universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

“Fundamental Mathematical Concepts for Problems Arising in Robotics”

Verfasserin

Bettina Ponleitner

angestrebter akademischer Grad

Magistra der Naturwissenschaften (Mag.rer.nat)

Wien, im April 2013

Studienkennzahl lt. Studienblatt:	A 405
Studienrichtung lt. Studienblatt:	Mathematik
Betreuer:	ao. Univ.-Prof. Dipl.-Ing. Dr. Hermann Schichl

Acknowledgements

First and foremost, my heartfelt and deepest thanks go to my family: to my parents Alice Stohl and Ulrich Ponleitner, who were throughout my studies always encouraging and supporting me, and in particular, to my sister Nathalie Ponleitner, who provided me with invaluable advice and motivation, and always found time for talking with me–sometimes literally at night and day. Their absolute and on-going belief in me and my abilities always have been and still are of great significance to me. Further, I sincerely thank Tina Schrutka for her friendship, for always listening to me, and if needed, for cheering me up. And, last but not least, I express my deep gratitude to my advisor Hermann Schichl, for his great support, patient guidance and enthusiastic inspiration.

Abstract

Robotics summarizes the aspects of designing, building and analyzing robots. It poses a large number of challenging highly nonlinear, often algebraic problems in a moderate number of variables. Due to the nonlinearities the relevant (systems of) equations usually have multiple solutions. Further, safety considerations or the demand for a complete analysis require a worst case analysis of the possible scenarios. Additionally, uncertainties like manufacturing tolerances on the mechanical structure, sensor measurement errors, or numerical round-off errors have to be taken into account.

A large number of research results is available in the field of robotics. This thesis gives a survey of recent results concerning several geometrical and kinematic problems (in particular for parallel robots), with emphasis on the mathematical tools used for solving them. Due to the high interdisciplinary character of this research area, a wide variety of mathematical methods is used for solving the posed problems, which cannot be covered here in total. Therefore, the mathematical part of the thesis is restricted to giving an overview of several methods for solving systems of nonlinear equations, and to outline main concepts of interval analysis.

Zusammenfassung

Das Forschungsfeld der Robotik umfasst sämtliche Aspekte, die die Entwicklung, Konstruktion und Analyse von Robotern betreffen. Es wirft eine Vielzahl von anspruchsvollen Problemstellungen auf, die im Allgemeinen hochgradig nichtlinear mit einer überschaubaren Anzahl von Variablen sind. Aufgrund ihres nichtlinearen Charakters haben die relevanten Gleichungen bzw. Gleichungssysteme im Normalfall mehrere Lösungen. In vielen Fällen ist eine „worst case"-Untersuchung sämtlicher möglicher Szenarien erforderlich, zum Beispiel wenn Sicherheitsaspekte geprüft oder eine komplette Analyse des Problems durchgeführt werden sollen. Da es im realen Leben keinen idealen „perfekten" Roboter gibt, müssen zusätzlich noch diverse Unsicherheitsfaktoren berücksichtigt werden, wie zum Beispiel Fertigungstoleranzen an der mechanischen Struktur eines Roboters, Sensor-Messfehler oder numerische Rundungsfehler.

Es gibt eine große Anzahl von Forschungspublikationen, die sich mit verschiedensten Fragestellungen der Robotik beschäftigen. Die vorliegende Arbeit gibt einen Überblick über Forschungsergebnisse der jüngeren Vergangenheit, die sich auf verschiedene kinematische und Modellierungs-Problemstellungen beziehen, insbesondere für parallele Roboter. Dabei liegt der Fokus auf den mathematischen Methoden, die zur Lösung herangezogen werden. Aufgrund des starken interdisziplinären Charakters der Robotik sind die verwendeten mathematischen Techniken sehr breit gefächert. Nachdem eine detaillierte Beschreibung sämtlicher Methoden mehrere Bücher füllen würde, beschränkt sich diese Arbeit darauf, einen Überblick über verschiedene Methoden zum Lösen nichtlinearer Gleichungssysteme zu geben, und darüber hinaus wesentliche Konzepte der Intervall Analysis zu präsentieren.

Contents

Introduction

Robotics summarizes the aspects of designing, building and analyzing robots. It poses a large number of challenging highly nonlinear, often algebraic problems in a moderate number of variables. Due to the nonlinearities the relevant (systems of) equations usually have multiple solutions. Further, safety considerations or the demand for a complete analysis require a worst case analysis of the possible scenarios. Therefore, the problems may be formulated as constraint satisfaction problems or global optimization problems. Additionally, uncertainties like manufacturing tolerances on the mechanical structure, sensor measurement errors, or numerical round-off errors have to be taken into account.

A large number of research results[1] is available in the field of robotics. When we restrict our focus on the mechanical architecture of robots rather than on the artificial intelligence aspect (therefore see, e.g., Russell & Norvig [267]), we can identify two main classes of mechanical architectures: *serial* and *parallel* robots. This thesis gives a survey of recent results concerning geometrical and kinematic problems arising in robotics. As parallel manipulators have become more and more popular in the last two decades, and have taken the center stage in research, the focus is laid on problems related to parallel robots. Since parallel robots may be designed in various different ways with specific properties, preferably results with general character are presented here; in particular, so-called Gough-type manipulators are outlined, since these probably pose the most demanding problems in the field. Thereby, the problems are looked at from a "mathematical" point of view, i.e., emphasis is placed on the formulation of the problems and the mathematical methods used for solving them. Robotics is a field with high interdisciplinary character, and therefore, a wide variety of mathematical methods is used for solving the posed problems. Due to lack of space, only a few methods are outlined.

The goal of the thesis is to present several problems arising in robotics in such a way, that they are accessible to a mathematician without specific engineering knowledge, and further, to provide an overview of the major computational challenges in the field, with the intention to pave the way for further research.

The thesis is organized as follows. Basically, it is divided in two main parts: **Part I** covers fundamental mathematical concepts necessary for solving certain problems

[1]An extensive robotics bibliography library can be found at `http://www-sop.inria.fr/members/Jean-Pierre.Merlet/references_robotique.html`.

arising in robotics; **Part II** then provides the contents particularly concerning robotics.

In general, the kinematic analysis of robots relies on solving systems of nonlinear, and frequently algebraic equations. In **Chapter 1** three different approaches are presented, in particular polynomial continuation, Gröbner bases and elimination, in order to provide a sufficient background on methods for achieving a solution of such a system.

The methods discussed in the previous chapter may be insufficient, in particular, if all solutions of a certain problem are searched for, in the presence of uncertainties (such as round-off errors or measurement errors) or, if certified results are demanded. Therefore, in **Chapter 2**, main concepts of interval analysis are outlined. Due to the specific structure of the field, the theory is introduced more thoroughly than the methods before.

In **Chapter 3** the general structure of robot mechanisms is described, and basic definitions are given. Further, Denavit–Hartenberg parameters and screw displacements for the description of open-loop kinematic chains are introduced.

Chapter 4 deals with the geometry of robot manipulators. The notion of the equation of state is introduced and several sets and maps are defined, which are convenient for formulating the main types of geometrical problems. Additionally, generalized coordinates are introduced and a parametrization in terms of quaternions is presented. Finally, a description of several types of the workspace of a robot is given, as well as results concerning their determination.

A significant part of the kinematic analysis of a robot consists of deriving the solution(s) of the forward and inverse kinematics problem, i.e., calculating the state of a robot from measurements or poses. Therefore, **Chapter 5** provides main results about obtaining solutions to these problems.

Another main task of the kinematic analysis of a robot is to study the changes of geometry and the resulting translational and angular velocities and accelerations. Velocity, acceleration and the crucial notion of the kinematic Jacobian are discussed in **Chapter 6**.

For applications it is important to know about the flexibility and accuracy with which the end-effector of a robot is placed in a given pose. Several methods have been proposed to quantify these features. So, in **Chapter 7** an approach for the modeling of the accuracy of a pose, as well as a critical review of several dexterity indices are given. Additionally, the relationship between positioning errors and (global) optimization problems is summarized.

Finally, the singularity analysis of parallel robots is addressed in **Chapter 8**. The concept of singular configurations is considered briefly from different points of view, and a summary of results concerning the determination of singularities is given.

Part I

Mathematical Methods

Chapter 1

Solving Systems of Nonlinear Equations

1.1 Introduction

The kinematic analysis of a robotic manipulator usually depends on the solution of sets of nonlinear equations. In Chapter 5 two basic problems where such systems of nonlinear equations typically occur, the inverse kinematics problem for serial manipulators and the forward kinematics problem of parallel manipulators, are described. Furthermore, results concerning the number of (real) solutions and the specific solutions are presented. There is a variety of both numerical and algebraic techniques available to solve such systems of equations and to give bounds on the number of solutions. This chapter provides a summary of three often used techniques for solving algebraic systems, i.e., systems of polynomial equations

$$f(x) := \begin{pmatrix} f_1(x_1, \dots, x_m) \\ \vdots \\ f_n(x_1, \dots, x_m) \end{pmatrix} = 0, \tag{1.1}$$

where $f\colon \mathbb{C}^m \to \mathbb{C}^n$. The techniques presented are *polynomial continuation*, *Gröbner bases*, and *dialytic elimination*. To begin with, some notation and remarks about polynomials and the number of solutions are given.

1.2 Preleminaries

1.2.1 Polynomials

A **polynomial** f in variables $x_1, \dots, x_n$ with coefficients $a_\alpha \in \mathbb{K}$ is a finite linear combination of monomials, i.e.,

$$f = \sum_{|\alpha| \leq m} a_\alpha x^\alpha,$$

where $\alpha = (\alpha_1, \dots, \alpha_m)$ is a multiindex with $\alpha_i \geq 0$ for all $i = 1, \dots, m$ and

$$x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_m^{\alpha_m}.$$

The set of all polynomials in $x_1, \dots, x_n$ and coefficients in a field $\mathbb{K}$ together with pointwise addition and multiplication is an integral domain, called the **ring of polynomials**, denoted by $\mathbb{K}[x_1, \dots, x_n]$. The **degree** of each monomial is the sum of its exponents, i.e., $|\alpha| := \sum_i \alpha_i$. For example, for $f = 5x_1^2 + 3x_1x_2^2 + x_2 - 3$ the degrees of the monomials x_1^2, $x_1x_2^2$, and x_2 are 2, 3, and 1, respectively. The degree of the polynomial is equal to the degree of its highest degree term, which is the so-called **leading term**, denoted by $\textsc{lt}(f)$. The coefficient of the leading term of the polynomial is called the **leading coefficient**, denoted by $\textsc{lc}(f)$. The expression **power product** refers to the monomials in a polynomial, e.g., for $x_1^3x_2 + 4x_1^2x_3 + 2x_3 + 1 = 0$ the power products are $x_1^3x_2$, $x_1^2x_3$, x_3, and 1. The **leading power product**, denoted by $\textsc{lp}(f)$, is the power product of the leading term of a polynomial; in the previous example, $\textsc{lp} = x_1^3x_2$.

The **total degree** of a system as in (1.1) is the product of the degrees of its individual polynomial equations [253].

Definition 1.2.1 (Affine variety). *Let $f_1, \dots, f_s$ be polynomials in $\mathbb{K}[x_1, \dots, x_n]$. Then*

$$V(f_1, \dots, f_s) := \{(a_1, \dots, a_n) \in \mathbb{K}^n \mid f_i(a_1, \dots, a_n) = 0 \quad \forall 1 \leq i \leq n\}$$

is called the ***affine variety*** *defined by $f_1, \dots, f_s$.*

For example, $f = x_1^2 + x_2^2 - x_3^2$ describes the cone, which is an affine variety given by $V_c := V(f) = \{(x_1, x_2, x_3) \in \mathbb{C}^n \mid x_1^2 + x_2^2 - x_3^2 = 0\}$. In particular, the set of polynomials in (1.1) defines an affine variety.

Definition 1.2.2 (Ideal). *A subset $\mathcal{I} \subset \mathbb{K}[x_a, \dots, x_m]$ is an* ***ideal*** *if it satisfies*

(i) $0 \in \mathcal{I}$,

(ii) if $f, g \in \mathcal{I}$, then $f + g \in \mathcal{I}$,

(iii) if $f \in \mathcal{I}$ and $h \in \mathbb{K}[x_1, \dots, x_m]$, then $h \cdot f \in \mathcal{I}$.

E.g., for $f = x_1^2 + x_2^2 - x_3^2$ the set $\mathcal{I}_c = \{f\}$, consisting of all polynomials vanishing on V_c, is an ideal in the ring of polynomials $\mathbb{C}[x, y, z]$. It can be shown that for $f_1, \ldots, f_s \in \mathbb{K}[x_1, \ldots, x_m]$,

$$\langle f_1, \ldots, f_n \rangle := \left\{ \sum_{i=1}^{s} h_i f_i \mid h_1, \ldots, h_n \in \mathbb{K}[x_1, \ldots, x_m] \right\}$$

is an ideal; $\langle f_1, \ldots, f_n \rangle$ is said to be generated by $f_1, \ldots, f_n$. For further information on varieties and ideals see, e.g., [56].

Definition 1.2.3 (Multiplicity [285])**.** *The **multiplicity** $\mu(f, x^*)$ of an an isolated solution x^* of (1.1) is defined as the dimension of the finite dimensional vector space*

$$\mathcal{O}_{\mathbb{C}^n, x^*} / \mathcal{I}_{x^*},$$

where $\mathcal{O}_{\mathbb{C}^n, x^*}$ is the ring of convergent power series centered at x^*, and $\mathcal{I}_{x^*}$ denotes the ideal of $\mathcal{O}_{\mathbb{C}^n, x^*}$ generated by the polynomials $f_1, \ldots, f_n$.

A solution x^* being an **isolated solution** means that there are no other solutions in the vicinity, i.e., the solution point x^* does not belong to a higher dimensional solution set (such as a curve or a surface). For $n = m = 1$ this definition agrees with the commonly used notion of multiplicity, i.e., given $p(x) \in \mathbb{C}[x]$, a not identically zero polynomial in one variable, the multiplicity of a point $x^* \in V(p)$ is the integer $\mu > 0$ such that $p(x) = (x - x^*)^\mu q(x)$ for a polynomial $q(x)$ with $q(x^*) \neq 0$. In the case $n = m > 1$, μ has a simple geometrical interpretation. If x^* is an isolated solution of (1.1) (with $n = m > 1$) and $v \in \mathbb{C}^n$ is a generic vector sufficiently near 0, then the system $f(x) = v$ has exactly μ nonsingular isolated solutions $x_1^*, \ldots, x_\mu^*$ near x^*. Thereby, nonsingular means that the Jacobian of the system is invertible at each $x_1^*, \ldots x_\mu^*$. However, for $n \neq m$, the meaning of multiplicity is not so closely connected to geometric intuition (see Sommese & Wampler [285]).

1.2.2 Number of Solutions

Systems of nonlinear polynomial equations generally have multiple solutions, in contrast to systems of linear equations. In a kinematic problem, the unknowns in such a system of equations represent information such as the joint angles or spatial displacements of a robot manipulator or platform. Thus, multiple solutions of the system represent different possible poses of the mechanism under the stated constraints. Bezout's Theorem [301] states that the number of solutions of (1.1) is equal to the total degree of the system, if the multiplicity of each solution is counted properly. Therefore, in general, two algebraic curves of degrees m and n intersect in $m \cdot n$ points and cannot meet in more than $m \cdot n$ points, unless they have a common component (see, e.g., [56, Ch. 8]). The number stated by Bezout is called the **Bezout number** or **Bezout bound**. It contains both the finite solutions as well as the so-called solutions at infinity and is a loose upper bound on the number of solutions of a polynomial system of equations.

A useful alternative to the total-degree Bezout number is the so-called **multi-homogeneous (m-homogeneous)** Bezout number. MORGAN & SOMMESE [210] introduced the use of multi-homogeneous structures (see also, e.g., LI [164]). It is based on a partition on the set of unknowns, where m equals the number of sets in the partition. Consider a system of n polynomial equations into n unknowns. The unknowns are divided in m groups, $S_1, \dots, S_m$, where S_j contains n_j elements, such that $n = n_1 + n_2 + \cdots + n_m$. The degree of f_i $(i = 1, \dots, n)$ as a polynomial in only the variables in S_j is denoted by d_{ij} for $j = 1, \dots, m$. Then the m-homogeneous Bezout number is the coefficient of the monomial $\prod_{j=1}^{m} \alpha_j^{n_j}$ in the polynomial $\prod_{i=1}^{n}(\sum_{j=1}^{m} d_{ij}\alpha_j)$. It gives an upper bound on the number of geometrically isolated solutions of (1.1) in a product of projective spaces $\times_{i=1}^{m} P^{n_i}$ [212]. Note, that the value of the m-homogeneous Bezout number depends on the grouping of the variables [253]. The following example illustrates the computation of the m-homogeneous Bezout number.

Example 1.2.4 (Bezout number [253]). *Consider the system of equations*

$$\begin{aligned} f_1 &\colon a_{11}x_1x_2 + a_{12}x_1 + a_{13} = 0 \\ f_2 &\colon a_{21}x_1x_2 + a_{22}x_1 + a_{23} = 0. \end{aligned} \tag{1.2}$$

The (total-degree) Bezout number of (1.2) is 4*, because both* f_1 *and* f_2 *are containing the term* x_1x_2 *and therefore are of degree two. For grouping the variables, there are only two possibilities, either leading to one group* $\{x_1, x_2\}$ *or to two groups,* $\{x_1\}, \{x_2\}$*. If* x_1 *and* x_2 *are assigned to two separate groups,* $n_1 = n_2 = 1$ *and* $m = 2$*. The degrees* d_{11} *and* d_{21} *of* f_1 *in the variable of group one and group two, respectively, are clearly one; as are the degrees* d_{12} *and* d_{22} *of* f_2*. Thus,* $\prod_{i=1}^{n}(\sum_{j=1}^{m} d_{ij}\alpha_j) = (\alpha_1 + \alpha_2)(\alpha_1 + \alpha_2)$ *and the* 2*-homogeneous Bezout number is the coefficient of* $\alpha_1^1\alpha_2^1$ *in this product, which is equal to* 2*.*

The use of multi-homogeneous variables may reduce the number of solutions at infinity in various situations and provides a good bound on the number of finite solutions [253]. Especially for polynomial continuation approaches, the m-homogeneous Bezout number has shown to be a useful tool, as it equals the number of solution paths used to compute all geometrically isolated solutions of a system using multi-homogeneous continuation [210, 285, 309] . WAMPLER [309] has given an algorithm for an efficient computation of Bezout numbers. However, MALAJOVICH & MEER [175] proved that the computation of the *optimal* m-homogeneous Bezout number is an NP-hard problem. An alternative for computing bounds on the number of solutions is the so-called **BKK bound**, according to BERNSTEIN, KHOVANSKII and KUSHNIRENKO. For details see, e.g., RAGHAVAN & ROTH [253], where a summary and further references are given.

1.3 Polynomial Continuation

The polynomial continuation method, also called *homotopy method*, is a numerical process for solving (1.1). In particular, isolated roots of the system are determined.

Polynomial continuation uses the fact that small changes in the parameters of the system of equations result in small changes in the numerical values of the solutions. Thus, the basic concept of polynomial continuation is to start with a suitable system of equations which is similar to the given one and of which the solutions are known, and track the solution paths as the coefficients of the start system are continuously transformed to the ones of the target system. A continuation algorithm has three essential components [64]:

(i) a homotopy defining a collection of algebraic curves,

(ii) a set of start points on these curves, and

(iii) a prescription for how to advance along the curves from the start points to a set of endpoints.

Since the early 1990s, the polynomial continuation method developed into a convenient, reliable tool for solving problems in kinematics [283]. For example, Raghavan [251] uses polynomial continuation for showing that the forward kinematics problem for a Stewart platform of general geometry (see Chapter 5) has 40 distinct solutions. According to Sommese et al. [283], the continuation approach has first been applied to kinematics problems by Roth & Freudenstein [82, 264] in 1963 as a heuristic. A modern approach with solid mathematical background has been given by Tsai & Morgan [299] in 1985. The modern approach makes essential use of complex numbers for avoiding singularities and other degeneracies and is presented in, e.g., [209] for engineering and kinematic problems.

Continuation is naturally applied to square systems, i.e., where the number of equations equals the number of unknowns. But especially in the past two decades, advances in the non-square case, i.e., when $n \neq m$ in (1.1), have been made, see, e.g., Wampler & Sommese [313, Ch. 7] or Sommese & Wampler [285]. A particularly important case for non-square systems in robotics is that of overconstrained mechanisms, which have more degrees of freedom of movement (cf. Chapter 3) than expected from the usual mobility calculation (for details on mobility indices see, e.g., Gogu [86]). These mechanisms may have a mixture of isolated solutions and motion curves of various dimensions [283]. Sommese et al. [283] introduced some advances in solving such problems and presented examples of mechanisms as well. In polynomial equations derived from kinematics problems often extraneous or degenerate (positive dimensional) solutions appear, which usually do not have physical meanings [293]. One method to identify such positive dimensional solutions is the so-called *regeneration method*, introduced by Hauenstein et al. [101], and further developed [102]. Another approach, the *constrained homotopy method*, is presented by Tari et al. [293]. Here, the basic idea is to augment the original system by additional constraint polynomials and map the unwanted solutions to solutions at infinity in the augmented system.

In the following, the basics of the continuation method for finding isolated roots of a square system of polynomial equations are briefly discussed. The explications are based on the work of WAMPLER & SOMMESE [313]; for a more detailed description see, e.g., [164, 285, 313]. For information about solving non-square systems see, e.g., [284, 285, 313]. There the main idea is to reduce the non-square system to a square one using a probabilistic method, which then allows to compute much of the information contained in the original system.

Definition 1.3.1 (Homotopy)**.** *A **homotopy** for (1.1) is a system of polynomials*

$$h(x,t) := \begin{pmatrix} h_1(x,t) \\ \vdots \\ h_n(x,t) \end{pmatrix},$$

with $h\colon \mathbb{C}^n \to \mathbb{C}^n$, $(x,t) \in \mathbb{C}^{n+1}$, and $h(x,0) = f(x)$.

According to WAMPLER & SOMMESE [313], one of the most classical homotopies is the **total-degree homotopy** which is defined by

Definition 1.3.2 (Total-degree homotopy)**.**

$$h(x,t) := (1-t)f(x) + tg(x), \tag{1.3}$$

with $g(x) = (g_1(x), \ldots, g_n(x))^T$, where g_i (for $i = 1, \ldots, n$) is a polynomial of degree $\deg(g_i) = \deg(f_i) = d_i$, whose solutions are known.

Here, for $t = 0$ the original system (1.1) is obtained while $t = 1$ produces the simplified system $g(x) = 0$. Note, that in several publications the parameters in (1.3) are chosen the opposite way, e.g., in [195, 253].

Bezout's Theorem states that the solution set of the system $g(x) = 0$ of general polynomials g_i with $\deg(g_i) = d_i$ consists of $\prod_{i=1}^n d_i$ nonsingular isolated solutions. The most general polynomials of degree d_j may have a large number of terms. But this level of complexity can be avoided. According to RAGHAVAN & ROTH [253], the polynomials g_i may be defined as $g_i = p_i^{d_i} x_i^{d_i} - q_i^{d_i}$, where p_i and q_i are random (non-zero) numbers. MORGAN & SOMMESE [211] states the γ-trick[1] which justifies the use of the very simple start system defined as

$$g_i(x) = x_i^{d_i} - 1 \quad \text{for } i = 1, \ldots, n.$$

For computing the solution set of the system $h(x,0) = f(x) = 0$, consider there is a function $x(t)\colon (0,1] \to \mathbb{C}^n$ such that

[1] The $\gamma-$trick states that with probability one, the homotopy

$$h(x,t) := (1-t)f(x) + \gamma t g(x)$$

with $f(x), g(x)$ as in (1.3) and γ a random complex number satisfies:

(i) $\{(x,t) \mid t \in (0,1], x \in \mathbb{C}^n, h(x,t) = 0)\} = \bigcup_{j=1}^D x(t)$, where $x_j(t)$ is a good path starting at the solutions of $h(x,1) = 0$ and $D := \prod_{i=1}^n d_i$, and

(ii) the set of limits $\lim_{t\to 0} x_j(t)$ that are finite include all the isolated solutions of $h(x,0) = f(x) = 0$ [313].

(i) $x(1) = y^*$, with y^* is an isolated solution of $h(x,1) = 0$,

(ii) $h(x(t), t) = 0$, and

(iii) the Jacobian $J_x h(x,t)$ of $h(x,t)$ with respect to the variables x is of maximal rank at $(x(t), t)$ for $t \in (0,1]$.

Then the path $x(t)$ satisfies the Davidenko differential equation

$$\sum_{j=1}^{n} \frac{\partial h(x,t)}{\partial x_j} \frac{dx_j(t)}{dt} + \frac{\partial h(x,t)}{\partial t} = 0. \tag{1.4}$$

Definition 1.3.3 (Good path)**.** *A path $x(t)\colon (0,1] \to \mathbb{C}^n$ satisfying conditions (i)–(iii) stated above and (1.4) is called a* ***good path*** *starting at y^*.*

Given the initial condition $x(1) = y^*$, the differential equation (1.4) can be solved numerically. This is typically done using a **pathtracking** procedure: an appropriate predictor step in the variable t is done and then Newton's method is used on $h(x,t) = 0$ in the variables $x_1, \ldots, x_n$ to compute a correction step (for details about pathtracking see, e.g., [2]). A solution x^* of (1.1) can then be computed as

$$x^* := \lim_{t \to 0} h(x(t), t),$$

or, in case of a diverging path, it can be concluded that the limit does not exist.

Different branches may be followed on a different node of a computer cluster, thus leading to parallel algorithms. A classical result is that a total-degree homotopy starting at a system $g(x)$, with the g_i being sufficiently general polynomials of degree $\deg(g_i) = d_i$, has good paths for the $D := \prod_{i=1}^{n} d_i$ nonsingular isolated solutions of $g(x) = 0$ [209]. Furthermore, the finite limits $\lim_{t\to 0} h(x(t), t)$ of the D paths contain all the isolated solutions of (1.1), though they often contain some points lying on positive-dimensional components of the solution set of $f = 0$ as well. The determination of the exact set of isolated solutions is a highly non-trivial task; an algorithm for doing so was first presented by Sommese et al. [282].

The selection of homotopy can make quite a difference in the number of paths; consider, e.g., a 3-RPR planar parallel manipulator (see Chapter 3) and the forward kinematics problem (see Chapter 5): a total-degree homotopy has 12 paths, while a 2-homogeneous homotopy has only 6 paths, which is the same as the generic number of solutions [313]. The m-homogeneous form often eliminates solutions at infinity [253]. Thus, if the m-homogeneous Bezout number for a given system $f(x)$ is smaller than the general Bezout number, it is wanted to construct a start system $g(x)$ with the same m-homogeneous Bezout number. This leads to fewer paths which have to be tracked than by using the general Bezout number [253]. In this context the notion of **good homotopies** $h(x,t)$ of an isolated solution x^* of $h(x,0) = f(x) = 0$ shall be mentioned.

Definition 1.3.4 (Good homotopy)**.** *A* ***good homotopy*** *is a homotopy with the properties that* *(i) there is at least one good path $x(t)$ with $x^* = lim_{t\to 0} h(x(t), t)$, and*

(ii) every path $x(t)\colon (0,1] \to \mathbb{C}^n$ with $x^* = lim_{t\to 0} h(x(t), t)$ is a good path.

In the end of the 20th century it was an important research topic to construct good homotopies where the number of paths is not too different from the number of isolated solutions of (1.1) (see [285]). If $h(x,t)$ is a good homotopy for a solution x^* of $h(x,0) = 0$, it can be shown that the number of paths ending in x^* equals the multiplicity of x^* as a solution of $h(x,0) = 0$ (see [285, 313]).

Continuation has been used to solve a variety of problems arising from kinematic analysis and synthesis of robot manipulators. Its earliest known appliance is the so-called *Bootstrap Method* developed by ROTH & FREUDENSTEIN [264] in the 1960s. Two highly important examples of a successful application of polynomial continuation are the numerical proof that the inverse kinematics problem of a general 6R serial manipulator has 16 solutions [299] (see Chapters 3, 5), and the proof that the direct kinematics problem of a generalized Stewart-platform has 40 solutions [251] (see Chapter 5). For the relatively new homotopy method for the non-square case, SOMMESE et al. [283] report a few applications for solving overdetermined polynomial systems in kinematics, e.g., an overconstrained planar mechanism or Griffis-Duffy examples of Stewart-Gough platform structures. For the planar seven-bar mechanism the results of the non-square homotopy method are consistent with known theory, while for the Griffis-Duffy II platform (see Fig. 5.4) a sixth-degree component was found by continuation which was not reported before.

But although the achieved results are highly valuable, a few difficulties arise when using polynomial continuation for kinematics problems. According to MERLET [198], the main drawback of the continuation method is that the auxiliary system $g(x)$ should have at least the same number of possible complex solutions as the final system $f(x)$, which is usually very large. Therefore, the number of branches followed will be large (e.g., 960 in the algorithm of RAGHAVAN [251]), which has strong influence on computation time. However, research has shifted towards lowering the number of tracks that must be followed [293].

Another problem is the difficulty of ensuring numerical robustness of the method if a singularity, i.e., a point (x', t') at which the Jacobian of (1.1) is singular, is encountered. Singular solutions are difficult to handle numerically. With fixed precision, numerical methods as homotopy will fail in the proximity of a singularity, due to the ill-conditioning of the Jacobian [11]; the convergence properties of Newton's method are destroyed near the singular point due to the vanishing derivatives. Therefore, tracking paths to x^* from a good homotopy for x^* is computationally expensive and often impossible in double precision [313]. Methods for dealing with singular points effectively are *endgames*, *adaptive precision* and *deflation* (see, e.g., [313]). BATES et al. [11] provide the first known set of heuristics for applying adaptive precision methods in the setting of higher-order predictor methods.

Finally, a well-known disadvantage of numerical methods for computing approximate solutions to systems of polynomials is that the output is not certified [103].

However, recently Beltrán & Leykin [13] have shown how to certify path-tracking by using Smale's α-theory (see, e.g., [19, 103]), and hence, the output of numerical homotopy algorithms [103]. Another algorithm (`alphaCertified`) has been proposed recently by Hauenstein & Sottile [103], which implements elements of α-theory to certify numerical solutions of systems of polynomial equations (i.e., the output of a numerical computation) using both exact rational and arbitrary precision floating point arithmetic.

There are several software packages available for computing isolated solutions of polynomial systems, e.g., `Bertini` [12], `HOM4PS-2.0` [159, 165], `Hompack90` [290, 320, 329] and `PHCpack` [303, 304] (see [313] for more details).

1.4 Gröbner Bases

The Gröbner basis approach gives a method for reformulating a system of algebraic equations (1.1) with finitely many solutions into a system $g(x)$ which has the same roots, but is in triangular form. The concept of Gröbner bases together with the characterization theorem were introduced by Buchberger in 1965 [32, 33]. It has become an important field of computer algebra, is implemented in every software system which does symbolic computation, and is applied to a large number of different seemingly unrelated problems [35]. To begin with, a few algebraic definitions are given.

Let $\mathbb{K}[x_1, \ldots, x_m]$ be the set of polynomials in $x_1, \ldots, x_m$ with coefficients in $\mathbb{K}$. A key ingredient for the triangularization of a set $f_1, \ldots, f_n$ of polynomials is the notion of **ordering of terms** in polynomials. A monomial $x^\alpha = \prod_{i=1}^m x_i^{\alpha_i}$ can be reconstructed from the n-tuple of exponents $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{Z}^m_{\geq 0}$. Therefore, any ordering established on $\mathbb{Z}^m_{\geq 0}$ gives an ordering on monomials (for a formal definition see [56, §2]). An important example for a monomial ordering is the **lexicographic order**, which states that

$$x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_r^{\alpha_r} <_{lex} x_1^{\beta_1} x_2^{\beta_2} \ldots x_r^{\beta_r}$$

if the leftmost nonzero entry in the difference of the exponent-vectors, i.e.,

$$[\alpha_1, \alpha_2, \ldots, \alpha_r] - [\beta_1, \beta_2, \ldots, \beta_r],$$

is negative. The lexicographic order works analogously to the alphabetical ordering in dictionaries. For example, $x_1^1 x_2^4 x_3^1 >_{lex} x_1^1 x_2^2 x_3^5$ since $\alpha - \beta = [0, 2, -4]$. Other examples for monomial orderings are the *graded lexicographic order* or the *graded reverse lexicographic order* (see [56]). For further discussion, it will be assumed that one particular monomial ordering has been selected; leading terms, etc., are computed relatively to that order.

If a monomial ordering is chosen, each $f \in \mathbb{K}[x_1, \ldots, x_m]$ has a unique leading term $\textsc{lt}(f)$. Thus, an **ideal of leading terms** can be defined for any ideal $\mathcal{I} \neq 0$ as $\langle \textsc{lt}(\mathcal{I}) \rangle$, the ideal generated by the elements of

$$\textsc{lt}(\mathcal{I}) := \{cx^\alpha \mid \exists f \in \mathcal{I} \text{ with } \textsc{lt}(f) = cx^\alpha\}.$$

Note, that for a finite generating set for $\mathcal{I} = \langle f_1, \ldots, f_s \rangle$ the ideals $\langle \text{LT}(f_1), \ldots, \text{LT}(f_s) \rangle$ and $\langle \text{LT}(\mathcal{I}) \rangle$ may be different; in particular,

$$\langle \text{LT}(f_1), \ldots, \text{LT}(f_s) \rangle \subsetneq \langle \text{LT}(\mathcal{I}) \rangle. \tag{1.5}$$

But it can be shown that there are $g_1, \ldots, g_t \in \mathcal{I}$ such that

$$\langle \text{LT}(\mathcal{I}) \rangle = \langle \text{LT}(g_1), \ldots, \text{LT}(g_t) \rangle.$$

This leads to the following important result

Theorem 1.4.1 (Hilbert basis theorem)**.** *Let $\mathcal{I} \subset \mathbb{K}[x_1, \ldots, x_m]$ be an arbitrary ideal. Then there exist $g_1, \ldots, g_t \in \mathcal{I}$ such that $\mathcal{I} = \langle g_1, \ldots, g_t \rangle$, i.e., every ideal $\mathcal{I} \subset \mathbb{K}[x_1, \ldots, x_m]$ has a finite generating set.*

Proof. See [56, Ch. 2,§5, Thm. 4] □

The basis $\{g_1, \ldots, g_t\}$ from Theorem 1.4.1 has the special property that equality holds in (1.5). Therefore, these bases are given a name:

Definition 1.4.2 (Gröbner basis I)**.** *A finite subset $G = \{g_1, \ldots, g_t\}$ of an ideal $\mathcal{I}$ is said to be a **Gröbner basis** if*

$$\langle \text{LT}(g_1), \ldots, \text{LT}(g_t) \rangle = \langle \text{LT}(\mathcal{I}) \rangle.$$

A more informal way of formulating the definition above is, that $G \subset \mathcal{I}$ is a Gröbner basis of $\mathcal{I}$ if and only if the leading term of any element of $\mathcal{I}$ is divisible by a $\text{LT}(g_i)$ for $i \in \{1, \ldots, t\}$. Alternatively, the following property is given for defining a Gröbner basis:

Definition 1.4.3 (Gröbner basis II)**.** *Let $\mathcal{I} \subset \mathbb{K}[x_1, \ldots, x_m]$, $f \in \mathbb{K}[x_1, \ldots, x_m]$ and $G = \{g_1, \ldots, g_t\} \subset \mathcal{I}$. Then G is called a **Gröbner basis** if the following property holds:*

$$f \in \mathcal{I} \iff \text{the normal form of } f \text{ modulo } G \text{ is zero.}$$

The normal form of f modulo G is introduced as follows [7, 56]:

Definition 1.4.4 (Reduction modulo and normal form modulo G)**.** *f is said to be **reducible modulo G**, iff there exists $g_i \in G$ such that $\text{LP}(g_i)$ divides $\text{LP}(f)$. Otherwise, f is **irreducible modulo G** and is said to be in **normal form** modulo G, denoted by $\bar{f}^G$. If f is reducible modulo g_i and*

$$\text{LT}(f) = t\ \text{LT}(g_i) \frac{\text{LC}(f)}{\text{LC}(g)},$$

*for a suitable power product t, then f is said to be **reduced modulo G** to*

$$f' = f - t\ \frac{\text{LC}(f)}{\text{LC}(g)}.$$

Consider for example [253]

$$f = 3x_1^2x_2 + 5x_1x_2^2 + 2x_1 + 8,$$
$$g = 7x_1^2 + 4x_2 + 14.$$

f may be reduced modulo g as follows:

$$f' = f - \frac{3}{7}x_2 g = 5x_1x_2^2 + 2x_1 - \frac{12}{7}x_2^2 - 6x_2 + 8.$$

As a consequence of Theorem 1.4.1 it can be shown that every ideal $\mathcal{I} \neq 0$ has a Gröbner basis. To decide wether a given basis of an ideal $\mathcal{I}$ is a Gröbner basis, the following key result was proved by BUCHBERGER [32, 33]. The presented version of computing the S-**pair** is taken from [253].

Theorem 1.4.5 (Buchberger's criterion)**.** *A basis $G = \{g_1, \ldots, g_t\}$ for a polynomial ideal $\mathcal{I}$ is a Gröbner basis for $\mathcal{I}$ if and only if for all pairs $i \neq j \in \{1, \ldots, t\}$ the normal form $\overline{\mathrm{S}(g_i, g_j)}^G$ of $\mathrm{S}(g_i, g_j)$ modulo G is zero. The* ***S-polynomial*** *$\mathrm{S}(g_1, g_2)$ of two nonzero polynomials g_1, g_2 is defined as*

$$\mathrm{S}(g_1, g_2) := \frac{b}{\mathrm{LP}(g_1)} \cdot g_1 - \frac{\mathrm{LC}(g_1)}{\mathrm{LC}(g_2)} \cdot \frac{b}{\mathrm{LP}(g_2)} g_2,$$

where $\mathrm{LP}(g_i)$ is the leading power product of g_i, $\mathrm{LC}(g_i)$ is the leading coefficient of g_i and b is the least common multiple of $(\mathrm{LP}(g_1), \mathrm{LP}(g_2))$.

