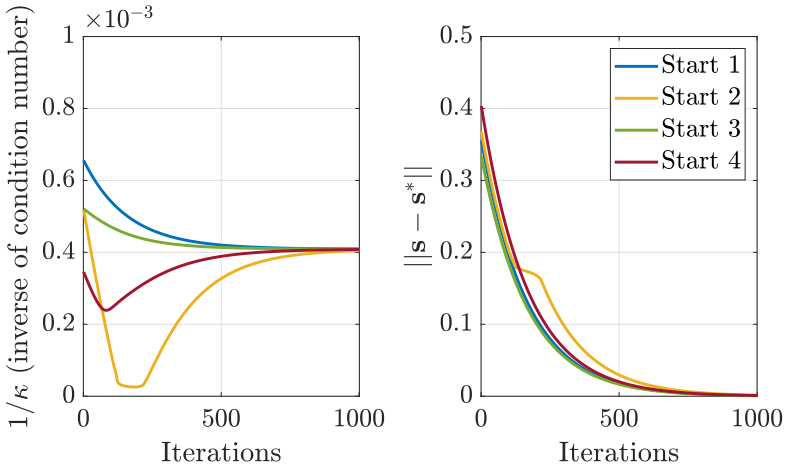


Figure 4.7: Inverse of the condition number κ of the interaction matrix $\mathbf{M}_{(4)}$ (left) and norm of the error vector $||\mathbf{s} - \mathbf{s}^*||$ (right).

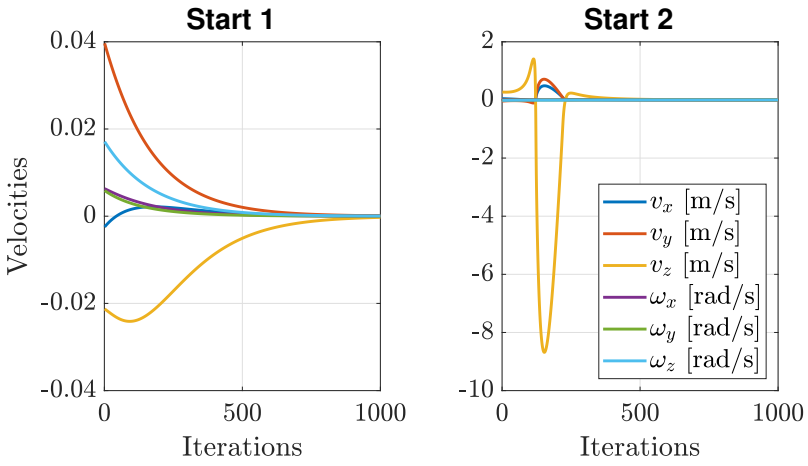


Figure 4.8: Velocity inputs $\boldsymbol{\tau}_c$ for the camera in a stable control scheme (left), and when crossing a singularity (right).

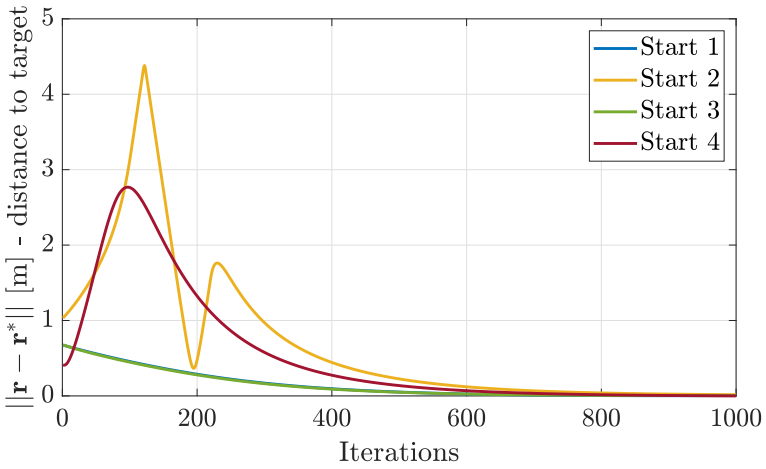


Figure 4.9: Distance to the target during the visual servo.

Trajectory following across singularity line $\mathcal{L}_N$

With the same four lines defined by the parameters (4.69), a simple trajectory along a cubic curve defined by $\mathbf{r}_1^*(t) = [X_1^*(t)\ Y_1^*(t)\ Z_1^*(t)]^T$ with

$$\begin{aligned} X_1^*(t) &= -s(t)^3 - 1.7551, \quad Y_1^*(t) = s(t) + 0.3992 \\ Z_1^*(t) &= 0.7s(t) + 10.0. \end{aligned} \tag{4.72}$$

for $s(t) = 0.04t - 1$ and $0 \leq t \leq 50$ s, was designed to cross the singularity line $\mathcal{L}_N$ at $s(t) = 0$. A second, very similar trajectory $\mathbf{r}_2^*(t) = [X_2^*(t)\ Y_2^*(t)\ Z_2^*(t)]^T$ with

$$\begin{aligned} X_2^*(t) &= -s(t)^3 - 3.2551, \quad Y_2^*(t) = s(t) + 0.3992 \\ Z_2^*(t) &= 0.7s(t) + 10.30. \end{aligned} \tag{4.73}$$

should not cross any singular points.

Trajectory following can be performed using visual servoing by introducing a time-dependent vector $\mathbf{s}^*(t)$ of desired visual features in the control law (4.71), which can be computed by forward projection (4.2) of the 3D line coordinates in the camera frame $\mathcal{F}_c$ as the camera moves along the trajectory.

As before, we assume that the parameters in the interaction matrix $\mathbf{M}_{(4)}$ can be measured such that we can obtain an estimate of its pseudoinverse $\mathbf{M}_{(4)}^+$ to use in the control law (4.71). However this time we added white Gaussian noise of standard deviation $\sigma = 2 \cdot 10^{-3}$ to the 3D coordinates of the observed lines, in order to simulate the impact of errors in the measurements. The gain factor λ was set to 1.

The camera behaviour is shown in Fig. 4.10. For trajectory $\mathbf{r}_2^*$ (4.73), away from the singularity line, the camera follows the prescribed path with relative accuracy. The velocity inputs are mild and there is a small, approximately constant tracking error of about 0.1 (Fig. 4.11), which could be reduced with a more sophisticated controller, for example by introducing an integral term in the control law (4.71).

For the trajectory $\mathbf{r}_1^*$ (4.72), the camera is unable to follow the desired path accurately in the vicinity of the singularity line. Around $s(t) = 0$, the velocity commands become very high in magnitude, inducing instability, and the translation error becomes as high as 1 (Fig. 4.11).

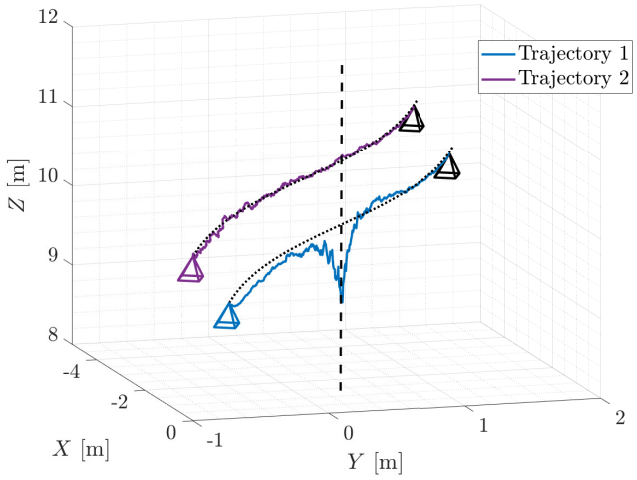


Figure 4.10: VS from four image lines: Trajectory described by the camera when controlling it along a prescribed path (thin dotted line). The control becomes unstable along the trajectory that crosses the singularity line $\mathcal{L}_N$ (black dashed line).

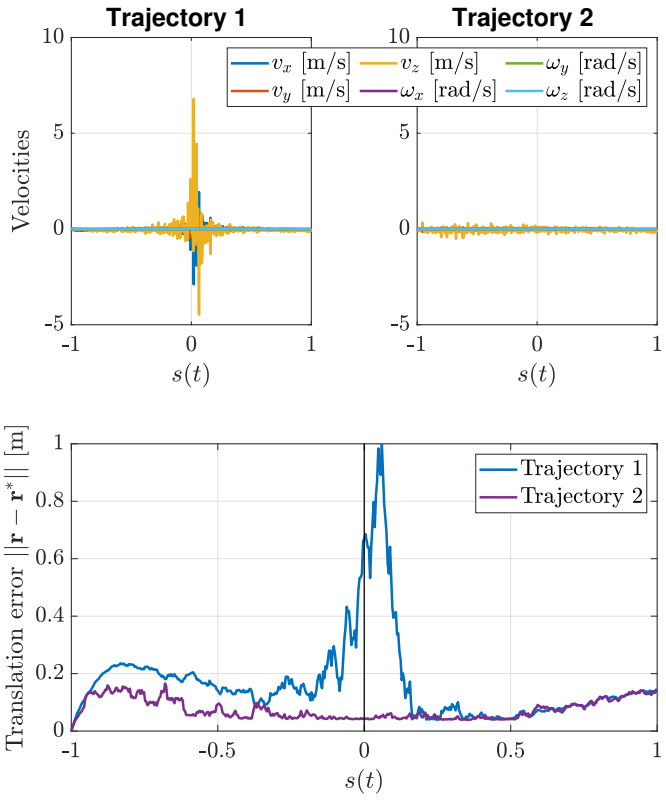


Figure 4.11: Camera velocity inputs $\boldsymbol{\tau}_c$ (top) and position error $||\mathbf{r}(t) - \mathbf{r}^*(t)||$ (bottom) along the trajectories. The vertical step in the bottom figure indicates where Trajectory 1 crosses the singularity line $\mathcal{L}_N$.

VS around the isolated point singularities

For the configuration (4.69), there also exist 6 isolated points $\mathcal{P}_i$ such that a singularity occurs when the camera is on any of them. They are the consequence of Proposition 4.5, and they have the following coordinates (see Section 4.4.3):

$$\begin{aligned}
\mathcal{P}_1 &= [-9.858 \;\; -2.473 \;\; -1.841],\\
\mathcal{P}_2 &= [0.0541 \;\; 0.0092 \;\; 1.8422],\\
\mathcal{P}_3 &= [-0.3203 \;\; 0.0105 \;\; 0.2205],\\
\mathcal{P}_4 &= [1.0113 \;\; 0.7947 \;\; -0.8850],\\
\mathcal{P}_5 &= [0.9387 \;\; 0.5681 \;\; -2.0225],\\
\mathcal{P}_6 &= [65.09 \;\; -96.57 \;\; -0.03639].
\end{aligned} \tag{4.74}$$

Another point $\mathcal{P}_0$ is chosen arbitrarily and away from any singularities:

$$\mathcal{P}_0 = [-8.858 \;\; -2.473 \;\; -1.841]. \tag{4.75}$$

For each of these locations, we simulated a trajectory with the shape of a *quadrifolium* centred at $\mathcal{P}_i$, given by the following equations:

$$\begin{aligned}
&X^*(t) = \mathcal{P}_{ix}, \qquad Y^*(t) = \mathcal{P}_{iy} + 0.3\cos(s)\cos(2s),\\
&Z^*(t) = \mathcal{P}_{iz} + 0.3\sin(s)\cos(2s).
\end{aligned} \tag{4.76}$$

where $\mathcal{P}_{ix}$, $\mathcal{P}_{iy}$, $\mathcal{P}_{iz}$ are the coordinates of each point, and $s = 2\pi\, t/20$ with $0 \leq t \leq 20$ s, and we applied the control law (4.71) with $\lambda = 5$. Once again, Gaussian noise of standard deviation $\sigma = 10^{-3}$ was added to the Plücker coordinates of the lines to simulate measurement errors.

In Fig. 4.12 we show the results for the trajectory centred at the first of these points $\mathcal{P}_1$. The translation error along this trajectory is compared in Fig. 4.13 with that around $\mathcal{P}_0$, which does not come near a singularity. An oscillating tracking error is present in both cases due to the delay of the camera position relative to the desired point at a given time. However the presence of the singularity in the first trajectory results in destabilizing velocity commands as the camera approaches the centre of the quadrifolium (for $s = \frac{\pi}{4} + n\frac{\pi}{2}$), as shown in Fig. 4.13. As a consequence, a significantly larger deviation occurs around this point.

The maximum translation errors for all the experiments (4.74), displayed in Fig. 4.14, occur always when the camera approaches the singularity point. Meanwhile the difference between the maximum and median errors indicate that the greatest part of the trajectory is completed with relative accuracy.

A particularly large error occurs in the example around $\mathcal{P}_2$. In this case, the point of singularity $\mathcal{P}_2$ is located very near both the singularity line $\mathcal{L}_M$ (see Section 4.6.1) and one of the observed lines $\mathcal{L}_2$ (we recall that the camera is at a singularity when it lies on $\mathcal{L}_i$ because it loses visibility of the line). When the camera approaches point $\mathcal{P}_2$, it diverges and is pulled towards the unstable regions around $\mathcal{L}_2$ and $\mathcal{L}_M$, resulting in a very large translation error.

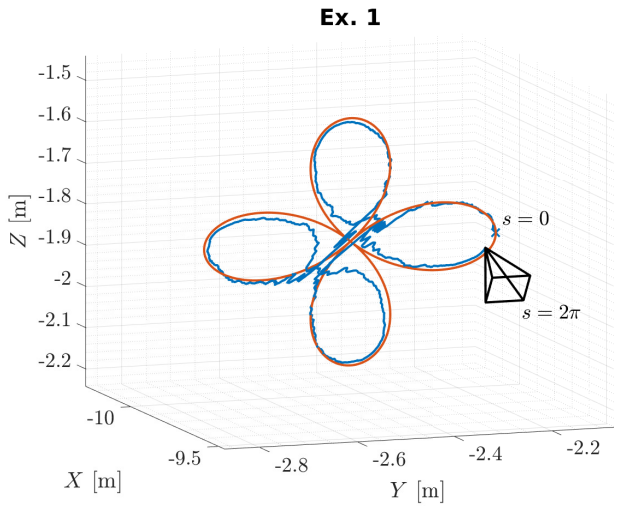


Figure 4.12: Visual servoing along a trajectory with the shape of a quadrifolium (red) centred at the singularity point $\mathcal{P}_1$. The true end camera position is drawn in black. A large translation error occurs every time the camera approaches the singularity.

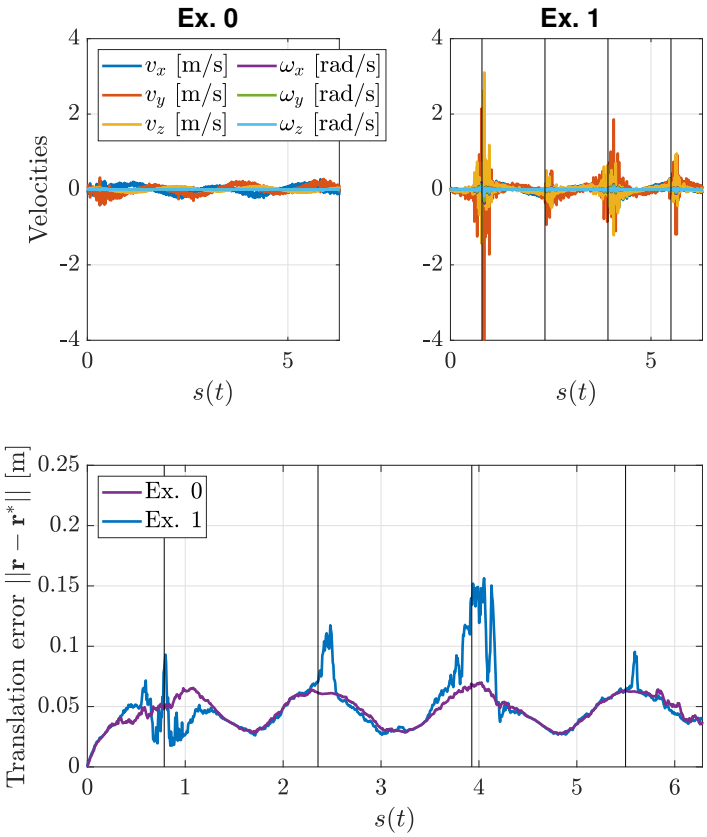


Figure 4.13: Velocity inputs $\boldsymbol{\tau}_c$ (top) and translation error $||\mathbf{r} - \mathbf{r}^*||$ (bottom) during experiments 0 and 1. The vertical steps indicate where the trajectory in Ex. 1 traverses the singularity point $\mathcal{P}_1$.

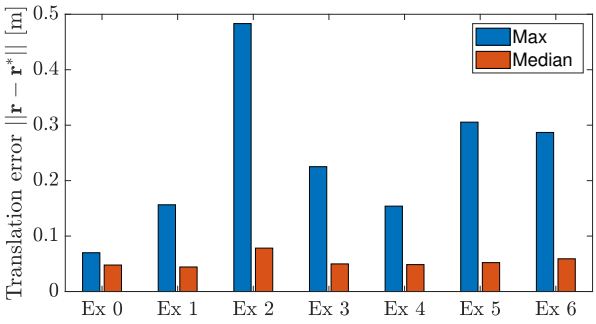


Figure 4.14: Maximum and median error along the quadrifolium trajectory for all experiments.

Pose Estimation along a quadrifolium trajectory

This section illustrates the impact that the singularities have when performing pose estimation in their neighbourhood. Typically, pose computation algorithms can be classified in *iterative* and *non-iterative methods*. Iterative methods are usually more efficient and accurate than the non-iterative ones but, in contrast to them, they require the estimated pose to be initialized and their convergence is very sensitive to a bad initialization.

For the following results we used our own implementation of the classical (non-iterative) Robust Perspective-n-Line (RPnL) algorithm [ZXLK12], that combines several classical methods for the solution of P3L [WXCY19], as well as an improved version, the Accurate Subset-based PnL (ASPnL) [XZCK16], regarded as one of the state-of-the-art direct solvers for pose estimation from n lines, and which is available as open-source code. Both RPnL and ASPnL solve the P3L problem for $(n-2)$ different triplets of lines and then select the solution that yields the smallest reprojection error.

The previous methods are best suited for small sets of lines with no outliers (they assume there are no feature mismatches). We consider here $n = 4$, the minimal number of lines for which the pose estimation problem has a unique solution, in order to test the behaviour of pose estimation in the vicinity of the exposed singularities. The pose computed from the direct methods (RPnL and ASPnL) can be used as an initial estimate to be refined using a first-order iterative solver. Here we use Virtual Visual Servoing (VVS) ([MC02]), implemented in ViSP [MC02], which minimizes the reprojection error of the lines by performing visual servoing on a virtual camera such that the desired image matches the image recorded by the real camera.

We consider four lines in the same configuration used in Section 4.6.1, defined by a point and a direction (4.20) and with the parameters fixed as in (4.69). Three experiments were performed, based on the three points whose coordinates are shown in Table 4.4. Centred at each of these, we simulated an open-loop trajectory, defined thereafter, along which the pose computation methods were assessed. For Example 1 from Table 4.4 a generic point far from any singularities was chosen as a benchmark for the efficiency of the pose estimation algorithms. Example 2 corresponds to the point singularity P_1 in (4.74) - we demonstrate

Table 4.4: Point coordinates used for simulations of pose determination.

Example	Coordinates	Note
1	[5.0 2.0 3.0]	Away from singularities.
2	[−9.858 − 2.473 − 1.841]	Isolated point singularity.
3	[0.7809 − 2.024 3.50]	On singularity line $\mathcal{L}_M$.

here only the behaviour of pose estimation near one of the points in (4.74) because in practice the results are very similar around the other five isolated singularities. Finally, the point for Example 3 lies on the singularity line $\mathcal{L}_M$.

In all three cases the prescribed trajectory has the shape of a *quadrifolium* or four-leaved clover, defined by (4.76) - for Example 2 in Table 4.4, the pattern is rotated by 90 degrees around the Y axis, such that the "leaves" of the quadrifolium do not lie too close to the line $\mathcal{L}_M$. A constant camera orientation was chosen such that there is good visibility of the lines at all times (with the focal axis roughly pointing towards the origin).

Since the pose computation methods should be very sensitive to numerical noise in the proximity of a singularity, we added Gaussian noise with standard deviation $\sigma = 10^{-4}$ to the Plücker coordinates of the 3D lines.

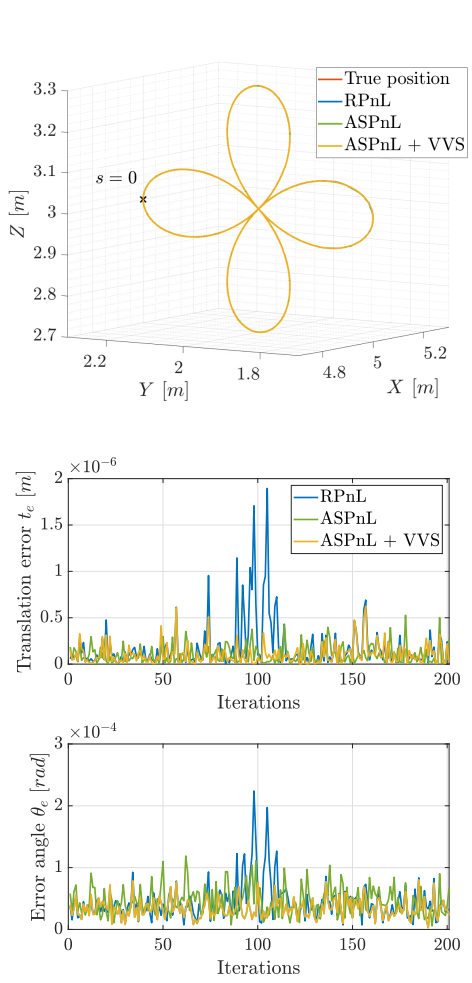
Two parameters are measured from the simulations: the *translation error* t_e, defined as the Euclidean distance between the true and estimated camera positions, and the *rotation error*: the absolute value of the error angle

$$\theta_e = \left| \arccos\left(\frac{1}{2}\left[\mathrm{tr}\left({}^c\mathbf{R}_o \, \widehat{{}^c\mathbf{R}_o}^T \right) - 1 \right] \right) \right|, \tag{4.77}$$

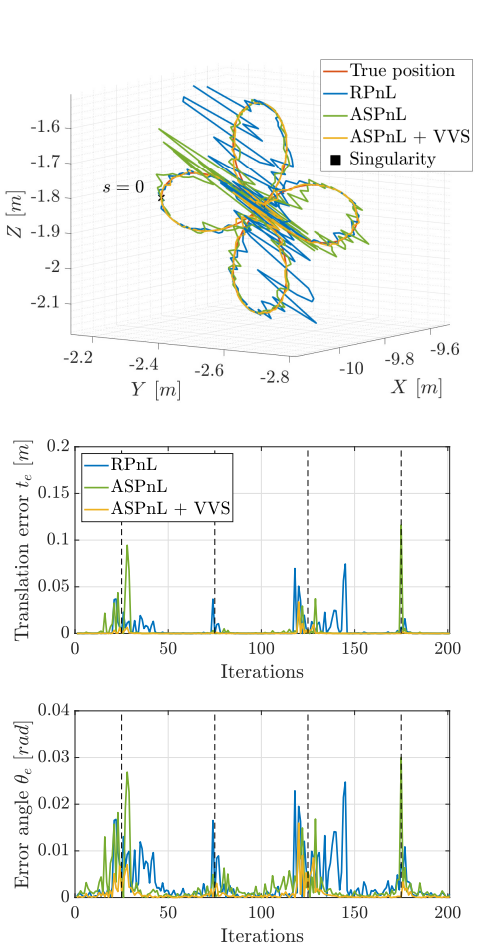
where ${}^c\mathbf{R}_o$ and $\widehat{{}^c\mathbf{R}_o}^T$ are respectively the rotation matrices representing the true and the estimated orientation of the camera frame.

The results from RPnL and ASPnL as the camera moves along each of the trajectories are depicted in Fig. 4.15; along with the refinement by VVS when initialized from ASPnL. The corresponding error metrics are displayed below. Away from the singularities, all three methods yield near perfect estimations for both position and orientation (see Fig. 4.15a).

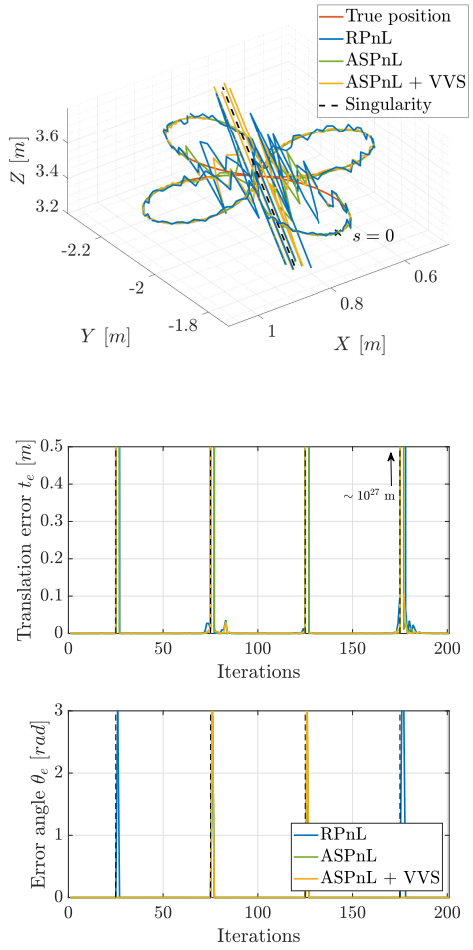
In the second experiment (Fig. 4.15b), centred at an isolated singularity, the direct methods become very inaccurate. A large translation error, reaching up to 0.1, occurs particularly as the camera crosses the singularity at the centre of the pattern, but also in the vertical direction of the clover. This is explained by the fact that both RPnL and ASPnL solve the P3L problem for two triplets of lines (in this case $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ on one hand, and $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_4$ on the other), and that for the P3L problem the singularity loci is a surface. The refinement from VVS generally allows reducing the translation error to below 10^{-4}, except very near the singular point, where a persistent error of about 0.03 remains.



(a) Example 1: Away from singularities.



(b) Example 2: Centred at an isolated singularity.



(c) Example 3: Around the singularity line $\mathcal{L}_M$.

Figure 4.15: Pose estimation from four image lines along a trajectory with the shape of a quadrifolium centred at different points: a generic point away from singularities (left), an isolated point singularity (centre) and a point on a line singularity (right). The top images show the true camera position (red), the estimation from the non-iterative RPnL (blue) and ASPnL (green), and the refinement by VVS when initialized from the ASPnL estimate (yellow). In the bottom are displayed the translation error and the absolute error angle (4.77). The vertical steps indicate the points where the camera passes through the singularity. Far away from any singularities all three methods have a near-zero error; only the yellow plot is visible in the left image because the three trajectories overlap. In a large area around a point of singularity, the direct methods become very sensitive to noise in the data, while VVS is quite effective in refining the result from ASPnL except when very near the singularity. In the near proximity of the line singularity, all methods output an abhorrent estimation, with the errors tending to infinity.

In the Example 3 from Table 4.4, the direct methods and VVS all fail catastrophically to give an acceptable estimation near the singularity (see Fig. 4.15c). As the camera approaches the line $\mathcal{L}_M$, the translation error blows up by several orders of magnitude ($\sim 10^{27}$). Since the RPnL and ASPnL algorithms compute several local minima and then select the best solution based on the reprojection error, we verified that the true solution does not lie among the local minima that were discarded. It seems that the ill-conditioning of the interaction matrix causes numerical issues in the direct solvers such that they are incapable of locating where along this line the true solution lies.

Overall, these experiments demonstrate that **a rank-deficiency of the interaction matrix can significantly impact the performance of pose computation methods**, leading to a considerable loss of accuracy or, in some cases, a completely wrong solution for camera poses near a singular location. This is true even in the case of direct solvers such as RPnL and ASPnL which do not explicitly involve the interaction matrix.

4.6.2 Singularities in P5L

We now consider the case of five lines determined by the following parameters:

$$\begin{aligned} &s_1 = 4,\ s_2 = -5,\ s_3 = 7,\ s_4 = 3,\ s_5 = -2,\ s_6 = 13, s_7 = -11,\ s_8 = 6,\\ &d_1 = 2,\ d_2 = 3,\ d_3 = 5,\ d_4 = 1,\ d_5 = 7,\ d_6 = -4,\\ &d_7 = \frac{128893236}{7630285} - \frac{24\sqrt{8508173023861}}{7630285} \approx 7.7178. \end{aligned}$$

For this configuration, the polynomial condition from Proposition 4.7 is satisfied, and there is one transversal line $\mathcal{L}_M$ that intersects all five lines (see Section 4.5.3), defined by its Plücker coordinates:

$$\mathcal{L}_M = [0.0830\ 0.4989\ -0.8627\ 0\ -0.9641\ -0.5575] \tag{4.78}$$

From Theorem 4.3, we know that a singularity will occur when the camera lies on this line. Note that the lines $\mathcal{L}_1, \ldots, \mathcal{L}_4$ are defined identically as in Section 4.6.1, and that $\mathcal{L}_5$ is chosen so as to intersect the first of the lines of singularity (4.70).

VS near the singularity line

We considered a point $\mathcal{P}_M = [0.3962\ -4.337\ 7.50]$ that lies on the transversal $\mathcal{L}_M$ and defined four starting camera positions in its surroundings. From each of these points, we performed a simulation of visual servoing towards a target point. The initial and desired positions are the same as those considered in Section 4.6.1, whose coordinates relative to point $\mathcal{P}_M$ are given in Table 4.3. The camera velocity inputs are computed according to (4.71) with the factor $\lambda = 0.1$. Gaussian noise of standard deviation $\sigma = 2 \cdot 10^{-3}$ was added to the 3D coordinates of the lines.

In all four experiments, the magnitude of the error vector decreases exponentially (Fig. 4.18). However, the trajectories displayed in Fig. 4.16 show very different behaviours. Starts 1 and 3 converge rapidly, with small camera velocities and displacements.

On the contrary, Starts 2 (directly opposed to desired point) and 4 (further away from the singularity), lead to unstable motion and large deviations. The velocity input profiles for Starts

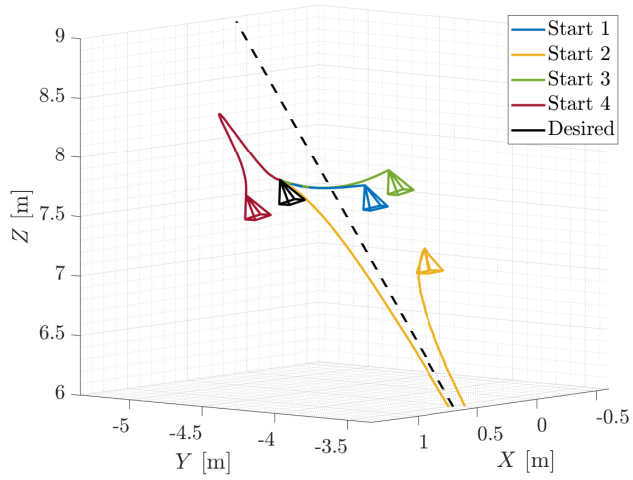


Figure 4.16: Visual servoing from five lines starting from different positions (coloured) towards a desired pose (black). The line $\mathcal{L}_M$ (dashed line) that intersects all five lines is a singularity of the interaction matrix.

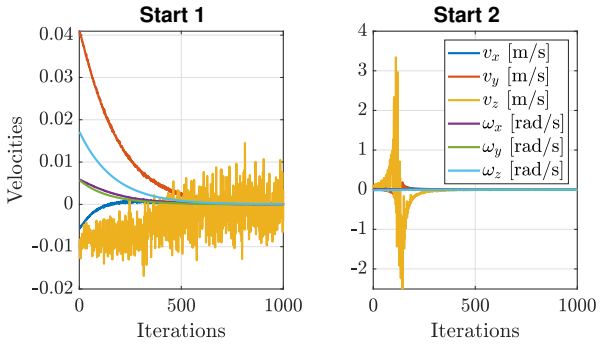


Figure 4.17: Camera velocity inputs $\boldsymbol{\tau}_c$ in a stable situation (left), and when crossing a singularity (right).

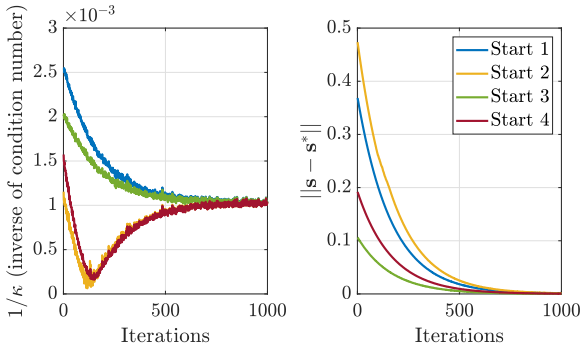


Figure 4.18: Inverse of the condition number κ of the interaction matrix $\mathbf{M}_{(4)}$ (left) and norm of the error vector $||\mathbf{s} - \mathbf{s}^*||$ (right).

1 (stable) and 2 (unstable) are compared in Fig. 4.17. In the latter, the inputs become very high in magnitude as the inverse of the condition number of the interaction matrix drops near zero, when the camera crosses $\mathcal{L}_M$. Again we notice that there exists a region of instability around the singularity line, particularly strong in one direction which affects the trajectory of Start 4, where the interaction matrix becomes ill-conditioned.

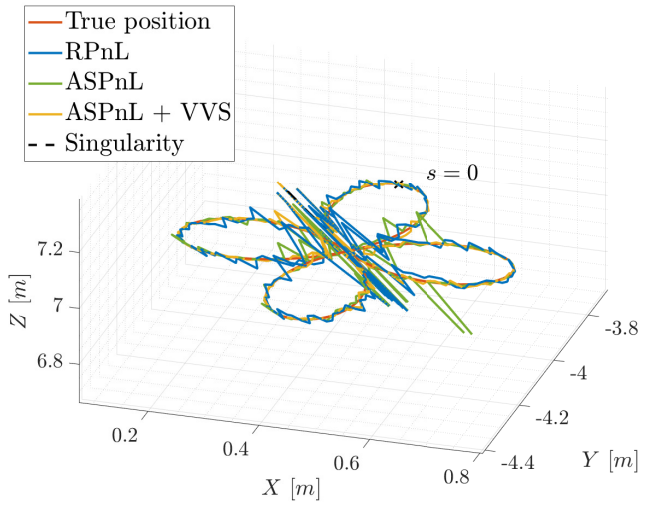


Figure 4.19: Pose computation from RPnL, ASPnL and ASPnL refined by VVS using five image lines along a quadrifolium trajectory centred at a point on the line singularity $\mathcal{L}_M$. In the proximity of the singularity the error in the estimation grows unbounded.

Pose Estimation near the singularity line

For completeness, we conclude by demonstrating the behaviour of pose estimation from the observation of five lines near a singularity. We simulated a trajectory along a pattern with the shape of a quadrifolium (with equations (4.76)) centred at a point with coordinates $[0.7809\ -2.024\,3.50]$, which lies on the singularity line $\mathcal{L}_M$. Along this trajectory, the camera pose was computed using the RPnL ([ZXLK12]) and ASPnL ([XZCK16]) algorithms and the estimation was further refined using VVS ([MC02]) initialized at the pose computed by ASPnL. We considered Gaussian noise of standard deviation $\sigma = 10^{-4}$ on the $3D$ coordinates of the observed lines.

The true pose is shown along with the estimations from the three methods in Fig. 4.19, while the translation error t_e and the rotation error θ_e defined by (4.77), are depicted in Fig. 4.20. The observed behaviour is very similar to the case of four lines when pose reconstruction is performed near the singularity line: along the leaves of the quadrifolium, the direct solvers (RPnL and ASPnL) are quite sensitive to numerical noise, which is mitigated by the refinement through VVS. However, very near the singularity the errors blow up in magnitude and all methods prove totally unsuccessful. For comparison, along a similar trajectory, but centred at coordinates $[0.5\,4.0\,1.0]$, far from any singularities, all methods are accurate up to 0.1 in the estimation of position and up to $2 \cdot 10^{-3}$ rad in orientation throughout.

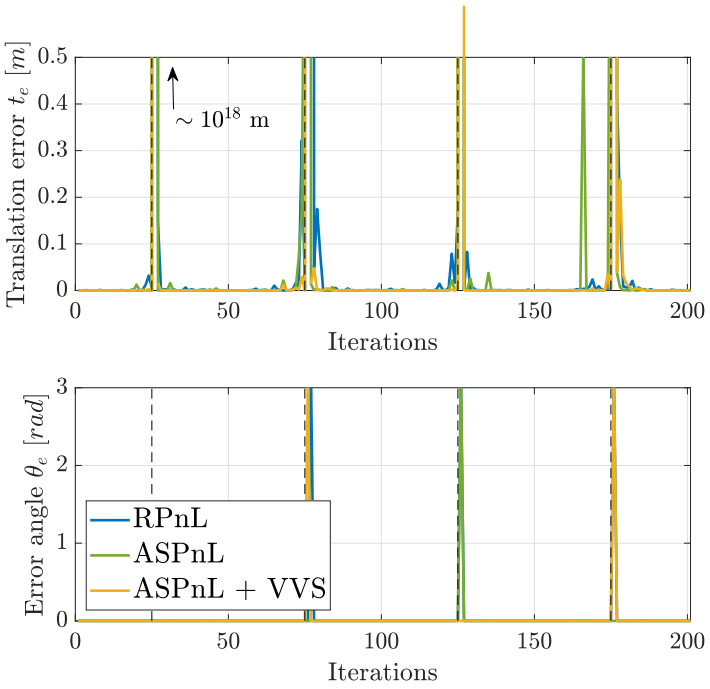


Figure 4.20: Translation (top) and rotation (bottom) errors in pose estimation from five lines along a quadrifolium trajectory. The vertical steps indicate where the camera crosses the singularity.

4.7 Conclusions

In this chapter, the singularities in the perspective four and five line problems were determined. Finding these singularities is crucial since they lead to controllability issues in visual servoing of image-lines and in large errors in pose estimation for PnL. To do so, a basis of the interaction matrix was found such that its rows are Plücker coordinates of n affine and n ideal lines for PnL.

First, it was recalled that the singularities in P3L are due to the vanishing of the determinant of the (6×6) interaction matrix which factors as a quadric and a cubic surface in terms of the position coordinates of the optical centre of the camera. It was then proved that the quadric surface is essentially the hyperboloid of one sheet uniquely defined by the three observed lines.

This fact was further used in the case of P4L to understand different cases such that the 28 principal minors of the (8×6) interaction matrix vanish simultaneously, leading to singularities. One of those cases is when the camera lies on two transversals intersecting the observed four lines. It was also shown that this one dimensional singularity could be avoided by choosing the fourth line to not intersect the hyperboloid defined by the other three lines. Additionally, Gröbner basis computations were used to determine that there can exist up to ten real isolated singularities of P4L.

Furthermore, in the case of P5L, no singularities were found for a generic choice of five

lines. Nonetheless, some conditions of the relative configurations of the five lines were shown to yield a transversal line of singularities that intersects the five observed lines.

The same analysis was done for four and five lines that are constrained to be orthogonal or parallel to each other to corroborate the results for the generic case. It turned out that the singularities for P4L consist of two transversal lines, whereas for P5L, it is a unique transversal line.

The geometric interpretation of the one dimensional singularities of P4L and P5L was provided, by extending the result that a hyperboloid of one sheet is a singularity in the P3L case. These one dimensional singularities appeared when the affine lines in the interaction matrix are linearly dependent by being parallel to the same plane. In the future, we will try to obtain the geometric interpretation of the isolated point singularities in the generic P4L and P5L.

The results are supported with experimental simulations of Visual Servoing control of a camera and of pose determinations algorithms from the observation of lines in the proximity of the singularities. As expected, the ill-conditioning of the interaction matrix near a singular point results in unstable behaviour of the control law from VS and in significant losses of accuracy for the pose estimation methods.

In Appendix A we present a similar singularity analysis for the observation of minimal combinations of point and line features.

Chapter 5

Critical points of IBVS

Abstract. When performing Image-Based Visual Servoing of a camera, it is a known issue that using a redundant set of visual features (i.e. more than the number of degrees of freedom of the system) may lead to the appearance of local minima, that is, stable camera configurations for which the error function is non-zero. As a consequence, the camera may converge to one of these equilibria, instead of the desired final position, resulting in a steady state error that can be critical for applications requiring high-precision. The challenge of determining whether any such points exist, and computing their location, has hitherto remained an open problem, due to the complexity of the equations involved. Further, identifying all the local minima is a necessary step for an analysis of the global stability properties of IBVS controllers and, in particular, for determining whether the camera will converge to the desired position from an initial configuration in the workspace.

For many IBVS strategies, the control input velocity is defined to make the camera move in a direction that decreases the magnitude of the error. Then, the points of equilibrium are the critical points of a potential function representing the error norm. In this chapter we address the problem of computing the equilibrium points of IBVS control from N reference points. We model the problem as a system of polynomial equations arising from the gradient of the error potential, in the variables representing the projected coordinates of the points and their depth along the focal axis of the camera. Using `msolve`, a polynomial-based system solver, we solved this system for $N = 4$ reference points in some different configurations. However, due to the complexity of the equations, the computing times are exceedingly long (over several weeks over 12 processors running in parallel). Since the local minima must be recomputed for every configuration of the reference points and for every desired end pose of the camera, we find that this formulation is insufficiently effective.

We then present an alternative modeling of the equations, by defining a change of variables that exploits the symmetries of the solution set of the original system, and then performing algebraic elimination by means of Gröbner bases with respect to an elimination ordering, leading to a new polynomial ideal with lower degree. Additionally, we also find that, in the case of planar markers (i.e. the reference points lying on the same plane), we can further reduce the degree of the polynomial ideal by imposing the coplanarity condition in the space of the state variables.

With these reformulation, we can compute the critical points in the case of 4 generic reference points in a matter of 2-3 days, as well as in a few hours for coplanar points.

The work presented in this chapter was done in collaboration with Alessandro Colotti and Alexandre Goldsztejn.

Keywords Visual servoing · Critical points · Global asymptotic stability ·

5.1 Introduction

5.1.1 Motivation and problem statement

Image-Based Visual Servoing methods are known to be, at least, locally asymptotically stable in the neighbourhood of the global minimum of the system [CH06]. However, in general, the controlled dynamical system may have other local minima, corresponding to configurations where the system has converged to a state with non-zero error. Further, the local minima may be located anywhere, and particularly, arbitrarily near the actual global minimum. This can be critical for applications with high-precision requirements, such as industrial or medical robotics. In addition, the existence of these minima raises the question of delimiting the regions in the configuration space where the camera can be guaranteed to converge towards the desired position. The challenge of accurately identifying the points of equilibrium of IBVS controllers has thus far remained unsolved, as has the broader problem of characterizing the global stability properties of the controller and, in particular, of deciding whether the system will converge to the desired state from an initial configuration.

In this chapter we focus on the classical IBVS control of a camera with six degrees of freedom, with an error function $(\mathbf{s} - \mathbf{s}^\star)$ defined as the difference between the vector of visual features $\mathbf{s} \in \mathbb{R}^k$ and the reference value of these features $\mathbf{s}^\star$, and the input control velocity proportional to the error vector as $\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}^+(\mathbf{s} - \mathbf{s}^\star)$, where $\widehat{\mathbf{L}}^+$ is the control matrix and λ a gain factor. The matrix $\widehat{\mathbf{L}}^+$ is typically chosen as a generalized inverse (such as the Moore-Penrose pseudo-inverse) of the interaction matrix $\mathbf{L}$ or of an estimation $\widehat{\mathbf{L}}$ of it, if not all of the parameters involved in $\mathbf{L}$ are known (refer to [CH06, HHC96] or to Chapter 2, Section 2.4). For this input velocity, the error vector evolves as

$$\frac{d}{dt}(\mathbf{s} - \mathbf{s}^\star) = -\lambda \mathbf{L}\widehat{\mathbf{L}}^+(\mathbf{s} - \mathbf{s}^\star). \tag{5.1}$$

The global asymptotic stability of this system around its global minimum is guaranteed if the following condition is ensured throughout the whole parameter space except at $\mathbf{s} = \mathbf{s}^\star$ (see Chapter 2, Section 2.6):

$$\mathbf{L}\widehat{\mathbf{L}}^+ \succ 0. \tag{5.2}$$

However, when a redundant number of visual features $k > 6$ is used, the condition (5.2) can never be ensured, since $\mathbf{L}\widehat{\mathbf{L}}^+ \in \mathbb{R}^{k \times k}$ is of maximum rank 6. For this case, configurations such that $(\mathbf{s} - \mathbf{s}^\star) \in \ker(\widehat{\mathbf{L}}^+)$ correspond to *fixed points* of the system: configurations for which the camera velocity is null but the error is not zero.

Choices of the control matrix $\widehat{\mathbf{L}}^+$ for which the product $\mathbf{L}\widehat{\mathbf{L}}^+ \succeq 0$, i.e. it is at least always positive semi-definite, satisfy that the system (5.1) always evolves in a direction that decreases the norm of the error function. If we define the error potential as $V = \frac{1}{2}||\mathbf{s} - \mathbf{s}^\star||^2$, then $\frac{dV}{dt}$ is always non-positive, and such a system is said to be of *gradient-descent* type with respect to V. This is the case for a transpose controller $\widehat{\mathbf{L}}^+ = \mathbf{L}^T$, which is known to be quite robust in general, although it has poor convergence properties. It is also the case when using the classical pseudo-inverse matrix, when all the parameters can be measured or computed from the image: $\widehat{\mathbf{L}}^+ = \mathbf{L}^+(\mathbf{p})$. Note that it is not the case, for instance, for the choice $\widehat{\mathbf{L}}^+ = \mathbf{L}^+(\mathbf{p}^\star)$, where the matrix is evaluated at the desired position, a choice commonly used in practice.

According to Lyapunov's theory (see, for instance, [SL+91]), convergent trajectories of a system that is *gradient-like* with respect to a function V will always converge towards a critical point of V [Sas13]. The computation of the critical points is thus a first step towards a global stability analysis of IBVS controllers, the next step being the characterization of the regions of attraction around each of the points of equilibria, in particular the global minimum.

Recently, Alessandro Colotti and Alexandre Goldsztejn have developed a strategy to estimate the regions of attraction by solving an optimization problem with polynomial constraints. Their method relies on comparing the relative value of the error potential at the different critical points and, as a consequence, a crucial part of it is identifying precisely *all* of these points. Given that the dynamics of visual servoing are governed by systems of highly non-linear polynomial equations, and the need of exact solutions, computer algebra is particularly well-suited for the task.

5.1.2 Main results

This chapter deals with the computation of the critical points of IBVS from N reference points. First, we give the equations describing the critical points of the error potential V as the vanishing of the gradient of V. By differentiating with respect to the camera pose as an element of $SE(3)$, we show that the critical points of V are the camera configurations for which the error vector is in the left kernel of the interaction matrix: $\mathbf{L}^T \mathbf{e} = \mathbf{0}$. In the case of N reference points, it leads to a system of six equations that are naturally parametrized in the space of variables $(\mathbf{s}, \mathbf{Z})$, representing the coordinates of the points on the image and their depths along the focal axis of the camera. Solving the equations in terms of these variables is more computationally tractable than trying to solve them in terms of a parametrization of the camera pose parameters $\mathbf{p}$, although it requires that additional constraints be considered, by imposing that the distance between each of the points does not change.

For $N > 3$ points, the resulting system of equations defines a finite set of solutions with complex coordinates. For instance, for $N = 4$ generic points, we obtain a polynomial ideal of degree 3656, i.e. there are 3656 complex solutions counted with multiplicity. Using `msolve` [BES21], a computer algebra based polynomial system solver, we succeeded in computing the critical points for $N = 4$ points, both in generic and in particular configurations of interest, such as planar targets. However, the computing times are exceedingly high, well over several weeks for most cases, making it ineffective for the study of more than a few different configurations. Nevertheless, this is, to our knowledge, the first time that these results have been obtained using exact methods, and is therefore a noteworthy result. Alternatively, numerical methods, such as homotopy continuation, could be used; however we show that, due to the high degrees of the polynomials involved, methods relying on floating point arithmetic are not reliable for this problem. In particular, homotopy often fails to capture all the solutions of the system, thus justifying the use of more expensive methods for exact polynomial system solving.