Proof. See [56, Ch. 2, §6, Thm. 6]. □

The above S-pair criterion leads to an algorithm for computing Gröbner bases [34]. The main idea of the algorithm is to extend an arbitrary basis B of an ideal $\mathcal{I}$ to a Gröbner basis. Therefore remainders $\overline{\mathrm{S}(f_i, f_j)}^B \neq 0$ are added successively to B. Let $F = [f_1, \ldots, f_n]$ be a given set of polynomials and assume that the terms of F are arranged due to a monomial monomial ordering and decreasing from left to right. Then BUCHBERGER's algorithm for computing a Gröbner basis of F proceeds as follows [253]:

Step 1. $G := F$.

Step 2. Construct a set B of all pairs of polynomials in G, i.e.,
$B = \{(f_i, f_j) \mid f_i, f_j \in G, f_i \neq f_j\}$.

Step 3. While $B \neq \{\}$, perform the following steps

(3a) Pick an element (f_i, f_j) of B.

(3b) Remove this element from B, i.e., $B = B \backslash \{(f_i, f_j)\}$.

(3c) Compute $\mathrm{S}(f_i, f_j)$.

(3d) Compute $\overline{\mathrm{S}(f_i, f_j)}^G$.

(3e) If $\overline{\mathrm{S}(f_i, f_j)}^G \neq 0$, perform

(i) $G = G \cup \{\overline{\mathrm{S}(f_i, f_j)}^G\}$.

(ii) Update B, i.e., $B = B \cup \{(g, \overline{\mathrm{S}(f_i, f_j)}^G) \mid g \in G\}$.

Finding a Gröbner basis for a given system $f(x)$ of polynomial equations simplifies the form of the equations considerably. In particular, equations are obtained where the variables are eliminated successively, therefore resulting in a system of equations in triangular form in some sort. The last equation is a univariate polynomial, and each subsequent equation adds at most one new variable [222]. Such a system is easy to solve by applying one-variable techniques and substituting back and solve for the other variables. An example is given below [253]:

Example 1.4.6. *Consider the system of equations*

$$f(x_1, x_2) = \begin{pmatrix} x_2^2 + x_1^2 - 10x_1 \\ x_2^2 + x_1^2 - 16 \end{pmatrix} = 0. \tag{1.6}$$

Now, a Gröbner basis for the system (1.6) is computed. Consider $G = \{f_1, f_2\}$, $B = \{(f_1, f_2)\}$.

To begin with, consider the pair (f_1, f_2). $B \to B\backslash\{(f_1, f_2)\} = \{\}$ *and*

$$\mathrm{S}(f_1, f_2) = f_1 - f_2 = -10x_1 + 16,$$

which is in normal form modulo G*, i.e.,* $\overline{\mathrm{S}(f_1, f_2)}^G = \mathrm{S}(f_1, f_2) =: f_3$. *Then,* G *and* B *are updated:* $G \to \{f_1, f_2, f_3\}$, $B \to \{(f_1, f_3), (f_2, f_3)\}$.

Next, considering the pair (f_1, f_3) *leads to* $B \to B\backslash\{(f_1, f_3)\} = \{(f_2, f_3)\}$ *and*

$$\mathrm{S}(f_1, f_3) = x_1 f_1 - \frac{x_2^2}{10} f_3 = \frac{16}{10} x_2^2 + x_1^3 - 10x_1^2 =: f_4,$$

which may be further reduced modulo the elements of G*:*

$$f_4 - \frac{16}{10} f_1 + \frac{x_1^2}{10} f_3 - x_1 f_3 = 0.$$

Now, the remaining pair of B, (f_2, f_3) *shall be examined.* $B \to B\backslash\{(f_2, f_3)\} = \{\}$, *and*

$$\mathrm{S}(f_2, f_3) = x_1 f_2 + \frac{x_2^2}{10} f_3 = \frac{16}{10} x_2^2 + x_1^3 - 16x_1^2 =: f_5,$$

which may be reduced modulo G*:*

$$f_5 - \frac{16}{10} f_1 + \frac{x_2^2}{10} f_3 = 0.$$

Therefore, the normal forms of the S*-polynomials of* (f_1, f_3) *and* (f_2, f_3) *are zero. Since* B *is empty, the algorithm terminates and* $G = \{f_1, f_2, f_3\}$ *is a Gröbner basis.*

Therefore, the system (1.6) can be solved by calculating the roots of

$$G(x_1, x_2) = \begin{pmatrix} x_2^2 + x_1^2 - 10x_1 \\ x_2^2 + x_1^2 - 16 \\ -10x_1 + 16 \end{pmatrix} = 0,$$

leading to $x_1 = 1.6$*,* $x_2 = \pm 3.67$*. Thus, the solutions of (1.6) are* $(1.6, \pm 3.67)$.

The calculation time of Gröbner bases heavily depends on the size of the system of equations, i.e., the number of equations and their degree. Thus, to improve the efficiency of the algorithm, several improvements on Buchberger's algorithm have been made; some of these are discussed in [56], but it is still an active area of research. Also, the computation of a Gröbner basis may consume a huge amount of storage space. The algorithm may produce quite large intermediate polynomials before converging to the Gröbner basis; Mayr & Meyer [179] construct a Gröbner basis for an ideal generated by polynomials of degree less or equal to some d where polynomials of degree of the order 2^{2^d} are involved.

Another drawback is that the calculation with floating point numbers is numerically unstable. To fix this problem, initial floating point coefficients of the starting system may be converted to rational numbers, so Gröbner calculations are done over integers; but then the coefficients of the Gröbner basis may become huge [198]. Nevertheless, the technique has been proven quite useful in kinematic analysis; notable applications of the Gröbner basis approach to mechanism analysis have been done, e.g., by [213]. Faugère et al. [74, 75, 78, 80] have provided several algorithms (e.g., $\mathtt{F}_4$, $\mathtt{F}_5$, FGb) for the calculation of Gröbner bases. Further, recent developments of the Gröbner basis method have been conducted, e.g., by Mourrain & Trébuchet [214, 215]. A method is presented for computing border bases of a zero-dimensional ideal; the given algorithm weakens the monomial ordering requirement for Gröbner basis computations.

See Chapter 5 for further discussion of recent applications of Gröbner bases to kinematic problems.

1.5 Elimination

In the following, the basic idea for solving systems of polynomial equations as in (1.1) by Sylvester's dialytic elimination procedure is described. Consider an algebraic system of n equations in the n unknowns $\{x_1, \ldots, x_n\}$ ($m = n$ in (1.1)), where each equation is a sum of monomials. The main idea of the elimination method is to first suppress one variable (x_1), so the system is recast in polynomials in $\{x_2, \ldots, x_n\}$ with coefficients depending on x_1 [198, 312]. (Since the unknowns may be renumbered at will, there is no loss of generality by assuming x_1 as hidden variable.) In the next step, new polynomials are formed by algebraically combining the original polynomials, until a square system of K equations in K unknown monomials is obtained. The augmented system of equations may be written in matrix form as linear system

$$[A(x_1)]m = 0, \tag{1.7}$$

where m is a column vector of the monomials (including the constant monomial 1). For this system to have a nontrivial solution, $m \neq 0$, it is necessary that

$$\det A(x_1) = 0. \tag{1.8}$$

If the equations of (1.7) are linearly independent, the solutions of (1.8) contain the solutions of the original system. In general, some extraneous solutions may be contained as well. After solving the univariate polynomial (1.8), a backtracking procedure is used to determine all x_i. The elimination procedure is illustrated by a simple example [222].

Example 1.5.1. *Consider the system*

$$\begin{aligned} 3x_1^2x_2^2 + 2x_1 + 9 &= 0 \\ 6x_1x_2 + x_1 + 8 &= 0 \end{aligned} \tag{1.9}$$

Now the system is rewritten with one variable "hidden" in the coefficient field. So, by suppressing x_2, (1.9) becomes

$$\begin{aligned} (3x_2^2)x_1^2 + (2)x_1 + (9)1 &= 0 \\ (6x_2 + 1)x_1 + (8)1 &= 0, \end{aligned} \tag{1.10}$$

where the coefficients are in parentheses, and the constant 1 is treated as an unknown. The system in (1.10) is now linear in the unknowns $\{x_1^2, x_1, 1\}$, so there are two equations in three unknowns (note that the monomials x_1^2 and x_1 are treated as separate linear variables). Thus, one more equation is needed to properly constrain the system; it may be obtained by multiplying the second equation by x_1. Rewriting the set of equations in matrix form yields a system which is homogeneous in the three "linear" unknowns:

$$\begin{pmatrix} 3x_2^2 & 2 & 9 \\ 0 & 6x_2+1 & 8 \\ 6x_2+1 & 8 & 0 \end{pmatrix} \begin{pmatrix} x_1^2 \\ x_1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{1.11}$$

Thus, for solutions to exist, the determinant of the coefficient matrix, must vanish. Finding the zeros of the determinant yields all values of x_2 for which a solution of the original system exists. The determinant of the coefficient matrix for (1.10) is

$$g(x_2) = -516x_2^2 - 12x_2 + 7,$$

with zeros $x_{2_1} = -0.12868$ and $x_{2_2} = 0.10542$. The values for x_1 can be obtained from (1.11) by omitting the last equation and treating the 1 as coefficient. Thus, the solutions of (1.9) are $(x_{1_1}, x_{2_1}) = (-0.12868, -35.100)$, and $(x_{1_2}, x_{2_2}) = (0.10542, -4.9003)$.

For two equations it is straightforward to specify the multiplying terms for the elimination procedure. The result is the classical Sylvester resultant of two binary forms [222]; an example for three polynomials in three variables and an extension of the procedure for solving the inverse kinematics of a 6 d.o.f. serial manipulator (see Chapter 3) is given in [216, p. 108f]. Raghavan & Roth [252, 253] introduced a modification of Sylvester's Dialytic Elimination procedure for solving a set of nonlinear equations. It consists of six basic steps [253]:

Step 1. Rewrite all equations with one variable suppressed (w.l.o.g. x_1).

Step 2. Define the new power products of remaining variables as new "linear" unknowns.

Step 3. Generate as many new linearly independent equations as number of linear unknowns from the original equations, i.e., a $K \times K$ system is obtained.

Step 4. Set the determinant of the coefficient matrix to zero. A polynomial in the suppressed variable is obtained.

Step 5. Determine the roots of the characteristic polynomial or the eigenvalues of the matrix. As a result all possible values for the suppressed variable x_1 are derived.

Step 6. Substitute one of the roots (or eigenvalues) for x_1 into the original system of equations and solve the linear system for the remaining unknowns. Repeat the procedure for each value of x_1.

In principle, the described procedure will always work if enough new equations can be determined in *Step 1*. Note, that it is important that the value of K, i.e., the size of the new system of polynomials, is as small as possible. In kinematic analysis one possible approach for generating a minimal K is to use the trigonometric relations that exist among the coefficients of the governing equations [253] (see, e.g., [158, 252]).

The computation of the determinant of A in *Step 4* usually is quite challenging. In general, gathering an analytical form of the determinant is difficult, and as a rule of thumb it cannot be obtained for $K > 5$ [198]. Additionally, round-off errors may affect the coefficients of the characteristic polynomial. Here, rationals can be used instead of floating point numbers but this will lead to large integers, and using software which allows multiple precision is necessary. One way to avoid this drawback of the elimination method is to compute the determinant of A numerically for random values of x_1; for a polynomial of degree m it is sufficient to compute the determinant at $m+1$ random values for x_1 to get a system of $m+1$ linear equations in the $m+1$ coefficients of the polynomial. These are solved numerically to get the coefficients. However, the resulting Vandermonde system is extremely ill-conditioned. So this approach is very sensitive to round-off errors and hence far from being numerically robust [198].

The procedure of expanding the determinant of the coefficient matrix (*Step 4*) can be omitted if the problem is solved as eigenvalue problem [253]. The final matrix equation of *Step 3* may be written as

$$[A_n x^n + A_{n-1} x^{n-1} + \cdots + A_0] m = 0, \tag{1.12}$$

where x is the suppressed variable, and m is a vector of unsuppressed power products including 1. All values of x for which the determinant of the matrix in (1.12) vanishes can be found by solving the generalized eigenvalue problem

$$Gz - xHz = 0, \tag{1.13}$$

where

$$H = \begin{pmatrix} I & 0 & 0 & \cdots & \\ 0 & I & 0 & \cdots & \\ 0 & 0 & I & \cdots & \\ & \vdots & & \ddots & \\ A_{n-1} & A_{n-2} & A_{n-3} & \cdots & A_0 \end{pmatrix},$$

$z = (x^{n-1}m, x^{n-2}m, \ldots, m)^T$, and x is the suppressed variable [222]. If H is invertible, (1.13) is a regular eigenvalue problem, i.e.,

$$H^{-1}Gz - xz = 0,$$

otherwise (1.13) may be solved using the QZ-algorithm. Note, that the corresponding eigenvectors contain the values for all power products. Thus, there are two ways for deriving the values of x after performing *Steps 1–3*; either expand the determinant of an $K \times K$ matrix which contains the suppressed variable of degree n, or solve a generalized eigenvalue problem of size Kn. According to Nielsen & Roth [222], the eigenvalue routine is much easier to implement. Additionally, robust eigenvalue routines are available. The eigenvalue approach is illustrated for a polynomial system in two variables in [253, p. 73]. Further, e.g., Wampler [312] proposes a modified elimination method using standard eigenvalue procedures for solving the loop equations of the displacement analysis of spherical mechanisms (see Chapter 3) for up to three loops.

The most crucial step of the dialytic elimination process is *Step 3*, where the linear equations are generated, usually by multiplying the original equations by powers of one ore more of the variables. A more structured and rigorous way to construct the determinantal polynomial equation in one variable might be taken by using an entity known as **resultant**. Results about resultants may be found in Gelfand et al. [85]. Algorithms for constructing a resultant have been proposed, e.g., by Pedersen & Sturmfels [234] or Canny & Emiris [37]. An overview of uni- and multivariate resultants is availably in Wee & Goldman [322, 323]. Resultants proceed to construct the determinantal equation similarly to the dialytic process, but the power products used as multipliers to generate new equations are selected in a systematic manner. This approach ensures that the final determinantal equation in one variable contains no extraneous solutions.

The main disadvantage of Sylvester's dialytic elimination method is the challenge of finding an appropriate multivariate eliminant for a particular problem. For more than two equations there are various ways to produce the matrix A, which leads to different polynomials with different degrees. Theoretically, Sylvester's dialytic elimination method will reduce any system of multivariate polynomial equations to a single polynomial in one unknown. But practically it only is applicable to small sets of polynomial equations, because the resulting polynomial may explode exponentially with number and degree of equations, and may produce a large number of extraneous solutions [298].

The strength of elimination-based solution methods is computation time. If such a method can be found for a particular problem, it is normally much faster than polynomial continuation or methods based on Gröbner bases [222]. Raghavan & Roth [253] indicates that particularly dialytic procedures in combination with eigenvalue routines often produce computationally fast algorithms. Besides, the elimination method gives greater insight into a problem than purely numerical solution methods by permitting studies of the solution space as a function of the structural parameters of a linkage [253].

Sylvester's dialytic elimination method is widely used in kinematics. In particular, the application to serial mechanisms is quite successful, because certain functions of the equations are a composition of the same power products as the original equations. Thus, the extra equations are obtained without introducing new power products and so the eliminant calculation becomes feasible [222]. An example for an important application of elimination theory to kinematics is the solution of the inverse kinematics problem for serial manipulators by Raghavan & Roth [252] in 1991 (see also Chapters 3, 5). According to Raghavan & Roth [253], dialytic elimination methods are very useful for small problems with up to 6 variables, while polynomial continuation is more appropriate for large problems with many solutions. Combinations of these methods are recommended as well, e.g., using polynomial continuation for determining the number of finite solutions of a larger problem and then, if this number is not too large, constructing a solution using dialytic elimination. More applications of the various methods are given in Chapter 5.

Chapter 2

Interval Analysis

2.1 Introduction

The basic idea of interval analysis is to enclose numbers in intervals, and vectors and matrices in boxes. Then, classical arithmetic and analysis are extended to or replaced by corresponding interval concepts. Intervals and ranges of functions arise naturally in several applications. For example, measured variables (e.g., a joint angle between two adjacent links of a robot arm) are only known approximately. Hence, it is a realistic description, that the variable lies in a certain range, i.e., in an interval, whose width depends on the accuracy of the measurement. Further, in machine computations, numbers are represented with a finite number of digits (finite precision arithmetic). This implies the presence of roundoff errors for the input data, the computations and, for mathematical constants. For example, with five significant digits, $\pi \approx 3.1416$. But a sharper representation is to give the tightest representable lower and upper bounds. So, with five significant decimal digits, $\pi \in [3.1415, 3.1416]$ [218]. Thus, interval arithmetical tools and methods are useful for solving problems in the presence of uncertainties in certain parameters and finite precision computer arithmetic. As uncertainties are (due to, e.g., measurement errors) omnipresent in robotics and computations are done with finite precision arithmetic, interval analysis methods are quite useful for this field of applications (see, e.g., [100, 149, 157, 195, 196, 203, 334])

Another aspect in the use of interval analysis is the certification of a derived result. For several applications in robotics for which safety is a crucial issue, e.g., medical or space robotics, it is indispensable to ensure rigorous bounds on a solution (see [200, 201]). Interval algorithms provide rigorous and reliable enclosures for the solution of numerical problems, even in the presence of round-off errors and measurement uncertainties.

In the following, basic properties of interval arithmetic as well as an overview of several tools essential for solving linear and nonlinear systems of equations is given. The considerations are based on the book of Neumaier [218], where more detailed explanations can be found.

2.2 Basic Properties of Interval Arithmetic

2.2.1 Intervals

A (real) **interval** is a set
$$\mathbf{a} := [\underline{a}, \overline{a}],$$
with $\underline{a} \leq \overline{a} \in \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$, i.e., $\mathbf{a} = \{\tilde{a} \in \mathbb{R} \mid \underline{a} \leq \tilde{a} \leq \overline{a}\}$. The **set of real intervals** is denoted by $\overline{\mathbb{IR}}$ and the **set of nonempty bounded real intervals** by $\mathbb{IR}$.

An unknown number may be represented by an approximation plus/minus an error bound. Therefore, the **midpoint** $\check{a}$ of an interval $\mathbf{a}$ is introduced as

$$\check{a} := \frac{(\underline{a} + \overline{a})}{2},$$

and the **radius** of $\mathbf{a}$ is given by

$$\operatorname{rad}(\mathbf{a}) := \frac{(\overline{a} - \underline{a})}{2}.$$

With these definitions the membership relation can now be expressed as

$$\tilde{a} \in \mathbf{a} \iff |\tilde{a} - \check{a}| \leq \operatorname{rad}(\mathbf{a}),$$

i.e.,
$$\mathbf{a} = [\check{a} - \operatorname{rad}(\mathbf{a}), \check{a} + \operatorname{rad}(\mathbf{a})].$$

For the extension of the absolute value of a real number, several interval extensions are possible. Two useful extensions are the **magnitude**

$$|\mathbf{a}| := \max\{|\tilde{a}| \mid \tilde{a} \in \mathbf{a}\},$$

and the **mignitude**
$$\langle \mathbf{a} \rangle := \min\{|\tilde{a}| \mid \tilde{a} \in \mathbf{a}\}$$
of an interval $\mathbf{a}$. Usually, the terms "magnitude" and "absolute value" are used synonymously.

For an nonempty bounded subset $S \in \mathbb{R}$ we denote by

$$\square S := [\inf(S), \sup(S)]$$

the **hull** of S, i.e., the tightest interval enclosing S.

To define **elementary operations**, i.e., $\circ \in \Omega := \{+, -, *, /, ^\wedge\}$, on the set of intervals, let $\mathbf{a}, \mathbf{b} \in \overline{\mathbb{IR}}$. Then

$$\mathbf{a} \circ \mathbf{b} := \square\{\tilde{a} \circ \tilde{b} \mid \tilde{a} \in \mathbf{a}, \tilde{b} \in \mathbf{b} \text{ and } (\tilde{a}, \tilde{b}) \in D[\circ]\},$$

where $D[\circ]$ denotes the domain of definition, in particular $D[/] = \overline{\mathbb{IR}} \times (\overline{\mathbb{IR}} \setminus \{0\})$ and $D[\hat{}\,] = \overline{\mathbb{IR}}_+ \times \overline{\mathbb{IR}}$, where $\overline{\mathbb{IR}}_+ = \{\mathbf{a} \in \overline{\mathbb{IR}} \mid \mathbf{a} \geq 0\}$.

The extension of **elementary functions** $\varphi \colon \mathbb{R} \to \mathbb{R}$ to $\varphi : \overline{\mathbb{IR}} \to \overline{\mathbb{IR}}$ is defined as

$$\varphi(\mathbf{a}) := \Box\{\varphi(\tilde{a}) \mid \tilde{a} \in \mathbf{a}, \tilde{a} \in D(\varphi)\}.$$

We will only use this form of extending univariate functions for a fixed set Φ of elementary functions

$$\Phi = \{\log, \exp, \sin, \cos, \sinh, \cosh, \bullet^n (n \in \mathbb{N}), \sqrt[n]{\bullet} (n \in \mathbb{N}), \arctan, \text{abs}\}.$$

In most cases, the lower and upper bounds of the resulting intervals can be computed by using the endpoints of the given intervals. For example, the addition of two intervals is given by $\mathbf{a} + \mathbf{b} = [\underline{a} + \underline{b}, \overline{a} + \overline{b}]$. For multiplication and division, the result depends on the signs of $\mathbf{a}$ and $\mathbf{b}$, and for division the result additionally depends on whether $0 \in \mathbf{a}$ and/or $0 \in \mathbf{b}$. A detailed description is given in [218, Ch. 1.2].

2.2.2 Rounded interval arithmetic

When interval arithmetic is realized on a computer, intervals in general have to be rounded. This is because there is only a finite set $\mathbb{M} \in \overline{\mathbb{R}}$ of **machine-representable numbers** (in most cases the floating point numbers). Therefore, the **optimal directed rounding** operations are defined as

$$\begin{aligned} \nabla r &:= \sup\{x_m \in \mathbb{M} \mid x_m \leq r\}, \quad \nabla \colon \overline{\mathbb{R}} \to \mathbb{M}, \text{ and} \\ \Delta r &:= \inf\{x_m \in \mathbb{M} \mid x_m \geq r\}, \quad \Delta \colon \overline{\mathbb{R}} \to \mathbb{M} \end{aligned}$$

for **rounding downwards** and for **rounding upwards**, respectively. With this notations, the **optimal outward rounding** is given by

$$\begin{aligned} \diamond \colon \overline{\mathbb{IR}} &\to \mathbb{IM} \\ [\underline{a}, \overline{a}] &\mapsto [\nabla \underline{a}, \Delta \overline{a}], \end{aligned}$$

where $\mathbb{IM}$ denotes the subset of $\overline{\mathbb{IR}}$ such that $\mathbf{a} \in \mathbb{IM}$ if and only if $\underline{a}, \overline{a} \in \mathbb{M}$. So, all arithmetic operations $\circ \in \Omega$ and elementary functions $\varphi \in \Phi$ on a computer are defined by

$$\begin{aligned} \mathbf{a} \circ \mathbf{b} &:= \diamond(\mathbf{a} \circ \mathbf{b}), \\ \varphi(\mathbf{a}) &:= \diamond\varphi(\mathbf{a}). \end{aligned}$$

Note, that it implicitly is assumed that $\pm\infty \in \mathbb{M}$, $0 \in \mathbb{M}$ and so, rounding upwards and downwards, respectively, always exists.

2.2.3 Boxes, interval vectors, and interval matrices

To cover expressions in several variables, some notation for interval vectors and interval matrices is introduced. Let $\mathbb{IR}^n$ be the **space of interval vectors**, i.e., the space of column vectors of intervals, also called **boxes**. Then $\mathbf{x} \in \mathbb{IR}^n$ is given by

$$\mathbf{x} = [\underline{x}, \overline{x}] = \{\tilde{x} \in \mathbb{R}^n \mid \underline{x} \leq \tilde{x} \leq \overline{x}\}$$

with $\underline{x}, \overline{x} \in \overline{\mathbb{R}}^n$. In a similar way, interval matrices are defined. An **m × n interval matrix** is a rectangular array of intervals $\mathbf{A}_{ik} \in \overline{\mathbb{IR}}$. Let $\overline{\mathbb{IR}}^{m\times n}$ be the set of all $m \times n$ interval matrices, then

$$\mathbf{A} = [\underline{A}, \overline{A}] = \{\tilde{A} \in \mathbb{R}^{m\times n} \mid \underline{A} \leq \tilde{A} \leq \overline{A}\}$$

with $\underline{A}, \overline{A} \in \overline{\mathbb{R}}^{m\times n}$. If Σ is a bounded set of real $m \times n$ matrices, then its **hull** is defined as

$$\Box\Sigma := [\inf(\Sigma), \sup(\Sigma)],$$

which is the tightest interval matrix enclosing Σ.

An important abstract tool for the study of iterative methods for the solution of linear and nonlinear systems of equations is the notion of **sublinear mappings**, which are mappings $S\colon \mathbb{IR}^n \to \mathbb{IR}^n$ which satisfy the axioms

(S1) $\mathbf{x} \subseteq \mathbf{y} \implies S\mathbf{x} \subseteq S\mathbf{y}$ (inclusion isotonicity),

(S2) $\alpha \in \mathbb{R} \implies S(\alpha\mathbf{x}) = \alpha(S\mathbf{x})$ (homogeneity), and

(S3) $S(\mathbf{x} + \mathbf{y}) \subseteq S\mathbf{x} + S\mathbf{y}$ (subadditivity)

for all $\mathbf{x}, \mathbf{y} \in \mathbb{IR}^n$. If equality holds in (S3), the mapping is called **linear**. Two important tools for analyzing sublinear mappings are the **absolute value of S**, defined as

$$|S| := |S[-\mathbb{1}, \mathbb{1}]|,$$

and the **core of S**, given by

$$\operatorname{cor}(S) := S\mathbb{1} = (Se_1, \ldots, Se_2).$$

Theorem 2.2.1. *Let $S\colon \mathbb{IR}^n \to \mathbb{IR}^m$ be sublinear. Then*

$$\begin{aligned} |\operatorname{cor}(S)| &\leq |S|, \\ S\mathbf{x} &\subseteq \operatorname{cor}(S)\check{x} + |S|(\mathbf{x} - \check{x}) \quad \forall \mathbf{x} \in \mathbb{IR}^n. \end{aligned} \tag{2.1}$$

If S is linear, then equality holds in (2.1). In particular, a linear mapping is uniquely determined by its core and its absolute value.

Proof. See Neumaier [218, Thm. 3.5.4]. □

A **linear interval equation**

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

with coefficient matrix $\mathbf{A} \in \mathbb{IR}^{m\times n}$ and right-hand side $\mathbf{b} \in \mathbb{IR}^{m}$ is defined as the family of linear equations

$$\tilde{A}\tilde{x} = \tilde{b} \quad \text{with } \tilde{A} \in \mathbf{A}, \tilde{b} \in \mathbf{b}. \tag{2.2}$$

We are interested in enclosing its **solution set**, given by

$$\Sigma(\mathbf{A}, \mathbf{b}) := \{\tilde{x} \in \mathbb{R}^n \mid \tilde{A}\tilde{x} = \tilde{b} \text{ for some } \tilde{A} \in \mathbf{A}, \tilde{b} \in \mathbf{b}\}. \tag{2.3}$$

In general, $\Sigma(\mathbf{A}, \mathbf{b})$ is not an interval vector and may have a very complicated structure. In order to guarantee its boundedness, $\mathbf{A}$ is required to be **regular**, i.e., that $\operatorname{rank}(\tilde{A}) = n$ for all $\tilde{A} \in \mathbf{A}$. Thus, if $\mathbf{A} \in \mathbb{IR}^{n\times n}$ the **hull of the solution set**

$$\mathbf{A}^H\mathbf{b} := \square\Sigma(\mathbf{A}, \mathbf{b})$$

is defined and, in particular, defines a mapping $\mathbf{A}^H\colon \mathbb{IR}^n \to \mathbb{IR}^n$, the so-called **hull inverse** of $\mathbf{A}$. For rectangular matrices $\mathbf{A} \in \mathbb{IR}^{m\times n}$ (and $m \geq n$, because otherwise $\Sigma(\mathbf{A}, \mathbf{b})$ is either empty or unbounded and thus, $\mathbf{A}^H\mathbf{b}$ is not defined), it makes sense to consider instead of (2.3) the **least squares solution set**

$$\Sigma_{LS}(\mathbf{A}, \mathbf{b}) := \{\tilde{A}^\dagger\tilde{b} \mid \tilde{A} \in \mathbf{A}, \tilde{b} \in \mathbf{b}\},$$

where $\tilde{A}^\dagger = (\tilde{A}^T\tilde{A})^{-1}\tilde{A}^T$ is the pseudo-inverse of $\tilde{A}$, and the **least squares hull** of $\mathbf{A}$ and $\mathbf{b}$, given by

$$\mathbf{A}^L\mathbf{b} := \square\Sigma_{LS}(\mathbf{A}, \mathbf{b}).$$

Clearly,

$$\mathbf{A}^H\mathbf{b} \subseteq \mathbf{A}^L\mathbf{b}.$$

The solution set $\Sigma(\mathbf{A}, \mathbf{b})$ of (2.2) has the following neat characterization

Theorem 2.2.2 (Beeck)**.** *Let* $\mathbf{A} \in \mathbb{IR}^{m\times n}, \mathbf{b} \in \mathbb{IR}^m$. *Then*

$$\Sigma(\mathbf{A}, \mathbf{b}) = \{\tilde{x} \in \mathbb{R}^n \mid \mathbf{A}\tilde{x} \cap \mathbf{b} \neq 0\} = \{\tilde{x} \in \mathbb{R}^n \mid 0 \in \mathbf{A}\tilde{x} - \mathbf{b}\}.$$

Proof. See Neumaier [218, Thm. 3.4.3]. □

The **matrix inverse** of a regular, square interval matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ can be defined as

$$\mathbf{A}^{-1} := \square\{\tilde{A}^{-1} \mid \tilde{A} \in \mathbf{A}\}.$$

An **inverse positive matrix** is a regular square interval matrix with nonnegative inverse. The most important subclass of these consists of so-called M-matrices, which play a distinguished role in the context of interval calculations, as they behave particularly well in algorithms for the solution of linear interval equations. An ***M*-matrix** is a square matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ such that $\mathbf{A}_{ik} \leq 0$ for all $i \neq k$, and $\mathbf{A}u > 0$ for some positive vector $u \in \mathbb{R}^n$. Every M-matrix is regular. Furthermore, for inverse positive matrices the inverse $\mathbf{A}^{-1}$ and the hull inverse $\mathbf{A}^H\mathbf{b}$ can be computed explicitly (see [218, Ch. 3.6]).

A generalization of M-matrices, the so-called H-matrices, are obtained by lifting the restrictive sign condition in the definition of an M-matrix. Therefore, the **comparison matrix** $\langle\mathbf{A}\rangle$ with entries

$$\langle\mathbf{A}\rangle_{ik} := \begin{cases} \langle\mathbf{A}_{ik}\rangle & \text{if } i = k \\ -|\mathbf{A}_{ik}| & \text{if } i \neq k \end{cases}$$

is introduced for an arbitrary square matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$. The comparison matrix always fullfills the sign-condition of an M-matrix. Now, an ***H*-matrix** is a square matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ such that its comparison matrix $\langle\mathbf{A}\rangle$ is an M-matrix; equivalently, $\mathbf{A}$ is an H-matrix if and only if $\langle\mathbf{A}\rangle u > 0$ for some $u > 0$. In particular, every H-matrix is regular and $0 \notin \mathbf{A}_{ii}$ for all $i = 1, \ldots, n$.

Every M-matrix is an H-matrix. In the special case $u = (1, \ldots, 1)^T$, a matrix $\mathbf{A}$ satisfying $\langle\mathbf{A}\rangle u > 0$ is called a **strictly diagonally dominant matrix**. It is an important fact, that a matrix which is sufficiently close to the identity matrix, is strictly diagonally dominant, and therefore an H-matrix. For example, every regular (lower or upper) triangular matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ is an H-matrix.

2.2.4 Interval extensions of arithmetic expressions

If $D \subseteq \overline{\mathbb{R}}^n$, then $\mathbb{I}D$ is written for the set of intervals $\mathbf{x} \in \overline{\mathbb{IR}}^n$ with $\mathbf{x} \subseteq D$. An interval function $f\colon \mathbb{I}D \subseteq \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}^m$ is called **inclusion isotone** if,

$$\mathbf{a} \subseteq \mathbf{b} \implies f(\mathbf{a}) \subseteq f(\mathbf{b}) \quad \forall \mathbf{a}, \mathbf{b} \in \mathbb{I}D.$$

Further, an interval function $f\colon \overline{\mathbb{IR}}^n \to \overline{\mathbb{IR}}^m$ is said to be an **interval extension** of the real function $f_0\colon D \subseteq \mathbb{R}^n \to \mathbb{R}^m$ if

$$f(\tilde{x}) = f_0(\tilde{x}) \quad \forall \tilde{x} \in D, \tag{2.4a}$$

$$f_0(\tilde{x}) \in f(\mathbf{x}) \quad \forall \tilde{x} \in \mathbf{x} \in \mathbb{I}D, \tag{2.4b}$$

$$f(\mathbf{x}) = \mathrm{NaN} \quad \text{for } \mathbf{x} \notin \mathbb{I}D, \tag{2.4c}$$

where the symbol NaN stands for "not a number", which may be thought of as "undefined" or "unspecific value". If only (2.4b) and (2.4c) are satisfied, f is called a **weak interval extension** or an **interval enclosure** of f_0. A large class of inclusion isotone interval extensions of real functions can be defined through arithmetic expressions. An **arithmetic expression** in the formal variables $\xi_1, \ldots, \xi_n$ is a member of the set $\mathcal{A} = \mathcal{A}(\xi_1, \ldots, \xi_n)$ defined by

(i) $\mathbb{R} \in \mathcal{A}$,

(ii) $\xi_l \in \mathcal{A}$ for $l = 1, \ldots, n$,

(iii) $g, h \in \mathcal{A} \implies (g \circ h) \in \mathcal{A}$ for all $\circ \in \Omega$,

(iv) $g \in \mathcal{A} \implies \varphi(g) \in \mathcal{A}$ for all elementary functions $\varphi \in \Phi$,

(v) among the sets which satisfy (i)–(iv), $\mathcal{A}$ is minimal with respect to inclusion.

A subexpression of an arithmetic expression f is an expression occurring in the recursive definition of f, formally, k is a subexpression of f if either

(i) $f = k$, or

(ii) $f = g \circ h$ and k is a subexpression of g or h, or

(iii) $f = \varphi(g)$ and k is a subexpression of g.

If $f = f(\xi)$ is an arithmetic expression in n variables, the **value** $f(\mathbf{x})$ of f at $\mathbf{x} \in \mathbb{IR}^n$ is obtained by substituting the intervals $\mathbf{x}_i$ for the corresponding formal parameters ξ_i (**interval evaluation**). On a computer the interval evaluation is simulated by a rounded evaluation, the **outward rounded value** of f is denoted by $f^\diamond(\mathbf{x})$.

2.2.5 Enclosures for the range of a function

Useful general results about the interval evaluation of arithmetic functions can be obtained, if the evaluation at intervals are avoided, in which the expressions are not sufficiently smooth. Therefore, Lipschitz properties for interval functions and arithmetic expressions are introduced.

An interval function $f\colon \mathbb{IR}^n \to \mathbb{IR}$ is called **Lipschitz continuous** in $\mathbf{x}^0 \in \mathbb{IR}^n$, if $\mathbf{x}^0 \subseteq D$ and there exists $l^0 \in \mathbb{R}^n$ such that

$$q(f(\mathbf{x}^1), f(\mathbf{x}^2)) \le l^0\, q(\mathbf{x}^1, \mathbf{x}^2) \quad \forall\, \mathbf{x}^1, \mathbf{x}^2 \subseteq \mathbf{x}^0,$$

where $q(\mathbf{a}, \mathbf{b}) := |\check{a} - \check{b}| + |\operatorname{rad}(\mathbf{a}) - \operatorname{rad}(\mathbf{b})|$ is the distance function, which is used component-wise on vectors. An arithmetic expression f in n variables is called **Lipschitz** at $\mathbf{x}^0 \in \mathbb{IR}^n$ if for all subexpressions g, h of f, the relation $g = h^\alpha$ for $0 < \alpha < 1$ implies $h(\mathbf{x}^0) > 0$, and the relation $g = \varphi(h)$ for $\varphi \in \Phi$ implies that φ is defined and Lipschitz continuous in a neighborhood of $h(\mathbf{x}^0)$. f is called **locally Lipschitz** in $\mathbf{x}^0 \in \mathbb{IR}^n$ if f is Lipschitz at every $\tilde{x} \in \mathbf{x}^0$. For the standard elementary functions the property of being Lipschitz only is marginally stronger than the property of being defined.

The mean value form and other centered forms

The interval evaluation $f(\mathbf{x})$ of an arithmetic expression f in n variables at an interval $\mathbf{x} \in \mathbb{IR}^n$ is an enclosure for the range

$$f^*(\mathbf{x}) := \{f(\tilde{x}) \mid \tilde{x} \in \mathbf{x}\}.$$

However, the overestimation may be large. Therefore, methods are needed to enclose the range $f^*(\mathbf{x})$ in such a way that the overestimation remains small for sufficiently narrow boxes $\mathbf{x}$. A very useful approach to bound the range is to consider the deviation of $f(\tilde{x})$ from a fixed center $\tilde{z} \in \mathbf{x}$. Thus, the resulting formulae are called **centered forms**. The simplest one is the **mean value form**

$$f_m(\mathbf{x}) := f(\check{x}) + f'(\mathbf{x})(\mathbf{x} - \check{x}). \tag{2.5}$$

Note that different expressions for the derivative f' may change the value of (2.5) (sometimes quite drastically), while different expressions f for the same real function do not affect the value of (2.5). The mean value form can be shown to be inclusion isotone. Further, it satisfies the important **quadratic approximation property**

$$0 \le \operatorname{rad}(f_m(\mathbf{x})) - \operatorname{rad}(f^*(\mathbf{x})) \le 2\operatorname{rad}(f'(\mathbf{x}))\operatorname{rad}(\mathbf{x}),$$

which expresses the overestimation of the range by a product of two radii. In particular, for narrow intervals the mean value form in general gives a much better enclosure for the range than the interval evaluation.