We next present two computer algebra strategies specific to these systems that can be applied to simplify the equations and reduce significantly the computing times. The first of these techniques relies on exploiting the symmetries of the polynomial system by defining a change of variables invariant to this symmetry, and applying elimination theory through Gröbner bases. In fact, the set of solutions of the original system of equations presents point

symmetry through the focal centre of the camera and, as a consequence, half of the solutions are reflections of the other half. We exploit this property to derive a system with half the total degree (e.g. 1828 for $N = 4$ generic points). The other technique can be applied when all the reference points lie on the same plane. For this case, there is another kind of symmetry that appears in the mapping from the group of Euclidean transformations to the state of the system: with respect to the plane of the object, and as a result the solutions of the system appear with multiplicity. Using the Jacobian criterion ([Eis13], Theorem 16.19), the multiplicity of the solutions is reduced and, with it, the total degree of the polynomial ideal, that we can again reduce by half.

With these computational improvements, we were able to solve a larger variety of examples with `msolve` in more practical times - of the order of 2-3 days for the case of $N = 4$ generic, non-coplanar points, and from a few minutes to a few hours for the case of points lying on the same plane. We expect to be able to compute the solutions for $N = 5$ reference points in the future, by taking advantage of these techniques. However the case $N = 5$ has so far proven too computationally intensive, due to the amount of memory required.

For comparison, we also attempted to solve the reformulated systems of equations using homotopy continuation, albeit with the same, meagre, success. For planar objects, applying the Jacobian criterion technique improves the proportion of solutions captured correctly by homotopy, but we still find examples for which many solutions are not detected. On the other hand, when we derive a new system of polynomials by removing the symmetric solutions, the system becomes over-constrained, and homotopy methods fail altogether to compute a solution.

Structure of the chapter. In Section 5.2, we introduce the equations that describe the critical points of general IBVS controllers that are of *gradient-descent* type with respect to the error potential. Then, in Section 5.3, we consider the case of N reference points and model the problem as a system of polynomial equations in the projected coordinates of the points and their depths along the focal axis. We also present some preliminary computation results obtained by solving this system using `msolve`, leading to very long computing times. Section 5.3.3 describes how to reduce the total number of solutions that must be computed by applying a change of variables that is invariant to the symmetries of the system, and Section 5.3.4 focuses on the case of planar objects, and shows how the total degree of the ideal generated by the equations can be further reduced by removing the multiplicity of the solutions. Then in Sections 5.3.5 and 5.3.6 is explained how to retrieve the camera poses that correspond to the critical configurations from the solutions in the space of image parameters, and how to classify the solutions in local minima, maxima or saddle points. In Section 5.4 are presented the computation results for several different configurations, and the improvements in computing times obtained by applying the two techniques above. The results are compared with the results from numerical (homotopy continuation) methods applied to these systems.

5.2 Critical points of gradient controllers

Let us consider the classical IBVS control of a camera, where the vector of features $\mathbf{s}$ is a function of the relative pose $\mathbf{p}$ between the camera and the target ($\mathbf{s} = \mathbf{s}(\mathbf{p})$), and the reference value of the features $\mathbf{s}^\star$ is fixed. Then, the error of the controlled system $\mathbf{e} = (\mathbf{s} - \mathbf{s}^\star)$ evolves as (5.1). The camera pose $\mathbf{p}$ is an element of $SE(3)$, the Lie group of rigid-body transformations (see Chapter 2, Section 2.1).

We define the potential of the error $V : SE(3) \to \mathbb{R}$ as the function

$$V(\mathbf{s}(\mathbf{p})) = \frac{1}{2}||\mathbf{s}(\mathbf{p}) - \mathbf{s}^\star||^2 = \frac{1}{2}\mathbf{e}^T\mathbf{e}. \tag{5.3}$$

The time-derivative of V is thus

$$\dot{V}(t) = \mathbf{e}^T\dot{\mathbf{e}} = -\lambda\mathbf{e}^T\mathbf{L}\widehat{\mathbf{L}^+}\mathbf{e} \tag{5.4}$$

Any choice of controller (i.e. any choice of the matrix $\widehat{\mathbf{L}^+}$) for which the product $\mathbf{L}\widehat{\mathbf{L}^+}$ is positive semi-definite everywhere in the configuration space, i,e, $\mathbf{L}\widehat{\mathbf{L}^+} \succeq 0$ for all $\mathbf{p}$ in $SE(3)$, will always evolves in the direction of non-increasing V: $\dot{V}(\mathbf{p}(t)) \leq 0$ for all $\mathbf{p}$. Then, the controller is said to be of *gradient-descent type* or *gradient-like* with respect to the function V. For such systems, Lyapunov's theory dictates that all the convergent trajectories converge towards a critical point of the function V (see Section 2.7); in other words, all the stable points of the controlled system are critical points of the function V, and the function V is a **Lyapunov function** around each of the stable points.

The critical points of V are the camera configurations $\mathbf{p}$ for which its gradient with respect to the camera pose $\mathbf{p}$ vanishes. The set of all the critical points is denoted $\mathcal{C}$:

$$\mathcal{C} = \{\mathbf{p} \in SE(3) : \ \nabla_{\mathbf{p}} V(\mathbf{p}) = \mathbf{0}\}. \tag{5.5}$$

The pose $\mathbf{p}$ being an element of $SE(3)$, it is also a differentiable manifold. The differential of an element of $SE(3)$ is an element of the Lie algebra $se(3)$, which defines a vector space locally tangent to the manifold $SE(3)$. Differentiation with respect to $SE(3)$ is commonplace in Robotics (refer to [LP17], and to Chapter 2, Section 2.1), although we find it convenient to recall it here in the derivation of the following theorem.

Theorem 5.1. *For IBVS controllers that are gradient-like with respect to the error potential* $V : SE(3) \to \mathbb{R}, \quad V(\mathbf{p}) = \frac{1}{2}||\mathbf{s}(\mathbf{p}) - \mathbf{s}^\star||^2$ *, the critical points of* V *are the solutions of*

$$\nabla_{\mathbf{p}} V(\mathbf{p}) \triangleq \mathbf{L}^T(\mathbf{p})\,(\mathbf{s}(\mathbf{p}) - \mathbf{s}^\star) = \mathbf{0} \tag{5.6}$$

Proof. Let the camera pose $\mathbf{p} \in SE(3)$ be represented by a (4×4) **homogeneous matrix** $\mathbf{M} = {}^c\mathbf{M}_o$ as in (2.1) (following the subscripts convention, $\mathbf{M}$ represents the pose of the object frame $\mathcal{F}_o$ relative to the camera frame $\mathcal{F}_c$), and consider a change of the function V due to a small variation in the pose given by $\delta\mathbf{M}$:

$$\delta V = \sum_i e_i(\delta e_i), \tag{5.7}$$

where the infinitesimal error variation δe_i is related to $\delta\mathbf{M}$ by

$$\delta e_i = \operatorname{tr}\left(e_{i\mathbf{M}}^T \delta\mathbf{M}\right). \tag{5.8}$$

Here, $\operatorname{tr}(\cdot)$ is the matrix trace operator and the subscript $\{\cdot\}_{\mathbf{M}}$ denotes the partial derivative with respect to the matrix $\mathbf{M}$. The partial derivative of a scalar f by any matrix $\mathbf{A} \in \mathbb{R}^{a\times b}$ is defined, using the standard denominator layout, as

$$f_{\mathbf{M}} = \frac{\partial f}{\partial \mathbf{A}} = \begin{pmatrix} f_{A_{11}} & f_{A_{12}} & \cdots & f_{A_{1b}} \\ \vdots & \ddots & & \vdots \\ f_{A_{a1}} & \cdots & & f_{A_{ab}} \end{pmatrix}. \tag{5.9}$$

The trace pairing between the matrix partial derivative and the matrix variation in (5.8) amounts to the correct variation of the scalar function [TR17].

Additionally, the variation $\delta\mathbf{M}$ of the homogeneous matrix can be expressed as [LP17]

$$\delta\mathbf{M} = [\delta\boldsymbol{\Sigma}_c]\,\mathbf{M} \tag{5.10}$$

where $\delta\boldsymbol{\Sigma}_c$ is an element of $se(3)$ representing an instantaneous velocity twist in **camera frame coordinates**, and $[\delta\boldsymbol{\Sigma}_c]$ is its (4×4) matrix representation according to Definition 2.5.

Substituting (5.10) in (5.8) and using the fact that $\mathbf{s}^\star$ is fixed and, hence, $e_{i\mathbf{M}} = s_{i\mathbf{M}}$, we have

$$\delta e_i = \operatorname{tr}\left(s_{i\mathbf{M}}^T\,[\delta\boldsymbol{\Sigma}_c]\,\mathbf{M}\right). \tag{5.11}$$

This expression can be rearranged in terms of the components of $\delta\boldsymbol{\Sigma}_c$ and expressed as a matrix-vector product:

$$\delta e_i = \mathbf{L}_i\,\delta\boldsymbol{\Sigma}_c \tag{5.12}$$

where $\mathbf{L}_i \in \mathbb{R}^{1\times 6}$ is the usual interaction matrix (5.14) for each feature s_i. To show that the above is true, it suffices to expand the terms on both sides of the equality. Finally, injecting this expression in (5.7), we obtain

$$\delta V = \sum_i e_i\,(\mathbf{L}_i\,\delta\boldsymbol{\Sigma}_c) = \left(\mathbf{e}^T\mathbf{L}\right)\delta\boldsymbol{\Sigma}_c. \tag{5.13}$$

The product $\left(\mathbf{e}^T\mathbf{L}\right)$ is thus the vector gradient of V, $\nabla_{\mathbf{p}}V$. At a critical point of V, the variation (5.13) must be zero for any direction of $\delta\boldsymbol{\Sigma}_c$, therefore $\left(\mathbf{e}^T\mathbf{L}\right)$ must be the zero vector, thus completing the proof. □

Note The matrix $\mathbf{M} = {}^c\mathbf{M}_o$ represents the position and the orientation of $\mathcal{F}_o$ expressed in $\mathcal{F}_c$ coordinates. Accordingly, the instantaneous twist $\delta\boldsymbol{\Sigma}_c$, that left-multiplies the transformation matrix in (5.10), is also expressed in camera-frame coordinates: $\delta\boldsymbol{\Sigma} = \delta\boldsymbol{\Sigma}_c$. For a twist $\delta\boldsymbol{\Sigma}_o$ expressed in world coordinates, (5.10) is equivalent to right multiplying by the twist operator: $\delta\mathbf{M} = \mathbf{T}\,[\delta\boldsymbol{\Sigma}_o]$ (see Chapter 2 Section 2.1.3). Since the velocity vector is typically expressed in camera frame coordinates, the form used in the derivation of the interaction model is (5.10).

5.3 Critical points of IBVS from N points

5.3.1 Modeling

Let us consider a reference object consisting of N points in $\mathbb{R}^3$. Then, a 3D point with coordinates $\mathbf{X}_i = (X_i, Y_i, Z_i)$ in the frame of the camera, is projected on the image plane on coordinates $(x_i, y_i) = (X_i/Z_i, Y_i/Z_i)$ (see Chapter 2, Section 2.3). We take $\mathbf{s} = (x_1,\, y_1, \ldots, x_N,\, y_N)$ as the vector of coordinates of the image points, and $\mathbf{s}^\star = (x_1^\star, y_1^\star, \ldots, x_N^\star, y_N^\star)$ as the reference value of the visual features. The vector $\mathbf{Z} = (Z_1, \ldots, Z_n)$ contains the depth coordinates of the points, along the focal axis of the camera. The interaction matrix, which relates the variation of the visual features to the camera velocity twist by (2.23) is the usual for a set of points: $\mathbf{L}(\mathbf{s}, \mathbf{Z}) = \begin{bmatrix}\mathbf{L}_1^T & \ldots & \mathbf{L}_N^T\end{bmatrix}^T$ with $\mathbf{L}_i$ given by (2.31):

$$\mathbf{L}_i(\mathbf{s}_i, Z_i) = \begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{x_i}{Z_i} & x_i\, y_i & -(1+x_i^2) & y_i \\ 0 & -\frac{1}{Z_i} & \frac{y_i}{Z_i} & 1+y_i^2 & -x_i\, y_i & -x_i \end{bmatrix}, \tag{5.14}$$

The system (5.6) that describes the fixed points of the error potential $V = \frac{1}{2}||\mathbf{s} - \mathbf{s}^\star||$ leads to the following system of six equations, in $3N$ variables: $\mathbf{s} = (x_1,\, y_1, \ldots, x_N,\, y_N)$ and $\mathbf{Z} = (Z_1, \ldots, Z_N)$, and depending on the parameters $\mathbf{s}^\star = (x_1^\star, y_1^\star, \ldots, x_N^\star, y_N^\star)$:

$$\begin{aligned}
&\sum_{i=1}^{N} \frac{x_i - x_i^\star}{Z_i} = 0 \\
&\sum_{i=1}^{N} \frac{y_i - y_i^\star}{Z_i} = 0 \\
&\sum_{i=1}^{N} \frac{x_i\,(x_i - x_i^\star)}{Z_i} + \frac{y_i\,(y_i - y_i^\star)}{Z_i} = 0 \\
&\sum_{i=1}^{N} x_i\, y_i\,(x_i - x_i^\star) + (1+y_i^2)\,(y_i - y_i^\star) = 0 \\
&\sum_{i=1}^{N} (1+x_i^2)\,(x_i - x_i^\star) + x_i\, y_i\,(y_i - y_i^\star) = 0 \\
&\sum_{i=1}^{N} (x_i\, y_i^\star - y_i\, x_i^\star) = 0
\end{aligned} \tag{5.15}$$

It can be assumed that $Z_i \neq 0$ always for all i, since the contrary corresponds to the degenerate cases where the projection is undefined. Therefore the system (5.15) can be made polynomial by multiplying the first three equations by $\prod_i Z_i$.

A possible approach for computing the critical points of the function V is to define a parametrization for the pose of the camera $\mathbf{p}$ (e.g. by means of a translation vector $\mathbf{t}$ and a rotation matrix $\mathbf{R}$, or the components of a unit quaternion), and describe the system of equations (5.15) in terms of this parametrization.

We define the mapping that assigns the value of variables $(\mathbf{s}, \mathbf{Z})$ to a particular position and orientation of the camera frame as

$$\Phi : \ \mathbf{p} \in SE(3) \ \mapsto \ (\mathbf{s}, \mathbf{Z}) \in \mathbb{R}^{3N}. \tag{5.16}$$

Let the couple $(\mathbf{t}, \mathbf{R})$, with $\mathbf{t} = {}^c\mathbf{t}_o \in \mathbb{R}^3$ and $\mathbf{R} = {}^c\mathbf{R}_o \in SO(3)$ describe the pose $\mathbf{p}$ of a reference frame attached to the object, relative to the frame of the camera, following the convention for subscripts and superscripts (see Section 2.1). We can describe the variables $\mathbf{s} = (x_1, y_1, \dots, x_N, y_N)$ and $\mathbf{Z} = (Z_1, \dots, Z_n)$ in terms of the camera pose using

$$\begin{aligned} \mathbf{X}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} &= \mathbf{R} \cdot {}^o\mathbf{X}_i + \mathbf{t} \\ x_i = X_i/Z_i, \quad & y_i = Y_i/Z_i, \end{aligned} \tag{5.17}$$

where ${}^o\mathbf{X}_i$ is the vector of coordinates of the point in the object frame. The equations (5.17) describe the function $\Phi(\mathbf{p}) = (\mathbf{s}, \mathbf{Z})$.

However, substituting (5.17) in the system (5.15) yields, after properly algebraizing the equations, a system of polynomials in very high degrees, computing the solutions of which is outside of our computational capabilities.

Instead, a better strategy was proposed by Alessandro Colotti. We search for the solutions of the system above directly in the space of variables $(\mathbf{s}, \mathbf{Z})$, in which the equations (5.15) are naturally expressed. However this means there are $3N$ variables for 6 equations, and for $N \geq 3$ points the system is underconstrained. It is thus necessary to consider some additional constraints in order to construct a square system.

The extra relations are obtained from the problem geometry. The distance $||\mathbf{X}_i - \mathbf{X}_j|| = d_{ij}$ between any two points $\mathbf{X}_i$ and $\mathbf{X}_j$ must remain constant, when expressed in the space of variables $(\mathbf{s}, \mathbf{Z})$. Since $X_i = x_i Z_i$ and $Y_i = y_i Z_i$, each constrain has the form

$$c_{ij}(\mathbf{s}, \mathbf{Z}) := (x_i Z_i - x_j Z_j)^2 + (y_i Z_i - y_j Z_j)^2 + (Z_i - Z_j)^2 - d_{ij}^2 = 0, \tag{5.18}$$

where d_{ij} is known from the object configuration. We can derive $\frac{N(N-1)}{2}$ such relations for different pairs of points, although only $3N - 6$ are enough to constrain the system. The values of $(\mathbf{s}, \mathbf{Z})$ that satisfy (5.18) for all i and j are those compatible with the geometry of the object or, in other words, those for which the inverse of the transformation (5.16) is defined:

$$\exists\, \mathbf{T} \in E(3) : \Phi(\mathbf{T}) = (\mathbf{s}, \mathbf{Z}) \quad \Leftrightarrow \quad c_{ij}(\mathbf{s}, \mathbf{Z}) = 0 \tag{5.19}$$

Remark 1. *This inverse* $\mathbf{T}$ *belongs to the group of all Euclidean transformations* $E(3)$*, and not* $SE(3)$ *(see Section 2.1.1). In particular,* ***indirect isometries*** *of the Euclidean space, denoted by* $E(3)^-$ *that is, transformations including* ***reflections*** *by planes, or through axes and points, also satisfy* (5.18)*. They do not correspond to a feasible camera configuration, since they change the* ***handedness*** *of the reference frame.*

Remark 1 will become particularly relevant in Section 5.3, where we investigate the symmetries of the equations and reduce the complexity of the system by removing the spurious solutions *beforehand*.

Finally, we can remove the solutions of (5.15) that satisfy $Z_i = 0$ for some i (since they correspond to degenerate configurations), by enforcing a constraint of the form $1 - \ell\, Z_1 \dots Z_N = 0$, where ℓ is a new variable. Alternatively, the solutions with zero Z_i coordinate could be removed later, but we find that including this constraint reduces notably the computation time.

Theorem 5.2. *Consider an IBVS control scheme from N reference points and which is gradient-like with respect to the potential of the error $V = \frac{1}{2}\mathbf{e}^T\mathbf{e}$. The critical points of V are described by the system of equations*

$$\mathbf{F}(\mathbf{s}, \mathbf{Z}) = \left(\mathbf{L}^T\,(\mathbf{s} - \mathbf{s}^\star),\ c_{12},\ c_{13}, \dots,\ 1 - \ell\, Z_1 \dots Z_N\right) = \mathbf{0}, \tag{5.20}$$

which depends on the parameters $\mathbf{s}^\star = (x_1^\star,\, y_1^\star, \dots, x_N^\star,\, y_N^\star)$ and $\mathbf{d} = (d_{12}, d_{13}, d_{14}, \dots)$.

Lemma 5.3. *The solutions of the system (5.20) are symmetric with respect to the transformation $\mathbf{Z} \mapsto -\mathbf{Z}$. For every solution $(\mathbf{s}, \mathbf{Z})$ corresponding to a critical camera pose $\mathbf{p}_{crit} \in SE(3)$, the point $(\mathbf{s}, -\mathbf{Z})$ is also a solution, corresponding to a reflection of the scene about the focal point $\widehat{\mathbf{T}} \in E^-(3)$.*

Proof. The function $\mathbf{s}(\mathbf{p})$ described by (5.16) has an invariance with respect to a reflection of the scene through the focal point of the camera; that is, points with 3D coordinates $-\mathbf{X}_i$ map to the same image point: $\Pi(\mathbf{X}_i) = \Pi(-\mathbf{X}_i) = \mathbf{s}_i$. The solutions of the system (5.6) also present this symmetry, since the transformation $\mathbf{Z} \mapsto -\mathbf{Z}$ only changes the sign of some of the polynomials of $\mathbf{F}$. Since a reflection through a point is an isometry of the Euclidean space, distances between the points are preserved, therefore the distance constraints (5.18) remain invariant too.

In particular, for every solution $(\mathbf{s}, \mathbf{Z})$ corresponding to a critical point, there exists a symmetric solution $(\mathbf{s}, -\mathbf{Z})$. However the latter does not correspond to a real rigid-body transformation $\mathbf{p} \in SE(3)$; rather to its reflection through point C, which we denote $\widehat{\mathbf{T}}$. These spurious solutions can by identified *after* solving the system of equations. In Section 5.3.3, we will detail how to exploit this symmetry to our advantage, by applying a change of variables invariant to this transformation in order to derive a simplified system, with a lower total degree and a smaller number of solutions.

□

5.3.2 Solving the difficult system

Here we present some preliminary results, obtained by solving the system (5.20) directly. We can do this only for only a handful of configurations due to the long computing times (between 15 and 41 days from our tests). Then, over the next sections, we explain how to reformulate the system of equations to obtain a new polynomial ideal with lower degree, which allows us to obtain a wider collection of results, which are presented in Section 5.4.

In general the equation system (5.20) is of dimension zero for $N > 3$ points. It can be solved using any method for polynomial system solving: symbolic, such as approaches based on Gröbner bases, or numerical, such as homotopy continuation. In the following we use `msolve` [BES21], a software for computing the real solutions of zero-dimensional

Table 5.1: Computation of the critical points in IBVS for $N = 4$ points in different configurations using symbolic vs. numeric computation. In red are indicated the systems for which homotopy fails to find the correct number of solutions. The `msolve` computations were performed on 12 cores, while the `Julia` computations were done on a single core.

System	`msolve`			`Julia` (Homotopy Continuation)		
	$\#\text{sols}_{\mathbb{C}}$	$\#\text{sols}_{\mathbb{R}}$	Time ($\times$12 cores)	$\#\text{sols}_{\mathbb{C}}$	$\#\text{sols}_{\mathbb{R}}$	Time
1 square1	402	50	15 days	403	50	1630 s
3 square3	1016	44	24 days	1016	44	1495 s
7 rectangle3	1064	48	27 days	871	32	1950 s
13 generic1	3656	84	41 days	3537	95	2280 s

multivariate polynomial systems. `msolve` is based on the F4 [Fau99] algorithm for computing Gröbner bases and on the FGLM [FGLM93] algorithm for changes of monomial ordering (we refer to Section 3.2.3 for a review of the state-of-the-art algorithms for solving polynomial systems), and it produces a rational parametrization of the (finite) set of solutions, as introduced in Definition 3.20. A univariate real root isolation algorithm is then used to compute an approximation of the solutions to arbitrary precision. For a comparison between the performances symbolic and numerical approaches, we will later show the results obtained using real homotopy continuation, through the package [BT18] implemented in `Julia`. All the computations presented here, as well as the scripts used to perform them, will be made available in the github repository jorge-gf/thesis-archive [GF22].

Using `msolve`, we were able to solve the system of equations (5.20) for the observation of $N = 4$ points in a generic configuration, as well as for other configurations of practical interest, such as four coplanar points forming a square or a rectangle, with a desired final pose which is parallel to the plane of the object. Table 5.1 shows the number of complex ($\#\text{sols}_{\mathbb{C}}$) and real ($\#\text{sols}_{\mathbb{R}}$) solutions (including the mirrored solutions) of the system of equations (5.20) for $N = 4$ points in four different configurations, along with the computation times. The computations were performed on a machine equipped with an Intel Xeon Gold 6246R CPU (3.40GHz) and 1.5 TB RAM. The `msolve` computations were performed using 12 cores in parallel. The parameters $\mathbf{d}$, $\mathbf{s}^\star$ for these systems are given later, in Table 5.3 of Section 5.4, where we also present a larger sample of configurations and their solutions.

Unfortunately, the computing time for solving these systems is exceedingly high, way over several weeks for certain configurations, depending on the sizes of the coefficients in the polynomials. Given that the computation must be performed for every different geometry of the object, encoded by $\mathbf{d}$, and for every target pose for the camera, determined by $\mathbf{s}^\star$, the formulation used to compute the critical points by solving the system (5.20) is not effective for real life applications. Nevertheless, it is, to our knowledge, the first time that these results have been obtained, thanks to the use of state-of-the-art software and powerful Gröbner Bases algorithms.

The solutions from `msolve` are compared in Table 5.1 with the results and timings obtained using homotopy continuation, through the Homotopy Continuation package for `Julia`. Unsurprisingly, numerical methods are much faster than the exact ones (note also that the speed is

also roughly independent of the coefficient sizes). This is because polynomial based computer algebra methods suffer from the bit growth due to handling exact rational values, which is not a problem for floating-point arithmetic. However we observe that homotopy often fails to find the right number of complex and/or real solutions; these are the cases marked in red. On one hand, homotopy can sometimes miss solutions if one or more points are very close together and they are identified as being only one. On the other hand, solutions with multiplicity can be identified as being multiple different solutions. It is also possible that complex solutions with very small imaginary parts are interpreted as real solutions. As highlighted before, the correct determination of *all* the critical points of the system is crucial for the characterization of the regions of attraction of IBVS controllers. The method by Alessandro Colotti and Alexandre Goldsztejn based on optimisation relies on comparing the relative value of the error potential at all the different critical points; therefore missing a single critical point (or misidentifying a false one) could potentially spoil the estimation of the regions of attraction. This illustrates the limitations of numerical methods when applied to such type of highly non-linear equations, and the need to use exact polynomial system solving methods, even at a larger computational cost.

Over the next sections we describe two strategies to simplify the system of equations (5.20), which allowed us to reduce the computation time significantly, and to solve a wider variety of examples. One of these techniques relies on exploiting the symmetries of Lemma 5.3 by defining a change of variables invariant to this symmetry, and applying algebraic elimination theory through Gröbner bases. The other can be applied when all the reference points lie on the same plane, and it uses the Jacobian criterion to reduce the multiplicity of the solutions and therefore the total degree of the polynomial ideal. The results of these computations are shown in Section 5.4.

5.3.3 Exploiting the symmetries

As mentioned above, the set of solutions of the system $\mathbf{F}(\mathbf{s}, \mathbf{Z}) = \mathbf{0}$ in (5.20) presents point symmetry through the camera centre C; in other words, they are invariant to a transformation $\mathbf{Z} \mapsto -\mathbf{Z}$. As a consequence, half of the complex solutions (with non-zero Z coordinates) are trivial to compute if the other half are known. Furthermore, the reflected solutions are spurious, and do not correspond to a rigid body transformation $\mathbf{p} \in SE(3)$, but to a reflection $\widehat{\mathbf{T}} \in E^{-}(3)$.

In this section we detail how to exploit this property to our advantage, by deriving a new system that does not present this symmetry, where the total degree (the total number of solutions when counted with multiplicity) is half that of the original system, and which can be solved in much less time. This is done by defining a new set of coordinates that are invariant to the symmetry of the system, and deriving a set of equations in the new variables by means of Gröbner bases and algebraic elimination theory.

Let us define the following transformation, that maps the $\mathbf{Z}$ coordinates to new variables $\boldsymbol{\theta} \in \mathbb{R}^{\frac{(N-1)N}{2}}$:

$$\mathbf{Z} \mapsto \boldsymbol{\theta} = \left(\theta_{12}, \theta_{13}, \ldots, \theta_{(N-1)N}\right) \quad \text{where} \quad \theta_{ij} = Z_i Z_j. \tag{5.21}$$

This change of coordinates is invariant to the symmetry described above: $\boldsymbol{\theta}(\mathbf{Z}) = \boldsymbol{\theta}(-\mathbf{Z})$.

Therefore by projecting the solutions of the original system on the space of variables $(\mathbf{s}, \boldsymbol{\theta})$, each solution $(\mathbf{s}, \mathbf{Z})$ and its reflection $(\mathbf{s}, -\mathbf{Z})$ are mapped onto the same point.

From the definition of θ_{ij} we trivially obtain relations between the old and new variables of the form $\theta_{ij} - Z_i Z_j = 0$. For every polynomial in the original system $f_i \in \mathbf{F}(\mathbf{s}, \mathbf{Z})$, we consider the ideal formed by appending all these relations, along with the constraint $1 - \ell\, Z_1 \dots Z_N = 0$, where ℓ is an auxiliary variable, to enforce $Z_i \neq 0$ for all i:

$$I_i = \langle f_i,\ 1 - \ell\ Z_1 \dots Z_N,\ \theta_{12} - Z_1 Z_2, \theta_{13} - Z_1 Z_3, \dots \rangle \subseteq \mathbb{Q}[\ell, \mathbf{s}, \mathbf{Z}, \boldsymbol{\theta}]. \tag{5.22}$$

The ideal I_i is then projected onto the space of variables $(\mathbf{s}, \boldsymbol{\theta})$. This is done by computing a Gröbner Basis G_i for (5.22) with respect to an elimination order with $[\ell, \mathbf{Z}] \succ [\mathbf{s}, \boldsymbol{\theta}]$.

$$G_i \leftarrow I_i \cap \mathbb{Q}[\mathbf{s}, \boldsymbol{\theta}] \tag{5.23}$$

The set union of all the Gröbner bases obtained in this form defines the new system of equations in variables $(\mathbf{s}, \boldsymbol{\theta})$. However the resulting variety has a positive dimensional component that satisfies $\boldsymbol{\theta} = \mathbf{0}$, and corresponds to degenerate solutions with $Z_i = 0$ for some i. To remove this positive dimensional component we introduce again a constraint of the form $1 - \ell\ \theta_{12} \dots \theta_{1N} = 0$. The new system is then

$$\mathbf{G}(\mathbf{s}, \boldsymbol{\theta}) = (G_1,\, G_2,\, \dots,\, G_s,\, 1 - \ell\ \theta_{12} \dots \theta_{1N}) \tag{5.24}$$

The new system (5.24) contains more equations and variables (see the Table 5.2 in Section 5.3.4), but half the total degree and number of solutions of the ideal generated by the original equations $\mathbf{F}(\mathbf{s}, \mathbf{Z})$. We observe that, using this new formulation, the computation time is reduced by more than an order of magnitude in most cases, as will be shown in Section 5.4.

Once a solution $(\mathbf{s}, \boldsymbol{\theta})_{crit}$ for (5.24) has been found, it is necessary to reproject it in the space of variables $(\mathbf{s}, \mathbf{Z})$. This is done easily by

$$Z_i = \sqrt{\frac{\theta_{ij}\theta_{ik}}{\theta_{jk}}} \quad \text{for } i,\ j,\ k \text{ not equal.} \tag{5.25}$$

In principle, to every solution $(\mathbf{s}, \boldsymbol{\theta})_{crit}$ correspond two points $(\mathbf{s}, \mathbf{Z})$ and $(\mathbf{s}, -\mathbf{Z})$, and it is not possible to decide *a priori* which of the two corresponds to a rigid body transformation $\mathbf{p}_{crit} \in SE(3)$ and which one is the mirrored solution, so one needs to compute both and then classify them. However, the only solutions of interest are those with $Z_i > 0$ for all i (hence $\theta_{ij} > 0$ for all i and j); that is, those for which all the reference points lie in the semispace that is in front of the camera, so a large number of solutions can be disregarded at this stage.

5.3.4 Coplanar points

In this section we report on a computational improvement that can be applied when the observed points lie on the same plane, and which allows to divide the total degree (the total number of solutions counted with multiplicity) of the system of equations by two, in the case of $N = 4$, or by four in the case of $N = 5$.

The case of coplanar reference points merits a special focus, not only because planar images and targets are the most commonly used markers in visual servoing applications, but also because the equations that describe the critical points of IBVS present interesting properties that can be exploited. In particular, the improvements shown here are motivated by an empirical remark made during the computation of the critical points for $N = 4$ and $N = 5$ coplanar points. For these cases, the ideal generated by equations (5.20) seems to be always non-radical; that is, there is at least one solution with a multiplicity greater than 1. Specifically, we observe that, for $N = 4$, all the solutions have a multiplicity equal to 2, while for $N = 5$, all solutions have multiplicity 4.

Conjecture 5.4. *When the N feature points are contained in the same plane, the zero-dimensional ideal generated by (5.20), describing the critical points of the error potential $V = \frac{1}{2}\mathbf{e}^T\mathbf{e}$, is not radical (i.e. at least one solution has a multiplicity greater than 1). On the other hand, when the N points are all not coplanar, we observe that ideal generated by the polynomials (5.20) is always radical.*

The reason that this multiplicity appears is that planar objects present another kind of symmetry: with respect to the plane they are contained in. As a consequence, for every value $(\mathbf{s}, \mathbf{Z})$, there are two isometries $\mathbf{T} \in E(3)$ that satisfy (5.19): one corresponding to a feasible camera configuration and one corresponding to its reflection about the plane of the object. Note that this phenomenon is not the same as the invariance of the system of equations (5.20) with respect to the transformation $(\mathbf{s}, \mathbf{Z}) \mapsto (\mathbf{s}, -\mathbf{Z})$, which occurs for both planar and generic objects. In Section 5.3.5 we explain how to compute the correct camera configuration $\mathbf{p}_{crit} \in SE(3)$ from the solutions $(\mathbf{s}, \mathbf{Z})_{crit}$ of the system (5.20).

Let us consider four spatial points $\mathcal{P}_i$, $i = 1, \ldots, 4$, with homogeneous coordinates $\mathbf{X}_i = (X_i \; Y_i \; Z_i \; 1)^T$ in the reference frame of the camera. The determinant of the (4×4) matrix which has the homogeneous coordinates of the points as its columns is six times the volume of the tetrahedron defined by these points:

$$J_1 = \begin{vmatrix} X_1 & X_2 & X_3 & X_4 \\ Y_1 & Y_2 & Y_3 & Y_4 \\ Z_1 & Z_2 & Z_3 & Z_4 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 6 \cdot V_{tetrahedron} \tag{5.26}$$

After substituting $X_i = x_i Z_i$ and $Y_i = y_i Z_i$, this determinant is a polynomial of degree 5 in variables $(\mathbf{s}, \mathbf{Z})$, and it vanishes if and only if the four points lie on the same plane. We can therefore make use of this extra polynomial relation $J_1 = 0$ for the computation of the equilibria when the object is planar. This condition is also known as the Grassmann incidence condition of the four points.

In the case of $N = 4$ points, we observe that by including the condition that $J_1 = 0$ in our system, we obtain an ideal which is radical: with the same number of solutions, but half the total degree of the original system, and a reduction in the computation time of more than an order of magnitude (see Section 5.4).

Proposition 5.5. *Consider an IBVS scheme from $N = 4$ points. If the four points are coplanar, they satisfy that the polynomial $J_1(\mathbf{s}, \mathbf{Z})$, defined as the determinant (5.26), is zero.*

This condition can be exploited in the computation of the critical points of the function V from the system (5.20)

An intuition on why this improvement is achieved comes from the following theorem, which is a corollary of the Jacobian Criterion, stated in a more evolved form in ([Eis13], Theorem 16.19).

Theorem 5.6. *(Jacobian Criterion): Let $\mathbf{F} = (f_1, \dots, f_s)$, $f_i \in \mathbb{Q}[x_1, \dots, x_n]$ define a finite number of solutions in $\mathbb{C}^n$, and let $Jac(\mathbf{F})$ denote the Jacobian matrix associated to $\mathbf{F}$ with respect to $(x_1, \dots, x_n)$. Then, if the ideal generated by $\mathbf{F}$ is not radical, there exist points which cancel both $\mathbf{F}$ and the determinants of the minors of size (n, n) of $Jac(\mathbf{F})$.*

For $N = 4$, the Jacobian matrix $\text{Jac}(\mathbf{F}) = \frac{\partial \mathbf{F}}{\partial(\mathbf{s},\mathbf{Z})}$ is of size (12×12), so only its determinant $J = \det(\text{Jac}(\mathbf{F}))$, a polynomial of degree 30, must be considered. We observe that the polynomial J_1 in (5.26) always divides this determinant $(J_1|J)$. When the points are coplanar, all the solutions cancel this factor, making the determinant of the Jacobian vanish. We define the augmented system obtained by including this factor as

$$\mathbf{F}_1(\mathbf{s}, \mathbf{Z}) = [\mathbf{F},\ J_1] = \mathbf{0} \tag{5.27}$$

The solutions of the new system $\mathbf{F}_1(\mathbf{s}, \mathbf{Z})$ still present the point symmetry with respect to the camera centre C from Lemma 5.3, so the change of coordinates detailed there can be applied to (5.27), allowing us to derive a system $\mathbf{G}_1(\mathbf{s}, \boldsymbol{\theta})$ in variables $\theta_{ij} = Z_i Z_j$, where the degree of ideal it generates is further reduced by half. The reductions obtained in the degree are summarized in Table 5.2.

For the case of $N = 5$ coplanar points we observe that all the solutions appear always with a multiplicity of 4. The Jacobian $\text{Jac}(\mathbf{F})$ has in this case dimension (15×15), and its determinant is a polynomial of degree 45. This time, for the 5 points to all be coplanar, there are two polynomials $J_1 = \det([\mathbf{X}_1\ \mathbf{X}_2\ \mathbf{X}_3\ \mathbf{X}_4])$, and $J_2 = \det([\mathbf{X}_1\ \mathbf{X}_2\ \mathbf{X}_3\ \mathbf{X}_5])$ that must vanish, and we find that both J_1 and J_2 are factors of $J = \det(\text{Jac}(\mathbf{F}))$, i.e. $(J_1 \cdot J_2)\,|\,J$. We define a new system of equations by including these two factors $\mathbf{F}_{12} = [\mathbf{F}, J_1, J_2]$. The solutions of $\mathbf{F}_{12} = \mathbf{0}$ are the same as for the original system, but the degree of the ideal it generates is divided by four.

The same approach could be used to simplify the problem for any $N > 3$ coplanar points, and even for configurations where not all the points are on the same plane, but at least four of them are. For instance, for 5 points in a pyramidal shape, the coplanarity condition $J_1 = 0$ can be exploited for the points that form the base.

Proposition 5.7. *Consider an IBVS control scheme from N points $\mathbf{X}_i$, $i = 1, \dots N$. If at least four points $\mathbf{X}_1, \dots, \mathbf{X}_4$ are coplanar, then the polynomial $J_1 = det([\mathbf{X}_1\ \mathbf{X}_2\ \mathbf{X}_3\ \mathbf{X}_4]) = 0$, and this condition can be exploited in the computation of the critical points of V described by* (5.20).

The fact that the ideal generated by equations (5.20) is always non-radical when the object is planar, such that multiple solutions appear, remains a conjecture. However the condition that $\det([\mathbf{X}_1\ \mathbf{X}_2\ \mathbf{X}_3\ \mathbf{X}_4]) = 0$ relies only on the assumption that P_i, $i = 1, \dots, 4$ lie on the

Table 5.2: The polynomial system $\mathbf{G}(\mathbf{s}, \boldsymbol{\theta})$ is obtained by applying the change of variables (5.21) to remove the symmetric solutions. It contains more equations and variables than the original system $\mathbf{F}(\mathbf{s}, \mathbf{Z})$, but half the total degree. For the case of coplanar feature points, $\mathbf{F}_1(\mathbf{s}, \mathbf{Z})$ is the radical system obtained by including the coplanarity constraint as in (5.27), while $\mathbf{G}_1(\mathbf{s}, \boldsymbol{\theta})$ is obtained by applying the coordinate transformation to the latter.

		$\mathbf{F}(\mathbf{s}, \mathbf{Z})$	$\mathbf{G}(\mathbf{s}, \boldsymbol{\theta})$	$\mathbf{F}_1(\mathbf{s}, \mathbf{Z})$	$\mathbf{G}_1(\mathbf{s}, \boldsymbol{\theta})$
$N = 4$	#equations	13	59	14	65
	#variables	12	14	12	14
	degree ($\#\text{sols}_{\mathbb{C}}$)	D	D/2	D/2	D/4
$N = 5$	#equations	16	377	18	389
	#variables	15	20	15	20
	degree ($\#\text{sols}_{\mathbb{C}}$)	D	D/2	D/4	D/8

same plane, such that the algorithmic improvement shown above can always be applied under this condition, and we expect that it will always provide a similar time reduction in the computations. On the other hand, for non-coplanar points, according to our observations, the ideal is generically radical; that is, all solutions appear only with multiplicity 1, and therefore we cannot hope for any such improvement.

5.3.5 Retrieving the camera pose

The problem of computing the critical points of the the IBVS controller is modeled above as a system of equations in the space of variables $(\mathbf{s}, \mathbf{Z})$. Once a solution $(\mathbf{s}, \mathbf{Z})_{crit}$ has been found, using either the original (5.20) or the modified formulation (5.24), we can recover the corresponding critical pose $\mathbf{p}_{crit}$ for the camera, paying attention to the remark above that some of the solutions correspond to a mirrored camera frame, and classify the solutions in local minima, maxima, or saddle points. Note that, while exact computer algebra methods were used to obtain the solutions $(\mathbf{s}, \mathbf{Z})_{crit}$ above, in our method we then use floating point arithmetic to retrieve the camera pose parameters and classify the critical points.

The pose $\mathbf{p}$ can be represented by a (4×4) homogeneous matrix (see Section 2.1)

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{5.28}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$ are a (3×3) rotation matrix, and a position vector respectively. The matrix $\mathbf{T}$ maps the *homogeneous* coordinates of a point $\mathcal{P}_i$ in the world frame $\mathbf{P}_i = (P_{ix}\ P_{iy}\ P_{iz}\ 1)^T$, to its camera-frame coordinates $\mathbf{X}_i = (X_i\ Y_i\ Z_i\ 1)^T$. For each solution $\mathbf{s} = (x_1, y_1, \ldots, x_N, y_N)$ and $\mathbf{Z} = (Z_1, \ldots, Z_N)$, the vector $\mathbf{X}_i$ can be computed for each solution as $\mathbf{X}_i = (x_i Z_i\ y_i Z_i\ Z_i\ 1)^T$.

If we consider N points $\mathbf{P} = [\mathbf{P}_i \ \ldots \ \mathbf{P}_N]$, with camera frame coordinates $\mathbf{X} = [\mathbf{X}_i \ \ldots \ \mathbf{X}_N]$, we have

$$\mathbf{T}\ \mathbf{P} = \mathbf{X}. \tag{5.29}$$

We can reformulate the system (5.29) as a linear system $\widehat{\mathbf{P}}\widehat{\mathbf{T}} = \widehat{\mathbf{X}}$, where now $\widehat{\mathbf{T}}$ is a vector in $\mathbf{R}^{12}$ containing the components of $\mathbf{R}$ and $\mathbf{t}$:

$$\begin{bmatrix} P_{x1} & P_{y1} & P_{z1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & P_{x1} & P_{y1} & P_{z1} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & P_{x1} & P_{y1} & P_{z1} & 0 & 0 & 1 \\ & \ddots & & \ddots & & \ddots & & \ddots & & \\ P_{xN} & P_{yN} & P_{zN} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & P_{xN} & P_{yN} & P_{zN} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & P_{xN} & P_{yN} & P_{zN} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \\ r_{31} \\ r_{32} \\ r_{33} \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} x_1 Z_1 \\ y_1 Z_1 \\ Z_1 \\ \vdots \\ x_N Z_N \\ y_N Z_N \\ Z_N \end{bmatrix}, \tag{5.30}$$

In order to solve (5.30), we must distinguish the two following cases:

1. **Non-coplanar points.** If the N points do not lie on the same plane, then $\mathbf{P}$ and $\mathbf{X}$ are of full-rank, since their determinants are essentially the same as (5.26) (see Section 5.3.4). For $N = 4$, the system (5.30) is square, and the matrix $\widehat{\mathbf{P}}$ is invertible. For $N > 4$, the system is overconstrained, but it will always have a solution if the vector $\widehat{\mathbf{X}}$ is compatible with the geometry of the problem (i.e. compatible with the solutions of our problem $(\mathbf{s}, \mathbf{Z})_{crit}$). It can be solved using any linear algebra method, such as Gaussian reduction.

   Let us call $\widehat{\mathbf{T}}_c$ a solution of (5.30) corresponding to a critical point $(\mathbf{s}, \mathbf{Z})_{crit}$. Then, in principle, $\widehat{\mathbf{T}}_c$ may correspond either to a feasible camera configuration (i.e. $\mathbf{T} \in SE(3)$), or to a pose that is reflected through the camera centre ($\mathbf{T} \in E^-(3)$). It suffices to compute the determinant of the (4×4) matrix $\mathbf{T}$ to remove the mirrored solutions. The solutions that correspond to a true rigid-body transformation will satisfy $\det(\mathbf{T}) = 1$, while the spurious solutions will have $\det(\mathbf{T}) = -1$, and should be removed.

2. **Coplanar points.** For a planar object, the matrix $\widehat{\mathbf{P}}$ is no longer of full-rank, and the system (5.30) does not have a unique solution. In this case, it is necessary to consider the additional constraint for the transformation matrix that $\mathbf{T}^T\mathbf{T} = \mathbf{I}$. Without any loss of generality, we can assume that the points are contained in the plane $Z = 0$ of the world-frame; that is $\mathbf{P}_i = (P_{xi}\ P_{yi}\ 0\ 1)^T$ for all i; then the resulting rigid-body transformation will be defined from this frame to the camera coordinates. Next, the

system (5.30) can be reduced to the $(3N \times 9)$ system $\widehat{\mathbf{P}}'\widehat{\mathbf{T}}' = \widehat{\mathbf{X}}'$:

$$\begin{bmatrix} P_{x1} & P_{y1} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & P_{x1} & P_{y1} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & P_{x1} & P_{y1} & 0 & 0 & 1 \\ & \ddots & & \ddots & & \ddots & & \ddots & \\ P_{xN} & P_{yN} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & P_{xN} & P_{yN} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & P_{xN} & P_{yN} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \\ r_{21} \\ r_{22} \\ r_{31} \\ r_{32} \\ t_1 \\ t_2 \\ t_3 \end{bmatrix} = \begin{bmatrix} x_1 Z_1 \\ y_1 Z_1 \\ Z_1 \\ \vdots \\ x_N Z_N \\ y_N Z_N \\ Z_N \end{bmatrix}, \tag{5.31}$$

The system (5.31) always has a solution if the vector $\widehat{\mathbf{X}}'$ is compatible with the geometry of the problem, and can be solved using linear algebra. Let a solution of the reduced system be

$$\widehat{\mathbf{T}}' = (r_{11}^*, r_{12}^*, r_{21}^*, r_{22}^*, r_{31}^*, r_{32}^*, t_1^*, t_2^*, t_3^*). \tag{5.32}$$

The remaining components of the rotation matrix can be found from the constraint $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, leading to the following three equations in (r_{13}, r_{23}, r_{33}):

$$\begin{aligned} r_{11}^*\, r_{13} + r_{21}^*\, r_{23} + r_{31}^*\, r_{33} &= 0, \\ r_{12}^*\, r_{13} + r_{22}^*\, r_{23} + r_{32}^*\, r_{33} &= 0, \\ {r_{13}}^2 + {r_{23}}^2 + {r_{33}}^2 - 1 &= 0. \end{aligned} \tag{5.33}$$

The system (5.33) always has two solutions: $(r_{13}^*,\ r_{23}^*,\ r_{33}^*)$ and $(-r_{13}^*,\ -r_{23}^*,\ -r_{33}^*)$. One of them will satisfy $\det(\mathbf{T}) = 1$, and correspond to the true rigid-body transformation, and the other will result in $\det(\mathbf{T}) = -1$, and correspond to a solution where the camera frame is mirrored with respect to the plane of the object (see Section 5.3.4).

5.3.6 Classifying the solutions

Once the critical points have been computed in the space of camera configurations, we can classify the solutions in **local minima**, **maxima** or **saddle points**. To do this, we must evaluate the eigenvalues of the Hessian of the potential function V at each of these points. There are multiple ways to do this. Here, as proposed by Alessandro Colotti, we choose to represent the camera orientation from the components of a unit quaternion, and its position by a three dimensional vector. Then, the problem of classifying the critical points of V can be modeled as a constrained second-order optimization problem, with the constraint being the unit-norm of the quaternion.