To derive more general centered forms, the notion of slopes shall be introduced. For a Lipschitz continuous function $f\colon \mathbb{R}^n \to \mathbb{R}$ one can always write

$$f(x) = f(z) + f[z,x](x - z)$$

for any two points x and z with a suitable vector $f[z,x] \in \mathbb{R}^n$, called a **slope vector** for f at (z,x). In the univariate case $(n = 1)$

$$f[z,x] = \begin{cases} \frac{f(x)-f(z)}{x-z} & \text{if } x \neq z,\\ f'(z) & \text{if } x = z, \end{cases}$$

hence $f[z,x]$ is uniquely determined. In the multivariate case, however, the slope $f[z,x]$ is no longer unique while $f[z,z] = \nabla f(z)$, i.e., the gradient of f. As slopes obey a similar chain rule as derivatives, recursive procedures to calculate $f[z,x]$ given x and z can be developed (see, e.g., Krawczyk & Neumaier [150]). If we have a slope $f[\tilde{z},\tilde{x}]$ for all $\tilde{x} \in \mathbf{x}$ and $f[\zeta,\xi]$ is an arithmetic expression in ζ and ξ, then the range $f^*(\mathbf{x})$ can also be enclosed by the **slope form** with center $\tilde{z} \in \mathbf{x}$, defined as

$$f_{s,\tilde{z}}(\mathbf{x}) := f(\tilde{z}) + f[\tilde{z},\mathbf{x}](\mathbf{x} - \tilde{z}). \tag{2.6}$$

The slope form is inclusion isotone as well. More generally, formulae $f(\tilde{z}) + \mathbf{s}^T(\tilde{x} - \tilde{z})$ $(\mathbf{s} \in \mathbb{IR}^n)$, for which $f(\tilde{x}) = f(\tilde{z}) + \tilde{s}^T(\tilde{x} - \tilde{z})$ holds for all $\tilde{x} \in \mathbf{x}$ and some $\tilde{s} \in \mathbf{s}$, are called **centered form** for f with **center** $\tilde{z}$. All centered forms satisfy the overestimation bound

$$q(f(\tilde{z}) + s^T(\mathbf{x} - \tilde{z}), f^*(\mathbf{x})) \leq 2\,\mathrm{rad}(s)|\mathbf{x} - \tilde{z}|,$$

but need not to be inclusion isotone. However, for fixed centers inclusion isotonicity is a trivial consequence of (2.6).

A further improvement for the enclosure for the range is obtained by taking two different centers $\tilde{z}$ and $\tilde{w}$ intersecting the two corresponding slope forms and get so-called **bicentered forms**

$$f_b(\mathbf{x}) := f_{s,\tilde{z}}(\mathbf{x}) \cap f_{s,\tilde{w}}(\mathbf{x}).$$

If the slope vector $f[z, x]$ itself is Lipschitz continuous, it can be written as

$$f[z, x] = f[z, w] + (x - w)^T f[w, z, x] \tag{2.7}$$

for arbitrary $x, w, z \in \mathbb{R}^n$, with a **second order (bicentered) slope matrix** $f[w, z, x] \in \mathbb{R}^{n\times n}$ [273]. Using this the second order slope for f at z and w, a range estimate for f can be computed using the **second order slope form**, given by

$$f_{ss,\tilde{z},\tilde{w}}(\mathbf{x}) := f(\tilde{z}) + (f[\tilde{z}, \tilde{w}] + (\mathbf{x} - \tilde{w})^T f[\tilde{w}, \tilde{z}, \mathbf{x}])(\mathbf{x} - \tilde{z}).$$

These estimates in general are tighter than analogous ones computed by interval derivatives. All second order centered forms have a cubic approximation property. Slopes of arbitrary order can be derived by defining a slope of order k in the same way as in (2.7) as a slope of slopes of order $k - 1$ [273].

2.3 Solving Linear Systems of Equations

For solving linear systems of equations, the class of strongly regular matrices (see below) is an important feature. It can be shown, that for linear interval equations with a strongly regular coefficient matrix the solution set can be enclosed by the solution set of a related, preconditioned system whose coefficient matrix is an H-matrix. If $\tilde{x}$ is a solution of the system

$$\tilde{A}\tilde{x} = \tilde{b} \quad \text{for } \tilde{A} \in \mathbf{A} \text{ regular}, \tilde{b} \in \mathbf{b}, \tag{2.8}$$

and $\mathrm{rad}(\mathbf{A})$ is sufficiently small, then $\tilde{A} \approx \check{A}$, and $\tilde{x} = \tilde{A}^{-1}\tilde{b} \approx \check{A}^{-1}\tilde{b}$. The error made can be analyzed by multiplying (2.8) by $\check{A}^{-1}$.Thus, $\tilde{x}$ is the solution of a system with coefficient matrix $\check{A}^{-1}\tilde{A} \approx \mathbb{1}$ and right-hand side $\check{A}^{-1}\tilde{b}$. Hence,

$$\tilde{x} \in \Sigma(\check{A}^{-1}\mathbf{A}, \check{A}^{-1}\mathbf{b}) \subseteq (\check{A}^{-1}\mathbf{A})^H(\check{A}^{-1}\mathbf{b})$$

if $\check{A}^{-1}\mathbf{A}$ is regular. The transformation above is called **preconditioning by the midpoint inverse**, and $\check{A}^{-1}$ is called the **preconditioner** or **midpoint preconditioner**. A matrix $\mathbf{A} \in \mathbb{IR}^{n\times n}$ is called **strongly regular** if $\check{A}^{-1}\mathbf{A}$ is regular.

In particular, $\mathbf{A}$ being strongly regular is equivalent to $\check{A}^{-1}\mathbf{A}$ being an H-matrix. However, a useful preconditioning does not depend on $\check{A}^{-1}$ as preconditioner. Instead, $\mathbf{A}$ can be multiplied by any matrix $C \in \mathbb{R}^{n\times n}$ and further, it can also be post-multiplied by some matrix $C' \in \mathbb{R}^{n\times n}$, as long as the resulting matrix is regular.

In the following, two methods for solving linear interval systems of equations are outlined, namely Krawczyk iteration and interval Gauss-Seidel iteration.

Krawczyk iteration

Krawczyk's iteration method provides computable bounds for the overestimation of an enclosure of $\mathbf{A}^H\mathbf{b}$ as well as enclosures which have the quadratic approximation property. It can be shown that for a square system of linear equations with a strongly regular coefficient matrix, a (rather crude) enclosure for the hull $\mathbf{A}^H\mathbf{b}$ of the solution set is given by

$$\mathbf{A}^H\mathbf{b} \subseteq \|C\mathbf{b}\|_v[-u, u], \tag{2.9}$$

if $u > 0$, and $\langle C\mathbf{A}\rangle u \geq v > 0$. A better enclosure might be derived from (2.9), if an approximation $\tilde{x}$ to a particular solution is found first and then added to an enclosure for the residual correction $\mathbf{A}^H(\mathbf{b} - \mathbf{A}\tilde{x})$ (obtained using (2.9). In particular, $\mathbf{A}^H\mathbf{b} \subseteq \tilde{x} + \mathbf{A}^H(\mathbf{b} - \mathbf{A}\tilde{x})$. If $\mathbf{b} - \mathbf{A}\tilde{x}$ is of order $O(\epsilon)$, this causes that

$$\mathbf{A}^H\mathbf{b} \subseteq \tilde{x} + \|C(\mathbf{b} - \mathbf{A}\tilde{x})\|_v[-u, u] \tag{2.10}$$

has an overestimation of $O(\epsilon)$ and thus, it is an enclosure for $\mathbf{A}^H\mathbf{b}$ with linear approximation order. A method with quadratic approximation order can be realized by improving the initial enclosure (2.10) iteratively. Therefore, consider the relation

$$\tilde{A}^{-1}\tilde{b} = C\tilde{b} - (C\tilde{A} - \mathbb{1})(\tilde{A}^{-1}\tilde{b}) \in C\mathbf{b} - (C\mathbf{A} - \mathbb{1})\mathbf{x},$$

where $\mathbf{x}$ is an enclosure for $\mathbf{A}^H\mathbf{b}$. This holds for all $\tilde{A} \in \mathbf{A}$ and $\tilde{b} \in \mathbf{b}$, implying

$$\mathbf{A}^H\mathbf{b} \subseteq \mathbf{x} \implies \mathbf{A}^H\mathbf{b} \subseteq (C\mathbf{b} - (C\mathbf{A} - \mathbb{1})\mathbf{x}) \cap \mathbf{x}.$$

This relation is the input to the so-called **Krawczyk iteration**

$$\begin{aligned} \mathbf{z}^0 &:= \mathbf{x} \\ \mathbf{z}^{l+1} &:= (C\mathbf{b} - (C\mathbf{A} - \mathbb{1})\mathbf{z}^l) \cap \mathbf{z}^l. \end{aligned}$$

The $\mathbf{z}^l$ form a nested sequence of interval vectors, and by construction $\mathbf{A}^H\mathbf{b} \subseteq \mathbf{x}$ implies $\mathbf{A}^H\mathbf{b} \subseteq \mathbf{z}^l$ for all $l \geq 0$. It can be shown (see [218, Ch. 4.2]), that for interval input whose radii are small enough the Krawczyk iteration has quadratic approximation order, i.e., if the enclosure (2.10) is used as an initial enclosure for the iteration the Krawczyk iterates $\mathbf{z}^l$ with $l \geq 1$ enclose the hull $\mathbf{A}^H\mathbf{b}$ of the solution set $\Sigma(\mathbf{A}, \mathbf{b})$ with quadratic approximation order. For $C\mathbf{A}$ close to the identity, in the limit the Krawczyk iteration produces enclosures of $\mathbf{A}^H\mathbf{b}$ which have a bounded radius overestimation factor independent of the right-hand side. Therefore, after sufficiently many iterations, reliable enclosures are also obtained for wide right-hand sides $\mathbf{b}$ (see [218, Thm. 4.2.4]).

Interval Gauss-Seidel iteration

Krawczyk's iteration is a reliable method for deriving realistic enclosures of the solution set of linear equations whose coefficients are narrow intervals. However, there is a superior method, the so-called interval Gauss-Seidel iteration. For M-matrices $\mathbf{A}$, Gauss-Seidel iteration converges for every $\mathbf{b} \in \mathbb{IR}^n$ to the hull $\mathbf{A}^H\mathbf{b}$ of the solution set (see [218, Thm. 4.4.8]). If Gauss-Seidel iteration is applied to a preconditioned system, it always yields at least at tight intervals as the Krawczyk iteration (see [218, Thm. 4.3.5]).

For arbitrary $\mathbf{x} \in \mathbb{IR}^n$, good enclosures for the **truncated solution set**

$$\Sigma_{\mathbf{x}}(\mathbf{A}, \mathbf{b}) := \Sigma(\mathbf{A}, \mathbf{b}) \cap \mathbf{x} = \{\tilde{x} \in \mathbf{x} \mid \tilde{A}\tilde{x} = \tilde{b} \text{ for some } \tilde{A} \in \mathbf{A}, \tilde{b} \in \mathbf{b}\}$$

shall be found. Note, that $\Sigma_{\mathbf{x}}$ is bounded, even if $\mathbf{A}$ is singular. Gauss-Seidel iteration is based on writing the system $\tilde{A}\tilde{x} = \tilde{b}$ explicitly in components as

$$\sum_{k=1}^{n} \tilde{A}_{ik}\tilde{x}_k = \tilde{b}_i \quad (i = 1, \ldots, n),$$

and, assuming that $\tilde{A}_{ii} \neq 0$, the ith equation is solved for the ith variable. Hence,

$$\tilde{x}_i = \frac{1}{\tilde{A}_{ii}}(\tilde{b}_i - \sum_{k \neq i} \tilde{A}_{ik}\tilde{x}_k) \subseteq \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{k \neq i} \mathbf{A}_{ik}\mathbf{x}_k) \cap \mathbf{x}_i =: \mathbf{x}'_i. \tag{2.11}$$

If $\mathbf{A}_{ii} \neq 0$ for all i, relation (2.11) can be applied for all i and thereby, a new enclosure $\mathbf{x}'$ for $\tilde{x}$ can be computed from a known interval vector $\mathbf{x}$ containing $\tilde{x}$. However, the available information can be used in an improved way. In the ith step, an enclosure for $\tilde{x}_i$ and improved enclosures for $\tilde{x}_i, \ldots, \tilde{x}_{i-1}$ are already available and can be used to obtain an improved enclosure of the right-hand side of (2.11) to get an improved enclosure for $\tilde{x}_i$. Thus, $\tilde{x}$ is contained in the vector $\mathbf{y}$ given by

$$\mathbf{y}_i := \frac{1}{\mathbf{A}_{ii}}(\mathbf{b}_i - \sum_{k<i} \mathbf{A}_{ik}\mathbf{y}_k - \sum_{k>i} \mathbf{A}_{ik}\mathbf{x}_k) \cap \mathbf{x}_i \quad (i = 1, \ldots, n).$$

For intervals $\mathbf{a}, \mathbf{b}, \mathbf{x} \in \mathbb{IR}$ we define

$$\Gamma(\mathbf{a}, \mathbf{b}, \mathbf{x}) := \square\{\tilde{x} \in \mathbf{x} \mid \tilde{a}\tilde{x} = \tilde{b} \text{ for some } \tilde{a} \in \mathbf{a}, \tilde{b} \in \mathbf{b}\}.$$

Then

$$\Gamma(\mathbf{a}, \mathbf{b}, \mathbf{x}) = \begin{cases} \mathbf{b}/\mathbf{a} \cap \mathbf{x} & \text{if } 0 \notin \mathbf{a}, \\ \square(\mathbf{x} \setminus \,]\frac{\underline{b}}{\underline{a}}, \frac{\underline{b}}{\overline{a}}[) & \text{if } \mathbf{b} > 0 \in \mathbf{a}, \\ \square(\mathbf{x} \setminus \,]\frac{\overline{b}}{\overline{a}}, \frac{\overline{b}}{\underline{a}}[) & \text{if } \mathbf{b} < 0 \in \mathbf{a}, \\ \mathbf{x} & \text{otherwise.} \end{cases}$$

In extension of the notation we define the **(interval) Gauss-Seidel operator** $\Gamma(\mathbf{A}, \mathbf{b}, \mathbf{x})$ with entries

$$\Gamma(\mathbf{A}, \mathbf{b}, \mathbf{x})_i := \Gamma(\mathbf{A}_{ii}, \mathbf{b}_i - \sum_{k<i} \mathbf{A}_{ik}\mathbf{y}_k - \sum_{k>i} \mathbf{A}_{ik}\mathbf{x}_k, \mathbf{x}_i) \quad (i = 1, \ldots, n).$$

If $\Gamma(\mathbf{A},\mathbf{b},\mathbf{x}) \subsetneq \mathbf{x}$, we can hope to further improve the enclosure of $\mathbf{A}^H\mathbf{b}$ by reapplying Γ. Thus, the **interval Gauss-Seidel iteration** is defined by

$$\begin{aligned}\mathbf{x}^0 &:= \mathbf{x}\\ \mathbf{x}^{l+1} &:= \Gamma(\mathbf{A},\mathbf{b},\mathbf{x}^l) \quad \text{for } l = 0,1,2,\dots,\end{aligned}$$

and the **preconditioned Gauss-Seidel iteration** for $C \in \mathbb{R}^{n\times n}$ can be defined similarly with $\mathbf{x}^{l+1} := \Gamma(C\mathbf{A}, C\mathbf{b}, \mathbf{x}^l)$. We then have $\Sigma_{\mathbf{x}}(\mathbf{A},\mathbf{b}) \subseteq \Sigma_{\mathbf{x}}(C\mathbf{A},C\mathbf{b}) \subseteq \mathbf{x}^l$ for all $l \geq 0$.

2.4 Solving Nonlinear Systems of Equations

Now we consider the problem to determine whether a continuous function $F\colon D_0 \subseteq \mathbb{R}^n \to \mathbb{R}^n$ has a **zero** x^* in a given subset $D \subseteq D_0$ and when such a zero is unique. Therefore, the Newton operator, the Krawczyk operator and the Hansen-Sengupta operator are introduced, which provide simple computational tests for (unique) existence and nonexistence of a zero in a given box. To obtain uniqueness results, a form of Lipschitz continuity is assumed, while the existence problem can be treated assuming only that F is continuous.

A subset $A \subseteq \mathbb{R}^{m\times n}$ is called a **Lipschitz set** for F on $D \subseteq D_0$ if it is convex, compact, and

$$F(\tilde{x}) - F(\tilde{y}) = \tilde{A}(\tilde{x} - \tilde{y}) \quad \text{for some } \tilde{A} \in A$$

holds for every $\tilde{x}, \tilde{y} \in D$. If additionally $A \in \mathbb{IR}^{m\times n}$, it is called a **Lipschitz matrix**. In particular, for every interval extension of the derivative F' on D the matrix $\mathbf{A} = F'(\mathbf{x})$ is a Lipschitz-matrix, if F is continuously differentiable in D, $\mathbf{x} \in \mathbb{I}D$ and $\mathbf{A} \in \mathbb{IR}^{m\times n}$. A Lipschitz set $\mathbf{A}$ of matrices $\tilde{A} \in \mathbb{R}^{n\times n}$ is called **regular** if all $\tilde{A} \in \mathbf{A}$ are nonsingular.

Theorem 2.4.1 (Moore, Nickel). *Let $F\colon D_0 \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be Lipschitz continuous on $D \subseteq D_0$, and let $\mathbf{A}$ be a regular Lipschitz matrix on* D. *If $\tilde{x} \in \mathbf{x} \in \mathbb{I}D$ then every $\mathbf{x}' \in \mathbb{IR}^n$ satisfying*

$$\mathrm{N}(\mathbf{x},\tilde{x}) := \tilde{x} - \mathbf{A}^H F(\tilde{x}) \subseteq \mathbf{x}'$$

has the following three properties:

(i) Every zero $x^* \in \mathbf{x}$ of F *satisfies $x^* \in \mathbf{x}'$.*

(ii) If $\mathbf{x}' \cap \mathbf{x} = \emptyset$ then F *has no zeros in $\mathbf{x}$.*

(iii) If $\tilde{x} \in \mathrm{int}(\mathbf{x})$ and $\mathbf{x}' \subseteq \mathbf{x}$ then F *contains a unique zero in $\mathbf{x}$ (and hence in $\mathbf{x}'$).*

Proof. See Neumaier [218, Thm. 5.1.7]. □

Note, that the condition $\tilde{x} \in \operatorname{int}(\mathbf{x})$ in (iii) can be replaced by the more practical requirement $\mathbf{x}' \subseteq \operatorname{int}(\mathbf{x})$. $\mathrm{N}(\mathbf{x}, \tilde{x})$ is called the **(interval) Newton operator** and can be interpreted as a verified version of the Newton iterate, which yields enclosures for the zero of F (which is unique under the present assumptions). Furthermore, $\mathrm{N}(\mathbf{x}, \tilde{x})$ provides a nonexistence test (Theorem 2.4.1(ii)) and an existence and uniqueness test (Theorem 2.4.1 (iii)), which are not available in the classical Newton iteration. However, the Newton operator requires $\mathbf{A}$ to be regular.

There are two more operators, which basically have the same properties as the Newton operator, but can be defined for a Lipschitz-matrix $\mathbf{A}$ and thus are more widely applicable. These are the **Krawczyk operator**, which is given by

$$\mathrm{K}(\mathbf{x}, \tilde{x}) := \tilde{x} - CF(\tilde{x}) - (C\mathbf{A} - \mathbb{1})(\mathbf{x} - \tilde{x}),$$

and the **Hansen-Sengupta operator**, which is defined as

$$\mathrm{H}(\mathbf{x}, \tilde{x}) := \tilde{x} + \Gamma(C\mathbf{A}, -CF(\tilde{x}, (\mathbf{x} - \tilde{x})).$$

Those are nonlinear variants of the Krawczyk iteration and the interval Gauss-Seidel iteration. Note, that both operators involve a preconditioning matrix C. The inclusion

$$\mathrm{H}(\mathbf{x}, \tilde{x}) \subseteq \mathrm{K}(\mathbf{x}, \tilde{x}) \tag{2.12}$$

holds for all $\tilde{x} \in \mathbf{x} \subseteq \mathbf{x}^0$. Hence, the Hansen-Sengupta operator gives at least as good enclosures for a zero of F as the formally simpler Krawczyk operator. A result similar to Thm. 2.4.1 holds with $\mathrm{H}(\mathbf{x}, \tilde{x})$ and $\mathrm{K}(\mathbf{x}, \tilde{x})$, but with a weaker existence test than with $\mathrm{N}(\mathbf{x}, \tilde{x})$.

Let $\mathrm{P}(\mathbf{x}, \tilde{x})$ be any of $\mathrm{N}(\mathbf{x}, \tilde{x})$, $\mathrm{K}(\mathbf{x}, \tilde{x})$ and $\mathrm{H}(\mathbf{x}, \tilde{x})$. Then an iteration for improving interval enclosures of a zero x^* of a function F by repeated application of $\mathrm{P}(\mathbf{x}, \tilde{x})$ is defined as

$$\begin{aligned} \mathbf{x}^0 &:= \mathbf{x} \\ \mathbf{x}^{l+1} &:= P(\mathbf{x}^l, \tilde{x}^l) \cap \mathbf{x}^l \quad (l \geq 0,\ \tilde{x}^l \in \mathbf{x}^l). \end{aligned} \tag{2.13}$$

By construction,

$$\mathbf{x}^{l+1} \subseteq \mathbf{x}^l \subseteq \cdots \subseteq \mathbf{x}^0 = \mathbf{x}, \tag{2.14}$$

and we know

$$x^* \in \mathbf{x} \text{ with } F(x^*) = 0 \implies x^* \in \mathbf{x}^l \ \forall l \geq 0. \tag{2.15}$$

In particular, the limit

$$\mathbf{x}^\infty := \lim_{l \to \infty} \mathbf{x}^l$$

exists, and if the starting box $\mathbf{x}$ contains a zero x^* of F then $x^* \in \mathbf{x}^\infty$. Note that in finite precision arithmetic the expression $F(\tilde{x})$ in the operator $\mathrm{P}(\mathbf{x}, \tilde{x})$ needs to be evaluated with outward rounding. Otherwise the solution may suddenly disappear ($\mathbf{x}^{l+1} = \emptyset$) due to rounding errors. Under suitable conditions it can be shown that the following alternative holds:

(SC) *Either* F has a unique zero $x^* \in \mathbf{x}$ and $\mathbf{x}^\infty = x^*$,
or F has no zero in $\mathbf{x}$ and $\mathbf{x}^l = \emptyset$ for some $l \geq 0$.

In particular, if (SC) holds, the iteration is capable of deciding whether there is a zero of F in $\mathbf{x}$ or not, and, assuming sufficiently accurate arithmetic, the zero can be enclosed with arbitrary accuracy. A sequence $\mathbf{x}^l$ ($l = 0, 1, 2, \dots$) satisfying (2.14) and (2.15) is called **strongly convergent**, if (SC) holds. For the Krawczyk iteration ($\mathrm{P}(\mathbf{x}, \tilde{x}) = \mathrm{K}(\mathbf{x}, \tilde{x})$ in (2.13)) strong convergence can be proven for the choice $\tilde{x}^l = \check{x}^l$ and, under strengthened assumptions, for other choices of $\tilde{x}$ as well. Since (2.12) always holds, these results immediately extend to the Hansen-Sengupta iteration ($\mathrm{P}(\mathbf{x}, \tilde{x}) = \mathrm{H}(\mathbf{x}, \tilde{x})$ in (2.13)). In that case, much weaker assumptions are sufficient to prove strong convergence. However, results about convergence rates have not been obtained under these weaker conditions.

In order to consider convergence conditions for the Newton-iteration, we consider the **general preconditioned Newton-iteration**, i.e., $\mathrm{P}(\mathbf{x}, \tilde{x}) = \mathrm{N}_I(\mathbf{x}, \tilde{x})$ in (2.13) with the **general Newton operator**

$$\mathrm{N}_I(\mathbf{x}, \tilde{x}) := \tilde{x} - (C\mathbf{A})^I (CF(\tilde{x})),$$

where $C \in \mathbb{R}^{n \times n}$ is a preconditioning matrix and $(C\mathbf{A})^I$ denotes an inverse of $(C\mathbf{A})$, i.e., a sublinear mapping with $(C\mathbf{A})^H \mathbf{b} \subseteq (C\mathbf{A})^I \mathbf{b}$ for all $\mathbf{b} \in \mathbb{IR}^n$. If $(C\mathbf{A})^I$ is regular, the general Newton iteration is strongly convergent for every choice $\tilde{x}^l \in \mathbf{x}^l$. Moreover, for all $l \geq 0$,

$$\text{either } \tilde{x}^l \notin \mathbf{x}^{l+1}, \text{ or } \mathbf{x}^{l+1} = \tilde{x}^l \text{ and } F(\mathbf{x}^l) = 0. \tag{2.16}$$

Relation (2.16) implies a volume-reducing property of the Newton iteration. More specifically, the general Newton iteration with $\tilde{x}^l = \check{x}^l$ satisfies

$$\operatorname{vol} \mathbf{x}^{l+1} \leq \tfrac{1}{2} \operatorname{vol} \mathbf{x}^l, \tag{2.17}$$

where $\operatorname{vol} \mathbf{x} := (\overline{x}_1 - \underline{x}_1)(\overline{x}_2 - \underline{x}_2) \cdot \ldots \cdot (\overline{x}_n - \underline{x}_n)$ denotes the **volume** of a box $\mathbf{x} \in \mathbb{IR}^n$. Further, if $C\mathbf{A}$ is an M-matrix or $\|\mathbb{1} - C\mathbf{A}\| < \frac{1}{2}$, then the **optimal Newton iteration**, i.e., the general Newton iteration ($\mathrm{P}(\mathbf{x}, \tilde{x}) = \mathrm{N}_I(\mathbf{x}, \tilde{x})$)with $C = \mathbb{1}$ and $\mathbf{A}^I = \mathbf{A}^H$, is strongly convergent for every choice of $\tilde{x}^l \in \mathbf{x}^l$, and (2.16) and (2.17) hold.

Summarizing the results, Hansen-Sengupta iteration strongly converges under the mildest assumptions, but it may have a slow convergence rate, and not even a good approximation to the zero can accelerate the iteration. Newton iteration, on the other hand, considerably speeds up using good approximations to the zero. However, when the hull inverse is replaced by an easier computable inverse, the iteration may stop prematurely. But a combination of both methods, given by

$$\begin{aligned} \mathbf{x}^0 &:= \mathbf{x} \\ \mathbf{x}^{l+1/2} &:= \mathrm{N}_I(\mathbf{x}, \tilde{x}^l) \cap \mathbf{x}^l \\ \mathbf{x}^{l+1} &:= \mathrm{H}(\mathbf{x}^{l+1/2}, \tilde{x}^l), \end{aligned}$$

where $\tilde{x}^l \in \mathbf{x}^l$ for all $l \geq 0$, $C\mathbf{A}$ is an H-matrix, and $(C\mathbf{A})^I$ is an inverse of $C\mathbf{A}$, preserves the advantages and avoids the disadvantages of both methods. With this combination an algorithm can be produced, which decides whether a zero x^* of F is contained in a given box $\mathbf{x} \in \mathbb{IR}^n$ and if affirmative, finds a good enclosure of x^*. The relation $\mathbf{x}^{l+1} \subseteq \mathbf{x}^l$ and finite precision arithmetic imply finite termination of the iteration.

For finding zeros of continuous parameter-dependent functions which are not necessarily Lipschitz continuous, general existence results can be derived. However, due to the weaker assumptions, uniqueness results can no longer be proved. In a similar way as the centered forms are sharpened when slopes are used instead of derivatives, the use of slopes in place of Lipschitz sets improves the enclosure and iteration methods for deriving zeros (for details see Neumaier [218, Ch. 5.4].)

Finally, we have discuss methods for computing narrow enclosures for *all* solutions $x^* \in D$ of $F(x^*) = 0$ with $F\colon D_0 \subseteq \mathbb{R}^n \to \mathbb{R}^n$ continuous. The methods considered so far were constructive only in the case when D was a box and the Lipschitz matrix of F at this box was regular. If these assumptions are dropped, on the one hand, the problem has to be reduced to one with bounded domain. This can be handled using a priori bounds or suitable problem transformations. On the other hand, the existence of several solutions may be possible. Therefore bisection is used. Assume that zeros are sought in a box $\mathbf{x}^0 \in \mathbb{IR}^n$. If this box is narrow enough, one of the above described methods can be applied. In the case that $\mathbf{x}^0$ is too wide, bisection methods must replenish the former methods. The basic principle of **bisection** is to split boxes repeatedly, until they can be further reduced by means of interval iteration. Therefore, a list of currently unprocessed boxes are stored in a so-called **stack** or **queue**. If there are several isolated solutions in the initial box $\mathbf{x}^0$, they will be isolated by bisection.

Interval analysis based algorithms are frequently used in global optimization, where the absolute minimum of the objective function F is sought. Therefore, **branch & bound-algorithms** are used (see, e.g., Moore et al. [208]). However, these methods frequently have the difficulty that subboxes containing no solution cannot be eliminated easily if they are close to the global minimum. Thus, the so-called **cluster effect** occurs. That means, many small boxes are created near each global minimum (and, in the process of solving the problem also near the currently best found local minimum) by repeated splitting, and often the processing of these boxes dominates the total work spent on the global search. The cluster effect can be reduced by defining exclusion regions around each local minimum found. It is guaranteed, that exclusion regions do not contain other local minima, and can therefore safely be discarded. Recently, Schichl et al. [274] discussed exclusion regions, which are constructed using uniqueness tests based on the Krawczyk operator.

Part II

Robotics

Chapter 3

Architecture

3.1 Introduction

The architecture of a robotic mechanism is concerned with the defining topology, the basic connectivity pattern of a robotic system. For both serial and parallel robots, many different architectures are possible (see Figures 3.11, 3.10, 3.5, 3.4, 5.4, 5.3 and, e.g., [57, 198, 286]). For a quantitative treatment, realistic models of mechanisms are needed. Therefore, in the following the basic structural elements of robot mechanisms are described. Further, two commonly used descriptions of open-loop kinematic chains using Denavit-Hartenberg parameters and screw displacements, respectively, are given. Finally, the notion of *separable* in terms of robot manipulators is introduced.

3.2 General structure of robot mechanisms

A robotic system is composed of various interrelated subsystems, namely a *mechanical* subsystem, a *sensing* subsystem, an *actuation* subsystem, a *controlling* subsystem and an *information processing* subsystem [6]. In the following, only the first one, the mechanical subsystem, will be considered. Hence, a simplified model of a manipulator consists of a system of mechanical links and joints. These are designed in such a way that one or more end-effectors (e.g., grippers, fingers, drills) can perform certain tasks. To move the robot, some of the joints are actuated, so they can be moved within certain limits. The actual joint coordinates are measured by various kinds of sensors and fed into a controller. The controller uses information provided by the information processing system and sends appropriate orders about the forces and torques needed to move the joints toward the desired position, to the actuators.

A frequently used idealization of a manipulator consists of a number of rigid bodies called **links**; sometimes several links are assumed as deformable, e.g., cable-links, which complicate the analysis of the manipulator. In the following, only mechanisms with rigid links are considered. Usually, a manipulator is fixed to an initial link, the **base**; the final link is called the **end-effector**. Links are coupled by **joints** to so-called **kinematic pairs**. There are two basic types of joints (see Figure 3.1):

Prismatic pairs (P) are links joined by a sliding prism with a definite direction. The motions possible are extension and retraction.

Revolute joints (R) are links connected by a joint with a definite rotation axis.

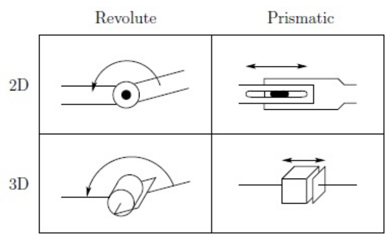


Figure 3.1. Symbolic representation of revolute and prismatic joints [286, p. 13].

The following joint types are commonly used as well. However, these types can be replaced by a concatenation of prismatic or revolute pairs.

Spherical joints (or **ball-and-socket joints**) are joints where three rotational axes intersect. They are equivalent to three revolute joints (RRR) whose axes have a common center.

Universal joints are equivalent to two revolute joints (RR) whose axes intersect.

Cylindrical joints are equivalent to one revolute and one prismatic pair with a common axis.

Kinematic types can also be distinguished by the type of contact between the two links. Two rigid bodies form an **upper kinematic pair**, when contact takes place along a line or at a point, e.g., in cam-and-follower mechanisms or gear trains [6]. In a **lower kinematic pair** contact takes place along a common surface between the two bodies. According to Angeles [6], six different forms of lower kinematic pairs can be distinguished (see Fig. 3.2).

If a kinematic pair can be moved by a motor (**actuator**), it is called **actuated joint**. An actuated kinematic pair is called a **linear** (P) or **rotary** (R) **actuator**. Pairs which are not actuated adjust themselves to forces imposed by loads and actuators, i.e., these pairs perform only passive motion and therefore are called **passive**. The motion of actuated joints is restricted by certain limits according to the actual construction of the robot.

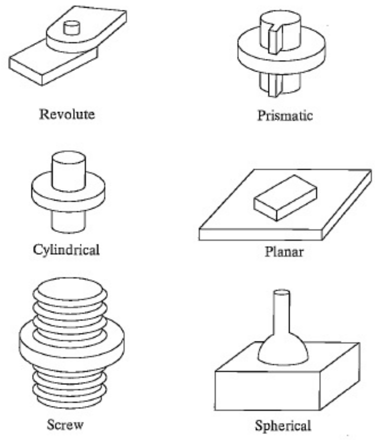


Figure 3.2. The six lower kinematic pairs [57, p. 63].

A rigid body in space can move in various ways by rotation or translation. These are called **degrees of freedom (d.o.f.)**. The total number of d.o.f. of a rigid body in 3-dimensional space is at most six (see Fig. 3.3). Each revolute and prismatic joint, respectively, in a kinematic chain is a 1-d.o.f. joint. In the former case the degree of freedom is given by the angle of rotation, in the latter case by the amount of linear displacement. Hence, the total number of degrees of freedom of a robot is the number of independent position variables needed to specify the spatial pose (i.e., position and orientation) of the end-effector (see also Chapter 4). For many applications, less than 6 d.o.f. are needed. Thus, the number of d.o.f. should match the number required by the desired task, at least for serial robots [57]. For parallel robots, Merlet [198, p. 19f] takes a critical point of view of the usual claim, that machines having only the necessary d.o.f. will be less costly than the usual 6 d.o.f. parallel manipulator. He states that various factors may increase the operating cost of a less than 6 d.o.f. robot, e.g., higher maintenance and fabrication cost because of different actuators/sensors involved in the chains, or exhibition of **parasitic motion** (i.e., motion that is not wanted) which affects the robots performance.

To view joints from an algebraic point of view, we need the special Euclidean group

$$SE(3) = \{(c, Q) \mid c \in \mathbb{R}^3,\ Q \in SO(3)\} = \mathbb{R}^3 \times SO(3),$$

with

$$SO(3) = \{Q \in \mathbb{R}^{3\times 3} \mid Q^T Q = QQ^T = 1,\ \det Q = 1\}$$

denoting the special orthogonal group, i.e., the group of rotations in $\mathbb{R}^3$. Then, joints are surfaces of contact between links that constrain the motion of the robot mechanism to a subset of $SE(3)^{n-1}$: without the joints, each link could move with 6 d.o.f. anywhere in $SE(3)$. Typically, the locations of the links are measured relatively to the base; thus, a concatenation of n links 'lives' in $SE(3)^{n-1}$. A joint which constrains a mechanism to an algebraic subset of $SE(3)^{n-1}$ is called **algebraic joint**. Five of the six lower-order pairs in Figure 3.2 are algebraic. The only exception is the screw (or helical) joint, whose transformation matrix consists of an angle α as well as $\sin\alpha$ and $\cos\alpha$, which make the motion truly trigonometric, hence non-algebraic [313].

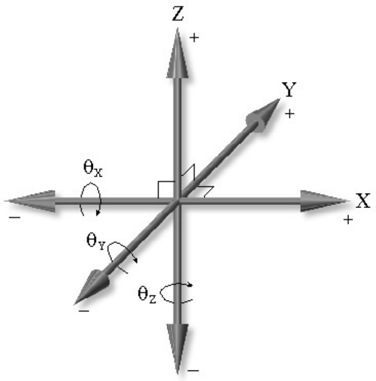


Figure 3.3. 6 degrees of freedom: 3 components of translation, 3 components of rotation [`www.dovermotion.com`].

An **open kinematic chain** is a chain of links, where each but the first and the last link is coupled to exactly two other links. For a quantitative treatment, it is helpful to characterize open- and closed-loop kinematic chains, respectively, by the **degree of connectivity** (or **connection degree**) of each link [88]. It is defined as the number of rigid bodies directly attached to a certain link by kinematic pairs. Therefore, in a **simple kinematic chain** each member possess a connection degree of at most 2. An open kinematic chain is defined as a simple kinematic chain, where exactly two links (the base and the end-effector) have connectivity 1. A closed-loop kinematic chain is obtained when the connection degree of one of the links (not the base) is at least 3 [198].

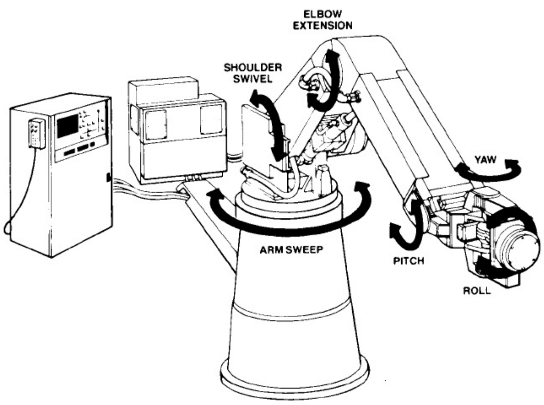


Figure 3.4. A serial manipulator with 6 d.o.f. (T3) [152].