We will use the following definitions and propositions, which are stated in a more evolved form in [NW99], Section 12. For our purposes we assume that all functions have continuous second-order partial derivatives.

Suppose we wish to identify the local minima of a scalar function $f : \mathbb{R}^n \mapsto \mathbb{R}$ in variables $\mathbf{x} \in \mathbb{R}^n$ subject to a set of equality constraints $g_i(\mathbf{x}) = 0$. Let $\mathbf{x}^*$ be a point satisfying

$$\nabla f(\mathbf{x}^*) = \sum_i \lambda_i \nabla g_i(\mathbf{x}^*) \tag{5.34}$$

for some scalars $\lambda_i > 0$. Then we say that $\mathbf{x}^*$ is a *critical point* of f with respect to those constraints; in other words, $\mathbf{x}^*$ is a candidate for being a local minimizer of f. The equations (5.34) are called the *first-order optimality conditions.*

Given the above, we define the *critical cone* as the set

$$C(\mathbf{x}^*) = \left\{ w \in \mathbb{R}^7 \mid \nabla g_i(\mathbf{x}^*)^T w = 0 \text{ for all } i \right\} \tag{5.35}$$

The critical cone defines the possible directions around the point $\mathbf{x}^*$ compatible with the constraints $g_i(\mathbf{x}) = 0$.

Then, from Theorem 12.5 of [NW99], we can derive the following proposition, leading to the *second-order optimality conditions.*

Proposition 5.8. *Let* $\mathrm{H}(f)$ *denote the Hessian of the function* f *with respect to variables* $\mathbf{x}$ *(i.e. the matrix of second-order partial derivatives). The point* $\mathbf{x}^*$ *is a local minimum of the constrained optimization problem if*

$$w^T \mathrm{H}(f)\, w > 0 \quad \textit{for all } w \in C(\mathbf{x}^*), \tag{5.36}$$

Let us return to the case at hand. The function $V = \frac{1}{2}\mathbf{e}^T\mathbf{e}$ representing the error norm can be regarded as a function defined in $\mathbb{R}^7$, in the components of the vectors $\mathbf{t} = [t_1\ t_2\ t_3]$ and $\mathbf{q} = [q_0\ q_1\ q_2\ q_3]^T$, with the variables subject to the constraint

$$c(\mathbf{q}) := q_0^2 + q_1^2 + q_2^2 + q_3^2 - 1 = 0. \tag{5.37}$$

To obtain the closed form of V in terms of $\mathbf{t}$ and $\mathbf{q}$, it suffices to substitute the representation of a rotation matrix in terms of the quaternion components (2.9) in (5.3) and (5.17).

Our candidate solutions are the points $(\mathbf{s}, \mathbf{Z})_{crit}$ computed by solving any of the systems proposed in (5.20), (5.24), (5.27). For every solution $(\mathbf{s}, \mathbf{Z})$, we compute the components of the position vector $\mathbf{t}$, and of the rotation matrix $\mathbf{R}$ as described in Section 5.3.5 by solving the linear system (5.31) or (5.33). Then, we compute the $\theta\mathbf{u}$ representation for the camera orientation (by an angle and an axis of rotation, see Section 2.1.2). The vector $\mathbf{u}$ is the eigenvector of $\mathbf{R}$ corresponding to an eigenvalue equal to 1. Then, the angle is computed as $\theta = \operatorname{acos}\left(\frac{\operatorname{tr}(\mathbf{R})-1}{2}\right)$. From the $\theta\mathbf{u}$-vector, we get the quaternion components as $q_0 = \cos(\theta/2)$, and $[q_1\ q_2\ q_3]^T = \mathbf{u} \cdot \sin(\theta/2)$. We call $(\mathbf{t}_c, \mathbf{q}_c)$ the critical points expressed in the space of the new variables.

Let us denote the Hessian of V with respect to the variables $(\mathbf{t}, \mathbf{q})$ by $\mathrm{H}(V)$, and let us define the matrix $\mathbf{Z}_c$ as a basis for the nullspace of the gradient of the constraint (5.37) with respect to $(\mathbf{t}, \mathbf{q})$:

$$\mathbf{Z}_c = [\mathbf{w}_1, \ldots, \mathbf{w}_6] \in \mathbb{R}^{7\times 6} : \quad \nabla c \cdot \mathbf{w}_i = 0 \quad \text{for all } i \tag{5.38}$$

with $\mathbf{w}_i$ linearly independent.

Then, by Proposition 5.8, a point $(\mathbf{t}_c, \mathbf{q}_c)$ is a **local minimum** of V if

$$\mathbf{Z}_c(\mathbf{t}_c, \mathbf{q}_c)^T \cdot \mathrm{H}(V)(\mathbf{t}_c, \mathbf{q}_c) \cdot \mathbf{Z}_c(\mathbf{t}_c, \mathbf{q}_c) \succ 0. \tag{5.39}$$

In other words, if the matrix (5.39) is positive-definite.

Another interpretation of (5.39) is as follows. Since the camera pose can be described by six independent parameters, and we are instead using seven parameters bound by a constraint, then the matrix Z_c represents a change of basis for the Hessian of V, that reduces the dimension of $\mathrm{H}(V)_{(\mathbf{t},\mathbf{q})}$ from (7×7) to the correct dimension (6×6). Then, we say that the matrix $\mathbf{Z}_c^T \mathbf{H}(V) \mathbf{Z}_c$ is the *reduced Hessian* of V. The local minima of our problem correspond to the values $(\mathbf{t}_c, \mathbf{q}_c)$ for which the eigenvalues of (5.39) have all a positive real part. Note that the computation of the eigenvalues of the matrix (5.39) is done in floating point arithmetic in our method.

5.4 Results

5.4.1 Computations

Section 5.3 describes how to compute the critical points of IBVS from N feature points as the real solutions of a zero-dimensional system of polynomial equations, where the parameters are the distances between each two points $\mathbf{d} = (d_{12}, \dots d_{(N-1)N})$ and the target value of the visual features $\mathbf{s}^\star = (x_1^\star, y_1^\star, \dots, x_N^\star, y_N^\star)$.

In this section, `msolve` was used to compute the solutions for a number of different examples of $N = 4$ points, both in generic, and in structured configurations; the computing times are compared using the different formulations detailed in Sections 5.3.1, 5.3.3 and 5.3.4. The results are later compared with results obtained from homotopy, using the Homotopy Continuation package for `Julia`. The computations are included in the repository jorge-gf/thesis-archive [GF22].

Table 5.3 displays the results for several configurations of $N = 4$ points, both planar and non-planar. We show the degree of the ideal generated by the original system of equations (5.20) (that is, the total number of complex solutions counted with multiplicity), and the number of different complex and real solutions. A brief description of the systems (the geometry of the points, and the desired final pose of the camera) is given, with the values of parameters $\mathbf{d}$ and $\mathbf{s}^\star$, in Table 5.4. The first column of the timings in Table 5.3 refers to the original system of equations $\mathbf{F}(\mathbf{s}, \mathbf{Z}) = \mathbf{0}$, containing 12 variables and 12 equations, described in (5.20), and contains the same results shown before in Table 5.1. The second column contains the timings for the system $\mathbf{G}(\mathbf{s}, \boldsymbol{\theta}) = \mathbf{0}$ in (5.24), derived by reducing the symmetries of the polynomial. The last two columns concern only the cases where the observed object is planar, and we solve the augmented system $\mathbf{F}_1 = [\mathbf{F}, J_1] = \mathbf{0}$ from (5.27), where the coplanarity condition (5.26) is included to remove the multiplicity of the solutions, and the system $\mathbf{G}_1(\mathbf{s}, \boldsymbol{\theta}) = \mathbf{0}$, after the change of variables. In the first of the two columns, the solutions are computed in the space of the original variables $(\mathbf{s}, \mathbf{Z})$, and include the mirrored, non-feasible solutions. In the second two it is in terms of variables $(\mathbf{s}, \boldsymbol{\theta})$, and only half of the solutions are computed. The computations were all performed on 12 cores on a machine equipped with an Intel Xeon Gold 6246R CPU (3.40GHz) and 1.5 TB RAM.

Between different configurations, the large variations in the computing times are mainly due to the different bit-size coefficients in the systems. Overall, for generic, non-planar objects,

we can compute the solutions in the order of 1-3 days, by using the formulation $\mathbf{G}(\mathbf{s}, \boldsymbol{\theta}) = 0$ from (5.24). This is an improvement of at least an order of magnitude with respect to trying to solve the original equations (5.20). For the coplanar cases, exploiting the coplanarity condition (5.26) and solving $\mathbf{F}_1(\mathbf{s}, \mathbf{Z}) = 0$ allows us to compute the critical points in less than one day for all the examples, in the space of the original variables $(\mathbf{s}, \mathbf{Z})$. If the change of variables (5.21) is applied to compute the system $\mathbf{G}_1(\mathbf{s}, \mathbf{Z}) = 0$, these times are reduced further, to the order of a few minutes to a few hours.

Table 5.6 presents the results obtained when computing the critical points using the Homotopy Continuation package implemented in `Julia`. We show the true number of real and complex solutions of the systems, compared against the number of solutions computed numerically.

The cells in red are cases for which homotopy fails to compute the right number of solutions. As before, homotopy can typically miss complex solutions when more than one are closely spaced; but it can also interpret solutions with multiplicity as different solutions, or complex solutions with small imaginary parts can be misinterpreted as real solutions.

We find that, by solving the system $\mathbf{F}_1(\mathbf{s}, \mathbf{Z})$ in (5.27) including the coplanarity constraint, in the cases where this is possible, homotopy tends to correctly identify more solutions than before, although still missing a large number of them for many cases. We also tried to solve the equations $\mathbf{G}(\mathbf{s}, \boldsymbol{\theta}) = \mathbf{0}$ from (5.24), but the homotopy package fails altogether at the initialization of the computation. This is because the reformulated system contains many more equations than variables. Homotopy methods are better suited for square systems (with the same number of variables and equations), but tend to perform worse for overconstrained systems.

We tried to compute the critical points in the case of IBVS from 5 points by exploiting the techniques from Sections 5.3.3 and 5.3.4, both for generic and for planar configurations. However for $N = 5$ the systems of equations become much more computationally difficult to solve. In the case of $N = 5$ coplanar points, the degree of the resulting ideal (after all the corresponding reductions) can be at least up to 2440, while for generic points it can be above 8600, and more variables are involved (Table 5.2). For the moment all of our attempts have ended with the computer running out of memory; however we expect to be able to solve the case $N = 5$ if we use a machine with more memory capacity.

Table 5.3: Computation of the critical points of IBVS using $N = 4$ points for different configurations using `msolve`. "Degree" is the total degree of the original system of equations. "#sols$_{\mathbb{C}}$" and "#sols$_{\mathbb{R}}$" are the number of different solutions after the symmetric points are removed. The computation times are compared for the original system of equations and the improvements from Sections 5.3.3 and 5.3.4: the change of variables invariant to the point symmetry, and the coplanarity constraint. All the computations were performed using 12 cores.

	System		Solutions**		Time (×12 cores)			
	Description	Degree*	#sols$_{\mathbb{C}}$	#sols$_{\mathbb{R}}$	$\mathbf{F}(\mathbf{s},\mathbf{Z})$	$\mathbf{G}(\mathbf{s},\boldsymbol{\theta})$	$\mathbf{F}_1(\mathbf{s},\mathbf{Z})$	$\mathbf{G}_1(\mathbf{s},\boldsymbol{\theta})$
Planar	1 square1	804	201	25	15 days	48.6 h	478 s	172 s
	2 square2	2000	500	26		43.5 h	8278 s	582 s
	3 square3	2032	508	22	24 days	52.4 h	21.2 h	10243 s
	4 square4	2032	508	24		44.1 h	29.4 h	27.2 h
	5 rectangle1	1560	390	26		11.2 h	3.3 h	1426 s
	6 rectangle2	1560	390	44		41 h	8671 s	2167 s
	7 rectangle3	2128	532	24	27 days	38.3 h	18.4 h	8035 s
	8 lozenge1	1608	402	18		192.8 h	2408 s	889 s
	9 genericPlanar1	2176	544	20		74.8 h	33.2 h	10 h
	10 genericPlanar2	2176	544	16		76.7 h	36.1 h	12.3 h
	11 genericPlanar3	2176	544	24		77.2 h	34.8 h	11.3 h
Non planar	12 tetrahedron1	3608	1804	68		66 h	-	-
	13 generic1	3656	1828	42	41 days	26 h	-	-
	14 generic2	3656	1828	42		104.7 h	-	-
	15 generic3	3656	1828	48		108 h	-	-

* Degree of original system $\mathbf{F}(\mathbf{s},\mathbf{Z})$ before any reductions.

** After removing the mirrored solutions from Lemma 5.3.

Table 5.4: Example configurations for the critical points of IBVS with $N = 4$ points.

	System	Description	Parameters
Planar	1 square1	Parallel to image plane.	$\mathbf{d} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}$
	2 square2	View from an angle	$\mathbf{d} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} -\frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & \frac{4}{5} & \frac{4}{5} \end{bmatrix}$
	3 square3	Generic orientation.	$\mathbf{d} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} \frac{19471}{581794} & \frac{404051}{1005316} & -\frac{19471}{841726} & -\frac{404051}{418204} \\ -\frac{242923}{290897} & -\frac{15581}{251329} & \frac{242923}{420863} & \frac{15581}{104551} \end{bmatrix}$
	4 square4	View from an angle	$\mathbf{d} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 1 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} \frac{7}{913} & \frac{3}{22} & -\frac{13}{1869} & -\frac{25}{147} \\ -\frac{53}{277} & -\frac{15}{713} & \frac{21}{121} & \frac{9}{343} \end{bmatrix}$
	5 rectangle1	Parallel to image plane.	$\mathbf{d} = \begin{bmatrix} \frac{49}{2500} & \frac{37}{1250} & \frac{1}{100} & \frac{1}{100} & \frac{37}{1250} & \frac{49}{2500} \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} -\frac{7}{30} & \frac{7}{30} & \frac{7}{30} & -\frac{7}{30} \\ \frac{1}{6} & \frac{1}{6} & -\frac{1}{6} & -\frac{1}{6} \end{bmatrix}$
	6 rectangle2	Parallel to image plane.	$\mathbf{d} = \begin{bmatrix} 400 & \frac{6401}{16} & \frac{1}{16} & \frac{1}{16} & \frac{6401}{16} & 400 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} -10 & 10 & 10 & -10 \\ \frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \end{bmatrix}$
	7 rectangle3	Generic orientation.	$\mathbf{d} = \begin{bmatrix} 400 & \frac{6401}{16} & \frac{1}{16} & \frac{1}{16} & \frac{6401}{16} & 400 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} \frac{144}{133} & \frac{62}{79} & \frac{87}{113} & \frac{791}{719} \\ \frac{43}{35} & \frac{281}{335} & \frac{76}{87} & \frac{87}{74} \end{bmatrix}$
	8 lozenge1	Parallel to image plane.	$\mathbf{d} = \begin{bmatrix} 5 & 4 & 5 & 5 & 16 & 5 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -2 & 0 & 2 \end{bmatrix}$

Table 5.5: Example configurations for the critical points of IBVS with $N = 4$ points - continued.

	System	Description	Parameters
Planar	9 genericPlanar1	Generic orientation.	$\mathbf{d} = \begin{bmatrix} \frac{1}{64} & \frac{45}{4096} & \frac{37}{4096} & \frac{157}{4096} & \frac{117}{4096} & \frac{1}{1024} \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} \frac{3}{32} & \frac{9}{112} & \frac{43}{330} & \frac{39}{304} \\ \frac{5}{64} & \frac{4}{1533} & \frac{115}{3507} & \frac{8}{511} \end{bmatrix}$
	10 genericPlanar2	Generic orientation.	$\mathbf{d} = \begin{bmatrix} \frac{1}{64} & \frac{53}{4096} & \frac{1}{1024} & \frac{229}{4096} & \frac{17}{1024} & \frac{49}{4096} \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} -\frac{1}{16} & -\frac{12}{167} & -\frac{304}{4255} & -\frac{21}{263} \\ -\frac{1}{32} & \frac{41}{568} & -\frac{37}{300} & -\frac{13}{658} \end{bmatrix}$
	11 genericPlanar3	Generic orientation.	$\mathbf{d} = \begin{bmatrix} \frac{1}{64} & \frac{41}{4096} & \frac{113}{4096} & \frac{41}{4096} & \frac{49}{4096} & \frac{5}{128} \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} 0 & \frac{67}{3383} & \frac{10}{121} & -\frac{46}{531} \\ -\frac{3}{32} & -\frac{34}{157} & -\frac{57}{407} & -\frac{45}{187} \end{bmatrix}$
Non planar	12 tetrahedron1	Generic orientation.	$\mathbf{d} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} -\frac{27}{211} & \frac{58}{179} & -\frac{61}{154} & -\frac{17}{57} \\ -\frac{48}{71} & \frac{71}{1288} & \frac{81}{158} & -\frac{25}{217} \end{bmatrix}$
	13 generic1	Generic points and orientation.	$\mathbf{d} = \begin{bmatrix} 1 & \frac{17}{32} & \frac{123}{128} & \frac{9}{32} & \frac{171}{128} & \frac{57}{128} \end{bmatrix}$ $\mathbf{s}^* = \begin{bmatrix} \frac{9}{34} & \frac{1699}{2550} & \frac{263}{510} & \frac{199}{446} \\ -\frac{443}{2550} & -\frac{443}{2550} & -\frac{827}{2550} & -\frac{1339}{2230} \end{bmatrix}$
	14 generic2	Generic points and orientation.	$\mathbf{d} = \begin{bmatrix} 1 & \frac{1}{32} & \frac{69}{4096} & \frac{25}{32} & \frac{3269}{4096} & \frac{117}{4096} \end{bmatrix}$ $\mathbf{s}^\star = \begin{bmatrix} \frac{1}{64} & \frac{63}{269} & \frac{55}{413} & \frac{29}{533} \\ \frac{1}{64} & -\frac{84}{61} & -\frac{22}{329} & -\frac{13}{127} \end{bmatrix}$
	15 generic3	Generic points and orientation.	$\mathbf{d} = \begin{bmatrix} 1 & \frac{25}{4096} & \frac{109}{4096} & \frac{3737}{4096} & \frac{3437}{4096} & \frac{81}{2048} \end{bmatrix}$ $\mathbf{s}^\star = \begin{bmatrix} -\frac{1}{16} & \frac{2}{6547} & -\frac{41}{596} & \frac{5}{326} \\ -\frac{1}{32} & -\frac{127}{84} & -\frac{21}{223} & -\frac{27}{320} \end{bmatrix}$

Table 5.6: Results from the Homotopy Continuation (**HC**) package for `Julia` and comparison with the true solutions. In red are indicated the systems for which `Julia` fails to capture the correct number of solutions. For the overconstrained systems (5.24) obtained after the change of variables, the homotopy methods fail at initialization.

System	True*		Original system $\mathbf{F}(\mathbf{s},\mathbf{Z})=\mathbf{0}$			Coplanarity condition $\mathbf{F}_1(\mathbf{s},\mathbf{Z})=\mathbf{0}$			Symmetrized (+ radical) $\mathbf{G}(\mathbf{s},\boldsymbol{\theta})=\mathbf{0}$
	$\#\text{sols}_{\mathbb{C}}$	$\#\text{sols}_{\mathbb{R}}$	$\#\text{sols}_{\mathbb{C}}$	$\#\text{sols}_{\mathbb{R}}$	Time	$\#\text{sols}_{\mathbb{C}}$	$\#\text{sols}_{\mathbb{R}}$	Time	
1 square1	402	50	403	50	1630 s	402	50	14499 s	N/A
2 square2	1000	52	1066	49	1353 s	994	52	15316 s	N/A
3 square3	1016	44	1069	43	1495 s	1016	44	15480 s	N/A
4 square4	1016	48	1120	48	1772 s	1012	48	22812 s	N/A
5 rectangle1	780	52	821	27	1968 s	776	48	23914 s	N/A
6 rectangle2	780	88	728	65	2296 s	623	72	19391 s	N/A
7 rectangle3	1064	48	1088	31	1950 s	871	32	20099 s	N/A
8 lozenge1	802	36	806	36	1499 s	803	36	14314 s	N/A
9 genericPlanar1	1088	40	1307	13	2191 s	1025	32	22616 s	N/A
10 genericPlanar2	1088	32	1347	10	2252 s	960	16	25476 s	N/A
11 genericPlanar3	1088	48	1223	15	1934 s	1066	42	24251 s	N/A
12 tetrahedron1	3608	136	3608	136	1500 s	-			N/A
13 generic1	3656	84	3656	84	1644 s	-			N/A
14 generic2	3656	84	3537	95	2280 s	-			N/A
15 generic3	3656	96	3548	96	2162 s	-			N/A

* Number of solutions **including the mirrored solutions** from Lemma 5.3.

5.4.2 Case studies

Example 1. Four points forming a square

Consider the example 1 (square1) from Table 5.4. The four points are at the corners of a square of size 1, and the desired camera pose is located centred over the square and with the image plane parallel to it, defined by $\mathbf{R}^\star$ and $\mathbf{t}^\star$:

$$\mathbf{R}^\star = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{t}^\star = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{5.40}$$

The values of the parameters $\mathbf{d}$ and $\mathbf{s}^\star$ are

$$\mathbf{d} = [1\ 2\ 1\ 1\ 2\ 1], \quad \mathbf{s}^\star = \begin{bmatrix} \mathbf{x}^\star \\ \mathbf{y}^\star \end{bmatrix} = \begin{bmatrix} -1/2 & 1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \end{bmatrix}.$$

For this configuration, we find, after having removed the symmetric solutions from Lemma 5.3, a total of 201 complex solutions, of which 25 are real. However, upon studying which of these solutions lie *in front* of the camera, that is, in the semi space defined by $Z_i > 0$ for all i, we find that there is only one: the global minimum, with value

$$\mathbf{s} = \mathbf{s}^\star = \begin{bmatrix} -1/2 & 1/2 & 1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & -1/2 \end{bmatrix}, \quad \mathbf{Z} = [\ 1\ 1\ 1\ 1\ ]$$

and corresponding to the desired pose (5.40).

Consider now the same object but a different target camera position, looking down at an angle, corresponding to case 4 (`square4`) in the Table 5.4. The camera pose is described (approximately) by

$$\mathbf{R}^\star = \begin{bmatrix} 0.54030 & -0.59500 & -0.59500 \\ 0.59500 & 0.77015 & -0.22985 \\ 0.59500 & -0.22985 & 0.77015 \end{bmatrix} \quad \text{and} \quad \mathbf{t}^\star = \begin{bmatrix} 0 \\ 0 \\ 15/4 \end{bmatrix} \tag{5.41}$$

In this case we compute a total of 508 complex solutions (after having removed the mirrored solutions), of which 24 are real. By removing the solutions with $Z_i < 0$ for some i, we are left with only four solutions. The values of $(\mathbf{s}, \mathbf{Z})$ are given in Table 5.7.

To each of these solutions corresponds two points in the space of transformations of the Euclidean space, the solutions of the systems (5.31) and (5.33): one is a feasible camera pose facing the object, and the other is its reflection through the plane of the points. Therefore, there are four equilibrium configurations for the camera. By studying the eigenvalues of (5.39), we determine that one of them corresponds to the global minimum, one to a local minimum, and two to saddle points. One of the saddle points is in between the two stable equilibria, while the other is located on the other side of the square. The critical points are displayed in Fig. 5.1.

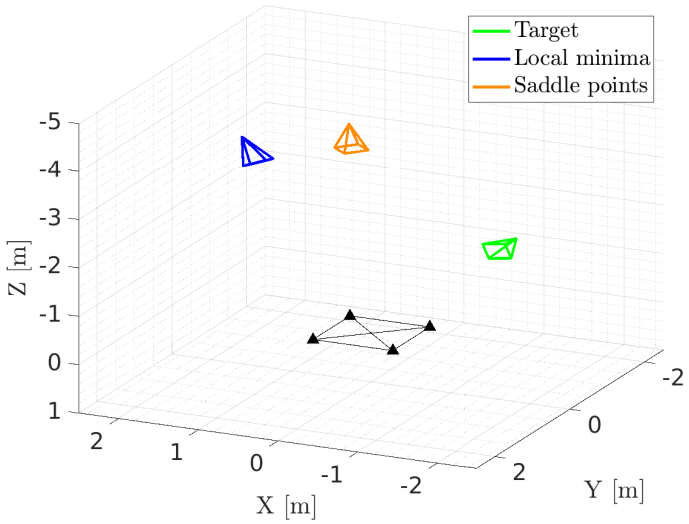


Figure 5.1: Critical points for example 4 (`square4`) from Table 5.4. One saddle point lies between the global minimum and local minimum, while the other is further away on the other side of the object.