A robot consisting of a single open kinematic chain is called a **serial robot** or **serial manipulator**. It is best visualized as a movable mechanical arm. A robot whose only end-effector (the **platform**) with n d.o.f. is connected to a fixed base by at least two independent open kinematic chains (the **legs**) is called **parallel**

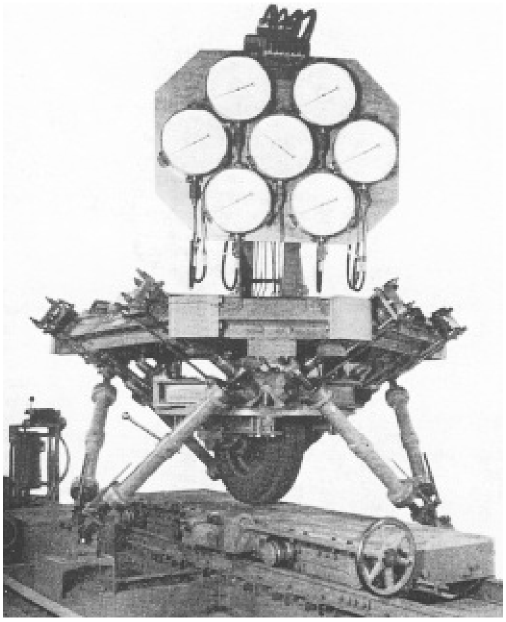
robot or **parallel manipulator** [198]. It is actuated by n simple actuators. A manipulator containing at least one closed-loop kinematic chain is called **hybrid**; see, e.g., [50, 292]. An important class of parallel architectures is the one of **fully parallel robots**, where the number of kinematic chains is strictly equal to the number of d.o.f. of the platform [88]. For example, a **6 d.o.f. fully parallel robot** consists of arbitrarily shaped base and platform connected by six legs. Each leg contains six kinematic pairs, of which one is linearly or rotary actuated, and the other five are passive joints. There exist various parallelism indices to characterize (fully) parallel robots; see, e.g., Gosselin [88], Rao [255].

Both in serial and parallel mechanisms, the type of the arm and each leg, respectively, usually is denoted by the sequence of its joint-types, with the actuated one underlined. For example, an **ideal leg** (which is described in more detail in Section 3.4) has type RR$\underline{\text{P}}$S, where S denotes a spherical joint (of type RRR).

A substantial example for a parallel manipulator is the so-called **Gough platform** (or **Gough-Stewart platform**). It consists of two rigid bodies, a fixed base and a moving platform, joined by six legs of adjustable length (see Fig. 3.5, 3.11). The basic principles of this platform were first established by Gough [93] in 1947 for a machine for testing tire wear and tear; the first prototype was built in 1955 [198]. Sometimes, the Gough platform is also referred to as **Stewart platform**. But the "true" Stewart platform is a mechanism consisting of a triangular mobile plate connected to the base by three identical legs and was designed to be used as a flight simulator (see [156] for further details). According to the six actuated legs, there are 6 d.o.f. in a Gough platform for the **pose** (i.e., position and orientation) of the end-effector. For some applications architectures with additional redundant actuators are considered, that give the platform additional flexibility. An extensive description of various parallel architectures of robot mechanisms is given by Merlet [198, Ch. 2]. (Stewart-)Gough-type parallel manipulators are often referred to as **m–n-platforms**, e.g., general 6–6 Stewart platform. Here, m refers to the number of attaching centers of S-joints in the base, and n to the number of attaching centers of S-joints in the mobile platform [115]; e.g., an exhaustive enumeration of Stewart-type manipulators can be found in [128].

3.3 Description of open-loop kinematic chains

Robotic manipulation implies a need for representing position and orientation of parts of the robot, the end-effector and the robot itself. Denavit & Hartenberg [59] introduced a way to uniquely describe the architecture of an open-loop kinematic chain. The DH-convention uses a minimal parameter representation of the kinematic chain. Another approach is the screw-based method, which is less known, but has become more and more popular in recent research. It is a more complex method, but has some advantages over the DH-method, especially considering differential kinematics (see Rocha et al. [257]). However, in the following the screw-based method will only be discussed very briefly to introduce



(a) 1954

(b) 2000

Figure 3.5. The original Gough platform (or octahedral hexapod) [24].

some fundamental terms; for a more detailed description see, e.g., Davidson et al. [58], Hunt [118].

3.3.1 Denavit-Hartenberg parameters

The architecture of an open-loop kinematic chain may be described uniquely by the relative location and orientation of the neighboring pair axes. Therefore, several coordinate systems, so-called **frames**, are defined. First, a **reference frame** is specified, i.e., a universal coordinate system to which everything is referenced to; secondly, a frame $\mathcal{F}$ is attached to each link of the open-loop kinematic chain. The links are numbered from 0 to n, the joints from 1 to n. Thereby the ith pair is coupling the $(i-1)$st and the ith link. Thus, a manipulator is composed of $n+1$ links and n pairs, each one either revolute or prismatic, with link 0 denoting the fixed base and link n the end-effector of the manipulator. In the convention used by Angeles [6], frame $\mathcal{F}_i$ with origin O_i and axes X_i, Y_i, Z_i is attached to the $(i-1)$st link $(i = 1, \ldots, n+1)$ (see Fig. 3.6). For further details about how to choose the axes see [6, Ch. 4].

A point with coordinates $p_i \in \mathbb{R}^3$ in the coordinate system of link $i > 0$ has coordinates

$$p_{i-1} = Q_i p_i + c_i, \quad i = 1, \ldots n, \tag{3.1}$$

where $c_i \in \mathbb{R}^3$ is a translation vector of special structure and $Q_i \in SO(3)$. The orthogonal matrix Q_i represents the matrix rotating frame $\mathcal{F}_{i+1}$ in an orientation coincident with that of $\mathcal{F}_i$; the vector c_i describes the origin of $\mathcal{F}_{i+1}$ in coordinates of $\mathcal{F}_i$. Transformations as in (3.1) are called **rigid motions** [286] or **Euclidean**

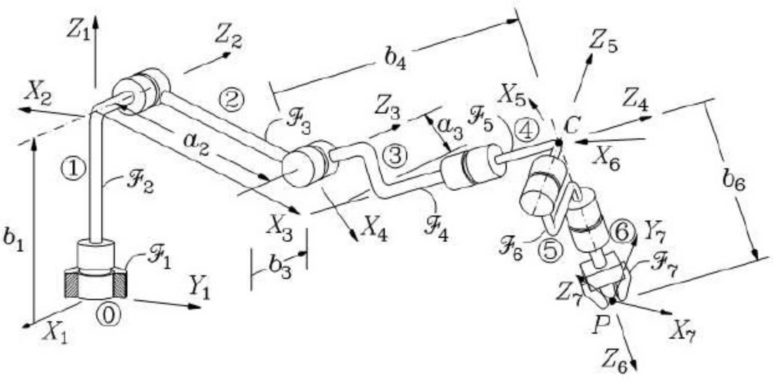


Figure 3.6. Coordinate frames of a Puma robot [6, p. 133].

motions [153] (see also, e.g., [180]). They completely determine the relative position of a link with respect to the former link. If homogeneous coordinates are introduced, then (3.1) may be written as

$$\begin{pmatrix} p_{i-1} \\ 1 \end{pmatrix} = T_i \begin{pmatrix} p_i \\ 1 \end{pmatrix}, \quad T_i = \begin{pmatrix} Q_i & c_i \\ 0 & 1 \end{pmatrix},$$

where the vector $(p_{i-1}, 1)^T$ is called **homogeneous representation** of p_i. The matrix T_i is the **homogeneous transformation matrix** with $Q_i \in SO(3)$. Hence, the pose of the end-effector with respect to the base can be written as

$$\begin{pmatrix} p_0 \\ 1 \end{pmatrix} = T \begin{pmatrix} p_n \\ 1 \end{pmatrix}, \quad p_0 = Qp_n + c,$$

where the product

$$T = T_1 \cdots T_n = \begin{pmatrix} Q & c \\ 0 & 1 \end{pmatrix} \tag{3.2}$$

and its inverse

$$T^{-1} = \begin{pmatrix} Q^T & -Q^T c \\ 0 & 1 \end{pmatrix}$$

determine the coordinate transformation. Expanding (3.2) we get more explicitly

$$Q = Q_1 \cdots Q_n, \quad c = c_1 + Q_1 c_2 + \cdots + Q_1 \cdots Q_{n-1} c_n,$$

in particular, Q is again a rotation.

In the following the matrices Q_i and T_i, respectively, are specified using Denavit-Hartenberg parameters. The homogeneous transformation matrices T_i can be written as a product of four basic transformations, two rotations about the axis Z_i and two translations along the axis X_i. In particular,

$$
\begin{aligned}
T_i &= \mathrm{Rot}_{Z_i\theta_i}\,\mathrm{Trans}_{Z_id_i}\,\mathrm{Trans}_{X_ia_i}\,\mathrm{Rot}_{X_i\alpha_i} \\
&= \begin{pmatrix} c_i & -s_i & 0 & 0 \\ s_i & c_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (3.3) \\
&= \begin{pmatrix} c_i & -s_ic_{\alpha_i} & s_is_{\alpha_i} & a_ic_i \\ s_i & c_ic_{\alpha_i} & -c_is_{\alpha_i} & a_is_i \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix},
\end{aligned}
$$

where $c_i = \cos\theta_i$, $s_i = \sin\theta_i$ and $c_{\alpha_i} = \cos\alpha_i$, $s_{\alpha_i} = \sin\alpha_i$. Thus, denoting the upper left 3×3-matrix of the rotational matrices in (3.3) by Q_{Z_i} and Q_{X_i}, respectively,

$$Q_i = Q_{Z_i}(\theta_i)Q_{X_i}(\alpha_i) \qquad (3.4)$$

$$c_i = Q_{Z_i}(\theta_i)\begin{pmatrix} a_i \\ 0 \\ d_i \end{pmatrix}. \qquad (3.5)$$

The axis Z_i is chosen to be the axis of actuation for joint i, i.e., the axis of revolution and the axis of translation, respectively, for the corresponding kinematic pairs. The X_i-axis is the common perpendicular to Z_{i-1} and Z_i, directed from the former to the latter; axis Y_i is chosen to complete the coordinate system to be a right hand frame.

The used parameters α_i, θ_i, d_i, and a_i have a natural geometric meaning and are called the **Denavit-Hartenberg parameters** (short **DH parameters**) of a kinematic pair [6] (see Fig. 3.7).

$\mathbf{a_i}$ is defined as the distance between the axes Z_i and Z_{i+1}. Therefore, it is non-negative and called **link length**.

$\mathbf{d_i}$ is defined as the Z_i-coordinate of the intersection of Z_i with X_{i+1}. Its absolute value is the distance between the axes X_i and X_{i+1} and is called the **link offset**.

$\boldsymbol{\alpha_i}$ is defined as the angle between Z_i and Z_{i+1} measured about the positive direction of X_{i+1}. It is called the **twist angle** between successive pair axes.

$\boldsymbol{\theta_i}$ is defined as the angle between X_i and X_{i+1} measured about the positive direction of Z_i. This item is known as the **joint angle**.

If the ith pair is revolute, the joint angle θ_i is variable, whereas a_i, d_i and α_i are constants. The former is termed **joint variable** (or **joint coordinate**, **articular coordinate**) of the ith pair, in the following denoted by Θ_i; the constants are

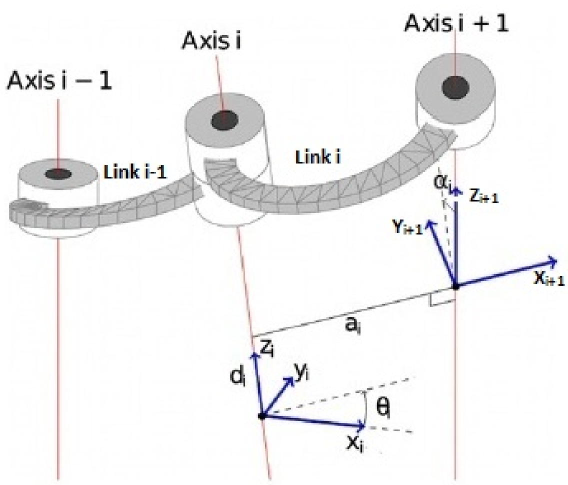


Figure 3.7. Denavit-Hartenberg parameters (mod. from [257]).

known as **joint parameters**, **design parameters** or **structural parameters**. For a prismatic pair the link offset d_i is the joint variable. The joint coordinate describes the degree of freedom in positioning the ith pair. An n-axis manipulator has n joint variables and $3n$ constant parameters. The former determine the configuration or the posture of the manipulator while the latter describe its architecture. The **joint space** of a robot consists of all possible joint configurations Θ, where Θ denotes a vector containing all joint variables of the mechanism. The **parameter space** of a manipulator is the set of joint parameter-vectors. For example, for a serial 3R planar robot as shown in Figure 3.8, the joint space consists of all possible joint angles $\Theta = (\theta_1, \theta_2, \theta_3)$, which is in absence of joint limits the three-torus T^3. The parameter space for the family of 3R planar manipulators consists of all link-length triplets (a_1, a_2, a_3), and thus is $\mathbb{R}^3$ [313].

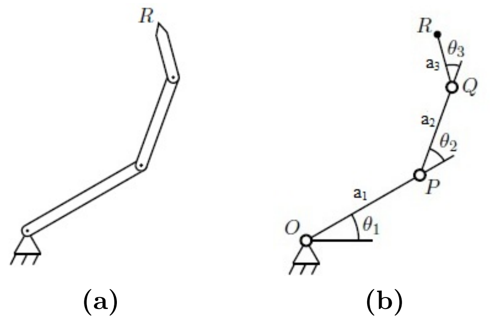


Figure 3.8. A 3R planar robot arm (a) and its kinematic skeleton (b) [313, p. 7]

For a prismatic pair in the variable linear position Θ_i one has

$$Q_i = Q_{Z_i}(\alpha_i), \quad c_i = \begin{pmatrix} a_i \\ 0 \\ \Theta_i \end{pmatrix},$$

following from (3.4) and (3.5) with $\theta_i = 0$. For a revolute joint in the variable angle position Θ_i, Q_i and c_i are derived directly from (3.3) by setting $\theta_i = \Theta_i$. Note that for prismatic pairs the matrix Q_i is constant, while for revolute joints

$$c_i' = Q_i^T c_i = \begin{pmatrix} a_i \\ d_i \sin \alpha_i \\ d_i \cos \alpha_i \end{pmatrix} \tag{3.6}$$

is a constant vector. The detailed conventions of DH parameters might differ slightly from the above by different authors, depending on the choice of the reference axes and the way the angles and signs are defined. For further details see, e.g., [6, 57, 286].

3.3.2 Screw displacements

Another approach for the kinematic modeling of robot manipulators is to use successive screw displacements. In contrast to the Denavit-Hartenberg convention, the screw-based method uses a non-minimal parameter representation of the kinematic chain. The method will not be discussed in detail, but a few notions are introduced at this point, including the Plücker coordinates of a line. A comparative analysis of the kinematic modeling using Denavit-Hartenberg parameters and the screw-based method is given by Rocha et al. [257].

A **screw** is a geometric entity representing both rotational and translational quantities. As stated in Chasle's Theorem [6], a general spatial displacement of a rigid body can be attained by a combination of a translation along an axis and a subsequent rotation about the same axis. The axis of action is called **screw axis $\mathcal{L}$**. The axis combined with a scalar **pitch**, which relates translation and rotation is called a **screw**.

A convenient way to define any line in three-dimensional space is by using the **Plücker coordinates of a line** [6, Ch. 3].

Definition 3.3.1 (Plücker coordinates)**.** *Let x_1 and x_2 denote the position vectors of two points X_1 and X_2 lying on a line $\mathcal{L}$. Then the **Plücker coordinates** of $\mathcal{L}$ are the entries of the 6-dimensional array $\gamma_{\mathcal{L}}$, given by*

$$\gamma_{\mathcal{L}} := \begin{pmatrix} x_2 - x_1 \\ x_1 \times (x_2 - x_1) \end{pmatrix}. \tag{3.7}$$

By setting

$$\mathbf{e} := \frac{x_2 - x_1}{\|x_2 - x_1\|}, \quad \mathbf{n} := x_1 \times \mathbf{e},$$

equation (3.7) can be written in ***normalized Plücker coordinates*** *as*

$$\gamma_{\mathcal{L}} = \|x_2 - x_1\| \begin{pmatrix} \mathbf{e} \\ \mathbf{n} \end{pmatrix}, \tag{3.8}$$

where **e** *determines the direction of* $\mathcal{L}$ *and* **n** *its location.*

The vector **n** can be interpreted as the moment of a unit force parallel to **e** and of line of action $\mathcal{L}$. The six components of the Plücker array, the **Plücker coordinates** are not independent, since

$$\mathbf{e} \cdot \mathbf{e} = 1, \quad \mathbf{e} \cdot \mathbf{n} = 0$$

hence, any line $\mathcal{L}$ has only four independent Plücker coordinates. Note further, that the Plücker coordinates are homogeneous coordinates (see, e.g., [275, Ch. 6] for details).

A rigid-body motion may now be described by 8 scalars, of which six parameters are independent. These are the four independent components of the Plücker coordinates defining the screw axis, the amount of linear translation along the axis and the rotation angle about the axis (**amplitude**). A motion described by these six parameters is called **screw displacement**. Thus, a general screw displacement for a rigid body may be given by Rodrigues's formula

$$x_2 = R(\phi)x_1 + d(d_0),$$

where $x_i = (x_{ix}, x_{iy}, x_{iz})^T$, $Q(\phi)$ is the rotation matrix corresponding to the rotation ϕ about the screw axis and $d(d_0)$ is a displacement vector corresponding to the translational distance d_0 along the screw axis [96] (see Fig. 3.9). Thus, a screw given by the screw parameters ϕ and d_0, can be used to describe the pose of a rigid body relative to a referential system [257].

An instantaneous rigid-body motion relative to a referential system is called an **instantaneous screw** (or **twist**). It is defined by the **instant screw axis** $\mathcal{L}'$, a pitch p' and an amplitude $\|\omega\|$, which represents the magnitude of the angular velocity. The instantaneous screw may alternatively be defined by six independent scalars, the three components of its angular velocity ω and the three components of the velocity v of an arbitrary point X_1. If the pitch equals zero, the twist is a pure rotation, whereas the twist is a pure translation for a pitch equal to infinity. (for further details see, e.g., [6]). So, screw theory associates physical meaning to a geometric entity by expressing (linear and angular) velocities as twists and forces/torques as **wrenches** [257]. Further discussion of instantaneous motion is presented in Chapter 6.

3.4 Separable (parallel) robots

An important class of serial manipulators are so-called **decoupled** manipulators (see Fig. 3.10). In a decoupled manipulator the last three joints have intersecting

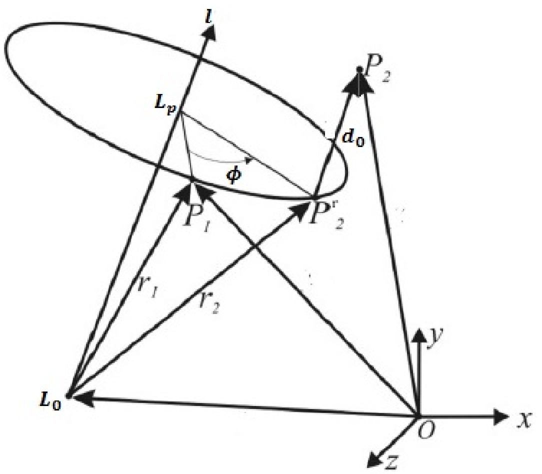


Figure 3.9. Screw displacement (mod. from [96]): A Point P is displaced from position P_1 to P_2 by a rotation ϕ about the screw axis followed by a translation of d_0 along the same axis. l denotes a unit vector along the direction of the screw axis $\mathcal{L}$.

axes and therefore constitute the wrist of the manipulator. If the point C of intersection is kept fixed, all points of the wrist move on spheres around C. Thus the wrist is called *spherical* [6]. Considering a general 6R manipulator, decoupled in terms of DH parameters means, that $a_4 = a_5 = d_5 = 0$, so the origins of frames $\mathcal{F}_5$ and $\mathcal{F}_6$ are coincident. All other DH parameters can assume arbitrary values [6].

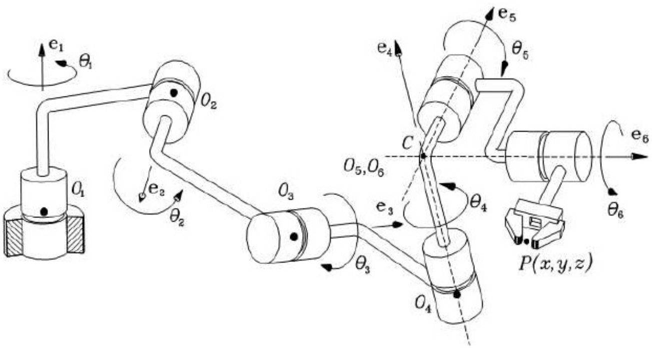


Figure 3.10. A general 6R manipulator with decoupled architecture [6, p. 142].

A more general notion is that of so-called **separable** legs [219]. Consider a leg with six kinematic pairs. Such a leg is called **separable** if the translation vector $c_5 = 0$ (see (3.5)). In terms of DH parameters, separable means that $a_5 = d_5 = 0$. In particular, a decoupled (serial) manipulator also is separable. Important examples for separable legs are the ones coupled to the platform by a spherical joint [219]. For separable legs we define the vector

$$\Delta c := c - Qc_6' - c_1 = Q_1 c_2 + Q_1 Q_2 c_3 + Q_1 Q_2 Q_3 c_4, \tag{3.9}$$

i	a_i	d_i	α_i	type
1	a_1	d_1	α_1	R
2	0	0	α_2	R
3	0	Θ_3	0	P
4	0	b_4	α_4	R
5	0	0	α_5	R
6	a_6	b_6	α_6	R

Table 3.1. DH parameters of an ideal leg [219]

which describes the position of the center of the forth joint relative to the center of the first joint. Given the pose (Q, c) of the end-effector, Δc is known and (3.9) defines three equations for the joint variables $\Theta_1, \Theta_2, \Theta_3$. Considering Θ_i as an actuated joint coordinate, the other two can be algebraically eliminated from (3.9) and an implicit equation

$$H_i(\Theta_i, \Delta c) = 0$$

can be derived. Note that the analysis of robots with separable legs can be restricted to the actuated joint coordinates if the properties of the passive joints are not significant or can be studied separately [219]. Consider for example an **ideal leg** of type RR$\underline{\text{P}}$S with DH parameters given in Table 3.1, and where the actuated pair is prismatic and the others are revolute. There one has

$$c_2 = c_5 = 0 \quad \text{and } Q_3 = \mathbb{1}.$$

Thus, the ideal leg is separable and (3.9) can be applied; taking norms gives

$$\|\Delta c\| = \|Q_1 Q_2 (c_3 + c_4)\| = \|c_3 + c_4\| = |\Theta_3 + d_4|,$$

so

$$\Theta_3 = \pm\|\Delta c\| - d_4, \tag{3.10}$$

where usually only the positive sign is physical. Takano (see [6, 219]) showed that for a general separable leg, (3.10) is a polynomial of degree ≤ 4 in Θ_i for a linear actuator, and in $\tan(\frac{\Theta_i}{2})$ for a rotary actuator.

Motivated by the above analysis a **separable parallel robot** can be defined as a parallel robot with six legs, where the pose (c, Q) of the end-effector is related to the actuated joint coordinates Θ_i by six independent scalar equations

$$H_i(d_i^c, \Theta_i, c + Q d_i^p - d_i^b) = 0 \quad (i = 1, \ldots, 6). \tag{3.11}$$

The local coordinates of the endpoints of leg i on the base and the platform are denoted by d_i^b and d_i^p, respectively, while d_i^c is a vector of coefficients determining the details of H_i [219]. Similar to (3.9) the expression $c + Q d_i^p - d_i^b$ codes the position of a reference point on the platform (e.g., the center of the platform-frame) relative to the center of the base-frame; see also Section 5.3.1. In general, the function H_i must be an algebraic function of the variables. Kempe [142] and

FREUDENSTEIN & ROTH [82] showed, that every algebraic curve can be realized by a closed kinematic chain with a single degree of freedom. Therefore, it is conceivable that, in principle, any set of algebraic functions H_i can be realized in this way by a sufficiently complex leg.

Depending on the considered class of robots, (3.11) can be formulated more specifically [219]. A Gough platform (see Fig. 3.11) with six ideal legs of type RR$\underline{\text{P}}$S can be characterized by the relation

$$\|c + Qd_i^p - d_i^b\|^2 - (l_i + \Theta_i)^2 = 0, \tag{3.12}$$

where l_i denotes the parts of the ith leg of fixed length (see also Ex. 5.3.1). Note, that in a general Gough platform neither the base nor the platform need to be planar (see Fig. 3.5).

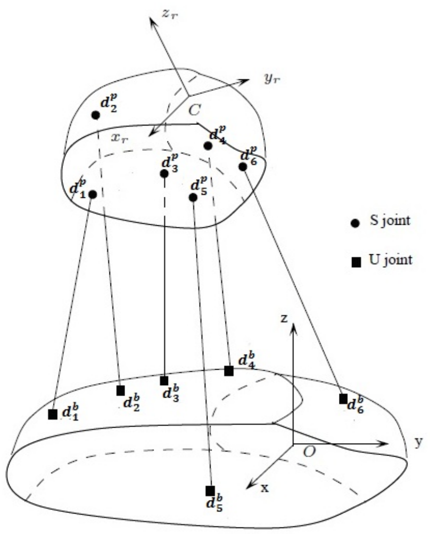


Figure 3.11. Scheme of a Gough platform (mod. from [196]).

Separable parallel robots with legs of type $\underline{\text{P}}$RRS are called **active wrist** or **hexaglide** [111] and satisfy equations of the form

$$\|c + Qd_i^p - d_i^b - \Theta_i u\|^2 - l_i^2 = 0. \tag{3.13}$$

Similarly, so-called **Hexa robots** [243], i.e., separable parallel robots with legs of type $\underline{\text{R}}$RRS, satisfy equations of the form

$$\|c + Qd_i^p - d_i^b - (\cos\Theta_i)u - (\sin\Theta_i)v\|^2 - l_i^2 = 0. \tag{3.14}$$

Summarizing, we can say that a large, practically relevant class of parallel robots can be completely described by (3.11) (and the special cases (3.12), (3.13), and (3.14), respectively), together with the six equations

$$(Q^TQ)_{ik} = \delta_{ik} = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{otherwise,} \end{cases} \tag{3.15}$$

and the orientation constraint

$$\det Q = 1. \tag{3.16}$$

Chapter 4

Geometry

4.1 Introduction

The geometry of a robot mechanism treats questions about numerical dimensions of the robot parts, limits on lengths and angles, interferences between legs (in parallel manipulators) or the accessible workspace. For coding the pose (i.e., position and orientation) of the end-effector of a robot, **generalized coordinates s** are used. The **equation of state**

$$F(s, \Theta) = 0 \tag{4.1}$$

relates the pose to the joint configuration (or posture) Θ, which describes the combined positions and orientations of all links and the end-effector (or mobile platform) [219]. The pairs (s, Θ) satisfying (4.1) are called **states**. A **measurement vector** y contains the values obtained by internal sensor measurements and is related to the joint configuration Θ by the **measurement equation**

$$y = P\Theta, \tag{4.2}$$

where P is a matrix with exactly one 1 per row and zeros elsewhere, which represents the projection to the vector of measured joint variables. The equation of state contains all information about the geometry of the measured joints and the end-effector. For finding limits of ranges on joint variables, global size constraints for a robot, or inequality constraints for ensuring the absence of interferences between different legs, however, more detailed information is necessary about how the robot is built; the equation of state does not cover these items. In the generic case, i.e., if the implicit function theorem is applicable, it is ensured by the dimensional constraint

$$\dim y \leq \dim s + \dim \Theta - \dim F, \tag{4.3}$$

that the measurements contain enough information to determine a local change of pose [219].

If equality holds in (4.3), the robot is called **minimally instrumented robot** [219]. Typically, this is the case if the actuators are instrumented and the passive joint variables are eliminated explicitly from the mathematical model, i.e., when Θ is reduced to the n actuated joint variables. Thus, in the case of independent generalized coordinates, $y = \Theta$ and $\dim F = \dim s$. For a minimally instrumented serial robot, the equation of state may be written explicitly as

$$s = F(\Theta), \tag{4.4}$$

and is obtained by writing relation (3.2) in terms of the generalized coordinates. For a minimally instrumented separable parallel robot, the equation of state is given by equation (3.11). In particular, for a Gough platform, (4.1) has the explicit form

$$\Theta = F(s). \tag{4.5}$$

It is obtained by writing (3.12) in terms of the generalized coordinates.

4.2 Sets and Maps

For the main types of geometric problems that typically arise in kinematics it is convenient to define several sets and the corresponding maps, which are shown in the following commutative diagram. The descriptions are given according to Wampler & Sommese [313].

$$\begin{tikzcd}
Y & M \arrow[l, "J"'] \arrow[r, "K"] \arrow[dl, "\hat{J}"'] \arrow[d, "\pi_M"'] \arrow[dr, "\hat{K}"] & O \\
Y \times P \arrow[u, "\pi_1"] \arrow[r, "\pi_2"'] & P & O \times P \arrow[l, "\pi_3"] \arrow[u, "\pi_4"']
\end{tikzcd}$$

The sets are described as follows:

Y denotes the **input** or **joint space** of the mechanism. Its coordinates are the joint variables Θ (or $y = P\Theta$) (which are typically controlled by actuators).

O denotes the **output** or **operational space** of the mechanism. Its coordinates are the poses s of the end-effector.

P denotes the **parameter space** of a family of mechanisms (e.g., the set of all 6R serial robots). It is the set of parameters necessary to describe the geometry of the joints in each link. Thus, each point in P corresponds to a specific mechanism with specified link lengths, etc.

M denotes the **mechanism space**, $M \subseteq Z \times P$, where $Z = SE(3)^{n-1}$ (or, for a planar mechanism family, $Z = SE(2)^{n-1}$) is the **link space** for an n link mechanism. M describes all possible configurations of a mechanism for all possible parameters. So, each point $(z, p) \in M$ represents a specific mechanism $p \in P$ in one of its assembly configurations $z \in Z$. The mechanism space M of a mechanism family is the subset of $Z \times P$ that satisfies the joint constraints. Thus, M characterizes a family of mechanisms, e.g., the family of spatial 6R serial robots.

Further, the following three maps are defined on M:

The **input map** $J\colon M \to Y$ extracts the input values from M.

The **output map** $K\colon M \to O$ extracts the output values from M.

The **projection** $\pi_M\colon M \to P$ extracts the parameters from M. It is the natural projection operator on $Z \times P$ restricted to M, given by $\pi_M\colon (z,p) \mapsto p$.

The above diagram is completed by defining $\hat{J} = (J, \pi_M)$ and $\hat{K} = (K, \pi_M)$ and the associated natural projections π_i for $i = 1, \ldots, 4$. Further, it is useful to define the inverses of the input/output maps,

$$J^{-1}(y) = \{(z,p) \in M \mid J(z,p) = y\}$$
$$K^{-1}(s) = \{(z,p) \in M \mid K(z,p) = s\},$$

which apply to a whole mechanism family, as $p \in P$ is not fixed. If one wants to address only one specific mechanism, the inverses of $\hat{J}$ and $\hat{K}$ have to be considered. In analysis problems, typically a specific mechanism $p^* \in P$ is given and the basic problems may be formulated using the above definitions. For example, the **forward kinematics problem** (see Chapter 5) is to find output (i.e., pose(s)) $s \in O$ of a manipulator which corresponds to a given input-vector of joint variables $y^* \in Y$ for a given mechanism $p^* \in P$,

$$\mathrm{FKP}(y^*, p^*) = K(\hat{J}^{-1}(y^*, p^*)). \tag{4.6}$$

The **inverse kinematics problem** goes from output to input and is to find the vector(s) of joint variables $y \in Y$ that correspond to a given pose $s^* \in O$ for a given mechanism $p^* \in P$,

$$\mathrm{IKP}(s^*, p^*) = J(\hat{K}^{-1}(s^*, p^*)). \tag{4.7}$$

For a more detailed discussion of geometric problems arising in kinematics see [313, Ch. 4].

4.3 Generalized coordinates

The **generalized coordinates** code the pose of the end-effector relative to the base. E.g., for a (serial) robot manipulator consisting of N rigid bodies, $6N$ coordinates are required to specify the position and orientation of all the bodies relative to a coordinate frame. But, because some of the links are joined together, there is a number of constraint equations which establish relationships between some coordinates. Therefore, the $6N$ coordinates can be expressed as functions of a smaller set s of coordinates, that are all independent. The coordinates of s are known as **generalized coordinates**; the joint variables Θ form a set of generalized coordinates [280].

There are multiple ways of choosing the set of generalized coordinates; in principle, their choice is a matter of convenience. The displacement vector is usually given in Cartesian form, although other parameterizations, e.g., the dual quaternion parameterization of Study [288] (see, e.g., [123]) are discussed in the literature (e.g., [122, 124, 235, 250, 310]). For a definite description, the vector of generalized coordinates s is defined as

$$s = \begin{pmatrix} r \\ c \end{pmatrix},$$

where c is the position of a specific point of the end-effector relative to the base, and r encodes the rotation $Q = Q[r]$ in such a way, that (3.15) and (3.16) hold for all admissible r. The rotation may be written in terms of a convenient parameterization. A common used one for three-dimensional rotations is the so-called **Euler angles** parameterization. According to Euler's rotation theorem, every rotation may be described using three angles, frequently denoted by φ, θ, ψ. The rotation Q may be achieved by composing three elemental rotations about certain axes of the coordinate frame. There are many significantly different ways of defining the Euler angles, depending on the posed problem. One highly used representation is pitch-roll-yaw (see, e.g., [132, 133, 289]); for further information on Euler angles and other Euler angles set conventions see, e.g., [57, 216, 286]. Other ways for parameterizing the rotation $Q[r]$ are the quaternion parameterization, or more redundant parameterizations involving components of Q and additional orthogonality constraints relating these. Taking for r simply the components of Q (restricted by (3.15) and (3.16)) produces the simplest but most redundant parameterization [219].

Using a trigonometric parameterization in terms of angles (e.g., Euler angles), its interpretation depends on a fixed coordinate system. Further, additional singularities for certain angle combinations are introduced. Therefore, the **quaternion parameterization** is a numerically more convenient and singularity-free parameterization of rotations. Most of the following statements about quaternion representation, further explanations, and proofs can be found in Neumaier [220].

Theorem 4.3.1 (Quaternion parameterization)**.** *Let $r \in \mathbb{R}^3$ with $r^T r \le 1$ and $r_0 = \sqrt{1 - r^T r}$, and let $X \in \mathbb{R}^{3\times 3}$ be a skew-symmetric matrix of the form*

$$X(r) := \begin{pmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{pmatrix}.$$

Then the matrix

$$Q[r] := \mathbb{1} + 2r_0 X(r) + 2X(r)^2, \tag{4.8}$$

is a rotation. Conversely, every rotation Q has the form $Q = Q[r]$ for some $r \in \mathbb{R}^3$ with $r^T r \le 1$.

Proof. See Neumaier [220]. □

Using the fact that

$$X(r)^2 = rr^T - r^T r \mathbb{1},$$

(4.8) can be expanded to

$$Q[r] = \mathbb{1} + 2 \cdot R, \quad R := \begin{pmatrix} -r_2^2 - r_3^2 & r_1r_2 - r_0r_3 & r_1r_3 + r_0r_2 \\ r_1r_2 + r_0r_3 & -r_1^2 - r_3^2 & r_2r_3 - r_0r_1 \\ r_1r_3 - r_0r_2 & r_2r_3 + r_0r_1 & -r_1^2 - r_2^2 \end{pmatrix}. \tag{4.9}$$

Relation (4.9) can be written in homogeneous form by

$$Q[r_0, r] = \mathbb{1} + \frac{2}{r_0^2 + r^2} \cdot R, \tag{4.10}$$

with $r_0 \in \mathbb{R}$, $r \in \mathbb{R}^3$ not both zero. Further, (4.10) reflects the fact that $SO(3)$ has the structure of a projective 3-space (see, e.g., [178]). Since $Q[r_0, r]$ satisfies

$$Q[r_0, r] = Q[\lambda r_0, \lambda r], \quad \lambda \neq 0,$$

i.e., scaling r_0 and r by the same nonzero factor does not change the rotation matrix, (r_0, r) may be normalized to

$$r_0 \geq 0, \quad r_0^2 + r^2 = e \quad (e > 0 \text{ fixed}).$$

Thus, the quaternion parameterization of rotations given in (4.8) may be written as $Q = Q_e[r]$ with

$$Q_e[r] = \mathbb{1} + \frac{2}{e} \cdot R, \tag{4.11}$$

where $r \in \mathbb{R}^3$, $r^2 \leq e$ and $r_0 = \sqrt{e - \sum_{i=1}^3 r_i^2}$. For $e = 2$, the factor $\frac{2}{e}$ can be deleted and $Q_2[r]$ can be computed with only 15 additions, 9 multiplications and a square root. However, the standard scaling in the literature has $e = 1$, which makes the computation of (4.11) more expensive, but saves the factor $e^{-1/2}$ in the multiplication rule (4.13) below. Different normalization conventions may be used as well, such as, e.g., $r_0 = 1$ or $r_0^2 + r^2 = 2r_0$, which work well for rotations by a limited angle [219].

For $r^2 = e$, we get $r_0 = 0$ and $Q_e[r] = Q_e[-r]$; thus, antipodal vectors r of length $\sqrt{e}$ are identified in the quaternion parameterization. This reflects the non-Euclidean structure of $SO(3)$ and has to be taken into account when constructing smooth motions joining two poses involving a small r_0 but nearly opposite r [219] (see, e.g., [10, 254, 279, 342]). Geometrically, $Q[r]$ describes a rotation around the vector r by the angle $\Phi = \arcsin(\frac{|r|}{\sqrt{e}})$.

Independent of the parameterization, the product of rotations again is a rotation. This fact implies the existence of a product formula

$$Q_e[r]Q_e[s] = Q[r \oplus s], \tag{4.12}$$

where $\oplus$ denotes a parameterization dependent, non-commutative but associative binary operation $\oplus$. One elegant way to derive (4.12) is using quaternions. A **quaternion** is a 4×4 matrix of the form

$$U(r_0, r) := \begin{pmatrix} r_0 & r^T \\ -r & r_0\mathbb{1} + X(r) \end{pmatrix},$$

for $r_0 \in \mathbb{R}$, $r \in \mathbb{R}^3$. Quaternions are used for proving the following theorem, which provides the product formula for the quaternion parameterization (4.11).

Theorem 4.3.2 (Product formula for rotations in quaternion parameterization)**.** *Let $r^2 \leq e$,$s^2 \leq e$. Then there exists a product formula (4.12), where*

$$r \oplus s := \pm e^{-1/2}(s_0 r + r_0 s + r \times s), \tag{4.13}$$

for $r_0 = \sqrt{e - r^2}$, $s_0 = \sqrt{e - s^2}$ and the sign chosen such that $\pm(r_0 s_0 - r \cdot s) \geq 0$. Further,

$$Q_e[r]^{-1} = Q_e[r]^T = Q_e[-r], \quad r \oplus (-r) = 0.$$

Proof. See NEUMAIER [220]. □

Applications for the quaternion representation are given in, e.g., [258, 312], where it is used for the forward displacement analysis of spherical parallel manipulators. WAMPLER [310] used Study's soma coordinates, which are, except for a factor of 1/2 identical to dual quaternions, for the forward displacement analysis of a general 6-SPS Stewart platform.

4.4 Workspace

Generally, the workspace of a mechanism is a region of points that can be reached by a reference point on the manipulator extremity [44]. The end-effector cannot reach every pose, due to its mechanical architecture (e.g., link-lengths) or technological constraints, such as limits on the joint motions. For both serial and parallel manipulators, the motions can be restricted by limits on the passive joints or limitation due to the actuators. For parallel robots, there are perhaps bounds on the angles between legs and platforms, which are chosen such that the legs do not interfere with each other within these limits. All these restrictions are taken into account in the modeling process. In the most general case, the constraints are defined by a vector of constraint inequalities,

$$C(s, \Theta) \in \mathbf{C}, \tag{4.14}$$

where $\mathbf{C}$ is a simple set, defined by lower and/or upper bounds on the components only [219]. The set of states (s, Θ) satisfying the equation of state (4.1) and (4.14) is a manifold. If the dimension of this manifold is d, then the robot is said to have d degrees of freedom [219]. The manifold can be connected or disconnected; in the latter case, the workspace is split into separate components, which means that the robot can be assembled in different, equivalent ways. Here, only the component $\mathcal{S}$ containing the nominal assembling position is of interest. The six dimensional manifold $\mathcal{S}$ is called the **state space** of the mechanism [219]. The set of all attainable poses of the end-effector is called **complete workspace**. In case of a 6 d.o.f. manipulator it is the **6D-workspace** $\mathcal{W} \subset SE(3)$, which is described by the projection

$$\mathcal{W} = \{s \mid (s, \Theta) \in \mathcal{S}\}$$

of the state space to the space of possible poses. Recall that s consists of a vector c, representing the coordinates of a specific point on the end-effector, and of a vector r, whose components define the end-effector's orientation. Thus, $\mathcal{W}$ can also be written as

$$\mathcal{W} = \{(c, r) \mid c \in W_c \subseteq \mathbb{R}^3; r \in \mathbb{R}^3 \text{ s.t. } Q[r] \in W_r \subseteq SO(3)\},$$

where W_c and W_r denote the maximal three-dimensional reachable position and orientation workspaces, respectively. Thus, $W_c(r)$ is the **position workspace under constant orientation** $Q[r]$, and $W_r(p)$ the **orientation workspace under constant position** c [134].

The 6D-workspace contains all poses of the end-effector which are realizable by the actuators within given constraints. The visualization of six dimensions is limited, hence workspace problems generally only involve some projection of the 6D-workspace; but note, that planning problems may involve the full 6D-workspace. Problems which consist of finding the extremal values of one of the pose parameters are known as **one-parameter problems**, while **three-parameter problems** assume nominal constraints on three of the joint variables; e.g., they may have a fixed value or must lie within given ranges. For one-parameter problems, a worst case analysis is required, therefore these are global optimization problems. These have never been fully investigated [219].

To formulate constraints on the pose, the positioning and orientational part of s (c and r, respectively), are treated separately. The robot has to work in an environment of given size. Thus, constraints on the position c are generally simple bounds resulting from this need. Constraints on the orientation r are usually expressed by constraints on the values of some angles. However, quaternion parameters can easily be used to encode geometric constraints. Consider the parameterization $Q = Q_e[r]$, given by (4.11). According to Neumaier [219], the constraints

$$r^2 \leq \frac{e}{2}(1 - \cos\alpha),$$

and equivalently,

$$r_0^2 \geq \frac{e}{2}(1 - \cos\alpha)$$

limit the rotation to an angle $|\theta| < \alpha$ around its axis. Further, the constraints

$$r_1^2 + r_2^2 \leq r_3^2 \tan^2\alpha,$$

and equivalently,

$$r_0^2 + \frac{r_3^2}{\cos^2\alpha} \geq 1$$

limit the rotation axis to a circular cone with half opening angle α around the Z-axis. In the same way,

$$r_1^2 \tan^2\alpha \leq r_2^2 + r_3^2,$$

and equivalently,

$$r_0^2 + \frac{r_1^2}{\cos^2\alpha} \geq 1$$

keep the rotation axis away from a circular cone with half opening angle α around the X-axis, without restricting it in the XY-plane. General homogeneous quadratic inequalities do the same for other preferred or avoided orientations.

Due to the six dimensions of the workspace, a graphical illustration of the robot workspace is difficult to provide. In contrast to serial manipulators, for parallel robots having more than 3 d.o.f. there exists no such graphical representation. Consider for example a serial manipulator with 6 d.o.f. with a concurrent axis wrist. It may be represented by the 3D-volume reachable by the center of the wrist, which illustrates the translations. Two of the rotational degrees of freedom may then be represented by the surface that may be reached by the extremity of the end-effector [198]. The graphical representation of the workspace of a parallel robot is possible only for robots with at most 3 d.o.f. Workspace representation for robots with $n > 3$ d.o.f. will only be possible, if $n - 3$ pose parameters are fixed. This assumption can be taken, because in many situations the platform either operates with a fixed orientation or rotates about a fixed point. Macho et al. [174] provide a new software tool which computes and visualizes different workspace-types of parallel manipulators, depending on the user chosen variables, onto which the workspace shall be projected. Depending on the choice of the pose parameters and the constraints imposed on these, different types of workspace will be obtained and impose certain types of three-parameter problems; for example, for a general Stewart platform, a total of twenty different slices of the 6D workspace can be obtained [21].

The typically relevant types of workspace are ([198]):

constant orientation workspace or **translation workspace**: all possible positions of a fixed point C at the robot that can be reached with a given orientation,

orientation workspace: all possible orientations that can be reached given that C is in a fixed location,

total orientation workspace: all the locations of C that may be reached with all the orientations among a set defined by ranges on the orientation angles,

dextrous workspace: important special case of the total orientation workspace, which contains all locations of C for which all orientations are possible,

maximal workspace or **reachable workspace**: all the locations of C that may be reached with at least one orientation of the end-effector.

But other specific workspace definitions may be useful for certain tasks; e.g., Bonev & Gosselin [25] give a fast algorithm for the so-called *constant-torsion workspace* of symmetrical spherical parallel mechanisms and presents a methodology for the analytical determination and representation of the workspace boundaries.

The corresponding three-parameter problems to the above described workspace types are listed below in increasing order of complexity [219].

Constant orientation workspace: Find the set of positions c such that $s \in \mathcal{W}$ for a given orientation r.

Orientation workspace: Find the set of orientations r such that $s \in \mathcal{W}$ for a given position c.

Total orientation workspace: Find the set of positions c such that $s \in \mathcal{W}$ for all $r \in \mathbf{r}$ for a given set of $\mathbf{r}$ of desired orientations.

Dextrous workspace: Find the set of positions c such that $s \in \mathcal{W}$ for all $r \in \mathbf{r}$, where $\mathbf{r}$ consists of all meaningful choices of orientations (in particular, for Euler angles $\mathbf{r}$ is the range $[0, 2\pi]^2 \times [0, \pi]$, and for quaternions it is the unit ball).

Maximal workspace: Find the set of positions c for which there exists at least one orientation r such that $s \in \mathcal{W}$.

4.4.1 Workspace for parallel robots

Constant orientation workspace

Three-parameter problems are constraint satisfaction problems of the covering type, and therefore usually expensive to solve. But if only limits on the actuated joints are considered, the computational effort can be significantly reduced. For example, the *constant orientation workspace* for separable parallel robots is the intersection of six circular rings and the workspace boundaries are given by circular arcs, and may be computed in a few milliseconds by an algorithm provided by Gosselin [89]. In the presence of constraints on joint limits in the passive joints, or possible leg interference, the problem becomes more complex [186]. The interior of the constant orientation workspace of a Gough-type platform was computed with interval methods by Merlet [193].

An alternative approach for accurate and computationally effective calculation of the constant orientation workspace of a constrained parallel manipulator was given by Shah et al. [276], using CAD modeling software. The used geometric programming methods (constraint analysis and Boolean volumetric operations), could potentially be much more insightful, efficient and accurate compared to conventional parameter sweep methods.

Orientation workspace

The computation and visualization of the *orientation workspace* turn out to be more difficult, due to more complex equations and the difficulty of representing orientations intuitively [21]. A possibility is to hold the orientation angles fixed [188], or to let the three angles vary [236]. But all methods rely on some sort of discretization and therefore, produce incomplete or less accurate output in some situations [21]. Jiang & Gosselin [133] gave a fast method providing appealing visualizations of the orientation workspace of a Gough-Stewart platform, but mechanical limits in the passive joints are neglected; thus, the computed workspace is an overestimation of the true workspace. Further, some methods only compute the connected component of the workspace which is achievable from a pre-defined initial assembly configuration [296].

Cao et al. [38] present a discretization method based on half-angle transformation and the inverse kinematics solution for computing the orientation workspace of a Gough-Stewart platform with similar semi-symmetrical hexagon base and platform. The method also takes limitations of active and passive joints and link interference into account.

Another numerical approach was given by Yang & Chen [336] for the orientation workspace analysis of a 3 d.o.f. spherical manipulator, using a novel partition scheme for dividing the sphere D^3 to which $SO(3)$ can be mapped, into finite elements with equal volume.

According to Bohigas et al. [21], a proposed method for workspace computation should ideally be *complete*, *accurate* and as *general* as possible. The property of being complete refers to the ability of finding all connected components of the workspace, rather than obtaining just one achievable from a recent configuration. Further, a precise representation of the workspace volume and of any motion barriers in the interior of the volume is required with high accuracy. Finally, every possible workspace slice that can be obtained by fixing three of the pose parameters, should be determined. A unified method satisfying the previous requirements was given in [21], valid for Stewart platforms consisting of base and platform of arbitrary geometry, connected by six legs of type UPS. First, a system of equations is formulated, defining the boundary $\partial\mathcal{W}$ of the workspace $\mathcal{W}$. Then, an iterative procedure based on linear relaxations is used to isolate slices of such boundary exhaustively at the required solution. In particular, a system of 33 equations in 39 variables represents the general 6D-workspace, using the joint variables and constraints introduced by the mechanical limits on all joints. The boundary $\partial\mathcal{W}$ is then obtained by adding an additional equation, representing the property that a point lies on $\partial\mathcal{W}$, whenever any of the active or passive joints reaches a mechanical limit. Thus, the $\partial\mathcal{W}$ has one dimension less then $mathcalW$ itself. However, it is still hard to compute the boundary exhaustively; but by setting three variables to a constant value, arbitrary two-dimensional slices of the boundary can be obtained. The algorithm for solving the system of equations numerically is based on a linear relaxation paradigm (described by Porta et al. [247]), starting with algebraizing the equations into a canonical form. An initial box containing all the solutions is computed, and then all boundary points at the desired resolution are isolated using a branch-and-prune method. Results with computations times of about one minute for computing the *constant position* and the *constant orientation* *workspace* are provided. However, possible leg interferences are not taken into account. The proposed method shows potential of being able to cope with additional constraints, like singularity-avoidance or leg-leg collisions, but further work seems required to provide a formulation ensuring an acceptable computational effort.

Maximal workspace

The most difficult task is the calculation of the *maximal workspace*, which may take hours of computation time for manipulators with large ranges for certain joint variables [191].

A complete analysis of the reachable workspace of spatial Gough-Stewart platforms with planar base and platform was given for the first time in 2006 by Pernkopf & Husty [236].

Jin et al. [134, 135] found optimal dimensions for the maximal 6D-workspace of a 3$\underline{\text{RP}}$PS parallel manipulator, using an approach of finite-partitioning of $SE(3)$ (based on [336]). The workspace optimization presented there is valid, but computational expensive, due to the large number of sampling features.

An algorithm for computing the reachable workspace of symmetric spatial parallel manipulators was proposed by Zhao et al. [346, 347], who investigated the analogous symmetry properties between the workspace and the mechanism structure. It has been proven, that the symmetry group (which for a polygon V in the plane consists of all motions ϕ for which $\phi(V) = V$) of the kinematic chain structure of the end-effector must be a subgroup of that of the end-effector. The computational effort may be reduced significantly, because only a reduced part of the workspace needs to be investigated; in particular, if the order of the symmetry group of the mechanism structure is n, then the search area can be reduced to $\frac{1}{n}$ of the initial one.

Further results

An analytic solution for the reachable workspace of axially symmetric hexapods was given by Agheli & Nestinger [1], based on a decomposition of the mechanism into three intersecting modified four-bar mechanisms. The method is extensible to other types of workspaces and manipulator geometries.

An overview of workspace determination for several types of parallel manipulators, e.g., for planar manipulators, 3-UPU-manipulators, and the general 6-UPS manipulator, are given by Merlet [198]. For other types of parallel manipulators, specific research has been done; for example Cervantes-Sánchez et al. [46] present a methodology for the determination of the workspace for 5R spherical symmetric manipulators or Hay & Snyman [105] a numerical multilevel optimization method for determining the dextrous workspace of a 3 d.o.f. planar parallel manipulator.

Kotlarski et al. [148] present a method to enlarge the *useable workspace*, i.e., the singularity-free workspace, guaranteeing a certain achievable accuracy for a given actuator resolution. The method is based on interval analysis and is operational for planar parallel manipulators with kinematic redundancy, realized by adding at least one actuated (prismatic) joint to one kinematic chain. Results show, that an additional (prismatic) actuator significantly increases the useable workspace.

As mentioned before, a constraint which enforces limitation on the workspace is a possible self-collision, i.e., the collision between the legs of the robot, and, eventually with the base or the platform. Collision detection may be formulated as a constrained optimization problem in which distance between the legs is minimized. But there is no need to compute an exact value of the minimum, it is enough to show whether it is lower or greater than a given value. Merlet & Daney [204] present an algorithm based on interval analysis, which exhibits an *efficient leg interference check* for 6 d.o.f. Gough-platforms, where the legs are modeled as finite cylindrical elements. The algorithm allows to detect an interference in a given workspace volume. Remaining problems are to determine the maximal interference-free workspace or the design parameters so that a given workspace will be interference free [204].

4.4.2 Workspace for serial manipulators

The determination of the workspace volume for serial manipulators is easier than the computation for parallel mechanisms. Consider for example a 3R planar manipulator. There, the reachable workspace problem is to find the boundaries of the real working volume of the given robot with design parameters $p^* \in P$, which can be accomplished by computing the singularity surface

$$S_p^* = \{\Theta \in T_{\mathbb{R}^3} \mid \operatorname{rank} \partial_\Theta K_{3R}(\Theta, p^*) < 3\},$$

where $T_{\mathbb{R}}^3$ denotes the set of real joint angles, i.e., the joint space consisting of all possible (real) joint angles $\Theta = (\Theta_1, \Theta_2, \Theta_3)$; in absence of joint limits, T^3 is the three-torus. $K_{3R}(\Theta, p^*)$ denotes the forward kinematics map (cf. Section 4.2) restricted to the set of parameters for given parameters p^* for the robot. Then $\partial_\Theta K_{3R}$ denotes the 3×3 Jacobian of K_{3R} with respect to the joint angles. The boundaries of the reachable workspace are contained in the image of the singularities [313]. Guo et al. [97] present a complete classification and workspace composition of 3R planar manipulators based on the characteristics chart method.

For serial manipulators, the robot manipulator structure can be divided into two separate groups. The regional structure consists of the arm moving the end-effector in a desired position; the orientation structure contains the link(s) rotating the end-effector to a desired orientation. Therefore, the positioning and orientation problem can be discussed separately. Gupta & Roth [98] have presented some valuable concepts regarding the workspace of manipulators. Different solution approaches for workspace calculations are possible, e.g., algebraic and geometric approaches [42, 43, 300], continuation [136], heuristic methods [335], or single- and multi-objective optimization [16, 45, 226]. In particular, for 3R manipulators a great number of publications can be found, due to their importance in industrial applications. So Zein et al. [343] present a classification of 3R manipulators with at least one DH parameter equal to zero, based on the topology of the workspace. A characterization of the workspace boundary of 3R manipulators can be found in [272]. Ottaviano et al. [225] characterize workspace topologies of 3R industrial manipulators by a level-set reconstruction; the proposed algebraic expressions are useful to classify industrial 3R manipulators. Further, conditions for avoiding singularities in the workspace are identified. Recently, a fast algorithm for numerical computation of the workspace has been presented by Shi et al. [277]. It is based upon the singularity analysis of a manipulator. The algorithm is operational for serial manipulators with up to six revolute joints.

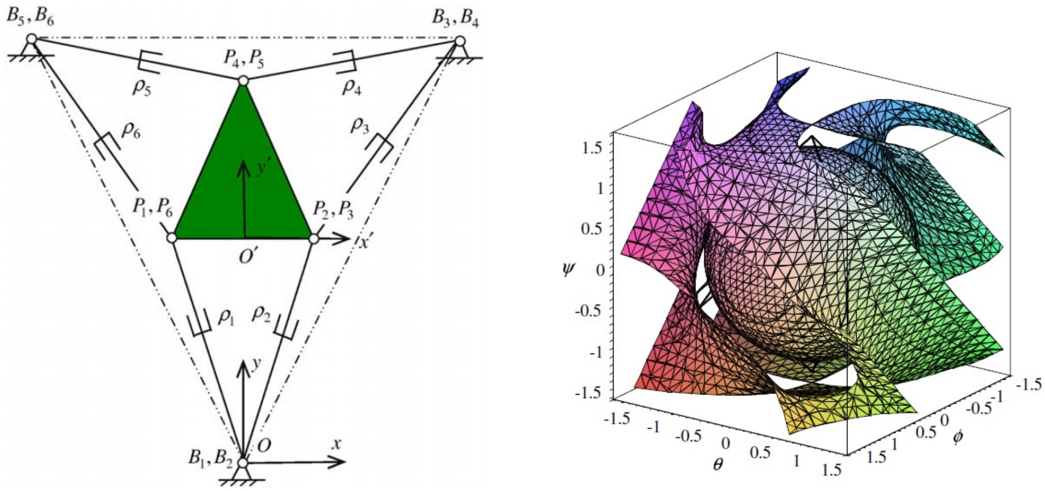
Since workspace quality and quantity have become most important performance indices [135], the workspace of a robot manipulator is an important criterion for the optimal design of the mechanism. Further, the property of being singularity-free is a crucial aspect of workspace analysis (see Chapter 8).

4.4.3 Related problems

An associated problem to workspace determination is to define a geometrical object (e.g., a cube or a sphere) and determine the largest such object enclosed in the workspace of the robot (see also section 8.3.1). For the Gough-Stewart platform of type MSSM (Maximal Simplified Symmetric Manipulator, see [198]), Jiang & Gosselin [132] provided an iterative algorithm for determining the maximal singularity-free sphere in the orientation workspace (see Fig. 4.1). The problem is to find the point on the singularity surface which is closest to a reference orientation; the distance of this point to the reference orientation then is taken as the radius of the maximal singularity-free sphere.

Chablat et al. [51] present an interval-analysis based algorithm for 3 d.o.f. parallel manipulators to determine the largest cube (or, with some parameter-modifications the largest sphere or other regular shaped object) enclosed in the dexterous workspace. [52] present the square workspaces of two specific 3 d.o.f. manipulators. An algorithm which is able to verify (very quickly) whether an arbitrary trajectory (or surface or volume) lies entirely within the workspace was proposed by Merlet [194]. The algorithm is based on interval arithmetic and therefore provides the validity of a trajectory (surface/volume) in a guaranteed matter. Further details about the singularity-free workspace are presented in Chapter 8.

Another problem is to **verify the reachability of a planned workspace**; in particular, to decide if a specific planned workspace $\mathcal{W}_0$ is contained in the full 6D workspace $\mathcal{W}$. Let the boundary of $\mathbf{C}$ in (4.14) be denoted by $\partial\mathbf{C}$; it consists of all vectors in which at least one bound defining $\mathbf{C}$ is attained. The reachability verification-problem then can be solved in two steps:



(a) The MSSM architecure (top view).

(b) Singularity surface, maximal singularity-free orientation workspace and maximal singularity-free sphere of the MSSM

.

Figure 4.1. A Gough-Stewart platform of type MSSM (Maximal Simplified Symmetric Manipulator) and its maximal singularity-free orientation workspace [132]

(i) Verify the existence $x_0 \in \mathcal{W}_0$. Such a point can frequently be found by local methods; a global search is only needed in hard cases.

(ii) Show that

$$F(s, \Theta) = 0 \quad \text{for } s \in \mathcal{W}_0, \quad C(s, \Theta) \in \partial \mathbf{C}, \tag{4.15}$$

has no solution.

The **boundary-freeness** condition in (4.15) is a global constraint satisfaction problem and therefore hard. If the two steps of the reachability verification-problem are solved successfully, continuity implies reachability [219]. (For more on boundary-freeness see, e.g., Neumaier [218], Zangwill & Garcia [340].)

Chapter 5

Geometric Kinematics: Forward and Inverse Kinematics

5.1 Introduction

Kinematics studies the motion of points or rigid bodies without reference to mass or forces. Motion is described by the trajectories of points, lines or other geometric objects. Furthermore, their differential properties like velocity and acceleration are also discussed. **Geometric kinematics** is an important part of the analysis of a robot mechanism. It is about solving the equation of state (4.1) for the states (s, Θ), given partial information. In addition, geometric kinematics is concerned with determining the accuracy with which the states can be assessed from noisy measurements. This topic is addressed in Chapter 7. **Differential kinematics** studies the relationship between the rate of change of pose and joint configurations of a robot performing a known motion (see Chapter 6).

The **direct** or **forward kinematics problem (FKP)** is to find the coordinates of the pose of the end-effector for given actuated joint coordinates. In particular, given are measurements $y = P\Theta$ for the sensed part $P\Theta$ of the joint configuration Θ. Then the problem is to find the set of possible states (s, Θ) for the measurement-vector y, and it is to be determined which of these states corresponds to the actual pose of the robot.

For the **inverse kinematics problem (IKP)**, on the other hand, the pose of the end-effector is known while the actuated joint coordinates are wanted. Therefore, the task is to find the set of possible states (s, Θ) for given generalized coordinates s, and to determine which one corresponds to the actual joint configuration of the manipulator.

Thus, for a robot in which all modeled joint variables are sensed, the FKP means to derive s from Θ while the IKP means to get Θ from s [219].

5.2 Serial Manipulators

For serial manipulators the pose of the end-effector is uniquely determined by the joint configuration. Therefore solving the direct kinematic problem for serial manipulators is straightforward, evaluating the **kinematic displacement** (or **loop closure**) **equations** (3.2)

$$T = T_1 \cdots T_n,$$

where the homogeneous transformation matrix T contains the pose of the end-effector with respect to the base, and T_i $(i = 1, \ldots, n)$ denote the transformation matrices, which write the coordinates of frame $\mathcal{F}_i$ with respect to frame $\mathcal{F}_{i-1}$.

Example 5.2.1 (FKP: 3R planar manipulator [313]). *Consider a planar robot with two revolute joints, as shown in Figure 5.1. Without loss of generality the point of origin $O = (0, 0)$. The relative rotation angles $\theta_1, \theta_2, \theta_3$ are given, and the lengths a_1, a_2, a_3 of the three links are known. Now, the coordinates of the tip, $R = (r_x, r_y)$ and the absolute orientation angle ϕ_3 of the last link shall be computed. Since link offset and twist angle are zero for each link, the transformation matrices T_i derived from (3.3) are*

$$T_i = \begin{pmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

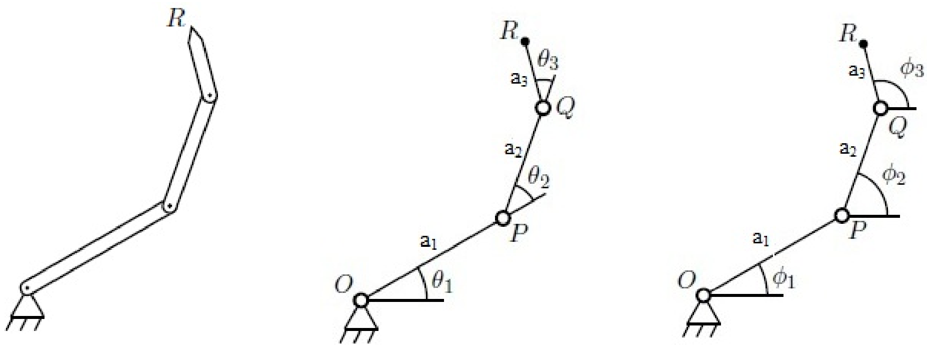
where $c_i = \cos\theta_i$ and $s_i = \sin\theta_i$. After converting the relative angles θ_i to absolute ones (ϕ_i) as

$$(\phi_1, \phi_2, \phi_3) = (\theta_1, \theta_1 + \theta_2, \theta_1 + \theta_2 + \theta_3),$$

the position of the tip is

$$\begin{pmatrix} r_x \\ r_y \end{pmatrix} = \begin{pmatrix} a_1 \cos\phi_1 + a_2 \cos\phi_2 + a_3 \cos\phi_3 \\ a_1 \sin\phi_1 + a_2 \sin\phi_2 + a_3 \sin\phi_3 \end{pmatrix},$$

which has a unique solution.



(a) (b) (c)

Figure 5.1. A 3R planar robot arm (a), and its kinematic skeleton (b), (c) [313, p. 7].

However, the inverse kinematics problem for serial manipulators might be difficult to solve. The IKP leads to a system of multivariate non-linear polynomials (i.e., n polynomials in n unknowns) which has multiple solutions, depending on the number of joints, the link parameters and the allowable ranges of motion of the joints [57]. A basic approach for solving the IKP of general (non-redundant) serial manipulators is to disassemble the kinematic chain of the robot at two joints to obtain two subchains or subloops [6]; the procedure is described by, e.g., ANGELES [6, Ch. 9].

Example 5.2.2 (IKP: 3R planar manipulator [313])**.** *Consider a 3R planar manipulator as in Example 5.2.1. Now, the absolute rotation angle ϕ_3 of the last link and the position of the tip, $R = (r_1, r_2)$ are given. Thus, the position of the intermediate point Q can be easily derived as*

$$Q = \begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} r_x - a_3 \cos\phi_3 \\ r_y - a_3 \sin\phi_3 \end{pmatrix}.$$

However, to find the remaining angles ϕ_1, ϕ_2 a system of trigonometric equations, given by

$$\begin{pmatrix} q_x \\ q_y \end{pmatrix} = \begin{pmatrix} a_1 \cos\phi_1 + a_2 \cos\phi_2 \\ a_1 \sin\phi_1 + a_2 \sin\phi_2 \end{pmatrix} \tag{5.1}$$

has to be solved. There are several ways for doing this. One way is to convert (5.1) into a system of polynomial equations. This may be done by introducing the variables $x_i = \cos\phi_i$ and $y_i = \sin\phi_i$ and the trigonometric identity $\cos^2\phi_i + \sin^2\phi_i = 1$ for $i = 1, 2$, and substituting the variables in (5.1). Another possibility is to use the tangent half-angle relation for rationally parameterizing a circle. Here, the substitutions made are

$$\sin\phi_i = \frac{2t_i}{1+t_i^2}, \quad \cos\phi_i = \frac{1-t_i^2}{1+t_i^2}, \quad \text{for } t_i = \tan\frac{\phi_i}{2}.$$

An alternative way is to concentrate first on finding the intermediate point $P = (p_x, p_y)$ which can lie at either of the intersection points of two circles, centered at O with radius a_1 and centered at Q with radius a_2, respectively. This leads to a system of two quadratic polynomials in the variables p_x and p_y,

$$\begin{pmatrix} p_x^2 + p_y^2 - a_1^2 \\ (p_x - q_x)^2 + (p_y - q_y)^2 - a_2^2 \end{pmatrix} = 0.$$

By substracting the first from the second polynomial, a system of one quadratic and one linear equation in the unknowns (p_x, p_y) is obtained. Hence, the IKP of the 3R planar manipulator has at most two isolated solutions.

In the end of the 1980s and at the beginning of the 1990s, the IKP of a general 6R manipulator was considered to be the most difficult among all problems in kinematics. PIEPER [242] provided the earliest results and derived a closed-form solution for manipulators with special geometry. The primary upper bound of 32 solutions [265] was reduced to 16 in the end of the 1980s [158, 249, 299]. RAGHAVAN & ROTH [252] presented a solution procedure based on dialytic elimination which results in a 16th degree polynomial in the tangent of the half-angle of one of the joint variables; e.g., Manocha & Canny [176] expressed the polynomial as matrix determinant and computed its roots by reducing the problem to an eigenvalue problem. An exhaustive description of the IKP for decoupled manipulators is given by Angeles [6, Ch. 4]; further, in [6, Ch. 9] solution methods for the IKP of general (non-redundant) 6R serial manipulators are discussed in detail.

More recently, Selig [275] gave a simple but abstract proof based on intersection theory and a cell decomposition of the Study quadric, which is an elegant representation of $SE(3)$. Di Rocco et al. [64] turned Selig's proof into a concrete homotopy method for numerically solving the IKP. In the final topology, there are 16 paths used to find the 16 solutions. Pfurner & Husty [124, 240, 241] provided a new and efficient algorithm for the general 6R manipulator, where for the first time the geometric structure of the problem is revealed. The Study model of Euclidean displacements (also denoted as kinematic image) is used, which identifies a displacement with a point on a six dimensional quadric S_6^2. The 6R–chain is decomposed to two 3R–chains, of which the kinematic image turns out to be a Segre manifold consisting of a one parameter set of 3-spaces. The intersection of two of these manifolds and S_6^2 gives 16 points, which are the kinematic images representing the 16 solutions of the IKP. By algebraic means, there are a system of seven linear equations and a resultant to solve in order to derive the univariate 16th degree polynomial. There, the algorithm gives two out of the six joint angles and calculates the remaining by solving the IKP of two 2R–chains.

Another approach is presented by Qiao et al. [250], which uses double quaternions and a new elimination procedure based on Dixon resultants. The double quaternion formulation of the spatial displacements simplifies the solution procedure, compared to the traditional homogeneous matrix method [250].

For 6 d.o.f. manipulators of specific architectures, various research has been done (e.g., [110, 113, 221]. For example, Ho et al. [110] provide a fast and reliable closed-form solution for a mechanism which is similar to the human arm structure, which is fast enough for real-time control. Nie & Huang [221] use the method of sequential retrieval for deriving an analytic solution for a specific arm–structure, for which the IKP has not been solved analytically before.

Additionally to geometric, algebraic or iterative approaches, optimization methods for solving the IKP have been published (e.g., [61, 81, 87, 162, 167, 331]). Due to the higher complexity of the IKP of redundant manipulators, the usage of artificial neural networks has become more important [147]. For example, Köker [147] proposed a method using neural networks combined with a genetic algorithm to minimize the error at the end-effector of a 6 d.o.f. Stanford manipulator; micrometer levels of accuracy are produced.

5.3 Parallel manipulators

In contrast to serial manipulators, solving the inverse kinematics problem for parallel mechanisms proves itself to be quite simple, while the solution of the direct kinematics problem is usually quite difficult and may possess several solutions.

5.3.1 Inverse kinematics of parallel manipulators

Solving the IKP for parallel robots is essential for position control of parallel robots. The general approach for solving the IKP can be either analytical or geometrical, as described by MERLET [198]. The analytic method uses the vector between the endpoints A and B where the chains are fixed to the base and the moving platform as fundamental data for solving the IKP. On the one hand, the coordinates of A are by construction known in a fixed reference frame, while the coordinates of B may be derived from the pose (c, Q) of the end-effector. If s represents the pose of the moving platform in generalized coordinates,

$$\overrightarrow{AB} = H_1(s) \tag{5.2}$$

gives the positions of the extreme points of all the chains for which the joint coordinates are to be calculated. On the other hand, the chain from A to B can be determined by the chain joint coordinates Θ and, if needed, the pose of the moving platform. So, the vector AB can be written in terms of Θ and s as

$$\overrightarrow{AB} = H_2(s, \Theta),$$

and solution of the IKP can be found by solving the system of equations

$$H_1(s) = H_2(s, \Theta), \tag{5.3}$$

which may be written for separable robots as a system of univariate equations as in equation (3.11) (using similar notation as in Example 5.3.1). In general, the calculation for each chain involves the generalized coordinates of the moving platform and the joint coordinates of the actual leg, but not those of the other chains. Therefore the solution can be done in parallel for each of the chains (as long as the chains do not share an actuated joint variable). The number of unknowns in (5.3) is $3p$, if p is the number of chains; for planar robots, the number of unknowns is $2p$. The advantage of this analytic approach is that it can be fully automatized.

Example 5.3.1 (IKP: 6-U$\underline{\text{P}}$S manipulator)**.** *Considering a 6 d.o.f. separable parallel robot (i.e., a 6 U$\underline{\text{P}}$S) manipulator, see Fig. 5.2), the IKP is to determine the length Θ_i of $\overrightarrow{A_iB_i}$, being given the generalized coordinates s of the platform, i.e., the vector $\overrightarrow{OC}$ and a set r of parameters defining the orientation of the platform. Hence, the components of $\overrightarrow{A_iB_i}$ can be completely determined by writing (5.2) as*

$$\overrightarrow{A_iB_i} = \overrightarrow{A_iO} + \overrightarrow{OC} + R\,\overrightarrow{CB_{i_r}} = H_{1_i}(s), \tag{5.4}$$

where the vectors $\overrightarrow{A_iO}, \overrightarrow{CB_{i_r}}$ *are known, and* R *denotes the rotation matrix between the moving frame and the reference frame. Furthermore,* B_i *can be considered as the extreme point of an RRP chain. Let* $\gamma_i = (\alpha_{i_1}, \alpha_{i_2})$ *be the vector of the two rotation angles around the R joints. Then the unit vector* n_i *that defines the direction of the leg, may be calculated as a function of* γ_i*, so*

$$\overrightarrow{A_iB_i} = (l_i + \Theta_i)n_i = H_{2_i}(\gamma_i, \Theta_i), \tag{5.5}$$

where l_i *denotes the part of the* i*th leg of fixed length. Thus, by combining equations (5.4) and (5.5), for each leg a system of 3 equations in the unknowns* $\Theta_i, \alpha_{1_i}, \alpha_{2_i}$ *is derived. To solve this system one may notice, that the norm of* $\overrightarrow{AB}$ *in equation (5.4) is* Θ_i^2*, and that the norm may be calculated directly from equation (5.5). Therefore, by writing* $\overrightarrow{AO} = -d_i^p, \overrightarrow{OC} = c, R\,\overrightarrow{CB_r} = Qd_i^p$*, the joint coordinates* Θ_i *can be derived from (3.12), which is*

$$\|c + Qd_i^p - d_i^b\|^2 - (l_i + \Theta_i)^2 = 0.$$

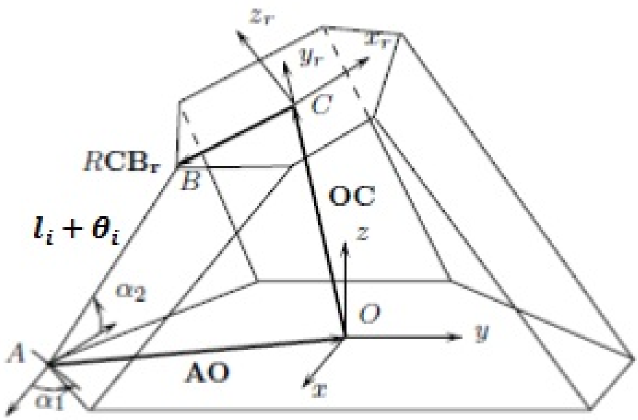


Figure 5.2. Fundamental vectors for establishing the inverse kinematics of a 6-U$\underline{P}$S platform [198, p. 99].

A geometrical approach for solving the IKP is to consider that the endpoints A and B of each chain have a known position in three-dimensional space [198]. Then, each leg is cut at a cutting point M. So the chains between A and M, respectively M and B, constitute two different mechanisms M_A and M_B. Under the assumption that the mechanisms have only classical lower pairs (i.e., revolute or prismatic joints, see Chapter 3), the point M is as a member of M_A considered to lie on an algebraic variety V_A; analogously, M considered as a member of M_B lies on an algebraic variety V_B. An algebraic variety with dimension d in three-dimensional space is defined through a set of $(3-d)$ independent equations. So, the varieties V_A and V_B with dimensions $\dim V_A$ and $\dim V_B$ are defined through $(3-\dim V_A)$ and $(3-\dim V_B)$ equations, respectively. The solution of the IKP lies on the intersection of the two varieties. An advantage of this method is the fact that the intersection of algebraic varieties in two- and three-dimensional space is a well-studied topic. Therefore, e.g., bounds on the number of intersection points may be determined without computing. The freedom of the choice of the cutting point is advantage and disadvantage at the same time. The advantage is, that it gives some freedom of the final system of equations and so is responsible for the complexity of the resulting equations. The drawback is that automation of this method is quite difficult [198].