Table 5.7: Critical points for the example 4 (`square4`) from Table 5.4.

Sol.	State vars	Camera pose
1. Saddle point	$\mathbf{s} = \begin{bmatrix} -0.017369 & 0.16014 & -0.0015384 & -0.17475 \\ -0.17073 & -0.012038 & 0.16524 & 0.0046934 \end{bmatrix}$ $\mathbf{Z} = [4.2231, 4.1476, 4.1821, 4.2573]$	$\mathbf{t} = \begin{bmatrix} -0.03989 & -0.01497 & 4.20253 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.73756 & -0.67065 & 0.07897 \\ 0.67106 & 0.74097 & 0.02509 \\ -0.07534 & 0.03449 & 0.99656 \end{bmatrix}$
2. Local min.	$\mathbf{s} = \begin{bmatrix} -0.013019 & 0.15333 & -0.018658 & -0.15545 \\ -0.17753 & -0.026060 & 0.17938 & 0.011110 \end{bmatrix}$ $\mathbf{Z} = [4.0238, 3.5108, 3.7354, 4.2482]$	$\mathbf{t} = \begin{bmatrix} -0.06104 & -0.02215 & 3.87952 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.59070 & -0.60801 & 0.53046 \\ 0.62284 & 0.76153 & 0.17929 \\ -0.51297 & 0.22449 & 0.82853 \end{bmatrix}$
3. Saddle point	$\mathbf{s} = \begin{bmatrix} 0.0072396 & -0.022847 & -0.023702 & 0.0063805 \\ -0.017752 & -0.018614 & 0.011474 & 0.012328 \end{bmatrix}$ $\mathbf{Z} = [33.230, 33.222, 33.225, 33.233]$	$\mathbf{t} = \begin{bmatrix} -0.27348 & -0.10433 & 33.22705 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} -0.99956 & -0.02852 & 0.00808 \\ -0.02849 & 0.99958 & 0.00444 \\ -0.00820 & 0.00421 & -0.99996 \end{bmatrix}$
4. Global min.	$\mathbf{s} = \begin{bmatrix} 0.0076676 & 0.13636 & -0.0069569 & -0.17007 \\ -0.19133 & -0.021044 & 0.17356 & 0.026236 \end{bmatrix}$ $\mathbf{Z} = [3.5676, 4.1626, 3.9328, 3.3378]$	$\mathbf{t} = \begin{bmatrix} -0.0 & -0.00001 & 3.75017 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.54028 & -0.59499 & -0.59505 \\ 0.59499 & 0.77016 & -0.22986 \\ 0.59505 & -0.22986 & 0.77012 \end{bmatrix}$

Example 2. Generic planar object

Let us now look at the case 11 (`genericPlanar3`) from Table 5.4. The four points have generic coordinates but are contained in the same plane, and the camera final camera position has coordinates:

$$\mathbf{R}^\star = \begin{bmatrix} -0.4237 & 0.8458 & -0.3242 \\ -0.9027 & -0.4237 & 0.07442 \\ -0.07442 & 0.3242 & 0.9431 \end{bmatrix}, \quad \mathbf{t}^\star = \begin{bmatrix} 0 \\ -\frac{3}{32} \\ 1 \end{bmatrix}. \tag{5.42}$$

The system parameters are

$$\mathbf{d} = \begin{bmatrix} \frac{1}{64} & \frac{41}{4096} & \frac{113}{4096} & \frac{41}{4096} & \frac{49}{4096} & \frac{5}{128} \end{bmatrix}, \quad \mathbf{s}^\star = \begin{bmatrix} 0 & \frac{67}{3383} & \frac{10}{121} & -\frac{46}{531} \\ -\frac{3}{32} & -\frac{34}{157} & -\frac{57}{407} & -\frac{45}{187} \end{bmatrix} \tag{5.43}$$

The critical points for this configuration were obtained by solving the equations $\mathbf{G}_1(\mathbf{s}, \boldsymbol{\theta}) = \mathbf{0}$ in the variables $(\mathbf{s}, \boldsymbol{\theta})$. We find 544 complex solutions in total, of which 24 are real. We retain only those solutions in the positive semispace (with $\theta_{ij} > 0$ for all i and j), of which there is four, with the values shown in Table (5.8).

We then recover the camera configurations corresponding to these points by solving the systems (5.31) and (5.33). We always find two solutions: one corresponding to a rigid-body transformation, and one to its reflection about the plane of the object, which is discarded. Finally, from the eigenvalues of (5.39), we conclude that the four solutions correspond to: the global minimum, with coordinates (5.42), a local minimum and two saddle points, one of which is located between the two stable equilibria, and the other which lies on the other side of the object (see Fig. (5.2)).

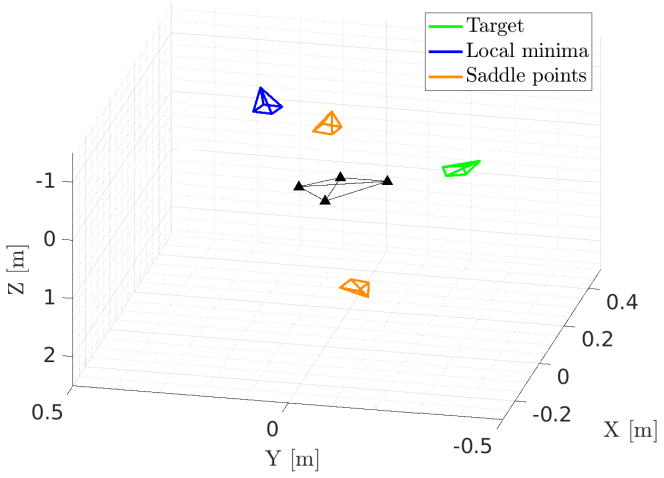


Figure 5.2: Critical points for the configuration 11 (`genericPlanar3`) from Table 5.4 - Four coplanar points in a generic configuration.

Table 5.8: Critical points for the configuration 11 (`genericPlanar3`) from Table 5.4.

Sol.	State vars	Camera pose
1. Global min.	$\mathbf{s} = \begin{bmatrix} 7.2157e-7 & 0.019801 & 0.082645 & -0.086627 \\ -0.093752 & -0.21657 & -0.14004 & -0.24064 \end{bmatrix}$ $\mathbf{Z} = [1.0, 1.0026, 1.0214, 0.97453]$	$\mathbf{t} = \begin{bmatrix} 0.0 & -0.09375 & 0.99997 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.15882 & 0.95337 & -0.25664 \\ -0.98708 & 0.15887 & -0.02066 \\ 0.02108 & 0.25661 & 0.96629 \end{bmatrix}$
2. Local min.	$\mathbf{s} = \begin{bmatrix} -0.0013218 & 0.018780 & 0.084421 & -0.086093 \\ -0.095159 & -0.21791 & -0.13868 & -0.23928 \end{bmatrix}$ $\mathbf{Z} = [1.0177, 0.97364, 0.98012, 0.99543]$	$\mathbf{t} = \begin{bmatrix} -0.00135 & -0.09684 & 1.01770 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.15704 & 0.95071 & 0.26738 \\ -0.92260 & 0.23782 & -0.30373 \\ -0.35234 & -0.19899 & 0.91447 \end{bmatrix}$
3. Saddle point	$\mathbf{s} = \begin{bmatrix} -0.0028405 & 0.020797 & 0.084555 & -0.086748 \\ -0.095040 & -0.21724 & -0.14009 & -0.23867 \end{bmatrix}$ $\mathbf{Z} = [1.0271, 0.99635, 1.0111, 0.99720]$	$\mathbf{t} = \begin{bmatrix} -0.00292 & -0.09761 & 1.02706 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.18911 & 0.98036 & 0.05589 \\ -0.95073 & 0.19704 & -0.23937 \\ -0.24568 & -0.00787 & 0.96932 \end{bmatrix}$
4. Saddle point	$\mathbf{s} = \begin{bmatrix} 0.044238 & -0.018686 & 0.023372 & -0.034192 \\ -0.17956 & -0.16198 & -0.13121 & -0.21824 \end{bmatrix}$ $\mathbf{Z} = [1.9137, 1.9182, 1.9312, 1.8970]$	$\mathbf{t} = \begin{bmatrix} 0.08466 & -0.34363 & 1.91371 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} -0.96401 & 0.26531 & 0.01711 \\ 0.26340 & 0.94436 & 0.19698 \\ 0.03611 & 0.19440 & -0.98026 \end{bmatrix}$

Example 3. Four generic points and and a generic target camera pose

Consider now the example 13 from Table 5.4, called `generic1`. For this configuration, the four points are in a generic, non-coplanar arrangement; the desired final pose is given by

$$\mathbf{R}^\star = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{t}^\star = \begin{bmatrix} \frac{675}{1024} & -\frac{443}{1024} & \frac{1275}{512} \end{bmatrix}^T, \tag{5.44}$$

giving the following parameters of the system

$$\mathbf{d} = \begin{bmatrix} 1 & \frac{17}{32} & \frac{123}{128} & \frac{9}{32} & \frac{171}{128} & \frac{57}{128} \end{bmatrix} \quad \mathbf{s}^\star = \begin{bmatrix} \frac{9}{34} & \frac{1699}{2550} & \frac{263}{510} & \frac{199}{446} \\ -\frac{443}{2550} & -\frac{443}{2550} & -\frac{827}{2550} & -\frac{1339}{2230} \end{bmatrix}$$

By solving the system $\mathbf{G}(\mathbf{s}, \boldsymbol{\theta}) = \mathbf{0}$ in (5.24), we find a total of 1828 complex solutions (after removing the symmetric solutions), of which 42 are real. After removing the solutions with coordinates $\theta_{ij} = Z_i Z_j < 0$ for any i or j, we are left with only 6 points, which may correspond either to a feasible camera pose or to a reflection about the camera centre. To determine this, we solve the linear system (5.30) and compute the pose parameters $\mathbf{t}$ and $\mathbf{R}$. We find that 4 of the 6 solutions correspond to a true rigid-body transformation, while the other 2 are indirect isometries of the Euclidean space.

Therefore, there are in total four camera configurations that correspond to a critical point of the error potential. In Table 5.9 we give the values $(\mathbf{s}, \mathbf{Z})_{crit}$ for these points, as well as the camera pose parameters. By studying the eigenvalues of the matrix (5.39), we conclude that the 4 solutions correspond to: the global minimum, a local minimum, and two saddle points,

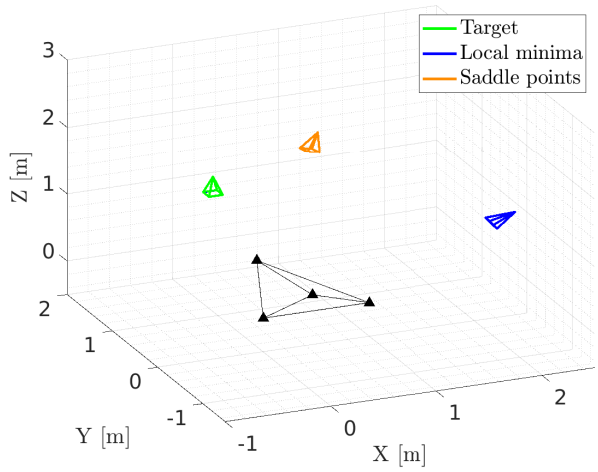


Figure 5.3: Critical points for the example 13 (`generic1`) from Table 5.4. One saddle point lies in between the global minimum and local minimum. Another lies further away on the other side of the object.

one of which is located between the two stable equilibria (see Fig. 5.3), and another that lies much further away from the object.

Table 5.9: Critical points for the configuration 13 (`generic1`) from Table 5.4.

Sol.	State vars	Camera pose
1. Global min.	$\mathbf{s} = \begin{bmatrix} 0.26471 & 0.66627 & 0.51569 & 0.44619 \\ -0.17373 & -0.17373 & -0.32431 & -0.60045 \end{bmatrix}$ $\mathbf{Z} = \begin{bmatrix} 2.49030 & 2.49030 & 2.49030 & 2.17780 \end{bmatrix}$	$\mathbf{t} = \begin{bmatrix} 0.65918 & -0.43262 & 2.49023 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 1. & 0. & 0. \\ 0. & -1. & 0. \\ 0. & 0. & -1 \end{bmatrix}$
2. Local min.	$\mathbf{s} = \begin{bmatrix} 0.25567 & 0.65830 & 0.54271 & 0.43529 \\ -0.17952 & -0.17008 & -0.33201 & -0.59192 \end{bmatrix}$ $\mathbf{Z} = [2.5971, 1.7360, 2.0868, 2.2360]$	$\mathbf{t} = \begin{bmatrix} 0.66398 & -0.46624 & 2.59707 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.47880 & 0.45139 & -0.75299 \\ 0.17099 & -0.88922 & -0.42432 \\ -0.86111 & 0.07441 & -0.50294 \end{bmatrix}$
3. Saddle point	$\mathbf{s} = \begin{bmatrix} 0.25550 & 0.67405 & 0.53338 & 0.42841 \\ -0.20294 & -0.16313 & -0.32637 & -0.58494 \end{bmatrix}$ $\mathbf{Z} = [2.8155, 2.2310, 2.4589, 2.3997]$	$\mathbf{t} = \begin{bmatrix} 0.71935 & -0.57138 & 2.81549 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} 0.78446 & 0.27176 & -0.55746 \\ 0.20743 & -0.96208 & -0.17712 \\ -0.58446 & 0.02331 & -0.81109 \end{bmatrix}$
4. Saddle point	$\mathbf{s} = \begin{bmatrix} 0.49909 & 0.45710 & 0.46513 & 0.46655 \\ -0.31833 & -0.29412 & -0.31588 & -0.34511 \end{bmatrix}$ $\mathbf{Z} = [23.429, 23.961, 23.640, 23.555]$	$\mathbf{t} = \begin{bmatrix} 11.69294 & -7.45794 & 23.42851 \end{bmatrix}$ $\mathbf{R} = \begin{bmatrix} -0.74075 & -0.62597 & 0.24380 \\ 0.41088 & -0.70930 & -0.57277 \\ 0.53147 & -0.32411 & 0.78262 \end{bmatrix}$

Chapter 6

Conclusions and future work

6.1 Conclusions

This PhD thesis dealt with the analysis of the failure cases of Image-Based Visual Servoing methods due to the singularities of the interaction model related to the visual features, or due to issues in the convergence of these controllers arising from the existence of multiple local minima. The main contributions of this thesis are twofold.

First, we provided a complete **singularity analysis** of the **Perspective-4-Line** (P4L) and **Perspective-5-Line** (P5L) problems, extending the previously existing results regarding the singularities for other sets of visual features ($N \geq 3$ image points, 3 lines). To do so, we computed a new basis for the interaction matrix corresponding to an image line. The degeneracy conditions of this basis are described by a system of polynomials arising from its maximal minors, depending on the camera pose parameters and on the configuration of the lines. We used algebraic geometry and **algebraic elimination** using **Gröbner bases** to characterize the locus of singularities. We conclude that, for any number of feature lines, a singularity occurs if there exists a line transversal to the 3D lines, and if the camera lies on it. For P4L, there are generically 0, 1 or 2 **transversal lines**, depending on the sign of a discriminant. For P5L there are in general no transversal lines, except for specific configurations. Further, we conclude that, in P4L, there can exist up to **10 additional singular camera poses**, which are unavoidable in general.

Secondly, we obtained the **points of equilibrium of IBVS** controllers using **four feature points**. It is, to our knowledge, the first result regarding the exact computation of the equilibria of IBVS systems. The problem is modeled as the computation of the **critical points** of a potential function equal to the norm of the error, leading to a system of polynomial equations depending on the projected point coordinates and on their depth in the camera frame. For a fixed configuration, and a given desired position, this system is **zero-dimensional** when at least four reference points are used. Solving this system is possible for $N = 4$ points, but not very effective, due to the long computing times (several weeks over 12 cores). We then propose an improved modeling that allows reducing the computational complexity of the system based on **exploiting the symmetries** in the polynomials, leading to a new ideal of lower degree. In the case of **planar objects**, the system complexity can be reduced further using the Jacobian Criterion. These improvements result in a more **effective modeling** for computing the local minima of IBVS from N reference points. We succeeded to compute the solutions of this system for $N = 4$ points in different configurations using `msolve`, a state-of-the-art software based on Gröbner bases for polynomial system solving.

Overall, the work presented in this thesis lies at the intersection of Robotics and computer algebra, and is an example of an application of exact techniques for polynomial system solving to real problems in engineering. Symbolic computation methods, such as the ones used throughout this thesis, are computationally more intensive than numerical ones; however their

outputs are exact, which is an advantage for applications requiring high-precision and/or certifiability of the results. Exact computer algebra methods are also useful for situations where numerical methods are unreliable, due for instance to systems of equations with high nonlinearities. Next, we propose some lines of future research that continue the work presented here.

6.2 Perspectives for future work

6.2.1 Singularity conditions for IBVS using image moments

From the previous existing works [MR93, BCM16, PENB+21, BMC16] and the results presented in this thesis, published also in [GFNBSED22], the singularities of the interaction matrix for IBVS have been characterized for the most elementary choices of visual features: points and straight lines in space.

A relevant topic for future research would be the analysis of the **singularity conditions** when using visual features computed from **image moments** [Cha04]. The image moments can be obtained for any object projected on the image over a region Ω, whose contour can be segmented digitally. Specifically, the moment m_{ij}, said to be of order $(i+j)$, is defined as

$$m_{ij} = \int\int_{\Omega} x^i y^j \, dx \, dy \tag{6.1}$$

where x and y are the pixel coordinates of the objects projection, integrated over the region Ω.

There is an active interest in designing IBVS features constructed from combinations of image moments [TC04, TC05] for several reasons. Image moments are defined for objects with any arbitrary shapes, and the feature extraction process is robust with respect to image measurement errors, as well as to changes in the visibility of the object. Furthermore, these features can convey some intuitive geometric information about the projected shape of the object. For instance, the area of the projection of the object on the image is $a = m_{00}$, and the centroid (the geometric centre) has coordinates (x_g, y_g) with $x_g = m_{10}/m_{00}$ and $y_g = m_{01}/m_{00}$; other information, such as the first and second moments of area, or the direction of the principal axes, can also be obtained from the moments of the image. Last, but not least, practical tests of IBVS based on visual features computed from the image moments have demonstrated good stability and convergence properties [BMHC06, KPD13, ZZGA21], and a nice decoupling between the translational and rotational degrees of freedom of the camera [Cha04, TC04], which helps the efficacy of robot tasks.

6.2.2 Estimating the regions of attraction of IBVS

In this thesis, we provided, to our knowledge, the first results to this date related to the computation of the local minima of IBVS systems. This is only a first step towards an analysis of the global stability behaviour of this class of controllers. A natural follow-up in this direction is to try to obtain an estimation of the **region of attraction** of the system around the desired final configuration. This is one of the objectives of the ANR Sesame project, and the research

subject of Alessandro Colotti and Alexandre Goldsztejn, with whom the work on the critical points computation was done in collaboration.

The method they propose relies on using the potential function of the error $V(\mathbf{p}) = \frac{1}{2}||\mathbf{s}(\mathbf{p}) - \mathbf{s}^\star||^2$ as a candidate Lyapunov function. For control laws that are *gradient-like* with respect to $V(\mathbf{p})$, i.e. those for which $\dot{V}(\mathbf{p}) \leq 0$ for any camera configuration $\mathbf{p}$, Lyapunov's theory says that, if a trajectory $\mathbf{p}(t)$ converges to a point, then it must converge towards a point in the set

$$\mathcal{C} = \{\mathbf{p} \in SE(3) : \ \nabla_{\mathbf{p}} V(\mathbf{p}) = \mathbf{0}\} \tag{6.2}$$

containing the critical points of $V(\mathbf{p})$ (see Section 2.7).

If the desired final configuration of the camera is $\mathbf{p}^\star$, and if one can determine a region $\mathcal{D} \subseteq SE(3)$ around $\mathbf{p}^\star$ that does not contain any other critical points of the function $V(\mathbf{p})$, then $\mathcal{D}$ is a **region of attraction** around the global minimum, and all the control trajectories that start off from $\mathcal{D}$ are guaranteed to converge to the desired point [SL+91, Sas13].

Furthermore, the sub-level sets of the Lyapunov function

$$\mathcal{V}_c = \{\mathbf{p} \in SE(3) : \ V(\mathbf{p}) \leq c\}. \tag{6.3}$$

are **positively invariant sets**. In other words, trajectories that start in one of these sets will never leave it. In fact, each connected component of such a set $\mathcal{V}_c$ is a positively invariant set [Bla99].

If all of the critical points $\mathbf{p}_{crit}$ of the potential function $V(\mathbf{p})$ are known, the problem of estimating a region of attraction can be approached as the problem of maximizing the value of the level set c in (6.3) while guaranteeing connectivity of the domain $\mathcal{V}_c$ around the global minimum. This can be modeled as a non-linear polynomial optimization problem, which can be solved using a variety of techniques: numerical, such as **sum-of-squares** of polynomials [Par03, PPW04, TPS08], as well as symbolic, such as **quantifier elimination** [BPR96, HED12, LSED21].

6.2.3 Improving computer algebra methods

This PhD thesis relied heavily on the use of exact, computer algebra methods for the resolution of polynomial systems of equations, that enabled us to solve problems that could not be solved with numerical computation. This was done largely thanks to the existence of efficient, state-of-the-art algorithms, such as F4 [Fau99] or FGLM [FGLM93], for Gröbner bases computations, and to their implementations on high-performance computing libraries such as `msolve` [BES21], allowing us, for instance, to run computations in parallel using multi-threading.

While illustrating the relevance of symbolic methods for real world applications, the problems we address here evidence also their limitations. Symbolic computation is expensive, thus limiting the sizes of the problems that are within our reach. Therefore, to conclude, we wish to highlight the importance of research aimed to develop improved, more effective algorithms in computer algebra, such as the ones in [BED22a, BED22b] for Gröbner bases computations of structured ideals or ideal saturations, [BND22], for fast change of ordering of Gröbner bases, or [LSED21, LED22, Le21], for quantifier elimination and real root classification.

Appendix A

Singularities in the combination of point and line features

In this Appendix we consider the use of combinations of points and lines as reference features for pose estimation and visual servoing. We provide the full characterization of the singularities of the interaction model related to two different configurations: two points and one straight line, and two lines plus one point. These are minimal configurations, in the sense that if any one feature is removed, the system becomes unconstrained. The Plücker vector representation was used to describe lines in space. The singularity conditions are simpler to compute and to describe than those exposed in Chapter 4 for P4L and P5L, because they are described by a single polynomial equation. We find that, for the case of two points and one line, the singularity conditions correspond to the camera centre lying on a quartic surface, while for two lines and one point it is a surface of degree five. All the codes used to perform the computations in this section are in the repository jorge-gf/thesis-archive [GF22]. Finally, we performed some basic numerical simulations in Matlab to certify the results, by evaluating the condition number of the interaction matrix as the camera moves along a trajectory that passes through a singularity.

A.1 Two points and one line

Modeling. Consider a line in space $\mathcal{L}_1$, and two points denoted by P_2 and P_3. As before, they are defined relative to a fixed object frame $\mathcal{F}_o : (\mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$ whose axis form a right-handed, orthonormal basis. We are free to define $\mathcal{F}_o$ such that $\mathcal{L}_1$ lies along the $\mathbf{x}_o$ axis. The line is then determined by its direction and the point P_1, which coincides with the origin:

$$\mathcal{L}_1 : (P_1, \mathbf{U}_1), \quad {}^o\overrightarrow{OP_1} = [0\ 0\ 0]^T, \quad \mathbf{U}_1 = [1\ 0\ 0]^T. \tag{A.1}$$

Further, the direction of axis $\mathbf{y}_o$ can be chosen such that it intersects the point P_2. This leaves us four parameters to define the coordinates of the points in the fixed frame:

$${}^o\overrightarrow{OP_2} = [0\ a\ 0]^T, \quad {}^o\overrightarrow{OP_3} = [b\ c\ d]^T. \tag{A.2}$$

We also define a camera frame $\mathcal{F}_c : (C, \mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c)$, centred at the focal point C, and also defining an orthonormal, right-handed basis. The position of C relative to the origin is given by the coordinates $[X\ Y\ Z]^T$, while the relative orientation of $\mathcal{F}_c$ and $\mathcal{F}_o$ is parametrized by a rotation matrix ${}^c\mathbf{R}_o$.