Example 5.3.2 (IKP: 3-$\underline{\text{R}}$RR planar manipulator - geometric approach [198]). *Consider a 3-$\underline{R}RR$ type planar robot with 3 d.o.f. as shown in Figure 5.3. For solving the IKP using the geometrical approach, the point M_i is chosen as cutting point. So, the mechanisms A_iM_i and B_iM_i impose the constraints, that on the one hand M_i must lie on a circle centered in A_i with radius l_1^i, and on the other hand on a circle with center B_i and radius l_2^i. Thus, M_i lies at the intersection of two circles, which produces either 2 solutions or none for each leg. So in total, the IKP has at most $2^3 = 8$ solutions.*

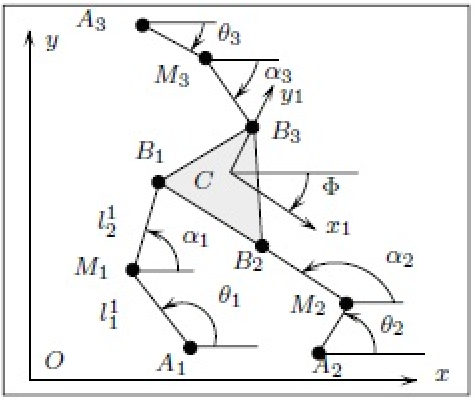


Figure 5.3. The 3-$\underline{\text{R}}$RR type planar parallel manipulator [198, p. 98].

In summary it may be said, that solving the IKP for parallel manipulators is usually quite simple, as long as the geometry of each leg is simple enough.

5.3.2 Direct kinematics of parallel manipulators

Solving the direct kinematics problem for a parallel manipulator is, contrary to serial manipulators, usually quite difficult. But it is essential for the control of the pose of the platform, and also for the velocity control of the end-effector. A detailed discussion of the topic and results for several architectures are given by Merlet [198].

Special architectures of parallel manipulators may have a unique solution to the FKP (e.g., a 3-$\underline{\text{P}}$RP planar parallel manipulator proposed by Zarkandi [341]), but in general, the solution of the FKP of parallel manipulators is not unique; for example the FKP for a Gough-type parallel manipulator (a general 6-U$\underline{\text{P}}$S platform) may posses up to 40 solutions. This was first proven by Ronga & Vust

[263], who gave an abstract proof for the existence of 40 complex or real solutions. This result was confirmed by RAGHAVAN [251] using polynomial continuation, LAZARD et al. [155] using Gröbner bases, MOURRAIN [213] using an approach based on elimination and resultants, and WAMPLER [310] and HUSTY [120], who both used parameterization based on dual quaternions. WAMPLER uses the principles of parameter continuation and 2-homogeneous Bezout numbers, while HUSTY derived a univariate polynomial of degree 40 using an elimination process, which requires intuition and thus cannot easily be automatized. DIETMAIER [66] gave the first proof of the existence of a 6-U$\underline{\text{P}}$S manipulator with 40 real assembly modes. HAUENSTEIN & SOTTILE [103] used a modification[1] of Dietmaier's formulation and verify that the 40 points correspond to distinct real solutions, using the algorithm `alphaCertified`, and hence give a rigorous mathematical proof of Dietmaier's result.

In addition to the lack of uniqueness of the FKP-solution, mostly there is no analytic expression of the generalized coordinates of the platform as functions of the actuated joint coordinates. There are (algebraic) methods (which are summarized below) for finding *all* the solutions, which may be regarded for singularity and workspace analysis in particular. These methods require a computation time too large to be used in real-time context; the computation times range between a few second and a minute, depending on the geometry of the mechanism [200]. But in real-time applications the the most important point may not be to determine all possible solutions but the one related to the current pose of the platform, i.e., when its joint configuration was measured. Algorithms which may eventually calculate the current pose are based on numerical methods and require a-priori information on this pose. If information about the possible pose for the end-effector is available, the problem is said to be local. For example, in a real-time controller, the current pose is computed every sampling time and in each control-cycle a solution close to the previous one is needed. For tasks where such information may not be available, for example when starting the robot, the problem is global and a solution method is required that does not need this information. One possibility is to compute all possible solutions but there is no known algorithm which allows the determination of the current pose among the solution set.

Example 5.3.3 (Direct kinematics of the Gough platform [219])**.** *Given numerical values for the d_i^p, d_i^b, l_i, the equations (3.12)*

$$\|c + Qd_i^p - d_i^b\|^2 - (l_i + \Theta_i)^2 = 0 \quad (i = 1, \ldots, 6)$$

need to be solved for the pose (c, Q). W.l.o.g. the base and platform origins may be chosen such that

$$d_1^p = d_1^b = 0.$$

By introducing the constants

$$\alpha_i = \frac{1}{2}((l_i + \Theta_i)^2 - \|d_i^p\|^2 - \|d_i^b\|^2),$$
$$a_i = d_i^p - d_i^b, \quad b_i = d_i^p + d_i^b,$$

[1]VERSCHELDE: *Polynomial formulation and solutions of Stewart-Gough platform with 40 real positions*, available at `http://www.math.uic.edu/~jan/Demo/stewgou40.html`

and the vectors

$$x = \frac{1}{2}(Q^T c + c), \quad y = \frac{1}{2}(Q^T c - c),$$

we get

$$c = x - y, \quad Q^T c = x + y, \tag{5.6}$$

$$x \cdot y = \frac{1}{4}(\|c\|^2 - \|Q^T c\|^2) = 0, \tag{5.7}$$

$$x \cdot x + y \cdot y = \|x + y\|^2 = \|c\|^2 = 2\alpha_1, \tag{5.8}$$

$$a_i \cdot x + b_i \cdot y + (d_i^b) \cdot Q d_i^p = \alpha_1 - \alpha_i \ (i = 2, \ldots, 6). \tag{5.9}$$

If the quaternion parameterization $Q = Q_e[r]$ (see Section 4.3) is used for the rotation, then equation (5.6) implies

$$r \times x = r_0 y, \tag{5.10}$$

where

$$r_0^2 + r^2 = e, \ r_0 \geq 0. \tag{5.11}$$

So, (5.7)–(5.11) are 11 quadratic equations for the 10 variables in r_0,r, x, y. Therefore, one of the equations is redundant; if $r_0 = 0$, then one of the three equations making up (5.10) is redundant, since $r \times x = 0$ only forces x to be parallel to r; and if $r_0 \neq 0$, then (5.7) is a consequence of (5.10).

Finding *all* solutions: The initial position problem.

For solving the FKP, consider the inverse kinematic equations (5.3); these form a non-linear system of equations, which may posses several solutions, corresponding to several possible poses of the platform for given joint variables. These different configurations of a manipulator for given joint coordinates are called **assembly modes**. If all assembly modes are computed, the so-called **initial position problem** is solved [239]. When using algebraic elimination methods to reduce the problem to finding the roots of a univariate problem, it is necessary to determine an upper bound for the number of assembly modes beforehand. This bound will then be used to guide the elimination process to find a polynomial of lowest possible degree (see Section 1.2.2).

Main advantages and disadvantages of the currently known methods for determining *all* the solutions of the FKP are summarized below, as well as important and recent research results. For details on the methods see Chapter 1 and Chapter 2. Note, that the use of elimination, polynomial continuation, and Gröbner bases requires algebraic functions for the inverse kinematics.

Elimination. The method may be fast, but it is not easy to get the final polynomial of minimal degree. The main drawback of the method is that it is numerically unstable and needs multi-precision arithmetic to produce reliable results. Hence, it does not provide certified solutions [198]. As mentioned before, Husty [120] was the first one to derive a 40th degree polynomial for the general 6-U$\underline{\text{P}}$S platform using an elimination procedure. Lee & Shim [160, 161] used a dialytic elimination procedure and proposed an improved algorithm, which derives the 40th degree polynomial from a final 15×15 Sylvester matrix. Innocenti [126] presents an elimination based algorithm, which is suitable for implementation in a standard floating-point computation environment. For a 6-6 Stewart platform with special properties, polynomials of degree less than 40 may be obtained. For example, Huang et al. [117] present an algorithm for a manipulator with planar base and platform which derives a 20th degree polynomial, leading to all solutions of the FKP; and Huang et al. [116] derive a univariate polynomial of degree at most 14 for a symmetrical 6-6 Stewart platform whose base and platform are hexagons and the joint centers satisfy certain conditions.

Elimination based approaches are also used for other specific parallel architectures. For example, Wampler [312] presents a modified Sylvester's elimination method for solving the FKP of spherical multi-loop structures, where the rotation is formulated either in rotation matrices or in quaternions. The numerical calculation is done using standard eigenvalue routines. Another example is given by Enferadi & Tootoonchi, who study a spherical star triangle manipulator [70] and a double-triangle spherical manipulator [71], and derive a minimal polynomial of degree 8 for the FKP using Bezout's elimination method. Rodriguez & Ruggiu [258] solve the forward displacement problem of several spherical parallel manipulators, which is to find the orientation of the moving platform for a given set of actuated joint variable values, using the Dixon determinant procedure. The Dixon determinant is a method for finding the eliminant of a system of polynomial equations which was adapted by Nielsen & Roth [223] to handle initial trigonometric equations [311].

Homotopy/polynomial continuation. This methods produce certified solutions, but are usually slow [198]. There is a possibly large number of branches which have to be followed; for example, Raghavan's method, needed 960 branches to derive 40 complex solutions of a general 6 d.o.f. Stewart platform [251]. However, more recent results show, that the FKP of a general Stewart-Gough platform may be solved in a few seconds up to about a minute of computation time, depending on the computer applied to the task [313]: according to Wampler & Sommese [313], solving the FKP of general Stewart-Gough platforms is equivalent to solving a system of seven quadrics in $\mathbb{P}^7$ (where $\mathbb{P}^n$ denotes the n-dimensional projective space). The solution of seven quadrics requires to track 2^7 homotopy paths. Furthermore, once the first general example has been solved to find just 40 solutions, all subsequent examples may be solved by tracking only 40 homotopy paths [313]. Wampler [310] presents a specific valid 40-path homotopy with which all non-singular solutions to the FKP of a general Gough-Stewart platform can be obtained.

Another problem of Polynomial Continuation methods was, that they were limited to finding isolated roots. In particular, overconstrained mechanisms which have more d.o.f. than expected from usual mobility calculations may have a mixture of isolated solutions and motion curves of various dimensions, and therefore are difficult to solve. But, indicated by resent results and software proposed by Sommese et al. [282, 283, 285, 303, 313], this problem has been overcome. In [283] applications of new methods to a seven-bar structure and to a special form of the Stewart-Gough platform, the so-called Griffis-Duffy platform (Fig. 5.4) are given. These mechanisms have triangular base and platform and ball joints at each vertex and along each side. In both cases, the method provides an irreducible decomposition of the solution set; for the seven-bar mechanism the derived results are consistent with the known theory. For the Griffis-Duffy platform on the other hand, some differences to the results published by Husty & Karger [122] are found, e.g., for the Griffis-Duffy II platform an additional sixth-degree component was found.

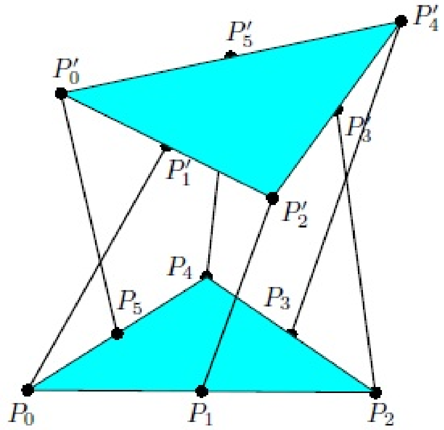


Figure 5.4. A Griffis-Duffy platform. Both base and platform are equilateral triangles with ball joints at each vertex and along each side [283].

Varedi et al. [302] use the homotopy method for solving the FKP of an offset 3-UPU translational parallel manipulator. There a combined Newton-Homotopy approach (as studied by Wu [330, 332]) is used, i.e., to start the system of equations with an auxiliary homotopy function and then solve it by the Newton-Raphson method. The proposed algorithm leads to all 16 real solutions and converges even with bad initial guesses.

Gröbner Bases. These need exact rational arithmetic and provide narrow rational bounds on the solutions. In particular, if the coefficients of the inverse kinematics equations are rational numbers, this method provides certified solutions [198]. It was used by Mourrain [213] for confirming the number of solutions for general Stewart-Gough platforms, by Lazard [154, 155] for solving the FKP of Stewart platforms, and by Rolland [262] for several generic 6-6 architectures. Faugère & Lazard [79] use a Gröbner basis approach to predict an upper bound on the generic number of solutions for special architectures. According to Merlet

[198], the Gröbner basis method using the `FGb`-algorithms of Faugère [76, 77] and Rouillier [266], is one of the fastest methods available to solve the FKP in a guaranteed manner, if the coefficients of the equations are rational. However, the implementation involves the use of large integers.

Dhingra et al. [63] use Gröbner bases for the closed-form displacement analysis of planar multi-loop mechanisms [63] and proposed a Gröbner-Sylvester hybrid method [62], which provides a systematic approach for generating (additional) equations for constructing Sylvester's matrix. First, a system of equations is transformed into its reduced Gröbner basis G using degree lexicographic ordering. Then, either using the entire set of generators in G or a subset, the Sylvester matrix is assembled. The hybrid approach directly yields the 40th degree polynomial, unlike Husty's derivation, where intermediate polynomials of degree up to 320 are generated [120]. The method is applied to a general 6-6 Stewart mechanism. On this basis, Gan et al. [84] derive a 40th degree polynomial for a general 6-6 Stewart mechanism from a constructed 13×13 Sylvester matrix using Gröbner bases, and Huang & He [115] derive a 20th degree univariate polynomial from the determinant of the 10×10 Sylvester matrix for a general planar Stewart platform.

Interval analysis. Methods based on interval analysis are numerically robust and do not necessarily need the inverse kinematic equations to be algebraic, but the unknowns have to be bounded (which usually is the case for kinematic problems). The interval analysis approach provides solutions in a fully reliable way, and as an additional advantage, inequalities restricting the possible poses (e.g., due to the limited speed of robot motion) and uncertainties in the model of the robot or in the measurements of leg lengths, can be taken into account. However, it is inconvenient that an estimation of the computation time is difficult to provide [198].

An interval analysis based method was applied, e.g., by Didrit et al. [65] to solve the FKP of a general 6 d.o.f. Stewart-type platform, implementing Hansen's algorithm. A more recent result was given by Merlet [195] for a general 6-U$\underline{\text{P}}$S manipulator. The algorithm finds all solutions and is one of the fastest available methods [198]. It is not intended to be used in real-time context for determining the current pose, and yet it is competitive in terms of computation time to the normally used Newton-Schemes (see below), but safer. If only one solution is found by Merlet's algorithm, this solution is (in contrast to the Newton-Schemes) guaranteed to be the current pose; if more than one solution is determined, for example, if the manipulator is close to a singularity (see Chapter 8), the robot will immediately be stopped, as it is not known for sure, what the current pose is. Further, the proposed method can be adapted to take physical constraints into account, which may have an decreasing effect on the computation time. However, the efficiency of the method relies heavily on a set of filtering methods [195].

Ad-hoc methods. Here, the FKP is first transformed into another simpler problem, which then is solved by applying one of the above methods, or an optimization procedure [198]. For example, Innocenti & Parenti-Castelli [127] use the solution for a 5-5 parallel manipulator (i.e., a Gough-type manipulator with 6 legs, which are fixed to base and platform by 5 joints) to reduce the problem of a 6 d.o.f. fully parallel mechanism to finding the roots of an univariate function. Ad-hoc methods may be quite fast, but they are bound to a specific architecture and cannot be extended easily to other problems.

Another example for an ad-hoc method is the systematic approach for special cases of UPS mechanisms proposed by Nair & Maddocks [217]. He shows that the inverse kinematics equations may be separated into a set of linear equations and a set of nonlinear constraint equations. For, e.g., a 6-UPS mechanisms with planar base and platform, Nair & Maddocks derived interesting results, for other types however, the approach may lead to highly complex equations and non-optimal results or, even no result at all [198]. Rojas & Thomas [261] use a method based on bilateration to derive the characteristic polynomials of all catalogued Baranov trusses (a special class of planar linkages) without relying on variable eliminations or trigonometric substitutions. The **bilateration problem** consists of finding the feasible locations of a point P, given its distance to two other points P_1 and P_2, whose locations are known (for details see, e.g., [259, 260]).

Zhang [345] presents a forward kinematic analysis of a planar generalized Stewart platform (planar GSP). The planar GSP can be considered as the most general form of planar manipulators with 3 d.o.f.; it consists of a fixed base and a moving platform connected by three distance or/and angular constraints between three pairs of points and/or lines on the base and platform [345]. Zhang classifies the forward kinematics of planar GSPs using Wu-Ritt's characteristic set method (see, e.g., Mishra [207], Wu [333]) to derive a triangular form of the FK-equations. Further, conditions are obtained to derive real solutions for the remaining triangular forms by a method described in [337]. Zhang also obtains explicit conditions on the parameters to have a given number of real solutions for each of the sixteen forms of planar GSPs.

Computing the *current* pose: the finite displacement problem.

Computing all the solutions of to the forward kinematics is only part of the solution of the FKP. In many applications, the *real FKP* [198] or **finite displacement problem** [239] has to be solved. That means, that the current pose of the platform has to be determined, i.e., the pose when the joint variables are measured. For the solution of the finite displacement problem, a-priori information is needed, but may not be available, e.g., when starting the robot. One possibility is to compute all solutions and select the current pose among these. However, no known algorithm for determining the current pose from the solution set exists [198]. Especially, designing an algorithm which provides a complete verification is difficult and still a challenging problem in kinematics. A possible approach is to transform the problem into a path-planning problem and assume that the pose s_{init} when the robot is initially assembled, is known. A solution s may be considered feasible for the current pose, if there is a singularity-free path from s_{init} to s such that all joint variables lie within their limits on the whole path, and legs, base, and platform do not interfere [198]. However, the singularity-freeness condition alone is not sufficient, as was shown by Innocenti & Parenti-Castelli [129] or Hunt & McAree [119], who proved that two different solutions may be connected through a singularity-free path. It is not known, whether the proposed approach will always lead to the solution, but it may work as long as the constraints are strong enough and the path-planning problem can be solved. Proving that it will lead to a unique solution is still an open problem [198].

Newton schemes. A common approach to find a new pose close to the former one is to solve the FKP using iterative numerical (Newton-type) algorithms. Assume that the generalized coordinates s of the platform are related to the joint configuration Θ by

$$\Theta = G(s),$$

and that s_0 is an estimate of the solution. Then the standard iterative Newton-scheme is of the form

$$s_{k+1} = s_k + A(\Theta - G(s_k)), \tag{5.12}$$

which stops when $\|\Theta - G(s_k)\| < \epsilon$ for a fixed threshold ϵ. According to the choice of the matrix A in (5.12), the scheme is called ([198])

Newton-Raphson scheme if $A = J = \frac{\partial G^{-1}}{\partial \Theta}(s_k)$, where J denotes the Jacobian of the robot,

Damped-Newton scheme if $A = \alpha^k \frac{\partial G^{-1}}{\partial \Theta}(s_k)$ with a gain factor α^k, and

Quasi-Newton scheme if, w.l.o.g., $A = \mathbb{1}$ at s_0, and is then updated in each iteration by rank-1-updates (Broyden et al. [31]) or rank-2-updates (Oren & Luenberger [224]).

Newton-type schemes have been fully studied (see, e.g., [141], [60]), highlighting the quadratic convergence of the Newton-method in the solution neighborhood. However, the algorithm may not converge at all, or not to the solution closest to the initial guess. Generally, the Jacobian J may not be available (see Section 6.3) and has to be computed numerically by inverting J^{-1} (which can be easily calculated) numerically in each iteration. This, on the one hand, increases the computation time, and on the other hand may cause additional convergence problems if $J^{-1}(s_k)$ is close to being singular. An orientation-representation via angles may then produce additional singularities. Hence, there are important reliability problems to be considered when dealing with Newton schemes, especially if they are used for control purposes (for further details see, e.g., [198, Ch. 4]).

A crucial ingredient for the Newton-iteration is a good initial value s_0. Thus, Parikh & Lam [230] propose a neural-network-based hybrid strategy that solves the FKP of a Stewart-Gough platform, where a neural-network concept achieves a starting point approximation for a standard Newton-Raphson iterative procedure. Results show that the hybrid concept outperformes the Newton-Raphson method by saving about 0.01s and two iterations, which is a significant improvement for real-time control. A similar approach is given by Wang & Wan [315], who transform the FKP of a 6 d.o.f. Stewart platform into an optimization problem, i.e., to minimize the difference between the given and the computed lengths of links, given a pose s_0. The optimal solution is gathered by an immune evolutionary algorithm based on information entropy, and then used as initialization for a numerical iterative scheme to improve the precision of the solution.

Other approaches. Wang [319] proposes a numerical method for general Gough-Stewart platforms which derives a linear relationship between small changes in the joint variables and the resulting small motion of the platform. The unique solution is then achieved through a series of small changes in joint variables. Vertechy & Parenti-Castelli [306] present an real-time algorithm which relies on the measurement of the location of three body points and minimizes the influence of measurement errors on the estimate. In comparison to other methods for certain manipulators, the method shows better performance in terms of estimation accuracy and computation time. Further different approaches to the finite displacement problem are for example a self-tuning iterative calculation method for the Stewart-Gough platform presented by Ren et al. [256], a fast online-offline data look-up paradigm proposed by Tarokh [294], or a concept based on the determination of basic regions (see, e.g., Wenger et al. [324]) given by Kamali & Akbarzadeh [138], Kamali & Tootoonchi [139].

Another way of gathering the current solution of the FKP of parallel manipulators is to use optimization metaheuristics. For example, Wang et al. [318] use differential evolution (DE) to obtain a globally optimal solution to the FKP of a 6-SPS parallel manipulator, which is transformed into an optimization problem with objective function

$$\min \sum_{i=1}^{6} |\Theta_i^2 - \Theta_{gi}^2|,$$

where Θ_{gi} denotes the given length of the ith leg and Θ_i the computed length of the ith link obtained from the solution of the inverse kinematics under the condition that the position and orientation of the upper platform are given by the DE-algorithm. The DE algorithm is sensitive to the choice of mutation factor: a smaller mutation factor speeds up convergence but the algorithm may end up in a local minimum, while a larger mutation factor helps to avoid a local solution but decreases the speed of convergence. Chandra & Rolland [53] use hybrid metaheuristics, in particular a combination of genetic algorithms and simulated annealing to solve the FKP of a 3-RPR parallel manipulator. The discussed methods give robust and high quality solutions without facing convergence problems. However, the average optimization time is 3-4s, which is not feasible for real-time applications. Further results are presented, e.g., by Sadjadian & Taghirad [269, 270], or by Zhang & Lei [344] for a 3 d.o.f. redundant robot.

Petuya et al. [239] state a main drawback of numerical techniques involving neural networks: training, testing, and validation for each specific platform demand enormous information from the workspace. Thus, they present a new general procedure to solve the non-linear equations arising in the FKP for spatial mechanisms of any topology. The geometrical-iterative method (GIM) gives a systematic procedure for mechanism modeling and obtainment of the position equation system on the one hand, and a specific numerical procedure for solving the system of equations on the other hand (see also [107, 108]). Any mechanism is modeled as a set of nodes and distance and angular constraints. A search algorithm produces an ordered sequence of constraint equations, which is the system to be solved, and throughout the procedure redundant mechanism constraints are eliminated [238]. The system obtained is an ordered sequence of constraint equations, called "iterative constraint sequence", which has $3N$ equations in $3N$ unknown quantities (for N mechanism nodes) of the form

$$\{f_a^i(x_j, y_j, z_j)\} = \{0\} \quad \forall j = 1, \ldots, n_{node}, \tag{5.13}$$

where a is the label of the constraint equation, i the node unknown in this constraint, (x_j, y_j, z_j) the node coordinates and n_{node} the number of nodes involved in every constraint [239]. The solution algorithm starts with substituting initial values (x_j^0, y_j^0, z_j^0) (not necessarily close to the solution value) for each mechanism node in system (5.13) and obtains

$$\{f_a^i(x_j^0, y_j^0, z_j^0)\} = \{r_a^0\}.$$

Here, $\{r_a^0\}$ is a set of residuals, which appear because the initial values generally do not correspond to the correct coordinates of the pairs in the new position of the mechanism. The constraint equations are applied sequentially in a predetermined order, and each time a new position of node i is obtained, which is used to update the system of constraint equations (5.13). One iteration is finished, when the last geometrical constraint has been used, and thereby the last of the residuals has been converted to zero. This means, that the last constraint applied is accurately verified and all mechanism nodes are closer to the solution. Successive iterations are applied until all residuals are below a given threshold. Then the solution of the finite displacement problem is obtained, where the residuals are used as an indicator of the convergence error. GIM converges globally to the solution and shows competitive results to the Newton-method (used in the multibody approach) with low computation times [239].

Extra sensors. Another possibility is to add sensors, i.e., to have more than n sensors for an n d.o.f. robot. This may allow to compute the current pose much faster, but the hardware is more complex. Additional sensors can be placed either on the passive joints of the manipulator, or on extra passive chains (or between existing chains, respectively). The former positioning may lead to drawbacks for accuracy, the latter risks a reduction of the workspace because of higher risk of leg interference [198].

The correct form of the direct kinematics problem then is to find all local minimizers of

$$\begin{aligned} \min \quad & \|P\Theta - y\| \\ \text{s.t.} \quad & F(\Theta, s) = 0, \quad \|P\Theta - y\| \le \epsilon, \end{aligned}$$

where ϵ is an accuracy threshold [219]. Suitable additional measurements may ensure that the FKP has a unique solution; see, e.g., [26, 27, 55, 72, 187, 229, 237]. An exhaustive literature overview of extra-sensor based methods for the FKP of 6 d.o.f. UPS parallel manipulators has been given by Vertechy & Parenti-Castelli [307]. However, additional sensors may induce some non-trivial problems. So, there is the question about types, locations, and minimal number of sensors required to grant uniqueness of the solution. Furthermore, the compatibility of the accuracy of the computed pose and the desired accuracy for the robot has to be considered, given the measurement errors of the extra sensors [198]. An overview of the maximal number of sensors and the relationship between sensor and pose accuracy for 6-UPS robots are presented in [198].

As a key result for the 6-UPS platforms it was shown by Parenti-Castelli & Di Gregorio [227] that the determination of the current pose of the platform is possible with only one additional angular sensor, if base and platform are both planar. Unfortunately, the method cannot be used for real-time purposes due to the complexity of the involved calculations. But for the two-sensor case the same authors give a real-time algorithm [228]. In addition, Vertechy & Parenti-Castelli [305] present an algorithm for the real-time evaluation of the current end-effector orientation of general parallel spherical wrists. Hesselbach et al. [109] present an approach for the Hexa-robot (see, e.g., [350]) to calculate all sensor configurations which are shown to lead towards analytical solutions of the FKP. More recently, Vertechy & Parenti-Castelli [307] proposed a novel extra-sensor-based method for the FKP-solution of 6 d.o.f. UPS parallel mechanisms of general geometry, based on the algorithms developed in [8, 9]. The method is robust, fast, and accurate since it always provides the actual platform pose in real-time, and the redundant extra-sensor information is used to reduce the influence of the measurement errors on the errors affecting the computed platform pose. The problem can be described as a constrained least-squares problem, for which an acceptable minimizer can be obtained by evaluating the orthogonal polar factor of the solution of the corresponding unconstrained least-squares problem.

Chapter 6

Differential kinematics

6.1 Introduction

While geometric kinematics is concerned with solving the equation of state for either given the state s or joint configuration Θ, differential kinematics studies the relationship between the rate of change of pose and joint configuration of a robot performing a known motion. For mapping the actuator rates in the joint space to the generalized velocity of the platform in the operational space, a linear map is defined, usually termed the (inverse) Jacobian of the manipulator. The Jacobian, velocity, and acceleration, are discussed in the following sections.

6.2 Velocity

Consider that the pose moves along a path

$$s(t) = \begin{pmatrix} r(t) \\ c(t) \end{pmatrix} \tag{6.1}$$

in generalized coordinate space, and let x^p denote the relative position of a fixed point on the moving platform (in the platform coordinate system). Then x^p moves with respect to the base along a curve

$$x^b(t) = Q(t)x^p + c(t) \tag{6.2}$$

in Cartesian space (in the base coordinate system). The parametric form of

$$Q(t) = Q[r(t)]$$

depends on the parameterization used for the pose. Differentiation of (6.2) with respect to time gives the velocity of point x^b,

$$\dot{x}^b = \dot{Q}x^p + \dot{c},$$

and with substituting $x^p = Q(x^b - c)$,

$$\dot{x}^b = \dot{Q}Q^T x^b + \dot{c} - \dot{Q}Q^T c. \tag{6.3}$$

Differentiation of $QQ^T = \mathbb{1}$ gives

$$\dot{Q}Q^T + Q\dot{Q}^T = 0,$$

so that the **angular velocity matrix**

$$\Omega(t) := \dot{Q}(t)Q(t)^T \tag{6.4}$$

is a skew-symmetric matrix, i.e., $\Omega(t)^T = -\Omega(t)$ [6]. Thus, there exists a vector $\omega(t)$ such that (in three dimensions)

$$\Omega(t)x^b = \omega(t) \times x^b. \tag{6.5}$$

The vector $\omega(t)$ is called **angular velocity** vector of a rigid body motion. Now the **platform velocity** v is defined by

$$v := \frac{Dc}{Dt}, \tag{6.6}$$

where

$$\frac{Dx(t)}{Dt} := \dot{x}(t) - \omega(t) \times x(t) \tag{6.7}$$

denotes the **covariant derivative** of a curve $x(t)$ in Cartesian 3-space [219]. Hence, the velocity (6.3) can be written as

$$\dot{x}^b = \omega \times x^b + v. \tag{6.8}$$

The relation

$$p' := \frac{\omega \cdot v}{\omega^2}$$

defines the pitch of the instantaneous spatial displacement, and

$$\|\omega\| = \sqrt{\omega^2}$$

is the magnitude of the angular velocity, i.e., the rotation speed. Note, that both ω and v are independent of the platform coordinate system. Furthermore, the inner products ω^2 and $\omega \cdot v$ are invariant under a change of the base coordinate system, while v^2 is invariant only if $\omega = 0$ [219].

The **twist** or **generalized velocity**

$$\mathbf{w} := \begin{pmatrix} v \\ \omega \end{pmatrix}$$

of the end-effector is composed of its translational and angular velocities. Note that the twist depends on the choice of the operating point, i.e., a specific point of the end-effector of which the Cartesian velocity v is computed [198]. For a robot with at least 2 rotational d.o.f. the twist is not the time-derivative of the generalized coordinates s, as there is no representation of the orientation whose derivatives correspond to the angular velocities [197]. But there usually is a matrix $B(s)$ of rank 6 (and its pseudo-inverse $B(s)^+$) depending on the parameterization such that

$$\mathbf{w} = B(s)\dot{s}, \ \dot{s} = B(s)^+\mathbf{w}. \tag{6.9}$$

6.3 Jacobian and inverse Jacobian matrix

If the pose moves along a path (6.1), the motion is accompanied by a motion of the joint variables along a path $\Theta(t)$ in configuration space. Differentiating the equation of state (4.1) leads to

$$F_s\dot{s} + F_\Theta\dot{\Theta} = 0,$$

where $F_s = F_s(s,\Theta)$ and $F_\Theta(s,\Theta)$ denote the matrices of partial derivatives with respect to s and Θ, respectively, and $\dot{\Theta}$ the vector of **joint rates** or **articular velocities** $\dot{\Theta}_j$. With (6.9) the basic relation

$$F_s(s,\Theta)B(s)^+\mathbf{w} + F_\Theta(s,\Theta)\dot{\Theta} = 0 \tag{6.10}$$

is derived. If $\operatorname{rank} F_sB^+ = 6$, equation (6.10) can be solved for the generalized velocity in terms of the joint velocities and gives

$$\mathbf{w} = J(s,\Theta)\dot{\Theta}.$$

The matrix $J(s,\Theta)$ denotes the **kinematic Jacobian** of the manipulator, which represents (together with its inverse and its transpose) the mapping of both velocities and forces between the actuators and the end-effector [287]. It must be noted that the Jacobian of a robot is despite of the name not a true Jacobian, i.e., it is not the derivative of a vector function.

If (6.10) is solved for the joint rates,

$$\dot{\Theta} = J^{-1}(s,\Theta)\mathbf{w}, \tag{6.11}$$

then $J^{-1}(s,\Theta)$ is called the **inverse kinematic Jacobian** of the manipulator. It is named in analogy to the Jacobian, but is not necessarily invertible. For most parallel robots it is usually easy to establish an analytical form for J^{-1} while it is often impossible to obtain J analytically, or at least related to high computational effort and insufficient precision of the numerical evaluation. Thus, it is more convenient to rely on a numerical inversion method, which will be faster and less sensitive to round-off errors [198].

According to [219], the Jacobian of a serial manipulator with equation of state (4.4) is given by

$$J(s,\Theta) = B(s)F'(\Theta),$$

and for a minimally instrumented separable parallel robot with equation of state (4.5), the inverse Jacobian is given by

$$J^{-1}(s) = F'(s)B^+(s).$$

For manipulators with $n < 6$ d.o.f. it may seem sufficient to determine only the $n\times n$ inverse Jacobian, but as Merlet [198] pointed out, a Jacobian that involves the actuated joint velocities $\dot{\Theta}$ and the full twist of the end-effector may be essential for singularity analysis (see Ch. 8); such a Jacobian is called **full inverse kinematic Jacobian** [198].

6.3.1 Formulation of the (inverse) Jacobian of parallel robots

The (inverse) Jacobian of a manipulator is conventionally defined as the linear map between actuator rates in the joint space and generalized velocity in the operation space. However, the formulation of the Jacobian is not unique, since the generalized velocity of the end-effector depends on the parameterization of the rotation. Furthermore, for manipulators having coupled translational and rotational d.o.f. the Jacobian is composed of non-homogeneous physical units. Thus, the algebraic characteristics such as condition number or singular values of the Jacobian depend on the scaling of the Jacobian. This may cause a serious problem if these characteristics are used for kinematic performance evaluation (see Ch. 7). So, mainly two methods to formulate dimensionally homogeneous Jacobians have been developed, either normalizing all translational elements (e.g., [5, 6, 291]), or by using a point-based method linking the actuator rates with the linear velocities at several points on the end-effector (e.g., [3, 90, 146, 170, 244]). For example, Gosselin [90] derives a Jacobian for serial manipulators that transforms actuator velocities into the linear velocities of two or three points on the end-effector and shows that this Jacobian is invariant under scaling of the manipulator. Another example is the concept of a "characteristic length" proposed by Tandirci et al. [291], to normalize all translational elements in the Jacobian.

In the following, two examples of formulating the inverse Jacobian for parallel manipulators shall be given. For examples of Jacobians of serial manipulators see, e.g., [6, 216]. Note, that the use of the terms "inverse Jacobian" and "Jacobian" varies in publications; here, "inverse Jacobian" refers to the matrix relating the actuator velocites $\dot{\Theta}$ to the generalized velocity of the end-effector $\dot{x}$ in the form $\dot{\Theta} = J^{-1}\dot{x}$.

Example 6.3.1 (Inverse Jacobian of a Gough platform [196, 287].)**.** *Consider a Gough-type platform as shown in Figure 6.1.*

Let x_i^p denote the position of the center of the spherical joint connecting leg i to the platform with respect to the moving frame, then the position of x_i^p with respect to reference (base) frame is given by

$$x_i^b = Qx_i^p + c,$$

where Q is a rotation matrix expressing the orientation of the moving frame relative to the base frame, and c is a translation describing the position of the moving frame relative to the base frame. The vector w_i describing leg i is given by

$$w_i = (l_i + \Theta_i)e_i = c + Qx_i^p - a_i^b, \tag{6.12}$$

$(l_i + \Theta_i)$ denotes the length of leg i (with fixed length l_i and variable length Θ_i), $e_i = \frac{w_i}{\|w_i\|}$ is a unit vector in direction of leg i and a_i^b denotes the position of the center of the U-joint connecting leg i to the base. Using (6.4)–(6.7), differentiation of (6.12) with respect to time yields

$$\dot{\Theta}_i e_i + (l_i + \Theta_i)\dot{e}_i = v + \omega \times x_i^b. \tag{6.13}$$

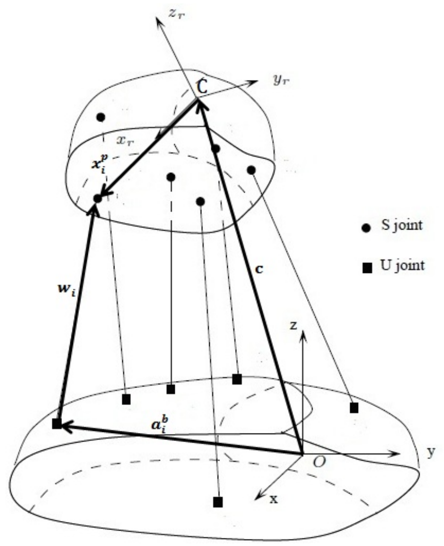


Figure 6.1. A Gough platform with base frame $(O; x, y, z)$ and moving frame $(C; x_r, y_r, z_r)$ (mod. from [196]).

Since e_i *is a unit vector,* $e_i^T \cdot e_i = 1$*, and* $e_i^T \cdot \dot{e}_i = 0$*. So, taking the inner product with* e_i *on both sides of (6.13) gives*

$$\begin{aligned}\dot{\Theta}_i &= v \cdot e_i + (x_i^b \times e_i) \cdot \omega \\ &= \begin{pmatrix} e_i^T & (x_i^b \times e_i)^T \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}.\end{aligned}$$

Thus, the inverse Jacobian of a Gough platform is constituted of the normalized Plücker vectors of the line associated to each leg:

$$J^{-1} = \begin{pmatrix} \frac{(w_i)^T}{\|w_i\|} & (x_i^b \times \frac{w_i}{\|w_i\|})^T \end{pmatrix}.$$

Note that the first three columns of J^{-1} *are dimensionless while the last three columns have units of length. As shown by* HAO & MCCARTHY *[99], the Plücker vector of the link connecting the end-effector to the remainder of the leg appears in the inverse Jacobian of a large class of spatial and planar parallel manipulators [198].*

Example 6.3.2 (Dimensionally homogeneous Jacobian formulation by three end–effector points [146, 244, 245].)**.** KIM & RYU *[146] propose a formulation for a dimensionally homogeneous (inverse) Jacobian of a parallel manipulator by using three points of the end-effector, which can be applied to very general mechanisms with more than 3 d.o.f., as long as the same type of actuator is used for each leg. Consider for example a Gough-type (6-6) platform as shown in Figure 6.1 with planar mobile platform. The pose of the platform with respect to the base coordinate system* $(O; x, y, z)$ *can be described by the motion of three distinct and non-collinear points* T_j *with coordinates* $\overrightarrow{OT_j} = t_j = (x_j, y_j, z_j)^T$ *[90]. Let* $q = (t_1^T, t_2^T, t_3^T)^T$ *the vector defined by the three points. The coordinates of the centers of the joints on the end-effector* (x_i^b) *can be expressed as linear combinations of the three points* *as* $x_i^b = \sum_{j=1}^{3} k_{ij} t_j$ *with* $\sum_{j=1}^{3} k_{ij} = 1$ *for dimensionless constants* k_{ij}*. Now, the vector* w_i *describing leg* i *is given by*

$$w_i = (l_i + \Theta_i)e_i = \sum_{j=1}^{3} k_{ij} t_j - a_i^b, \tag{6.14}$$

where l_i*,* Θ_i*,* e_i *and* a_i^b *are described analogously to (6.12). Differentiating eq. (6.14) and multiplying with* e_i^T *yields*

$$\dot{\Theta}_i = \sum_{j=1}^{3} k_{ij} e_i^T \dot{t}_j,$$

which can be represented in matrix form as

$$\dot{\Theta} = N\dot{q}, \tag{6.15}$$

where $N = J^{-1}$ *is a dimensionless* 6×9 *matrix with* $J_{ij}^{-1} = (k_{ij} e_i^T)$*, and* $\dot{q} = (\dot{t_1}^T, \dot{t_2}^T, \dot{t_3}^T)^T$ *is the vector of generalized end-effector velocities. Note, that this inverse Jacobian is an actual Jacobian as it is a matrix of partial derivatives of Cartesian coordinates with respect to the joint variables.*

For other parallel mechanisms with n d.o.f. ($n \leq 6$), a similar procedure can be applied, which leads to a general ($n \times 9$) inverse Jacobian $J^{-1} = M^{-1}N$, where M is a diagonal matrix and N is a matrix of the form as in (6.15). However, since for an n d.o.f. manipulator ($n \leq 6$) at most only n terms are independent and so dependent motions are included, the physical significance of the singular values (which are relevant for dexterity analysis, see Ch. 7) of such a Jacobian is not evident [146]. Pond & Carretero [244] further the proposed approach and obtain an $m \times n$ dimensionally homogeneous Jacobian, mapping the m actuator velocities Θ_i to n independent end-effector velocities. If the number of actuators equals the desired degrees of mobility, i.e., $m = n$ (as in most cases), the proposed inverse Jacobian matrix is square. The inverse Jacobian proposed in [244] is derived as $J^{-1} = J'P$ where J' is the $m \times 9$ inverse Jacobian of [146] and P a $9 \times n$ constraining matrix, which is derived from differentiating relations between n selected independent terms and the dependent terms of the end-effector with respect to the independent terms. This matrix relates the actuator velocities to the $n \times 1$ vector of independent end-effector velocities. This formulation of the Jacobian may be used reliably for calculating dextrous regions of the workspace, or several dexterity indices as the condition number or singular values (see Ch. 7 [244, 245].

Altuzarra et al. [3] develop a method for deriving a point-based, dimensionless, and comprehensive inverse Jacobian by presenting a novel numerical formulation of the velocity equation, which is to be compiled by general-purpose software and includes all kinematic variables of the mechanism. The method uses a set of kinematic restrictions applied to characteristic points of the mechanism and can be applied to spatial mechanisms of any topology. Recently, Huang et al. [114]

proposed a "generalized Jacobian", which makes it possible to integrate velocity, accuracy, and stiffness modeling of lower mobility serial and parallel manipulators with $n \leq 6$ d.o.f. into a unified mathematical framework. The method uses screw theory and the orthogonal and dual properties of wrench/twist subspaces and is used for a standardized and computational effective method for deriving a dimensionally homogeneous $n \times n$ Jacobian of parallel manipulators [170].

6.3.2 Tasks

In analogy to the FKP and IKP, the **forward instantaneous kinematic problem (FIKP)** is defined as determining the twist (and the passive joint velocities) as functions of the active joint velocities; the **inverse instantaneous kinematic problem (IIKP)** then is to determine the active (and passive) joint velocities as functions of the twist. As the twist depends on the choice of the operating point, any relation involved in FIKP and IIKP will as well be affected by the location of this point [198]. As the inverse Jacobian (6.11) may usually be determined analytically, the actuated joint velocities can be computed directly. For calculating the twist, one can use for example a numerical inversion algorithm or a linear equations solver to get the twist from (6.11); another method is to use a quasi-Newton Scheme

$$\begin{pmatrix} v_{k+1} \\ \omega_{k+1} \end{pmatrix} = \begin{pmatrix} v_k \\ \omega_k \end{pmatrix} + J_0 \left(\dot{\Theta} - J^{-1} \begin{pmatrix} v_k \\ \omega_k \end{pmatrix} \right), \tag{6.16}$$

where J_0 is the kinematic Jacobian calculated for a given pose [198].

The FIKP was solved, e.g., by Shi & Fenton [278], based on the velocities on three non-collinear points of the end-effector. More recently, Lu & Hu [173] presented a uni- and simplified approach for the forward/inverse velocity and acceleration analysis of (limited) n d.o.f. parallel manipulators with linear actuators.

If the whole workspace is considered to find the worst case bounds on either generalized velocity or joint rates, the FIKP and IIKP can be formulated as global optimization problems as ([219])

FIKP$_{global}$: Given bounds on the joint rates, find worst case bounds on the generalized velocity over a given type of workspace.

IIKP$_{global}$: Given bounds on the generalized velocity, find worst case bounds on the joint rates over a given type of workspace.

The solution of FIKP$_{global}$ provides the range of velocities that the robot can reach, while the solution of IIKP$_{global}$ allows the determination of the needed actuators for performing the task in the designing process [219]. The FIKP is also important for robot position control and trajectory generation.

Huynh & Arai [125] analyze the maximum velocity of a general 6 d.o.f. manipulator in operational space using a (graphical) method based on the intersection of hyperplanes which are defined by the algebraic inequalities describing the constraints on the kinematic model, and assuming that all articular velocities are bounded by the same limit, i.e., $|\dot{\Theta}_i| \leq \rho$ for all $i \in \{1, \ldots, 6\}$.

In the late 1990s, Merlet presented several algorithms which deal with certain problems related to the FIKP and IIKP (see also [198, Ch. 5]). In [189] he described an algorithm, which finds all possible parallel geometries such that the end-effector may perform a given translational velocity for any location of the end-effector in a given workspace, with the constraint that the actuator velocities are always lower than given bound. Another algorithm for efficiently computing the articular velocities necessary to perform a given translation/orientation velocity of the end-effector in the translational workspace is presented in [190]; and furthermore, in [192] Merlet proposes an efficient algorithm, which takes as assumption that the orientation of a parallel manipulator is constant over a given volume. It then computes, with a guaranteed error, the maximal and minimal articular velocities so that whatever the location of the end-effector is in the volume, it may perform a motion in a given Cartesian/angular velocity. The proposed algorithm is much faster and safer than the classical discretization method [192]. However, solving the general case is still an open problem [198]. More recently, an interval analysis-based algorithm was presented in [203], which allows the calculation of the maximal and minimal **full independent velocity** $\dot{X}_i^M$ and $\dot{X}_i^m$ with

$$\dot{X}_i := \sum_{j=1}^{n} |J_{ij}|$$

for all poses over an (almost) arbitrary workspace up to a given accuracy. However, the algorithm is computer-intensive and an efficient calculation of $\dot{X}_i^m$, $\dot{X}_i^M$ is still an open problem [198]. In a comparative study of two 3 d.o.f. parallel manipulators done by Chablat et al. [52], a method is described which allows to prove formally whether the velocity amplification factors in a subpart of of the Cartesian workspace remain within a predefined range or not.

6.4 Acceleration

To obtain similar relations for the joint and platform accelerations, the obtained equations from the velocity-analysis must be again differentiated. The acceleration of point x^b is given by differentiating (6.8) with respect to time to get

$$\ddot{x}^b = \dot{\omega} \times x^b + \omega \times x^b + a,$$

where

$$a := \frac{Dv}{Dt} = \ddot{c} - \dot{\omega} \times c - \omega \times \dot{c}$$

is the **platform acceleration** and $\dot{\omega}$ denotes the **angular acceleration** and is the axial vector of the **angular acceleration matrix** $\Lambda = \dot{\Omega} + \Omega^2$, i.e.,

$$\dot{\Omega}x = \dot{\omega} \times x.$$

Differentiation of (6.10) yields relations between the **joint acceleration** $\ddot{\Theta}$ and the **generalized acceleration** or **twist rate** $\begin{pmatrix} a \\ \dot{\omega} \end{pmatrix}$. In particular, for parallel manipulators the relation

$$\ddot{\Theta} = J^{-1} \begin{pmatrix} a \\ \dot{\omega} \end{pmatrix} + \dot{J^{-1}} \begin{pmatrix} v \\ \omega \end{pmatrix} \tag{6.17}$$

may be obtained directly by differentiating (6.11). In [6] and [198] a few examples for the acceleration analysis of serial and parallel manipulators, respectively, are given.

Example 6.4.1 (Acceleration analysis of a Gough platform)**.** *In Example 6.3.1, the joint rates $\dot{\Theta}$ of a Gough platform are given by*

$$\dot{\Theta} = \begin{pmatrix} e_i^T & (x_i^b \times e_i)^T \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix}.$$

Thus, the joint acceleration $\ddot{\Theta}$ given by (6.17) are

$$\ddot{\Theta}_i = \begin{pmatrix} \dot{e}_i^T & \dot{u}_i^T \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} + \begin{pmatrix} e_i^T & (x_i^b \times e_i)^T \end{pmatrix} \begin{pmatrix} a \\ \dot{\omega} \end{pmatrix},$$

where $\dot{u}_i = \dot{x}_i^b \times e_i + x_i^b \times \dot{e}_i$.

For the generalized acceleration, similar tasks as for the generalized velocity can be posed [219].

Chapter 7

Dexterity and Error Bounds

7.1 Introduction

The term **dexterity**[1] refers to the ability of the manipulator to arbitrarily change its position and orientation, or apply forces and torques in arbitrary directions [287]. In other words, dexterity summarizes the flexibility and accuracy with which the end-effector is placed in a given pose. The flexibility is quantified by the dexterous workspace (see Section 4.4). The positioning accuracy of a manipulator depends on various factors, as sensor errors, manufacturing errors or thermal errors. A short summary of several results concerning geometrical and thermal errors may be found in [196, 198]; most recently, a review of error sources and error compensation methods for serial and parallel manipulators was given by [181]. Usually, joint sensor errors are the largest source of error [196]. To quantify the accuracy, **dexterity indices** may be defined in numerous ways. These are widely used for the design of manipulators (e.g., [88, 90, 143, 205, 268]), but there are some concerns about the reliability of certain indices [197], due to the impossibility to define a single invariant metric for the special Euclidean group [268] (see Section 7.3 below). In this chapter, the modeling of the accuracy of a pose is discussed as well as several ways of measuring dexterity and tasks concerning accuracy and error bounds.

The joint configuration Θ is given by more or less accurate measurements $y \approx P\Theta$. Since it is based on these measurements, the pose of the robot, as estimated by the forward kinematics, is inaccurate. Thus, the manipulator accuracy Δs (or the error on the positioning of the robot) is induced by errors $\Delta\Theta$ on the sensor measurements (and joint sensor errors are usually the largest source of positioning errors [196]). These errors are approximately (in first order) linearly related by a relation of the form

$$A(s,\Theta)\Delta\Theta = B(s,\Theta)\Delta s, \tag{7.1}$$

[1]Note, that in several publications the term *dexterity* is synonymously used for the relative error amplification described by the condition number of the Jacobian.

with A, B matrices depending on the state of the robot. For a serial manipulator (7.1) has the form

$$\Delta s = J(s, \Theta)\Delta\Theta, \tag{7.2}$$

where J denotes the state-dependent kinematic Jacobian of the robot; for a Gough platform (7.1) is given by

$$\Delta\Theta = J^{-1}(s, \Theta)\Delta s, \tag{7.3}$$

where J^{-1} is the pose-dependent inverse Jacobian matrix of the robot. The determination of the (inverse) Jacobian is addressed in Section 6.3.

7.2 Modelling the accuracy of a pose

The pose of a robot depends on the chosen coordinate system for base and platform and the representation chosen for the rotation. Thus, the way of modelling the accuracy of a pose shall be discussed more detailed. The **geometric pose error** is given by

$$\Delta s = \begin{pmatrix} \Delta r \\ \Delta c \end{pmatrix}; \tag{7.4}$$

any single or aggregated error measure can be expressed as a function $\epsilon(\Delta s)$, depending on (7.4). The geometric pose errors may be expressed in terms of the error characteristics of the measurements. In robotics, a commonly used representation of sensor accuracies is by worst case error bounds. Therefore, the measurement equation (4.2) is replaced by an inclusion

$$y = P\Delta\Theta + \Delta y, \quad \Delta y \in \mathbf{d}, \tag{7.5}$$

where $\mathbf{d} = [-\bar{d}, \bar{d}]$ is a tiny box consisting of all measurement errors Δy with $|y_i| \leq \bar{d}_i$ for all i. The equation of state $F(s, \Theta) = 0$ is satisfied by the true pose s and joint configuration Θ. But the measured joint configuration $\widetilde{\Theta}$ and the pose $\tilde{s}$, which is calculated by direct kinematics from the measured data, satisfy instead

$$F(\tilde{s}, \widetilde{\Theta}) = 0, \quad P\widetilde{\Theta} = y,$$

which can be written in linearized form as

$$\begin{aligned} 0 &= F(\tilde{s}, \widetilde{\Theta}) \\ &\approx F(s, \Theta) + F_s(s, \Theta)(\tilde{s} - s) + F_\Theta(s, \Theta)(\widetilde{\Theta} - \Theta), \end{aligned} \tag{7.6}$$

where $F_s(s, \Theta)$ and $F_\Theta(s, \Theta)$ denote the matrices of partial derivatives with respect to s and Θ, respectively. The pose $\tilde{s}$ derived from inaccurate sensor measurements consists of a displacement part $\tilde{c}$ and an orientation part $\tilde{r}$. These may be expressed in the chosen parameterization for the rotation. In the following, a few details of modelling pose errors using the quaternion parameterization shall be given (for details see [219]). Note that similar formulas can be derived for other representations.

Let $Q[r]$ be the rotation of the platform as described in Section 4.3 and $Q[\tilde{r}]$ the rotation derived from inexact sensor readings with $\tilde{r}$ close to but different from r. The multiplication of $Q[r]^T$ and $Q[\tilde{r}]$ again gives a rotation, which has the form $Q[\Delta r]$ for some vector Δr. To derive a formula for $\tilde{r}$, the equation

$$Q[r]^T Q[\tilde{r}] = Q[\Delta r]$$

is multiplied by $Q[r]$, which results in

$$Q[\tilde{r}] = Q[r]Q[\Delta r]. \tag{7.7}$$

Using the product formula (4.12) and (7.7), orientation errors can be represented by

$$\tilde{r} = r \oplus \Delta r. \tag{7.8}$$

Thus, it can be seen from the nonstandard addition formula, that rotations in three dimensions usually do not commute (unlike in two dimensions, where angles simply add). The displacement error Δc can be written as

$$\Delta c = Q[r]^T(\tilde{c} - c).$$

Multiplying again by $Q[r]$, $\tilde{c}$ can be expressed in terms of c and Δc as

$$\tilde{c} = c + Q[r]\Delta c. \tag{7.9}$$

For the analysis of small errors only, higher order terms may be dropped and (7.8) and (7.9) may be written in linearized form as

$$\begin{aligned} \tilde{r} &\approx r + e^{-\frac{1}{2}}(r_0 \Delta r + r \times \Delta r) \\ \tilde{c} &\approx c + Q[r]\Delta c. \end{aligned} \tag{7.10}$$

(7.10) may then be written in the form

$$\tilde{s} \approx s + E(s)\Delta s, \quad E(s) = \begin{pmatrix} E[r] & 0 \\ 0 & Q[r] \end{pmatrix}, \tag{7.11}$$

where

$$E[r] = \frac{1}{\sqrt{e}} \begin{pmatrix} r_0 & -r_3 & r_2 \\ r_3 & r_0 & -r_1 \\ -r_2 & r_1 & r_0 \end{pmatrix}.$$

Using (7.11) and $\widetilde{\Theta} = \Theta + \Delta\Theta$, (7.6) can be written as

$$F_s(s,\Theta)E(s)\Delta s + F_\Theta(s,\Theta)\Delta\Theta \approx 0.$$

Note that $P\Delta\Theta = P\widetilde{\Theta} - P\Theta = \Delta y \in \mathbf{d}$. Thus, the errors satisfy (in the linear approximation) the constraints

$$DF(s,\Theta)\Delta s + F_\Theta(s,\Theta)\Delta\Theta \approx 0, \quad P\Delta\Theta \in \mathbf{d}, \tag{7.12}$$

where

$$DF(s,\Theta) = F_s(s,\Theta)E(s). \tag{7.13}$$

Equation (7.12) is the basic (linearized) formula for the quaternion representation relating known measurement errors $\Delta\Theta$ to pose errors Δs. It also holds in other representations, if the expressions in (7.11) are replaced by the corresponding expressions for the chosen representation. As an artifact of the normalization used for the quaternion representation, equation (7.12) is ill-defined at $r_0 = 0$ and numerically unstable at small values of r_0: on the one hand, due to the square root in the computation of r_0 the quaternion representation has a gradient singularity at $r_0 = 0$; on the other hand, the matrix $E(s)$ is singular at $r_0 = 0$, since $\det E(s) = \det E[r] = r_0(\sum_{i=0}^{3} r_i^2) = r_0 e$. This problem can be solved by treating r_0 as an additional parameter constrained by $r_0^2 + r^2 = e$. By writing $\tilde{r} = r + \Delta r$, it follows from equation (7.10) that

$$\begin{aligned}
\tilde{r}_0^2 &= e - \tilde{r}^2 \\
&\approx e - r^2 - 2e^{-\frac{1}{2}} r_0 r \cdot \Delta r \\
&= r_0^2 - 2e^{-\frac{1}{2}} r_0 r \cdot \Delta r \\
&\approx (r_0 - e^{-\frac{1}{2}} r \cdot \Delta r)^2,
\end{aligned}$$

so,

$$\tilde{r}_0 \approx r_0 - e^{-\frac{1}{2}} r \cdot \Delta r. \tag{7.14}$$

Equation (7.14) now can be written in the form

$$\tilde{r}_0 \approx r_0 - E_0(s)^T \cdot \Delta s,\ E_0(s) = \begin{pmatrix} e^{-\frac{1}{2}} r \\ 0 \end{pmatrix}.$$

Concluding, by writing the equation of state (4.1) more carefully as

$$F(r_0, s, \Theta) = 0,$$

the constraints (7.12) are obtained by replacing (7.13) with the numerically stable expression

$$DF(s, \Theta) = F_s(r_0, s, \Theta) - F_{r_0}(r_0, s, \Theta) E_0(s)^T.$$

Again, these formulas also hold for other representations, if the expression for $E_0(s)$ is replaced by corresponding expression for these representations [219].

7.3 Dexterity indices

As mentioned before, for applications several performance indices may be computed to measure the dexterity of a manipulator. Most discussed are "classical" dexterity indices: the *condition number* (of the Jacobian) [271] and the so-called *manipulability index* [338]. Due to their suffering from certain drawbacks and missing reliability for parallel robots in particular [197] (see below), other ways for measuring the accuracy are searched. It was shown that, especially for parallel manipulators, extremal positioning errors are more reliable than the condition number; however, to compare two different designs in terms of accuracy, mean value and variance will also be needed [196]. Just recently, *kinematic sensitivity* indices for rotation and translational displacement [39] or the *power manipulability* [177] have been introduced.

The dexterity indices discussed in the following sections evaluate (in a more or less reliable way) the sensitivity of a robots performance to variations in its actuated joints, but they are not suitable for other types of uncertainties, such as variations in geometric parameters. Thus, appropriate sensitivity measures were proposed by Caro et al. [40, 41] to evaluate the sensitivity of the pose of the end-effector of a Orthoglide 3-axis manipulator to variations in its design parameters and to compare planar parallel manipulators of the same architecture (3-RPR). Recently, Binaud et al. [17] introduced a methodology for 3 d.o.f. planar parallel manipulators to obtain the sensitivity coefficients of the pose of the moving platform to variations in their geometric parameters and actuated variables to compare manipulators of different architecture. Additionally, two aggregate sensitivity indices have been determined, one related to the orientation and the other one to the position of the moving platform. The described method is able to take the sensitivity to joint clearances and flexibility in the revolute joints into account [17].

7.3.1 Error bounds depending on the norm

In practice, it is assumed that the measurement errors are bounded, which consequently poses a bound on the positioning errors. Without loss of generality, usually a boundary value of 1 is chosen, so

$$\|\Delta\Theta\| \leq 1, \tag{7.15}$$

which yields

$$\Delta s^T J^{-T} J^{-1} \Delta s \leq 1. \tag{7.16}$$

The geometrical interpretation of (7.16) differs depending on the choice of the norm (see Figure 7.1 for the 2D case). By choosing the Euclidean norm, (7.15) represents a hyper-sphere in the joint error space which is mapped into the so-called **manipulability ellipsoid**. If the infinity norm is chosen, the absolute values of the joint errors are independently bounded by one and thus are restricted to lie in a hyper-cube within the joint error space, which is mapped into a convex polyhedron, the so-called **kinematic polyhedron** [197]. As Merlet pointed out, the infinity norm provides a more appropriate and realistic mapping due to the independency of the joint error bounds and additionally, computations involving a polyhedron can be done more easily than involving ellipsoids [197, 198].

The two indices described in the following paragraphs are used for both serial and parallel manipulators. For simplification of the notation, their definition will be introduced using the Jacobian J only.

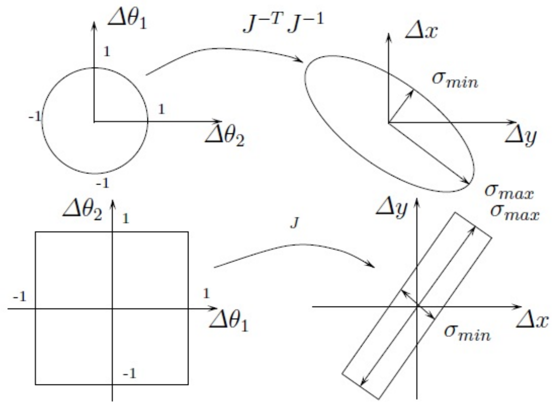


Figure 7.1. The mapping between the joint error space and the generalized coordinates error space induced by $J^{-T}J^{-1}$ according to the Euclidean norm (on top) and the infinity norm (on bottom) for the 2D case [197].

7.3.2 The manipulability index

Manipulability is defined as

$$\mu = \sqrt{\det J^T J}$$

for a square, non-singular matrix J by Yoshikawa [338] for serial manipulators and has been adopted as a dexterity measure for parallel robots as well. It is a local performance measure, since the Jacobian is pose dependent. Geometrically, the manipulability is (up to a proportionality constant) the volume of the ellipsoid described above. The only use of the manipulability as dexterity index may cause problems in automatic synthesis, as it may lead to manipulators with distorted geometries [39]. Further, a limitation of this index may be that it is independent from the position of the operating point on the end-effector, and thus cannot distinguish between a large and a small end-effector [39, 143]. Nevertheless, it is used for the dexterity analysis of several manipulators, e.g., in [166] for a 3-PRS parallel manipulator.

7.3.3 The condition number as error amplification factor

The error amplification factor in a system (7.3), or (7.2), expresses how a relative error in the joint coordinates Θ gets multiplied and yields a relative error in the pose s [197]. Salisbury & Craig [271] proposed the condition number

$$\kappa = \|J\|\|J^{-1}\|,$$

as a local performance index, where J^{-1} denotes either the inverse of the Jacobian or a pseudo-inverse (e.g., Moore-Penrose generalized inverse) [39]. The condition number gives an upper and a lower bound for the relative error amplification ([39]),

$$\frac{1}{\kappa}\frac{\|\Delta\Theta\|}{\|\Theta\|} \leq \frac{\|\Delta s\|}{\|s\|} \leq \kappa\frac{\|\Delta\Theta\|}{\|\Theta\|}.$$

Thus, a minimization of κ should minimize the maximum relative error amplification, but as a decreasing κ increases the lower bound on $\|\Delta s\|/\|s\|$, it does not provide the overall relative error amplification. The condition number is dependent on the choice of the matrix norm; the most used norms are the 2-norm and the Frobenius-norm [197]. In the former case, the condition number of the Jacobian is given by

$$\kappa(J) = \frac{\sigma_{max}}{\sigma_{min}} = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}},$$

where σ_{max}, σ_{min} denote the largest and smallest singular value of J, respectively, and λ_{max}, λ_{min} the largest and smallest eigenvalue of $J^T J$, respectively. If the Frobenius-norm is used, the condition number is given by

$$\kappa(J) = \frac{\sum \lambda_i^2}{\prod \lambda_i^2},$$

where λ_i are the eigenvalues of $J^T J$ [197].

Merlet [197] critically evaluated the efficiency of the condition number for accuracy evaluation for a Gough platform. He concluded for various choices of the matrix-norm, that none of the tested condition numbers exhibits a completely coherent behavior with respect to the accuracy of the robot. Thus, the use of the condition number must be carefully considered. However, it is used in various publications for describing the dexterity and/or optimizing the design of a robot (see, e.g., [172, 245, 246]). For serial manipulators, Kucuk & Bingul [151] proposed a combined performance index, which is a combination of the manipulability index and the condition number; it is used for workspace optimization via sequential quadratic programming for spherical 3-link manipulators.

A pose of a robot with $\kappa = 1$ is defined as **isotropic pose**, and a robot which has only isotropic poses is called an **isotropic robot** [197, 198]. Designing a parallel manipulator that is isotropic in a pose or over its full workspace is often considered as a design objective [73, 144, 297, 341].

7.3.4 Sensitivity indices

As discussed by several authors (e.g., [69, 169]), the described indices do not apply to Jacobians involving both translational and rotational velocities. Thus, several attempts have been made to formulate a dimensionally homogeneous Jacobian by using either multiple end-effector velocities [3, 90, 114, 146, 170, 244–246], or a normalization procedure of different kinds [143, 168, 338]. However, Cardou et al. [39] pointed out, that both approaches require some kind of arbitrariness or uncertainty (e.g., in choosing points or weighting factors), which in particular makes any comparison between different robotic manipulators difficult. Thus, the authors recently proposed two distinct indices for the rotation and the point-displacement sensitivity. The considered problem is stated as finding the actuator displacements Δs that generate actuator displacements of a fixed magnitude $\|\Theta\|_q$, while yielding global extrema of $\|\Delta r\|$ and $\|\Delta p\|$ [39]. The maximum solutions of this problem ,

$$\sigma_{r,q} := \max_{\|\Theta\|_q} \|\Delta r\|_q, \tag{7.17}$$

$$\sigma_{p,q} := \max_{\|\Theta\|_q} \|\Delta c\|_q, \tag{7.18}$$

give tight upper bounds on the amplitudes of rotations and operating-point displacements, and are defined as the **maximum rotation sensitivity** and **point-displacement sensitivity**, respectively. If for example $q = \infty$, i.e., the infinity-norm is chosen in (7.17) and (7.18), the sensitivity-indices for a parallel manipulator may be found by solving the optimization problem

$$\begin{aligned} \max_{\Delta s} \quad & \|\Delta \star \|_\infty \\ \text{s. t.} \quad & \|\Theta\|_\infty = \|J^{-1}\Delta s\|_\infty = 1, \end{aligned} \tag{7.19}$$

where $\star = r$ or $\star = p$, respectively, and under the assumption that all actuated joint assemblies are the same. Geometrically, the constraint forms the envelope of a polytope

$$\mathcal{P}_6(J^{-1}) := \{x \in \mathbb{R}^6 | \|J^{-1}x\|_\infty = 1\}.$$

Finding the supporting hyperplanes of a polytope may be cast into a linear program. Hence, as the objective function varies only along three independent directions, computing the solution of (7.19) amounts to finding the solution of three linear programs each, and choosing the maximal objective-function value among them [39].

The proposed sensitivity indices are “nothing but another method” [39], but have some characteristics not present in other indices. So they have a clear and definite physical meaning, rely on the Jacobian only, are unit-consistent, depend on the choice of the operating point (contrary to the manipulability), and apply to any uniformly actuated manipulator, regardless if redundantly actuated or not [39]. Saadatzi et al. [268] use these sensitivity indices for the optimization of 3-R$\underline{\text{P}}$R planar parallel mechanisms.

7.3.5 Global dexterity

If the dexterity of a robot over a given workspace $\mathcal{W}$ shall be evaluated, generally a **global conditioning index** (GCI) is used as proposed by Gosselin [88]:

$$\text{GCI} = \frac{\int_{\mathcal{W}} \frac{1}{\kappa} \,\mathrm{d}\,\mathcal{W}}{\int_{\mathcal{W}} \mathrm{d}\,\mathcal{W}},$$

where κ denotes the condition number (see also Gosselin & Angeles [92]). The GCI can be used for the optimal design of a robot [172], but a main problem is its calculation. Usually, the workspace is sampled using a regular grid, $1/\kappa_i$ is computed at each node N_i, and the GCI is approximated by

$$\mathrm{GCI}_{approx} = \frac{\sum_{i=1}^{n} \frac{1}{\kappa_i}}{n}.$$

The computational effort increases exponentially with respect to the number of d.o.f. of the robot and additionally, this method does not provide a bound on $|\,\mathrm{GCI} - \mathrm{GCI}_{approx}\,|$ [197]. Another evaluation method of the GCI may be using (quasi) Monte-Carlo integration. But a certified evaluation of the GCI is an open problem and will probably be computer intensive [197].

7.4 Errors and Global Optimization

As mentioned, most dexterity indices are not very adequate for parallel manipulators [196], or provide at least not a precise knowledge of the extremal pose errors [30]. A better way is the determination of the maximal positioning (and orientation) errors over a prescribed workspace (and further its mean value and variance [196]), or at a given configuration with known joint-sensor errors [30, 196]. Several tasks concerning the maximal pose error and the accuracy of a robot can now be formulated [219]:

- Find the maximal pose error over a specified workspace given bounds (7.5) on the sensor accuracy for one ore more error measures $\epsilon(\Delta s)$.
- Find the sensor accuracy needed, to guarantee a specified maximal pose error over a given workspace.
- Find the state (s, Θ) with the smallest maximal pose error.
- Find the set of states in which Δs lies within a specified region.
- Find the average value (and the variance) of the maximal pose error over the workspace, given a probability measure on the workspace (e.g. uniform).

The resulting optimization problems shall be discussed more detailed, following [219]. Consider the restricted case that all joint variables are measured, i.e., $P = \mathbb{1}$ and $\dim F \geq \dim s = 6$. In the generic case, i.e., if $\operatorname{rank} DF(s, \Theta) = 6$, (7.12) can be solved for Δs:

$$\Delta s \approx -DF(s, \Theta)^{+} F_{\Theta}(s, \Theta) \Delta\Theta, \tag{7.20}$$

where DF^{+} denotes a pseudo-inverse of DF. So, in a fixed state, the geometric pose errors are proportional to the measurement errors. However, since a pseudo-inverse is expensive to compute, it is preferable to solve instead of (7.20) an optimization problem for one or more error measures $\epsilon(\Delta s)$. For a robot in a fixed state these are given by

$$\begin{aligned} \max \quad & \epsilon(\Delta s) \\ \text{s.t.} \quad & DF(s,\Theta)\Delta s + F_\Theta(s,\Theta)\Delta\Theta \approx 0, \\ & \Delta\Theta \in \mathbf{d}, \end{aligned} \tag{7.21}$$

or, if nonlinear corrections are significant,

$$\begin{aligned} \max \quad & \epsilon(\Delta s) \\ \text{s.t.} \quad & F(\tilde{s}, \Theta + \Delta\Theta) - F(s,\Theta) = 0 \\ & \tilde{s} = \begin{pmatrix} r \oplus \Delta r \\ c + Q[r]\Delta c \end{pmatrix}. \end{aligned} \tag{7.22}$$

Both (7.21) and (7.22) are global optimization problems, since worst case errors are required. If the worst case accuracy over some workspace shall be computed, the formulations of (7.21) and (7.22) still hold; the constraints defining the workspace have to be added and s and Θ have to be treated as additional variables. However, the size of the search space is much bigger, so the difficulty increases [219].

Kim & Choi [145] provided methods for the forward and inverse error bound analyses (FEBA and IEBA)of the Stewart platform. Given the error bounds of the joints, the FEBA is used to find the error-bound on the position of the platform. The IEBA, on the other hand, determines the error bounds on the joints for a given error bound on the end-effector pose. The latter provides a designer with a way to calculate tolerance limits of the joints and is therefore important. Kim & Choi derived two eigenvalue problems corresponding to the FEBA and IEBA, where the eigenvalues from both analyses are reciprocal to each other.

Wang & Ehmann [317] present an error analysis for the Stewart platform, taking location errors and imperfect motion in the ball-joints as well as errors in the actuated joints, i.e., in the leg lengths, into account. They propose a first and second error model and provide a software package (written in `Mathematica`) which allows, for a given orientation of the platform, the determination of the workspace regions where the positioning error is larger than a given threshold. Jelenković & Budin [131] present an online error analysis tool[2] based on on the model presented by Patel & Ehmann [232], which computes the worst case accuracy for 6-UPS and 6-PUS hexapods. Pott & Hiller [248] give an approach to estimate the first-order influence of geometric errors on target quantities by considering the force transmission of the manipulator.

As pointed out by Merlet & Daney [203], several authors treated the problem of finding positioning errors of a given robot at some specific location within the workspace (e.g., [145]; for further references see [203, p. 942]). Thus, in [203] not only specific poses are considered but the whole workspace and further, the *error synthesis* problem is addressed, i.e., finding the geometries of a robot that have at

[2]available at `http://hexapod.zemris.fer.hr/engleski/Online_en.htm`)

least a required accuracy [203]. As a by-product variant of the interval analysis-based algorithm it is possible to compute the maximal positioning errors of a given (6 d.o.f.) robot in a specified workspace up to a predefined accuracy. However, the algorithm has a large computation time. MERLET [196] addresses a verification problem. Given the geometry of a robot (possibly with uncertainties), bounds on the sensor errors, and a prescribed workspace, the proposed algorithm checks if the maximum of the positioning errors does not exceed some threshold. As a result, the workspace is identified as safe, if for any pose within the workspace the positioning error will be lower than the threshold, or unsafe, if there is at least one pose with a positioning error exceeding the threshold (or, in some cases, the algorithm is unable to decide). Here, the main objective is not to calculate maximal positioning errors per se but to verify that they stay below some threshold. However, this method is relatively difficult to implement and gives no information on the evolution of the accuracy of the manipulator within the workspace [29]. Moreover, MERLET [196] points out, that in addition to the maximal pose errors it may be important to compute their mean value and the variance; but it is a challenging task to define an algorithm to compute those, at least approximately with a guaranteed error.

A geometric approach is given in [30, 339] to compute the exact value of the pose error of a 3 d.o.f. planar parallel manipulator at a given nominal position. YU et al. [339] introduce the method for 3-P̲RP planar parallel manipulators, replacing the existing dexterity maps by maximum position/orientation error maps. BRIOT & BONEV [30] generalize this work for all fully-parallel 3 d.o.f. planar parallel manipulators and prove that, when sufficiently far from singularities, the local maximum orientation/position errors can occur only when at least two active joint inputs suffer a maximal error. WANG et al. [314] recommend an exact error bound contrary

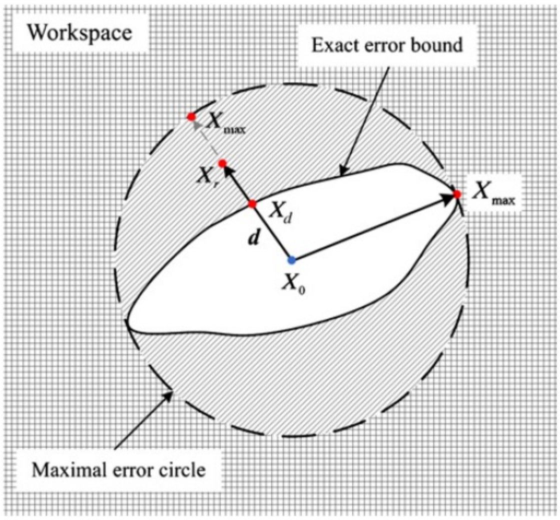


Figure 7.2. Comparison of the exact output error X_d to the maximal error X_{max} evaluation: along the operating direction d with allowed maximal output error X_r, the accuracy performance satisfies the required operating precision for $X_d(< X_r)$, but for $X_{max}(> X_r)$ not [314].

to a maximal error bound, which represents the extremal value of the maximal output errors in each direction, and thus may be limited to evaluate the accuracy performance in some case (see Fig. 7.2). The forward error bound analysis for an n d.o.f. manipulator at a given nominal configuration s_0 is stated as constraint optimization problem and then the exact output error bound, which can be indicated by a closed curve or surface, is computed using the level set method by tracking the propagation of the output error interface under the constraints of the active joints' input errors. The algorithm is quite easy to implement as it only requires the inverse kinematic functions of the manipulator. The proposed method not only provides a prediction of the exact error bound but also the best/worst error transformation direction and the output accuracy information along an arbitrary direction. Thus, it has been proven, that a manipulator may admit completely different accuracy properties along different directions. The method can also be applied to compute the exact error bound over a given trajectory or part of the workspace. For example, the exact error surface of a Delta parallel manipulator was computed, needing 15 iterations and about 226.5 seconds (see Fig. 7.3).