The set of visual features related to this configuration is $\mathbf{s} = [l_{x1}\ l_{y1}\ l_{z1}\ x_2\ y_2\ x_3\ y_3]^T \in \mathbb{R}^7$, with $[x_i\ y_i]^T$, and $\mathbf{l}_i = [l_{xi}\ l_{yi}\ l_{zi}]^T$ defined as in (2.26) and (4.2) respectively. The full interaction matrix $\mathbf{M} \in \mathbb{R}^{7\times 6}$ is obtained by stacking the interaction matrices corresponding

to the two points (2.31), and the one corresponding to the line (4.7):

$$\mathbf{M} = \begin{bmatrix} \frac{l_{x1}l_{y1}U_{z1}}{\Delta_1} & \frac{l_{y1}^2U_{z1}}{\Delta_1} & -\frac{l_{y1}(l_{x1}U_{x1}+l_{y1}U_{y1})}{\Delta_1} & l_{x1}l_{y1}l_{z1} & l_{y1}^2l_{z1} & -l_{y1} \\ -\frac{l_{x1}^2U_{z1}}{\Delta_1} & -\frac{l_{x1}l_{y1}U_{z1}}{\Delta_1} & \frac{l_{x1}(l_{x1}U_{x1}+l_{y1}U_{y1})}{\Delta_1} & -l_{x1}^2l_{z1} & -l_{x1}l_{y1}l_{z1} & l_{x1} \\ \frac{(U_{y1}+l_{y1}l_{z1}U_{z1})}{\Delta_1} & -\frac{(U_{x1}+l_{x1}l_{z1}U_{z1})}{\Delta_1} & \frac{l_{z1}(l_{x1}U_{y1}-l_{y1}U_{x1})}{\Delta_1} & l_{y1}(l_{z1}^2+1) & -l_{x1}(l_{z1}^2+1) & 0 \\ -\frac{1}{Z_2} & 0 & \frac{X_2}{Z_2^2} & \frac{X_2Y_2}{Z_2^2} & -(1+\frac{X_2^2}{Z_2^2}) & \frac{Y_2}{Z_2} \\ 0 & -\frac{1}{Z_2} & \frac{Y_2}{Z_2^2} & 1+\frac{Y_2^2}{Z_2^2} & -\frac{X_2Y_2}{Z_2^2} & -\frac{X_2}{Z_2} \\ -\frac{1}{Z_3} & 0 & \frac{X_3}{Z_3^2} & \frac{X_3Y_3}{Z_3^2} & -(1+\frac{X_3^2}{Z_3^2}) & \frac{Y_3}{Z_3} \\ 0 & -\frac{1}{Z_3} & \frac{Y_3}{Z_3^2} & 1+\frac{Y_3^2}{Z_3^2} & -\frac{X_3Y_3}{Z_3^2} & -\frac{X_3}{Z_3} \end{bmatrix}. \tag{A.3}$$

The matrix (A.3) depends on the projected coordinates of the points in the camera frame and their depths (x_i, y_i, Z_i), $i = 1, 2$, and on the components of vectors ${}^c\mathbf{U}_3 = [U_{x3}, U_{y3}, U_{z3}]^T$ and ${}^c\mathbf{l}_3 = [l_{x3}, l_{y3}, l_{z3}]^T$ parametrizing the straight line, also expressed in $\mathcal{F}_c$. We can express these quantities in terms of the camera position $[X, Y, Z]^T$ using

$$\begin{gathered} {}^c\overrightarrow{CP_i} = {}^c\mathbf{R}_o \cdot \left({}^o\overrightarrow{OP_i} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right), \\ {}^c\mathbf{U}_i = {}^c\mathbf{R}_o \cdot {}^o\mathbf{U}_i, \quad {}^c\mathbf{L}_i = {}^c\overrightarrow{CP_i} \times {}^c\mathbf{U}_i. \end{gathered} \tag{A.4}$$

For this problem, we will not assume a zero-orientation for the camera frame; we will see that the independence of the singularity conditions with respect to the orientation of the camera is a result which arises naturally from the computation of the determinant of $\mathbf{M}$. Instead, we will parametrize the matrix ${}^c\mathbf{R}_o$ by the four components of a unit-norm quaternion $\mathbf{q} = (t, u, v, w)$, as in (2.9).

Singularity conditions. We search the poses of the camera, determined by (X, Y, Z) and $\mathbf{q} = (t, u, v, w)$, for which the rank of the matrix $\mathbf{M}$ becomes smaller than 6. First, note that the first and second rows of $\mathbf{M}$, corresponding to the image line, are related by $L_{xi}\mathbf{M}_{i1} + L_{yi}\mathbf{M}_{i2} = 0$. This relation can be exploited to eliminate one row and study the determinant of a (6×6) matrix $\mathbf{M}'$. After substituting (A.4) in the expression for $\mathbf{M}'$, we use a computer algebra system like MAPLE to compute this determinant:

$$\det(\mathbf{M}') = \frac{(t^2+u^2+v^2+w^2)^9}{\Delta_1 \, Z_1^3 \, Z_2^3} \; F(X, Y, Z). \tag{A.5}$$

where Z_1 and Z_2 are the depths of points $\mathcal{P}_1$ and $\mathcal{P}_2$ along the focal axis, and $\Delta_3 = \sqrt{L_{x3}^2 + L_{y3}^2}$ is the depth factor of the image line. The term $F(X, Y, Z)$ is a polynomial in the variables (X, Y, Z) with coefficients that are functions of the parameters $\boldsymbol{\eta}$ defining the configuration of

the object:

$$F = 2d(a-c)\left(Z^4 - Y^4\right) + \left(2\left(d^2 - (a-c)^2\right)YZ - a\left(2b^2 + 3d^2\right)Z + 2bdXY + 2b(a-c)XZ \right. \\ \left. + 3ad(a-c)Y - abdX\right)\left(Y^2 + Z^2\right) \\ + ad(-ac + b^2 + c^2 + d^2)Z^2 + ac\left((a-c)^2 + b^2 + d^2\right)YZ - a^2d(a-c)Y^2. \tag{A.6}$$

The denominator in (A.5) vanishes in the degenerate cases where the projections of one or more of the geometric features are undefined:

- $Z_1 = 0$ or $Z_2 = 0$, corresponding to the case when the plane $Z = 0$ of the camera frame contains points $\mathcal{P}_1$ or $\mathcal{P}_2$, so that the projection mapping (2.26) is undefined.
- $\Delta_3 = 0$, when $L_{x3} = L_{y3} = 0$, for which (4.2) is undefined. This occurs when:
  - The focal point is on the line $\mathcal{L}_3$, such that the line projects on a point on the image.
  - The plane $Z = 0$ of $\mathcal{F}_c$ contains the line $\mathcal{L}_3$; as a result the line is projected at infinity on the image.

The variables (t, u, v, w) parametrizing the camera orientation vanish from the expression (A.5) because $||\mathbf{q}||^2 = t^2 + u^2 + v^2 + w^2 = 1$. Therefore the singularities of the interaction matrix depend only on the camera position and on the relative configuration of the features, and they occur when the camera centre C lies on the quartic surface defined by $F(X, Y, Z) = 0$. This surface is shown in Fig. A.1 for an example configuration.

An illustrative example. In order to verify the exactness of the results, we perform a numerical simulation in Matlab of a camera trajectory passing through a point of singularity. Consider the following values for the parameters

$$a = 1, \quad b = -1, \quad c = 1, \quad d = -3.$$

The singularity surface described by (A.6) is shown in Fig. A.1. We then impose a camera motion along a trajectory parametrized by the functions

$$X = -5/9 + s, \quad Y = 2 + s, \quad Z = 1 + s, \tag{A.7}$$

with $s \in [-0.5\ 0.2]$ a linearly increasing function. The trajectory crosses the singularity surface we computed in (A.6) at $s = 0$. Throughout the motion, a constant orientation is maintained, defined by its quaternion components:

$$\mathbf{q} = \left(\frac{-17}{32}, \frac{1}{64}, \frac{-51}{128}, \frac{9155^{1/2}}{128}\right). \tag{A.8}$$

At each time step, the interaction matrix $\mathbf{M}$ is recomputed from (A.4). Fig. A.2 shows the inverse of the condition number of the matrix $\kappa(\mathbf{M})$, as it evolves along the trajectory. As expected, $\mathbf{M}$ becomes rank-deficient when the camera is on the singularity surface, when $1/\kappa(\mathbf{M})$ becomes null at $s = 0$.

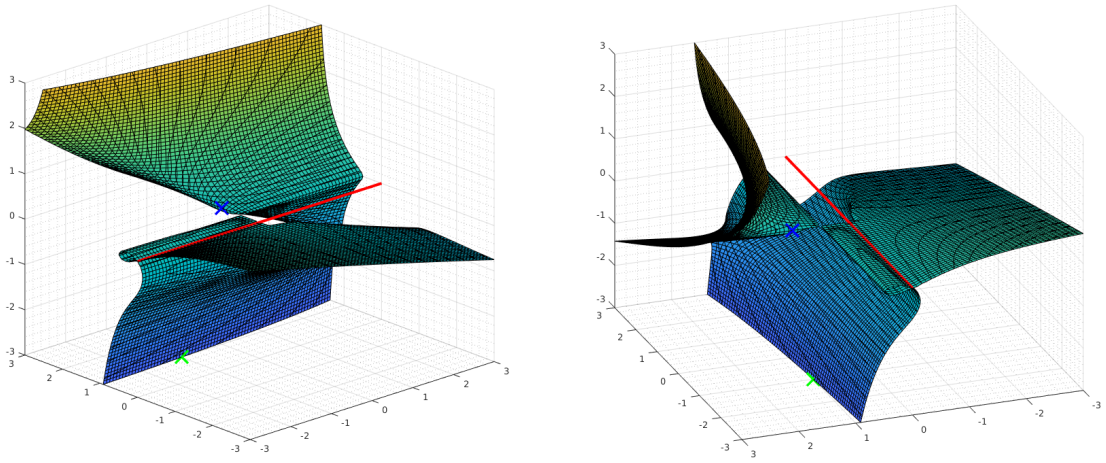


Figure A.1: Singularity loci from the observation of two points (blue and green) and a line (red) with parameters $a = 1$, $b = -1$, $c = 1$, $d = -3$. **Note:** The line is contained in the surface.

A.2 Two lines and one point

Modeling. The two straight lines $\mathcal{L}_1$ and $\mathcal{L}_2$, and the point $\mathcal{P}_3$, are parametrized as follows. We define the object frame $\mathcal{F}_o : (\mathbf{x}_o, \mathbf{y}_o, \mathbf{z}_o)$ such that the $\mathbf{x}_o$ axis has the same direction as $\mathcal{L}_1$; the line $\mathcal{L}_2$ is parallel to the plane $\mathbf{z}_o = 0$, and the $\mathbf{z}_o$ axis intersects both $\mathcal{L}_1$, at point P_1, and $\mathcal{L}_2$, at point P_2. The origin O is chosen at the same distance from both $\mathcal{L}_2$ and $\mathcal{L}_3$. The two lines are determined by the two points P_1 and P_2, and their directions $\mathbf{U}_1$, $\mathbf{U}_2$:

$$\begin{aligned} \mathcal{L}_1 : (P_1, \mathbf{U}_1), \quad {}^o\overrightarrow{OP_1} = [0\ 0\ -c/2]^T, \quad \mathbf{U}_1 = [1\ 0\ 0]^T, \\ \mathcal{L}_2 : (P_2, \mathbf{U}_2), \quad {}^o\overrightarrow{OP_2} = [0\ 0\ +c/2]^T, \quad \mathbf{U}_2 = [a\ b\ 0]^T, \end{aligned} \tag{A.9}$$

while the point P_3 has coordinates

$${}^o\overrightarrow{OP_3} = [d\ e\ f]^T \tag{A.10}$$

As before, the camera frame $\mathcal{F}_c$ is attached to point C with coordinates $[X, Y, Z]^T$, and its orientation is represented by the components of a quaternion with unit norm $\mathbf{q} = (t, u, v, w)$.

In this case, the vector of visual features is $\mathbf{s} = [l_{x1}\ l_{y1}\ l_{z1}\ l_{x2}\ l_{y2}\ l_{z2}\ x_3\ y_3]^T \in \mathbb{R}^8$, with $[x_i\ y_i]^T$ and $\mathbf{l}_i = [l_{xi}\ l_{yi}\ l_{zi}]^T$ defined as in (2.26) and (4.2) respectively. The interaction matrix $\mathbf{M}$ can be obtained by stacking the matrix corresponding to the point (2.31) and the two

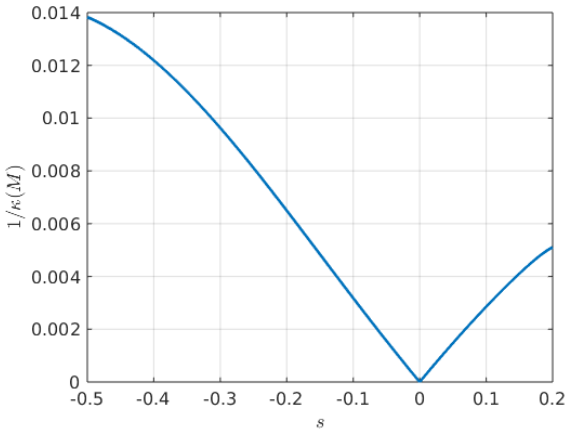


Figure A.2: Inverse of the condition number $\kappa(\mathbf{M})$ as the camera crosses a singularity.

matrices corresponding to the lines (4.7):

$$\mathbf{M} = \begin{bmatrix} \frac{l_{x1}l_{y1}U_{z1}}{\Delta_1} & \frac{l_{y1}^2U_{z1}}{\Delta_1} & -\frac{l_{y1}(l_{x1}U_{x1}+l_{y1}U_{y1})}{\Delta_1} & l_{x1}l_{y1}l_{z1} & l_{y1}^2l_{z1} & -l_{y1} \\ -\frac{l_{x1}^2U_{z1}}{\Delta_1} & -\frac{l_{x1}l_{y1}U_{z1}}{\Delta_1} & \frac{l_{x1}(l_{x1}U_{x1}+l_{y1}U_{y1})}{\Delta_1} & -l_{x1}^2l_{z1} & -l_{x1}l_{y1}l_{z1} & l_{x1} \\ \frac{(U_{y1}+l_{y1}l_{z1}U_{z1})}{\Delta_1} & -\frac{(U_{x1}+l_{x1}l_{z1}U_{z1})}{\Delta_1} & \frac{l_{z1}(l_{x1}U_{y1}-l_{y1}U_{x1})}{\Delta_1} & l_{y1}(l_{z1}^2+1) & -l_{x1}(l_{z1}^2+1) & 0 \\ \frac{l_{x2}l_{y2}U_{z2}}{\Delta_2} & \frac{l_{y2}^2U_{z2}}{\Delta_2} & -\frac{l_{y2}(l_{x2}U_{x2}+l_{y2}U_{y2})}{\Delta_2} & l_{x2}l_{y2}l_{z2} & l_{y2}^2l_{z2} & -l_{y2} \\ -\frac{l_{x2}^2U_{z2}}{\Delta_2} & -\frac{l_{x2}l_{y2}U_{z2}}{\Delta_2} & \frac{l_{x2}(l_{x2}U_{x2}+l_{y2}U_{y2})}{\Delta_2} & -l_{x2}^2l_{z2} & -l_{x2}l_{y2}l_{z2} & l_{x2} \\ \frac{(U_{y2}+l_{y2}l_{z2}U_{z2})}{\Delta_2} & -\frac{(U_{x2}+l_{x2}l_{z2}U_{z2})}{\Delta_2} & \frac{l_{z2}(l_{x2}U_{y2}-l_{y2}U_{x2})}{\Delta_2} & l_{y2}(l_{z2}^2+1) & -l_{x2}(l_{z2}^2+1) & 0 \\ -\frac{1}{Z_3} & 0 & \frac{X_3}{Z_3^2} & \frac{X_3Y_3}{Z_3^2} & -(1+\frac{X_3^2}{Z_3^2}) & \frac{Y_3}{Z_3} \\ 0 & -\frac{1}{Z_3} & \frac{Y_3}{Z_3^2} & 1+\frac{Y_3^2}{Z_3^2} & -\frac{X_3Y_3}{Z_3^2} & -\frac{X_3}{Z_3} \end{bmatrix}. \tag{A.11}$$

Singularity conditions. The matrix $\mathbf{M}$ is of size (8×6); however, the first and second rows of the submatrices (4.7) that correspond to each of the image lines are linearly related by $L_{xi}\mathbf{M}_{i1} + L_{yi}\mathbf{M}_{i2} = 0$. Therefore we can eliminate two rows and study the determinant of a square matrix $\mathbf{M}'$, as before. After introducing the parametrization (A.4), we compute the determinant using MAPLE:

$$\det(\mathbf{M}') = \frac{(t^2+u^2+v^2+w^2)^{12}}{\Delta_1\,\Delta_2\,Z_3^3} F_2(X,Y,Z). \tag{A.12}$$

The term $F_2(X,Y,Z)$ is a long polynomial (given in Appendix A.3) in the variables (X,Y,Z), with coefficients which depend on the parameters $(a,\ldots,f)$, and which describes a surface of degree five in $\mathbb{R}^3$. The denominator of (A.12) vanishes only in the degenerate cases described in section A.1. The term $(t^2+u^2+v^2+w^2)$ is equal to 1, the norm of the quaternion. Therefore, the singularity condition for this case is that the camera centre is contained in the surface of degree 5 defined by $F_2 = 0$. In Fig. A.3, this surface is shown for a specific configuration of the features.

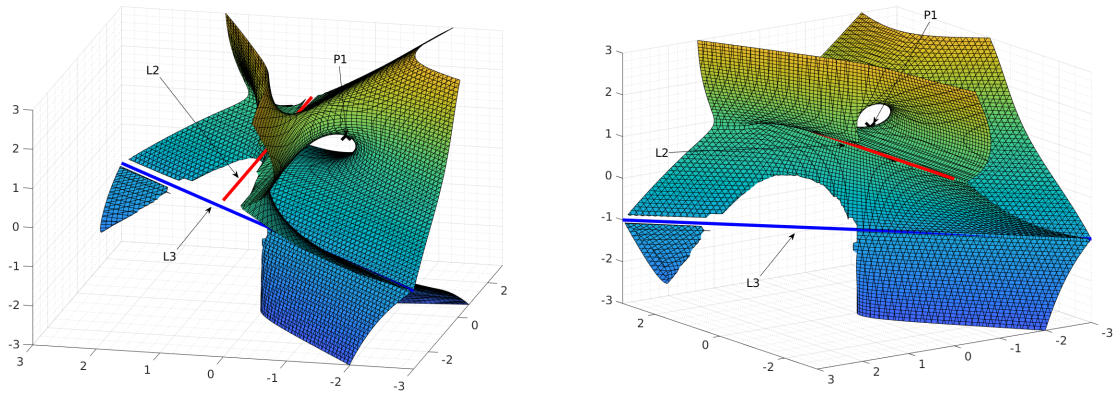


Figure A.3: Two views of the singularity loci from the observation of two lines and a point with parameters $a = 1$, $b = 1$, $c = -2$, $d = 1$, $e = -1$, $f = 1$. *Note*: the two lines are contained in the surface.

An illustrative example To corroborate the results, a camera motion that passes through a singularity point was simulated using Matlab. The parameters were fixed as

$$a = 1, \quad b = 1, \quad c = -2, \quad d = 1, \quad e = -1, \quad f = 1. \tag{A.13}$$

The surface of singularities is drawn in Fig. A.3. The camera motion is parametrized by the functions

$$X = -1 + s, \quad Y = -2 + s, \quad Z = 0 + s. \tag{A.14}$$

with $s \in [-0.3\ 0.3]$ a linearly increasing function, and by a constant orientation chosen randomly and defined by:

$$\mathbf{q} = (\frac{-73}{128}, \frac{-15}{32}, \frac{5}{32}, \frac{7055^{1/2}}{128}). \tag{A.15}$$

The trajectory passes through a point on the surface computed in (A.16) at $s = 0$. The inverse of the condition number of $\mathbf{M}$ as the camera moves along the path is displayed in Fig. A.4. As expected, the interaction matrix becomes rank-deficient at $s = 0$, where $1/\kappa(\boldsymbol{M})$ becomes null.

A.3 Coefficients of the polynomial F_2

The polynomial $F_2(X, Y, Z)$ computed in Section A.2 looks like:

$$\begin{aligned} F_2 = {} & a_{x3yz}X^3YZ + a_{x3y}X^3Y + a_{x3z2}X^3Z^2 + a_{x3z}X^3Z + a_{x3}X^3 + a_{x2y2z}X^2Y^2Z + a_{x2y2}X^2Y^2 \\ & + a_{x2yz2}X^2YZ^2 + a_{x2yz}X^2YZ + a_{x2y}X^2Y + a_{x2z3}X^2Z^3 + a_{x2z2}X^2Z^2 + a_{x2z}X^2Z \\ & + a_{x2}X^2 + a_{xy3z}XY^3Z + a_{xy3}XY^3 + a_{xy2z2}XY^2Z^2 + a_{xy2Z}XY^2Z + a_{xy2}XY^2 \\ & + a_{xyz3}XYZ^3 + a_{xyz2}XYZ^2 + a_{xyz}XYZ + a_{xy}XY + a_{xz4}XZ^4 + a_{xz3}XZ^3 + a_{xz2}XZ^2 \\ & + a_{xz}XZ + a_xX + a_{y4z}Y^4Z + a_{y4}Y^4 + a_{y3z2}Y^3Z^2 + a_{y3z}Y^3Z + a_{y3}Y^3 + a_{y2z3}Y^2Z^3 \\ & + a_{y2z2}Y^2Z^2 + a_{y2z}Y^2Z + a_{y2}Y^2 + a_{yz4}YZ^4 + a_{yz3}YZ^3 + a_{yz2}YZ^2 + a_{yz}YZ + a_yY. \end{aligned} \tag{A.16}$$

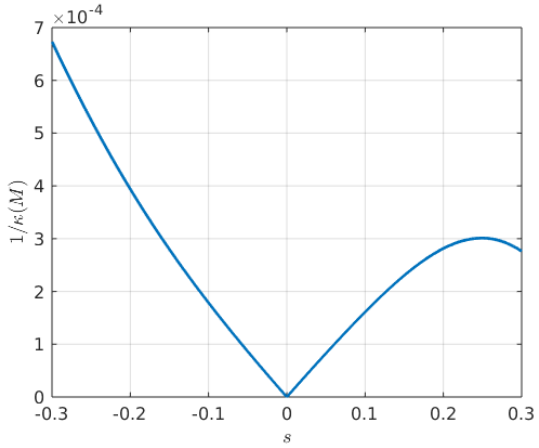


Figure A.4: Inverse of the condition number $\kappa(\mathbf{M})$ as the camera crosses a singularity.

where the coefficients a_{ijk} are as follows:

$$
\begin{aligned}
a_{x3yz} &= -b^3c - 2\,b^3f \\
a_{x3y} &= -1/2\,b^3c^2 - b^3cf \\
a_{x3z2} &= 2\,b^3e \\
a_{x3z} &= 2\,b^3ce \\
a_{x3} &= 1/2\,b^3c^2e \\
a_{x2y2z} &= 3\,ab^2c + 2\,ab^2f \\
a_{x2y2} &= 5/2\,ab^2c^2 + 3\,ab^2cf \\
a_{x2yz2} &= -4\,ab^2e \\
a_{x2yz} &= -7\,ab^2ce + 1/2\,b^3cd + b^3df \\
a_{x2y} &= -5/2\,ab^2c^2e + 1/4\,b^3c^2d + 1/2\,b^3cdf \\
a_{x2z3} &= ab^2c - 2\,ab^2f \\
a_{x2z2} &= 5/4\,ab^2c^2 - 3\,ab^2cf + ab^2e^2 + ab^2f^2 - b^3de \\
a_{x2z} &= 1/2\,ab^2c^3 - 3/2\,ab^2c^2f + ab^2ce^2 + ab^2cf^2 - b^3cde \\
a_{x2} &= 1/16\,ab^2c^4 - 1/4\,ab^2c^3f + 1/4\,ab^2c^2e^2 + 1/4\,ab^2c^2f^2 - 1/4\,b^3c^2de \\
a_{xy3z} &= -2\,a^2bc + b^3c - 2\,b^3f \\
a_{xy3} &= -4\,a^2bc^2 - 2\,a^2bcf - 1/2\,b^3c^2 + b^3cf \\
a_{xy2z2} &= 2\,ab^2d \\
a_{xy2Z} &= 7\,a^2bce - 1/2\,ab^2cd - ab^2df - 1/2\,b^3ce + b^3ef \\
a_{xy2} &= 9/2\,a^2bc^2e - 5/4\,ab^2c^2d - 3/2\,ab^2cdf + 1/4\,b^3c^2e - 1/2\,b^3cef \\
a_{xyz3} &= -2\,a^2bc - 4\,b^3f
\end{aligned}
$$

$$a_{xyz2} = -2\,a^2bc^2 + 6\,a^2bcf + 1/2\,b^3c^2 + 2\,b^3f^2$$
$$a_{xyz} = -a^2bc^3 + 4\,a^2bc^2f - 2\,a^2bce^2 - 2\,a^2bcf^2 + 2\,ab^2cde + b^3c^2f$$
$$a_{xy} = -1/4\,a^2bc^4 + 1/2\,a^2bc^3f - a^2bc^2e^2 - a^2bc^2f^2 + ab^2c^2de - 1/8\,b^3c^4 - 1/2\,b^3c^2f^2$$
$$a_{xz4} = 2\,ab^2d + 2\,b^3e$$
$$a_{xz3} = -a^2bce + 3/2\,ab^2cd - ab^2df + 1/2\,b^3ce - b^3ef$$
$$a_{xz2} = -1/2\,a^2bc^2e - 1/4\,ab^2c^2d - 1/2\,ab^2cdf - 3/4\,b^3c^2e - 1/2\,b^3cef$$
$$a_{xz} = 1/4\,a^2bc^3e - 3/8\,ab^2c^3d + 1/4\,ab^2c^2df - 1/8\,b^3c^3e + 1/4\,b^3c^2ef$$
$$a_{x} = 1/8\,a^2bc^4e - 1/16\,ab^2c^4d + 1/8\,ab^2c^3df + 1/16\,b^3c^4e + 1/8\,b^3c^3ef$$
$$a_{y4z} = -ab^2c + 2\,ab^2f$$
$$a_{y4} = 2\,a^3c^2 + 1/2\,ab^2c^2 - ab^2cf$$
$$a_{y3z2} = -2\,ab^2e + 2\,b^3d$$
$$a_{y3z} = -2\,a^2bcd + 5/2\,ab^2ce - ab^2ef - 2\,b^3cd$$
$$a_{y3} = -3\,a^3c^2e + 3/2\,a^2bc^2d + a^2bcdf - 3/4\,ab^2c^2e + 1/2\,ab^2cef + 1/2\,b^3c^2d$$
$$a_{y2z3} = -ab^2c + 2\,ab^2f$$
$$a_{y2z2} = 2\,a^3c^2 + 3/4\,ab^2c^2 + 3\,ab^2cf + ab^2e^2 - ab^2f^2 - b^3de$$
$$a_{y2z} = -4\,a^3c^2f - 1/2\,ab^2c^3 - 5/2\,ab^2c^2f - ab^2ce^2 - ab^2cf^2 + b^3cde$$
$$a_{y2} = 1/4\,a^3c^4 + a^3c^2e^2 + a^3c^2f^2 - a^2bc^2de + 3/16\,ab^2c^4 + 1/4\,ab^2c^3f + 1/4\,ab^2c^2e^2 + 3/4\,ab^2c^2f^2 - 1/4\,b^3c^2de$$
$$a_{yz4} = -2\,ab^2e + 2\,b^3d$$
$$a_{yz3} = -2\,a^2bcd - 3/2\,ab^2ce + ab^2ef - 1/2\,b^3cd - b^3df$$
$$a_{yz2} = a^3c^2e - 1/2\,a^2bc^2d + a^2bcdf + 5/4\,ab^2c^2e + 1/2\,ab^2cef - 3/4\,b^3c^2d + 1/2\,b^3cdf$$
$$a_{yz} = 1/2\,a^2bc^3d + 3/8\,ab^2c^3e - 1/4\,ab^2c^2ef + 1/8\,b^3c^3d + 1/4\,b^3c^2df$$
$$a_{y} = -1/4\,a^3c^4e + 1/8\,a^2bc^4d - 1/4\,a^2bc^3df - 3/16\,ab^2c^4e - 1/8\,ab^2c^3ef + 1/16\,b^3c^4d - 1/8\,b^3c^3df$$

Bibliography

[AEH02] Nicolas Andreff, Bernard Espiau, and Radu Horaud. Visual servoing from lines. *The International Journal of Robotics Research*, 21(8):679–699, 2002.

[AKNP14] Mahdi Azizian, Mahta Khoshnam, Nima Najmaei, and Rajni V Patel. Visual servoing in medical robotics: a survey. part i: endoscopic and direct vision imaging–techniques and applications. *The international journal of medical robotics and computer assisted surgery*, 10(3):263–274, 2014.

[ALM99] Philippe Aubry, Daniel Lazard, and Marc Moreno Maza. On the theories of triangular sets. *Journal of Symbolic Computation*, 28(1-2):105–124, 1999.

[AY78] JC Alexander and James A Yorke. The homotopy continuation method: numerically implementable topological procedures. *Transactions of the American Mathematical Society*, 242:271–284, 1978.

[BCM16] Sébastien Briot, François Chaumette, and Philippe Martinet. Revisiting the determination of the singularity cases in the visual servoing of image points through the concept of hidden robot. *IEEE Transactions on Robotics*, 33(3):536–546, 2016.

[BED22a] Jérémy Berthomieu, Christian Eder, and Mohab Safey El Din. New efficient algorithms for computing gröbner bases of saturation ideals (f4sat) and colon ideals (sparse-fglm-colon). *arXiv preprint arXiv:2202.13387*, 2022.

[BED22b] Jérémy Berthomieu and Mohab Safey El Din. Guessing gröbner bases of structured ideals of relations of sequences. *Journal of Symbolic Computation*, 111:1–26, 2022.

[BES21] Jérémy Berthomieu, Christian Eder, and Mohab Safey El Din. msolve: A Library for Solving Polynomial Systems. In *2021 International Symposium on Symbolic and Algebraic Computation*, 46th International Symposium on Symbolic and Algebraic Computation, pages 51–58, Saint Petersburg, Russia, July 2021. ACM.

[BHS06] Patricia Ben-Horin and Moshe Shoham. Singularity analysis of a class of parallel robots based on grassmann–cayley algebra. *Mechanism and Machine Theory*, 41(8):958–970, 2006.

[Bla99] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[BM13] Sébastien Briot and Philippe Martinet. Minimal representation for the control of gough-stewart platforms via leg observation considering a hidden robot model. In *2013 IEEE international conference on robotics and automation*, pages 4653–4658. IEEE, 2013.

[BMC16] Sébastien Briot, Philippe Martinet, and François Chaumette. Determining the singularities for the observation of three image lines. *IEEE Robotics and Automation Letters*, 2(2):412–419, 2016.

[BMG+09] Odile Bourquardez, Robert Mahony, Nicolas Guenard, François Chaumette, Tarek Hamel, and Laurent Eck. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, 25(3):743–749, 2009.

[BMHC06] Odile Bourquardez, Robert Mahony, Tarek Hamel, and François Chaumette. Stability and performance of image based visual servo control using first order spherical image moments. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4304–4309. IEEE, 2006.

[BMMT94] Eberhard Becker, Teo Mora, Maria Grazia Marinari, and Carlo Traverso. The shape of the shape lemma. In *Proceedings of the international symposium on Symbolic and algebraic computation*, pages 129–133, 1994.

[BMR15] S. Briot, P. Martinet, and V. Rosenzveig. The hidden robot: an efficient concept contributing to the analysis of the controllability of parallel robots in advanced visual servoing techniques. *IEEE Transactions on Robotics*, 31(6):1337–1352, 2015.

[BND22] Jérémy Berthomieu, Vincent Neiger, and Mohab Safey El Din. Faster change of order algorithm for gr\" obner bases under shape and stability assumptions. *arXiv preprint arXiv:2202.09226*, 2022.

[BPR96] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM (JACM)*, 43(6):1002–1045, 1996.

[BT18] Paul Breiding and Sascha Timme. Homotopycontinuation. jl: A package for homotopy continuation in julia. In *International Congress on Mathematical Software*, pages 458–465. Springer, 2018.

[Buc65] Bruno Buchberger. Ein algorithmus zum auffinden der basiselemente des restklassenringes nach einem nulldimensionalen polynomideal. *PhD thesis, Universitat Insbruck*, 1965.

[Buc70] Bruno Buchberger. Ein algorithmisches kriterium für die lösbarkeit eines algebraischen gleichungssystems. *Aequationes math*, 4(3):374–383, 1970.

[Buc79] Bruno Buchberger. A criterion for detecting unnecessary reductions in the construction of gröbner-bases. In *International Symposium on Symbolic and Algebraic Manipulation*, pages 3–21. Springer, 1979.

[BW98] Bruno Buchberger and Franz Winkler. *Gröbner bases and applications*, volume 17. Cambridge University Press Cambridge, 1998.

[CH06] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[CH07] François Chaumette and Seth Hutchinson. Visual servo control. ii. advanced approaches [tutorial]. *IEEE Robotics & Automation Magazine*, 14(1):109–118, 2007.

[CH08] F. Chaumette and S. Hutchinson. *Visual Servoing and Visual Tracking, chapter 24 of Handbook of Robotics*. Springer, 2008.

[Cha90] François Chaumette. *La relation vision-commande: théorie et application à des tâches robotiques*. PhD thesis, Rennes 1, 1990.

[Cha04] François Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723, 2004.

[CKY89] John F Canny, Erich Kaltofen, and Lakshman Yagati. Solving systems of nonlinear polynomial equations faster. In *Proceedings of the ACM-SIGSAM 1989 international symposium on Symbolic and algebraic computation*, pages 121–128, 1989.

[Cla90] R Clavel. Device for movement and displacing of an element in space, 1990. US Patent n4,976,582.

[CLO13] David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

[CM00] François Chaumette and Ezio Malis. 2 1/2 d visual servoing: a possible solution to improve image-based and position-based visual servoings. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 630–635. IEEE, 2000.

[DD95] Daniel F DeMenthon and Larry S Davis. Model-based object pose in 25 lines of code. *International journal of computer vision*, 15(1):123–141, 1995.

[DJSW02] Lingfeng Deng, Farrokh Janabi-Sharifi, and William J Wilson. Stability and robustness of visual servoing methods. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1604–1609. IEEE, 2002.

[DRLR89] Michel Dhome, Marc Richetin, J-T Lapreste, and Gerard Rives. Determination of the attitude of 3d objects from a single perspective view. *IEEE transactions on pattern analysis and machine intelligence*, 11(12):1265–1278, 1989.

[DSRC11] Albert Diosi, Siniša Segvic, Anthony Remazeilles, and François Chaumette. Experimental evaluation of autonomous driving based on visual memory and

image-based visual servoing. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):870–883, 2011.

[ECR92] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *ieee Transactions on Robotics and Automation*, 8(3):313–326, 1992.

[EF17] Christian Eder and Jean-Charles Faugère. A survey on signature-based algorithms for computing gröbner bases. *Journal of Symbolic Computation*, 80:719–784, 2017.

[Eis13] David Eisenbud. *Commutative algebra: with a view toward algebraic geometry*, volume 150. Springer Science & Business Media, 2013.

[Esp94] Bernard Espiau. Effect of camera calibration errors on visual servoing in robotics. In *Experimental robotics III*, pages 182–192. Springer, 1994.

[Fau99] Jean-Charles Faugere. A new efficient algorithm for computing gröbner bases (f4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[Fau02] Jean Charles Faugere. A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002.

[Fau10] Jean-Charles Faugère. Fgb: a library for computing gröbner bases. In *International Congress on Mathematical Software*, pages 84–87. Springer, 2010.

[FGLM93] Jean-Charles Faugere, Patrizia Gianni, Daniel Lazard, and Teo Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[FTC11] Romeo Tatsambon Fomena, Omar Tahri, and François Chaumette. Distance-based and orientation-based visual servoing from three points. *IEEE Transactions on Robotics*, 27(2):256–267, 2011.

[GF22] Jorge García Fontán. Repository for the phd thesis "singularity and stability analysis of vision-based controllers", 2022.

[GFNBSED22] Jorge García Fontán, Abhilash Nayak, Sébastien Briot, and Mohab Safey El Din. Singularity analysis for the perspective-four and five-line problems. *International Journal of Computer Vision*, 130(4):909–932, 2022.

[GHTC03] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):930–943, 2003.

[Gra08] Basile Graf. Quaternions and dynamics. *arXiv preprint arXiv:0811.2889*, 2008.

[HAC+09] Paul TP Ho, Pablo Altamirano, Chia-Hao Chang, Shu-Hao Chang, Su-Wei Chang, Chung-Cheng Chen, Ke-Jung Chen, Ming-Tang Chen, Chih-Chiang Han, West M Ho, et al. The yuan-tseh lee array for microwave background anisotropy. *The Astrophysical Journal*, 694(2):1610, 2009.

[HCLL89] Radu Horaud, Bernard Conio, Olivier Leboulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, 47(1):33–44, 1989.

[HED12] Hoon Hong and Mohab Safey El Din. Variant quantifier elimination. *Journal of Symbolic Computation*, 47(7):883–901, 2012.

[HHC96] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[Hun87] K. H. Hunt. *Kinematic Geometry of Mechanisms*. Clarendon Press, 1987.

[IOH03] Noriyasu Inaba, Mitsushige Oda, and Masato Hayashi. Visual servoing of space robot for autonomous satellite capture. *Transactions of the Japan Society for Aeronautical and Space Sciences*, 46(153):173–179, 2003.

[KB60] Rudolf E Kalman and John E Bertram. Control system analysis and design via the "second method" of lyapunov: I—continuous-time systems. 1960.

[KB78] Christoph Kollreider and Bruno Buchberger. An improved algorithmic construction of gröbner-bases for polynomial ideals. *ACM SIGSAM Bulletin*, 12(2):27–36, 1978.

[KLS14] Laurent Kneip, Hongdong Li, and Yongduek Seo. Upnp: An optimal o (n) solution to the absolute pose problem with universal applicability. In *European conference on computer vision*, pages 127–142. Springer, 2014.

[KMM+96] Djamel Khadraoui, Guy Motyl, Philippe Martinet, Jean Gallice, and François Chaumette. Visual servoing in robotics scheme using a camera/laser-stripe sensor. *IEEE Transactions on Robotics and Automation*, 12(5):743–750, 1996.

[KPD13] Olivier Kermorgant, Yvan Pétillot, and Matt Dunnigan. A global control scheme for free-floating vehicle-manipulators. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5015–5020. IEEE, 2013.

[Kro82] Leopold Kronecker. Grundzüge einer arithmetischen theorie der algebraische grössen. 1882.

[KSS11] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, page jun, Providence, RI, 2011.

[KWCC09] Daniel Kanaan, Philippe Wenger, Stéphane Caro, and Damien Chablat. Singularity analysis of lower mobility parallel manipulators using grassmann–cayley algebra. *IEEE Transactions on Robotics*, 25(5):995–1004, 2009.

[L+91] David G Lowe et al. Fitting parameterized three-dimensional models to images. *IEEE transactions on pattern analysis and machine intelligence*, 13(5):441–450, 1991.

[Laz14] Simon Lazarus. Basic algebraic geometry and the 27 lines on a cubic surface, 2014.

[Laz21] Daniel Lazard. Degree of a polynomial ideal and bézout inequalities. 2021.

[LBS18] Muyinatu A Lediju Bell and Joshua Shubert. Photoacoustic-based visual servoing of a needle tip. *Scientific reports*, 8(1):1–12, 2018.

[Le21] Huu Phuoc Le. *On solving parametric polynomial systems and quantifier elimination over the reals: algorithms, complexity and implementations*. PhD thesis, Sorbonne Université, 2021.

[LED22] Huu Phuoc Le and Mohab Safey El Din. Solving parametric systems of polynomial equations over the reals through hermite matrices. *Journal of Symbolic Computation*, 112:25–61, 2022.

[LHP80] Hugh Christopher Longuet-Higgins and Kvetoslav Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980.

[LL61] JP LaSalle and S Lefschetz. Stability by lyapunov's second method with applications. *book), Academic Press, New York, NY*, 1961.

[LNBT00] Michael H Loser, Nassir Navab, Benedicte Bascle, and Russell H Taylor. Visual servoing for automatic and uncalibrated percutaneous procedures. In *Medical Imaging 2000: Image Display and Visualization*, volume 3976, pages 270–281. SPIE, 2000.

[LP17] Kevin M Lynch and Frank C Park. *Modern robotics*. Cambridge University Press, 2017.

[LSED21] Huu Phuoc Le and Mohab Safey El Din. Faster one block quantifier elimination for regular polynomial systems of equations. In *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*, pages 265–272, 2021.

[LW93] TY Li and Xiao Shen Wang. Solving real polynomial systems with real homotopies. *mathematics of computation*, 60(202):669–680, 1993.

[LXX12] Shiqi Li, Chi Xu, and Ming Xie. A robust o (n) solution to the perspective-n-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1444–1450, 2012.

[Lya92] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

[Mal04] Ezio Malis. Improving vision-based control using efficient second-order minimization techniques. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pages 1843–1848. IEEE, 2004.

[MC02] Éric Marchand and François Chaumette. Virtual visual servoing: a framework for real-time augmented reality. *Computer Graphics Forum*, 21(3):289–297, 2002.

[MC08] Mohammed Marey and François Chaumette. Analysis of classical and new visual servoing control laws. In *2008 IEEE International Conference on Robotics and Automation*, pages 3244–3249. IEEE, 2008.

[MCB99] Ezio Malis, Francois Chaumette, and Sylvie Boudet. 2 1/2 d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, 1999.

[Mer05] Jean-Pierre Merlet. *Parallel robots*, volume 128. Springer Science & Business Media, 2005.

[MLS17] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[MMR09] Ezio Malis, Youcef Mezouar, and Patrick Rives. Robustness of image-based visual servoing with a calibrated camera in the presence of uncertainties in the three-dimensional structure. *IEEE Transactions on Robotics*, 26(1):112–120, 2009.

[MR93] Henri Michel and Patrick Rives. *Singularities in the determination of the situation of a robot effector from the perspective view of 3 points*. PhD thesis, INRIA, 1993.

[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 and Automation Magazine*, 12(4):40–52, December 2005.

[MUS15] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015.

[NW99] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[OC11] Ryuta Ozawa and François Chaumette. Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach. In *2011 IEEE international conference on robotics and automation*, pages 5670–5676. IEEE, 2011.

[OSG20] Boris Odenhal, Hellmuth Stachel, and Glaeser Georg. *The Universe of Quadrics*. Springer, 2020.

[Pap95] Nikolaos P Papanikolopoulos. Selection of features and evaluation of visual measurements during robotic visual servoing tasks. *Journal of intelligent and robotic systems*, 13(3):279–304, 1995.

[Par03] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[PENB+21] Beatriz Pascual-Escudero, Abhilash Nayak, Sébastien Briot, Olivier Kermorgant, Philippe Martinet, Mohab Safey El Din, and François Chaumette. Complete singularity analysis for the perspective-four-point problem. *International Journal of Computer Vision*, 129(4):1217–1237, 2021.

[PPW04] Stephen Prajna, Antonis Papachristodoulou, and Fen Wu. Nonlinear control synthesis by sum of squares optimization: A lyapunov-based approach. In *2004 5th Asian control conference (IEEE Cat. No. 04EX904)*, volume 1, pages 157–165. IEEE, 2004.

[PW01] Helmut Pottmann and Johannes Wallner. *Computational Line Geometry*. Springer-Verlag, Berlin, Heidelberg, 2001.

[RBM13] V. Rosenzveig, S. Briot, and P. Martinet. Minimal representation for the control of the Adept Quattro with rigid platform via leg observation considering a hidden robot model. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, Tokyo Big Sight, Japan, 2013.

[RBPD81] P Rives, P Bouthémy, B Prasada, and E Dubois. Recovering the orientation and the position of a rigid body in space from a single view. *INRS-Telecommunications, Verdun, QC, Canada, Tech. Rep*, 1981.

[RE87] Patrick Rives and Bernard Espiau. Estimation recursive de primitives 3d au moyen d'une caméra mobile, 1987.

[Rie14] Michael Q Rieck. A fundamentally new view of the perspective three-point pose problem. *Journal of mathematical imaging and vision*, 48(3):499–516, 2014.

[RZ04] Fabrice Rouillier and Paul Zimmermann. Efficient isolation of polynomial's real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2004.

[Sas13] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[SKK08] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

[SL+91] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ, 1991.

[TC04] Omar Tahri and François Chaumette. Image moments: Generic descriptors for decoupled image-based visual servo. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 2, pages 1185–1190. IEEE, 2004.

[TC05] Omar Tahri and Francois Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127, 2005.

[Tho66] EH Thompson. Space resection: Failure cases. *The Photogrammetric Record*, 5(27):201–207, 1966.

[TMCG02] Benoit Thuilot, Philippe Martinet, Lionel Cordesses, and Jean Gallice. Position based visual servoing: keeping the object in the field of vision. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, volume 2, pages 1624–1629. IEEE, 2002.

[TPS08] Ufuk Topcu, Andrew Packard, and Peter Seiler. Local stability analysis using simulations and sum-of-squares programming. *Automatica*, 44(10):2669–2675, 2008.

[TR17] John Till and D Caleb Rucker. Elastic stability of cosserat rods and parallel continuum robots. *IEEE Transactions on Robotics*, 33(3):718–733, 2017.

[Tsi16] Elias Tsigaridas. Slv: a software for real root isolation. *ACM Communications in Computer Algebra*, 50(3):117–120, 2016.

[WH06] Yihong Wu and Zhanyi Hu. Pnp problem revisited. *Journal of Mathematical Imaging and Vision*, 24(1):131–141, 2006.

[WSN87] LEEE Weiss, ARTHURC Sanderson, and CHARLESP Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417, 1987.

[WXC20] Ping Wang, Guili Xu, and Yuehua Cheng. A novel algebraic solution to the perspective-three-line pose problem. *Computer Vision and Image Understanding*, 191:102711, 2020.

[WXCY19] Ping Wang, Guili Xu, Yuehua Cheng, and Qida Yu. Camera pose estimation from lines: a fast, robust and general method. *Machine Vision and Applications*, 30(4):603–614, 2019.

[XZCK16] Chi Xu, Lilian Zhang, Li Cheng, and Reinhard Koch. Pose estimation from line correspondences: A complete analysis and a series of solutions. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1209–1222, 2016.

[ZGHL16] Yuanshen Zhao, Liang Gong, Yixiang Huang, and Chengliang Liu. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127:311–323, 2016.

[ZH06] Cai-xia Zhang and Zhanyi Hu. Why is the danger cylinder dangerous in the p3p problem? *Acta Automatica Sinica*, 32(4):504, 2006.

[ZXLK12] Lilian Zhang, Chi Xu, Kok-Meng Lee, and Reinhard Koch. Robust and efficient pose estimation from line correspondences. In *Asian Conference on Computer Vision*, pages 217–230. Springer, 2012.

[ZZGA21] Yang Zhou, Yuanxu Zhang, Jian Gao, and Xuman An. Visual servo control of underwater vehicles based on image moments. In *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 899–904. IEEE, 2021.