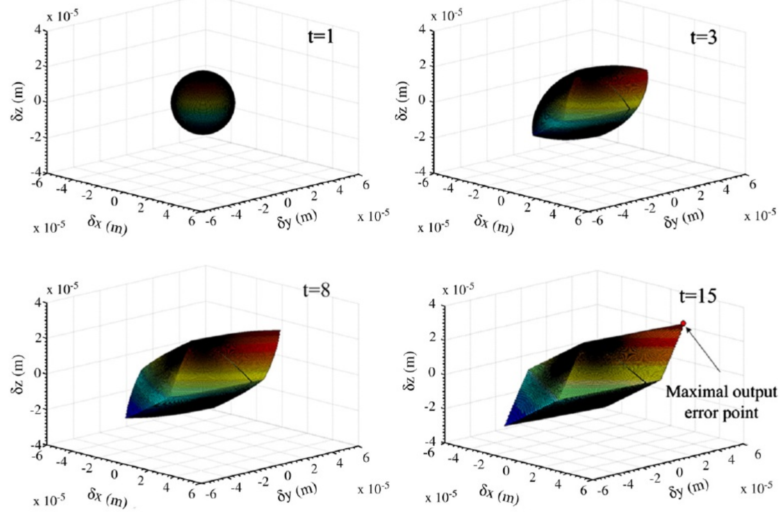


Figure 7.3. Propagation of the output error surface of the Delta manipulator for $t = 1, 3, 8, 15$ iterations [314]

A (so far) open problem is to locate a priori the poses with the worst positioning accuracy [198]: according to Merlet [198] it was shown by Hay & Snyman [104] that for planar robots the worst values of the condition number are obtained for poses on the boundary of the workspace. However, this result relies on numerical calculation and it is an open problem to determine if this is also true for spatial robots, and for the worst positioning error as well.

Another aspect that has to be considered is the influence of joint clearance on the position accuracy of robotic manipulators (see, e.g., [295]). Joint clearance, i.e., the distance between the mating surfaces of a joint, leads to uncertain flawed motions at an arbitrary pose of the mechanism and its effects are highly non-repeatable. For example, Meng et al. [183, 184] propose an efficient error evaluation method for parallel mechanisms which can be used to predict the global maximum pose errors of a clearance-affected mechanism in a prescribed workspace, by formulating the problem as a standard convex optimization problem. Chebbi et al. [54] develop an analytical predictive model for quantifying the pose error of a 3-UPU parallel manipulator as a function of the magnitude of the joint clearance, the external load acting on the platform, and the configuration of the manipulator. Furthermore, an algorithm is given which maps the pose error within the workspace of the robot, and thus allows a determination of the distribution of the error within the workspace, which may have a discontinuity due to the change of the configuration of the robot. More recently, Frisoli et al. [83] present a screw theory based method which computes the worst position accuracy for spatial parallel manipulators with revolute joint clearances and performs a maximization of the pose error function in a mixed analytical-numerical approach, converging to the exact maximum pose error.

Finally, a noteworthy comparison between serial and parallel manipulators regarding the accuracy done by Briot et al. [29] shall be mentioned. It is often claimed that parallel manipulators have a higher accuracy than serial ones (e.g., [54]) because their errors are averaged instead of cumulatively added [29]. A comparison of serial and parallel 2 d.o.f. manipulators seems to indicate that parallel mechanisms are less sensitive to input errors, but the study is too limited to state a general assumption. Furthermore, the authors state that a meaningful comparison between other pairs of serial and parallel mechanisms is virtually impossible.

Chapter 8

Singularities

8.1 Introduction

In this chapter, the notion of *singularities*[1] shall be defined and described in more detail. Singularities are particular poses of the end-effector in which either the forward or the inverse instantaneous kinematic problem becomes undeterminate, and thus, the properties of the mechanism change dramatically. In a singular configuration, the pose of the end-effector is no longer controllable, and forces, imposed on several links, may become very large, which may cause a breakdown of the robot. Therefore, singularity analysis is an important field of study in robotics, it plays a significant role in the design and control of robot mechanisms.

8.2 Singular configurations

The relationship between input (joint variables) and output (end-effector position) coordinates of any manipulator is given by the equation of state (4.1)

$$F(s, \Theta) = 0,$$

where s is a set of parameters coding the pose of the end-effector and Θ describes the joint configuration (including active and passive joint parameters). It allows to formulate a relation of the type (cf. (7.1))

$$A(s, \Theta)\Delta\Theta = B(s, \Theta)\Delta s,$$

where $\Delta\Theta$ denotes the error of the vector of joint positions, Δs the error in the pose of the end-effector, and $A = A(s, \Theta)$ and $B = B(s, \Theta)$ are both configuration dependent $n \times n$ matrices (for a n d.o.f. non-redundant manipulator). For a non-redundant parallel robot with n actuated joints (and n d.o.f.) for a total

[1]An extensive database of publications about singularities in robot kinematics can be found at `http://homepages.mcs.vuw.ac.nz/~donelan/cgi-bin/rs/main`.

of N joints, the equation of state (4.1) consists of a set of N equations in $N+n$ unknowns. A **kinematic singularity** exists if the rank of the kinematic Jacobian of the system is less than N, or equivalently, if there are less than N independent equations. Considering (7.1), that means that one of the matrices A or B becomes singular, i.e., $\det(A) = 0$ or $\det(B) = 0$. Thus, in principle three types of singularities may be distinguished [91]:

(1) A is singular (called **serial** or **type 1** singularity).

(2) B is singular (called **parallel** or **type 2** singularity).

(3) A and B both are singular (called **type 3** singularity).

In principle, at a serial singularity a robot *loses* degree(s) of freedom, while at a parallel singularity degree(s) of freedom are *gained*, i.e., the manipulator can move while the actuators are locked. As for many serial manipulators $B = \mathbb{1}$, a singularity only happens if $\det(A) = 0$. A parallel singularity lies within the workspace and happens in particular for parallel manipulators.

The general description of singularities by Gosselin & Angeles [91] is based on the input and output velocities. Park & Kim [231] applied methods from differential geometry and introduced three basic singularity types. The singularity-taxonomy is independent of the local coordinates used to describe the kinematics, and includes mechanisms that have more actuators than d.o.f. A more general classification was proposed by Zlatanov et al. [348], where not only the input and output velocities, but also the passive-joint velocities are included explicitly. A general manipulator model, in terms of differentiable mappings between manifolds, allows a rigorous mathematical definition of kinematic singularity. Input, output and configuration-space singularities are identified as the three basic types, but also a finer classification using six types is given, depending on the cause of the degeneracy.

There are different ways to introduce singular configurations, i.e. by studying the relation of input and output velocities, by studying the mechanical equilibrium or by studying the uniqueness of the solution to the FKP around a given solution [198]. In the following, we restrict our considerations to parallel manipulators; for results about serial manipulators see, e.g., [140, 316].

8.2.1 Singularities and velocities

Differentiating the equation of state (4.1) with respect to time gives the relation

$$A\dot{\Theta} + B\mathbf{w} = 0 \tag{8.1}$$

between the actuated joint velocities $\dot{\Theta}$ and the twist $\mathbf{w}$ of the end-effector. A *serial* singularity means that a zero twist of the end-effector is obtained for non-zero actuated joint velocities. In that case the IIKP (cf. Chapter 6) is not solvable. For a robot having revolute joints between the attachment points of the legs it can be shown that a pose on the boundary of the workspace causes a serial singularity [198]. However, the converse is not true; a counter-example has been given by Bonev et al. [28]. A *parallel* singularity means that a non-zero twist $\mathbf{w}$ of the end-effector is obtained although the actuators are locked. Locally (in the neighborhood of such a configuration), the end-effector is able to accomplish infinitesimal motion. As in the case of locked actuators the mobility of the end-effector should be zero, it is said that the robot *gains* some d.o.f. Thus, certain degrees of freedom of the robot cannot be controlled. In other words, kinematic singularities are configurations in which there is a change in the expected or typical number of instantaneous degrees of freedom [68]. A configuration is defined as non-singular when both the FIKP and the IIKP have unique solutions for any input or output [23].

8.2.2 Singularities and forces

A robot is in a mechanical equilibrium if the joint forces/torques τ balance the wrench $\mathcal{F}$ acting on the end-effector. The fundamental relation

$$\tau = J(s)^T \mathcal{F},$$

where $J(s)$ denotes the kinematic Jacobian, is valid for serial as well as parallel manipulators. In a singular configuration the joint forces may go to infinity as they are obtained by solving the linear system

$$\mathcal{F} = J^{-T} \tau.$$

Hence, there is a significant risk of a breakdown of the robot (as reported in [198]). Therefore, it is important to be able to find such singularities.

8.2.3 Singularities and kinematics

For fixed actuated joint variables Θ the inverse kinematics equation of a robot may be written as

$$F(s) = 0. \tag{8.2}$$

Singularities may now be introduced by studying the uniqueness of solutions of the FKP around a given solution. If the Jacobian is non-singular, there is only one solution in a neighborhood while for a singular Jacobian multiple solutions are obtained, i.e., the platform will be movable although the actuators are locked. Singularities are related to the study of **kinematic branches**. For a given set Θ_1 of actuated joint variables, a finite set of solutions of (8.2) is derived. If the actuated joint variables Θ_1 vary continuously to another set Θ_2, then the solutions of (8.2) also vary continuously: the poses of the end-effector follow different trajectories, the kinematic branches. A singularity occurs if two branches collapse, or if a branch goes to infinity [198].

8.3 Singularity Analysis

Singularity analysis is an important element in the process of designing and developing a robot and has been intensively studied in the past two decades. As pointed out by Merlet [199] the search of singular configurations of parallel robots depends on the study of the singularities of the *full* (6×6) inverse kinematic Jacobian J^{-1}(cf. Section 6.3), even for robots with less than 6 degrees of freedom. The full inverse Jacobian is usually derived through a velocity analysis or through an analysis of the mechanical equilibrium, as described above. The main problem is that although an analytical form of J^{-1} may be known, the expansion of the determinant leads to huge expressions. A closed form for the determinant can usually be calculated for planar or spherical robots with 3 d.o.f. but will be more complicated for a larger number of d.o.f. [198]. A number of authors have sought to apply methods of mathematical singularity theory to the examination of singularities (a survey can be found in [67]). Another approach (following [185]) uses Grassmann geometry [198, 326] and Grassmann–Cayley algebra [4, 14, 15, 326] to determine simpler singular conditions. But calculating the determinant is only the first stage of the singularity analysis; the second stage then is to find poses that cancel the determinant, which is an even more difficult task.

8.3.1 Determination of the singularity-free workspace

Ideally, the regularity of the inverse Jacobian should be checked for all poses of the useful workspace, i.e., the part of a robots workspace in which the end-effector will be most commonly located [199], or for any pose along a given trajectory. Furthermore, a suitable algorithm should be able to take uncertainties in the robot modeling and control into account and the singularity detection should be certified, i.e., the algorithm should provide safe answers even in the presence of numerical round-off errors. The singularity-detection problem has been addressed by Merlet [199], Merlet & Daney [202], where symbolic computation was used to derive an analytical form of the determinant of J^{-1} and its sign at a particular pose $s0$. Then, an interval analysis-based algorithm was applied, which permits to determine if the motion variety, i.e. a variety defined by the end-user within which the robot is constrained to move, includes a set of poses in which the determinant has a sign opposite to the one in s_0. However, in particular if uncertainties (e.g., in the geometrical parameters of the robot) are added as additional unknowns in the components of J^{-1}, obtaining a closed-form of the determinant is no longer possible. Thus, a numerical method may be applied to evaluate the determinant based on the interval form of the inverse Jacobian. However, such a numerical evaluation will lead to large overestimation of the true range of the determinant and hence, to large computation times for the singularity detection scheme. Therefore, it is necessary to develop methods that check the regularity of the set of matrices defined by an interval matrix without calculating its determinant, and thereby take into account, that the inverse Jacobian is a parametric matrix, i.e., that its components are not independent. Merlet &

Donelan [206] adress this problem and propose several methods to facilitate the regularity check. The algorithm given in [199] uses these methods and provides robust in-workspace detection of singularities for any type of robot and with almost any type of motion variety in acceptable computation time; additionally it is able to manage uncertainties in the control and the modeling of the robot.

For planar parallel manipulators Kotlarski et al. [149] present an improved method to obtain the guaranteed singularity-free workspace. In order to avoid the inconvenient interval form of the determinant of J^{-1} a geometrical-based solver is proposed. The method achieves significant improvements compared to a (preconditioned) interval evaluation of the $\det(J^{-1})$ in terms of the guaranteed singularity-free zones in the workspace and the number of bisections used in the branch-and-bound algorithm. However, the algorithm is restricted to special manipulator-geometries.

Recently, Bohigas et al. [23] presented a practical method to compute all types of singularities of non-redundant manipulators with no-helical lower pairs and designated instantaneous input and output speeds. A system of quadratic polynomial equations for each singularity-type described by Zlatanov et al. [349] is defined that encodes all singular configurations, and of which all solutions are initially bounded in a box $\mathbf{B}$. The algorithm for solving the system numerically recursively applies a box shrinking and splitting procedure, where the box shrinking is based on solving two linear programs. The method is applicable to manipulators with arbitrary geometry and will isolate singular configurations with a desired accuracy. In order to complete the task of exhaustive classification of the singularity set of a given mechanism, Bohigas et al. [22] give an efficient method to compute such a classification by means of a hierarchy of singularity tests.

It may be of interest to define a geometrical object such as a sphere or a cube and determine the largest such object that is singularity free and enclosed entirely in the workspace (see also Chapter 4.4). Algorithms to accomplish this task are given, e.g., by Li et al. [163] or Jiang & Gosselin [132].

8.3.2 Singularity index

Often one wants to define a "distance" to measure how "far" a given non-singular pose is from a singularity. The definition of such a distance in a mathematical sense of a metric only is possible if the degrees of freedom are either translational or rotational. In the presence of mixed d.o.f. only a weaker concept of **singularity indices** may be defined. Voglewede & Ebert-Uphoff [308] state a few conditions a singularity index should fulfill and define various such indices in the velocity domain as well as in the force domain. The generic form of a singularity index in the velocity domain is formulated (using the joint velocities $\dot{\Theta}$ and the twist $\mathbf{w}$) as constrained optimization problem

$$\begin{aligned} \min \quad & \dot{\Theta}^T U \dot{\Theta} \\ \text{s.t.} \quad & \dot{\Theta} = J^{-1}\mathbf{w} \\ & \mathbf{w}^T V \mathbf{w} = 1. \end{aligned}$$

Various (somewhat arbitrary) choices of the matrices U, V are proposed such that $\dot{\Theta}^T U \dot{\Theta}$ and $\mathbf{w}^T V \mathbf{w}$ have a physical meaning. The calculation of the index is equivalent to solving a generalized eigenvalue problem. Liu et al. [171] propose several indices to measure closeness to a singularity by taking motion/force transmissibility into account. Hong [112] presents an index which provides a unique invariant measure of closeness to a singular configuration for non-redundant serial and parallel manipulators, based on the screw-based Jacobian.

However, a major drawback of singularity indices is that they usually do not have a closed form. The indices may be computed numerically for a given robot and pose, but minimum, average value, and standard deviation over a given workspace or trajectory are difficult to acertain (but may be important in practice) [198].

8.3.3 FKP and path planning

Singularities may also be useful in some cases, e.g., to sort the solutions of the FKP. As we have seen in Chapter 5, there are methods available to determine all solutions of the problem, but the determination of the solution corresponding to the current pose still poses a challenging problem. The workspace of the robot may be separated into different regions that are singularity-free and separated by singularity varieties. Wenger et al. [324] introduce these so-called **aspects** for parallel manipulators (see also [48, 325]). Usually a manipulator operates within the aspect which includes the initial assembly mode. Thus, a forward kinematic solution will be a candidate to be the current pose only if it can be reached from an initial assembly mode by a singularity-free trajectory. However, it is a non-trivial task to determine the existence of such a trajectory. For a symmetrical planar parallel manipulator with equilateral base and platform triangle the existence of such a non-singular assembly-mode changing trajectory was shown by Chablat & Wenger [49]. Husty [121] then proved for general planar 3-RPR parallel manipulators the ability to pass from one solution of the FKP to another without crossing a singularity, and gave a technique how to construct singularity-free trajectories that join all assembly modes of one connected component (see also Chablat et al. [47]).

Just recently, Bohigas et al. [20] present a singularity-free path planning algorithm that attempts to connect two given non-singular configurations of any non-redundant closed-chain manipulator. The method relies on defining a system of equations whose solution manifold corresponds to the singularity-free subset of the configuration space. This manifold then is explored systematically by a higher-dimensional continuation strategy (as given in [130]), until a path joining the two configurations is found (or the non-existence of such a path is determined).

Conclusion

The science of robotics is a field with high interdisciplinary character. It poses a large number of complex and challenging problems–interesting not only from a mechanical but also from a mathematical point of view.

In the last two decades parallel manipulators have become more and more popular, especially for applications demanding high load carrying capacity or precise positioning. Further advantages of parallel manipulators over their serial counterparts for certain applications are low inertia, higher structural stiffness, and reduced sensitivity to certain errors. Parallel robots are intrinsically more accurate than serial robots as their errors are averaged instead of added cumulatively. However, in terms of size of the (dexterous) workspace or with respect to the suffering from singularities, the performance of parallel robots is poor compared to serial manipulators.

The application area of parallel robots in particular is a broad field; for example, they are used for industrial tasks (e.g., tyre testing, Gough & Whitehall [94]), space applications (e.g., the Canterbury satellite tracker, Jones & Dunlop [137]), or in medical science, Sima'an et al. [281] (e.g., for prostheses with parallel topology, Mendoza-Vázquez et al. [182], or a micro-manipulator for ophthalmic surgery, Grace et al. [95]); see also, e.g., Patel et al. [233], Weck & Staimer [321]. Due to the higher level of complexity problems concerning parallel manipulators are more difficult to solve than problems for their serial counterparts. In principle, the problems become more ambitious for an increasing number of degrees of freedom. Over the past years the number of publications has been growing for parallel manipulators. The benchmark for books concerning parallel manipulators is the one written by [198].

Since a various number of (parallel) architectures and geometries is possible with mostly quite distinct properties, each class of manipulators poses its own specific problem formulations, and proposed solution methods and algorithms are restricted to the special robot type. However, recent publications (e.g., Bohigas et al. [21, 22, 23], Cardou et al. [39], Liu et al. [170], Merlet [199], Petuya et al. [239], Wang et al. [314]) aim to constitute unifying frameworks for several problems whose application spectra are as general as possible.

The problems surveyed in this thesis only cover a small amount of fundamental questions related to robotics. Further topics are concerned with, e.g., trajectory planning, dynamical aspects, statics and forces, synthesis and design of robots. The distinct areas often heavily depend on each other, so it is usually quite difficult to treat one single problem without dealing with others at the same time. The probably most challenging problems are concerned with designing a robot that provides certain properties specified by the tasks it shall accomplish. During the design process from the initial stages of the conceptual definition of the robot to the real prototype, most of the tasks presented in the previous chapters (and some additional ones) have to be taken into account. Due to lack of writing space, the design-aspect was not discussed here. Hernández et al. [106], Merlet & Daney [205] give a design methodology for parallel manipulators.

Considering the mathematical aspects in robotics, the highly interdisciplinary character can be seen as well. The set of displacements which represent the motion of rigid bodies, has the special structure of a group and is directly related to the special Euclidean group $SE(3)$, which is a differentiable manifold, a so-called Lie group. Thus, *algebraic and differential geometry* as well as the concept of *Lie group theory* plays an important role for the analysis and formulation of the problems (see e.g., Selig [275]). Most of the problems arising in robotics heavily depend on the solution of highly non-linear and often algebraic (systems of) equations. Several solution methods, i.e., homotopy, elimination, Gröbner bases, and interval analysis, were outlined in previous chapters, and their main advantages and disadvantages explained. In recent research certain algorithms have been developed which improve the performance of Gröbner basis- and homotopy-methods in particular. Furthermore, interval analysis methods are getting more popular due to improved computer performance, which provides decreasing computation times. Due to the non-linearity, there are multiple solutions to the problems, and hence, methods for selecting the right solution have to be found. Up to now, this problem has not been solved in a satisfactory way, and hence, may be addressed in future work.

In many cases it is computational expensive or not possible to derive closed-form formulations of the equations. In either case, or, if uncertainties such as measurement errors or manufacturing tolerances have to be taken into account, numerical methods are required. To derive certified and reliable results even in the presence of uncertainties and round-off errors, methods of rigorous computing are needed, e.g., round-off error analysis (Wilkinson [327, 328]), interval arithmetical tools (Moore et al. [208], Neumaier [218]), or automatic differentiation (Bliek et al. [18], Bücker et al. [36], Schichl & Márkót [273]). However, the aspect of deriving rigorous solutions often is ignored, and thus, future work may address this problem.

For several applications, a worst-case analysis is required, and thus, global problems have to be solved. Hence, (global) optimization problems need to be considered. In the past few years, global optimization methods have been improved quite significantly. However, to my best knowledge, up to now rigorous global optimization methods are used quite rarely in robotics. Therefore, this opens a wide field for future research. For solving global optimization problems, various relaxation methods, techniques for constraint propagation and higher order constraint propagation, need to be utilized. In addition, inclusion/exclusion regions are required for reducing the so-called cluster effect (Schichl et al. [274]) which causes big performance problems when solving systems of equations. For robotics problems it is a necessity to use various equivalent mathematical formulations of the same problem (Lie group, trigonometric functions, polynomial systems) since some solution techniques work better on some of the formulations than on others. It may be essentially important to improve the methods for rigorous computation in Lie groups, using a generalization of interval arithmetic to Lie groups.

From a personal point of view, robotics is a quite fascinating topic, in particular because it interfaces multiple disciplines. Additionally, robotics is a field of research which yields practical results that may have a positive impact and benefit on society. The work on this thesis was a challenging task for me. On the one hand, because I have no engineering background, and thus, it took quite a while to understand and interrelate the research results; especially the apparently countless number of different manipulator types got confusing for me at times. Furthermore, due to different interpretations and varying use of the same terms and notation, it sometimes was a non-trivial task to draw contextual conclusions. On the other hand, the work on this thesis required the study and combination of several mathematical methods, which I never have studied in detail up to now, or only in a theoretical context. However, this thesis provides only a starting point for me and I intend to do further research in this area.

Appendix

List of Figures

References

[1] M. Agheli and S.S. Nestinger. Closed-form solution for reachable workspace of axially symmetric hexapod robots. In *Mechatronics and Embedded Systems and Applications (MESA), 2012 IEEE/ASME International Conference on*, pp. 75–80. IEEE, 2012.

[2] E.L. Allgower and K. Georg. Numerical path following. *Handbook of Numerical Analysis*, 5(3):207, 1997.

[3] O. Altuzarra, O. Salgado, V. Petuya, and A. Hernández. Point-based Jacobian formulation for computational kinematics of manipulators. *Mechanism and Machine Theory*, 41(12):1407–1423, 2006.

[4] S. Amine, M. Tale Masouleh, S. Caro, P. Wenger, and C. Gosselin. Singularity analysis of 3T2R parallel mechanisms using Grassmann–Cayley algebra and Grassmann geometry. *Mechanism and Machine Theory*, 52:326–340, 2012.

[5] J. Angeles. Is there a characteristic length of a rigid-body displacement? *Mechanism and Machine Theory*, 41(8):884–896, 2006.

[6] J. Angeles. *Fundamentals of robotic mechanical systems*. Springer, 3rd ed., 2007.

[7] J. Apel. A Gröbner approach to involutive bases. *Journal of Symbolic Computation*, 19(5):441–457, 1995.

[8] L. Baron and J. Angeles. The direct kinematics of parallel manipulators under joint-sensor redundancy. *Robotics and Automation, IEEE Transactions on*, 16(1):12–19, 2000.

[9] L. Baron and J. Angles. The kinematic decoupling of parallel manipulators using joint-sensor data. *Robotics and Automation, IEEE Transactions on*, 16(6):644–651, 2000.

[10] A.H. Barr, B. Currin, S. Gabriel, and J.F. Hughes. Smooth interpolation of orientations with angular velocity constraints using quaternions. In *ACM SIGGRAPH Computer Graphics*, vol. 26, pp. 313–320. ACM, 1992.

[11] D.J. Bates, J.D. Hauenstein, and A.J. Sommese. Efficient path tracking methods. *Numerical Algorithms*, 58(4):451–459, 2011.

[12] D.J. Bates, J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Bertini: Software for numerical algebraic geometry. Webpage, 2012. URL `http://www.nd.edu/~sommese/bertini`.

[13] C. Beltrán and A. Leykin. Certified numerical homotopy tracking. *Experimental Mathematics*, 21(1):69–83, 2012.

[14] P. Ben-Horin and M. Shoham. Singularity analysis of a class of parallel robots based on Grassmann–Cayley algebra. *Mechanism and Machine Theory*, 41(8):958–970, 2006.

[15] P. Ben-Horin and M. Shoham. Application of Grassmann–Cayley Algebra to Geometrical Interpretation of Parallel Robot Singularities. *The International Journal of Robotics Research*, 28(1):127–141, 2009.

[16] P.R. Bergamaschi, S. Saramago, and L.S. Coelho. Comparative study of SQP and metaheuristics for robotic manipulator design. *Applied numerical mathematics*, 58(9):1396–1412, 2008.

[17] N. Binaud, S. Caro, and P. Wenger. Sensitivity comparison of planar parallel manipulators. *Mechanism and Machine Theory*, 45(11):1477–1490, 2010.

[18] C. Bliek, C. Spellucci, L.N. Vicente, and A. et al. Neumaier. Algorithms for solving nonlinear constrained and optimization problems: The state of the art. Online, 2001. URL `http://www.mat.univie.ac.at/~neum/glopt/coconut/StArt.html`.

[19] L. Blum, F. Cucker, M. Shub, and S. Smale. Complexity and real computation. Foreword by Richard M. Karp. 1998.

[20] O. Bohigas, M.E. Henderson, L. Ros, and J.M. Porta. A singularity-free path planner for closed-chain manipulators. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 2128–2134. IEEE, 2012.

[21] O. Bohigas, L. Ros, and M. Manubens. A unified method for computing position and orientation workspaces of general Stewart platforms. *ASME Conference Proceedings*, 2011(54839):959–968, 2011.

[22] O. Bohigas, D. Zlatanov, M. Manubens, and L. Ros. On the numerical classification of the singularities of robot manipulators. In *Proc. of the ASME International Design Engineering Technical Conferences*, 2012.

[23] O. Bohigas, D. Zlatanov, L. Ros, M. Manubens, and J.M. Porta. Numerical computation of manipulator singularities. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1351–1358. IEEE, 2012.

[24] I.A. Bonev. para*lle*MIC. Webpage, 2012. URL `http://www.parallemic.org`.

[25] I.A. Bonev and C.M. Gosselin. Analytical determination of the workspace of symmetrical spherical parallel mechanisms. *Robotics, IEEE Transactions on*, 22(5):1011–1017, 2006.

[26] I.A. Bonev and J. Ryu. A new method for solving the direct kinematics of general 6-6 Stewart Platforms using three linear extra sensors. *Mechanism and Machine Theory*, 35(3):423–436, 2000.

[27] I.A. Bonev, J. Ryu, S.G. Kim, and S.K. Lee. A closed-form solution to the direct kinematics of nearly general parallel manipulators with optimally located three linear extra sensors. *Robotics and Automation, IEEE Transactions on*, 17(2):148–156, 2001.

[28] I.A. Bonev, D. Zlatanov, and C.M. Gosselin. Singularity analysis of 3-DOF planar parallel mechanisms via screw theory. *Journal of Mechanical Design*, 125:573, 2003.

[29] S. Briot, I. Bonev, et al. Are parallel robots more accurate than serial robots? *CSME Transactions*, 31(4):445–456, 2007.

[30] S. Briot and I.A. Bonev. Accuracy analysis of 3-dof planar parallel robots. *Mechanism and Machine Theory*, 43(4):445–458, 2008.

[31] C.G. Broyden, J. Dennis, and J.J. Moré. On the local and superlinear convergence of quasi-Newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

[32] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenrings nach einem nulldimensionalen Polynomideal*. PhD thesis, Universität Innsbruck, 1965.

[33] B. Buchberger. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems. *Aequationes mathematicae*, 4(3):374–383, 1970.

[34] B. Buchberger. *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory*, chapter 6. Reidel, 1985.

[35] B. Buchberger. Introduction to Gröbner bases. *Gröbner bases and applications*, pp. 3–31, 1998.

[36] M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris. *Automatic differentiation: applications, theory, and implementations (Lecture notes in computational science and engineering)*. Springer-Verlag New York, Inc., 2006.

[37] J. Canny and I. Emiris. An efficient algorithm for the sparse mixed resultant. *Applied Algebra, Algebraic Algorithms and Error-correcting Codes*, pp. 89–104, 1993.

[38] Y. Cao, Z. Huang, H. Zhou, and W. Ji. Orientation workspace analysis of a special class of the Stewart–Gough parallel manipulators. *Robotica*, 28(07): 989–1000, 2010.

[39] P. Cardou, S. Bouchard, and C. Gosselin. Kinematic-sensitivity indices for dimensionally nonhomogeneous Jacobian matrices. *IEEE Transactions on Robotics*, 26(1):166–173, 2010.

[40] S. Caro, N. Binaud, and P. Wenger. Sensitivity analysis of 3-RPR planar parallel manipulators. *Journal of mechanical design*, 131(12), 2009.

[41] S. Caro, P. Wenger, F. Bennis, and D. Chablat. Sensitivity analysis of the orthoglide: A three-DOF translational parallel kinematic machine. *Journal of mechanical design*, 128(2):392–402, 2006.

[42] M. Ceccarelli. A synthesis algorithm for three-revolute manipulators by using an algebraic formulation of workspace boundary. *Journal of mechanical design*, 117(2):298–302, 1995.

[43] M. Ceccarelli. A formulation for the workspace boundary of general n-revolute manipulators. *Mechanism and Machine Theory*, 31(5):637–646, 1996.

[44] M. Ceccarelli. Workspace evaluation for analysis and synthesis of manipulators. *Advances in Mechanisms Design*, pp. 289–301, 2012.

[45] M. Ceccarelli and C. Lanni. A multi-objective optimum design of general 3R manipulators for prescribed workspace limits. *Mechanism and Machine Theory*, 39(2):119–132, 2004.

[46] J.J. Cervantes-Sánchez, J.C. Hernández-Rodrıguez, and E.J. González-Galván. On the 5R spherical, symmetric manipulator: workspace and singularity characterization. *Mechanism and machine theory*, 39(4):409–429, 2004.

[47] D. Chablat, G. Moroz, and P. Wenger. Uniqueness domains and non singular assembly mode changing trajectories. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3946–3951. IEEE, 2011.

[48] D. Chablat and P. Wenger. Working modes and aspects in fully parallel manipulators. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, pp. 1964–1969. IEEE, 1998.

[49] D. Chablat and P. Wenger. The kinematic analysis of a symmetrical three-degree-of-freedom planar parallel manipulator. *arXiv preprint arXiv:0705.0959*, 2007.

[50] D. Chablat, P. Wenger, and J. Angeles. The kinematic design of a 3-DOF hybrid manipulator. *arXiv preprint arXiv:0705.0961*, 2007.

[51] D. Chablat, P. Wenger, F. Majou, and J.P. Merlet. An interval analysis based study for the design and the comparison of three-degrees-of-freedom parallel kinematic machines. *The International Journal of Robotics Research*, 23(6): 615–624, 2004.

[52] D. Chablat, P. Wenger, J.P. Merlet, et al. A comparative study between two three-dof parallel kinematic machines using kinetostatic criteria and interval analysis. In *11th World Congress in Mechanism and Machine Science*, pp. 1–6, 2004.

[53] R. Chandra and L. Rolland. On solving the forward kinematics of 3RPR planar parallel manipulator using hybrid metaheuristics. *Applied Mathematics and Computation*, 217(22):8997–9008, 2011.

[54] A.H. Chebbi, Z. Affi, and L. Romdhane. Prediction of the pose errors produced by joints clearance for a 3-UPU parallel robot. *Mechanism and Machine Theory*, 44(9):1768–1783, 2009.

[55] Y.J. Chiu and M.H. Perng. Forward kinematics of a general fully parallel manipulator with auxiliary sensors. *The International Journal of Robotics Research*, 20(5):401–414, 2001.

[56] D.A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, vol. 10. Springer, 2007.

[57] J. Craig. *Introduction to robotics (Mechanics and control)*. Pearson, 3rd ed., 2004.

[58] J.K. Davidson, K.H. Hunt, and G.R. Pennock. *Robots and screw theory: applications of kinematics and statics to robotics*. Oxford University Press, 2004.

[59] J. Denavit and R.S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME J. Applied Mechanics*, (77):215–221, 1955.

[60] J.E. Dennis and R.B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Society for Industrial and Applied Mathematics (Philadelphia), 1996.

[61] A.S. Deo and I.D. Walker. Overview of damped least-squares methods for inverse kinematics of robot manipulators. *Journal of Intelligent & Robotic Systems*, 14(1):43–68, 1995.

[62] A.K. Dhingra, A.N. Almadi, and D. Kohli. A Gröbner-Sylvester hybrid method for closed-form displacement analysis of mechanisms. *ASME Journal of Mechanical Design*, 122(4):431–438, 2000.

[63] A.K. Dhingra, A.N. Almadi, and D. Kohli. Closed-form displacement and coupler curve analysis of planar multi-loop mechanisms using Gröbner bases. *Mechanism and Machine Theory*, 36(2):273–298, 2001.

[64] S. Di Rocco, D. Eklund, A.J. Sommese, and C.W. Wampler. Algebraic C^*-actions and the inverse kinematics of a general 6R manipulator. *Applied Mathematics and Computation*, 216(9):2512–2524, 2010.

[65] O. Didrit, M. Petitot, and E. Walter. Guaranteed solution of direct kinematic problems for general configurations of parallel manipulators. *Robotics and Automation, IEEE Transactions on*, 14(2):259–266, 1998.

[66] P. Dietmaier. The Stewart-Gough platform of general geometry can have 40 real postures. *Advances in Robot Kinematics: Analysis and Control*, pp. 1–10, 1998.

[67] P. Donelan. Singularities of robot manipulators. *Singularity Theory*, pp. 189–217, 2007.

[68] P. Donelan. Kinematic singularities of robot manipulators. *Advances in Robot Manipulators*, pp. 401–416, 2010.

[69] K.L. Doty, C. Melchiorri, E.M. Schwartz, and C. Bonivento. Robot manipulability. *IEEE Transactions on Robotics and Automation*, 11(3):462–468, 1995.

[70] J. Enferadi and A.A. Tootoonchi. A novel spherical parallel manipulator: forward position problem, singularity analysis, and isotropy design. *Robotica*, 27(05):663–676, 2009.

[71] J. Enferadi and A.A. Tootoonchi. A novel approach for forward position analysis of a double-triangle spherical parallel manipulator. *European Journal of Mechanics-A/Solids*, 29(3):348–355, 2010.

[72] K. Etemadi-Zanganeh and J. Angeles. Real-time direct kinematics of general six-degree-of-freedom parallel manipulators with minimum-sensor data. *Journal of robotic systems*, 12(12):833–844, 1995.

[73] A. Fattah and A.M. Hasan Ghasemi. Isotropic design of spatial parallel manipulators. *The International Journal of Robotics Research*, 21(9):811–824, 2002.

[74] J.C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139(1):61–88, 1999.

[75] J.C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, pp. 75–83. ACM, 2002.

[76] J.C. Faugère. FGb: A library for computing Gröbner bases. In K. Fukuda, J. Hoeven, M. Joswig, and N. Takayama, editors, *Mathematical Software - ICMS 2010*, vol. 6327 of *Lecture Notes in Computer Science*, pp. 84–87, Berlin, Heidelberg, September 2010. Springer Berlin / Heidelberg.

[77] J.C. Faugère. FGb library. Webpage, 2013. URL `http://www-calfor.lip6.fr/~jcf/Software/index.html`.

[78] J.C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Grobner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.

[79] J.C. Faugère and D. Lazard. Combinatorial classes of parallel manipulators. *Mechanism and Machine Theory*, 30(6):765–776, 1995.

[80] J.C. Faugère, C. Mou, et al. Fast algorithm for change of ordering of zero-dimensional Gröbner bases with sparse multiplication matrices. In *ISSAC*, vol. 11, pp. 1–8, 2011.

[81] Y. Feng, W. Yao-nan, and Y. Yi-min. Inverse kinematics solution for robot manipulator based on neural network under joint subspace. *International Journal of Computers and Communications*, 7(3):459–472, 2012.

[82] F. Freudenstein and B. Roth. Numerical solution of systems of nonlinear equations. *Journal of the ACM (JACM)*, 10(4):550–556, 1963.

[83] A. Frisoli, M. Solazzi, D. Pellegrinetti, and M. Bergamasco. A new screw theory method for the estimation of position accuracy in spatial parallel manipulators with revolute joint clearances. *Mechanism and Machine Theory*, 46(12):1929 – 1949, 2011.

[84] D. Gan, Q. Liao, J.S. Dai, S. Wei, and L.D. Seneviratne. Forward displacement analysis of the general 6–6 Stewart mechanism using Gröbner bases. *Mechanism and Machine Theory*, 44(9):1640–1647, 2009.

[85] I.M. Gelfand, M. Kapranov, and A. Zelevinsky. *Discriminants, resultants, and multidimensional determinants.* Birkhäuser Boston, 2008.

[86] G. Gogu. Mobility of mechanisms: a critical review. *Mechanism and Machine Theory*, 40(9):1068–1097, 2005.

[87] C. Gonzalez, D. Blanco, and L. Moreno. A memetic approach to the inverse kinematics problem. In *Mechatronics and Automation (ICMA), 2012 International Conference on*, pp. 180–185. IEEE, 2012.

[88] C.M. Gosselin. *Kinematic analysis, optimization and programming of parallel robotic manipulators.* PhD thesis, McGill University, Montral, 1988.

[89] C.M. Gosselin. Determination of the workspace of 6-DOF parallel manipulators. *Journal of Mechanical Design*, 112(3):331–336, 1990.

[90] C.M. Gosselin. The optimum design of robotic manipulators using dexterity indices. *Robotics and Autonomous Systems*, 9(4):213–226, 1992.

[91] C.M. Gosselin and J. Angeles. Singularity analysis of closed-loop kinematic chains. *Robotics and Automation, IEEE Transactions on*, 6(3):281–290, 1990.

[92] C.M. Gosselin and J. Angeles. A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design*, 113(3): 220–226, 1991.

[93] V.E. Gough. Contribution to discussion of papers on research in automobile stability, control and tyre performance. In *Proc. Auto Div. Inst. Mech. Eng*, vol. 171, pp. 392–394, 1956.

[94] V.E. Gough and S.G. Whitehall. Universal tyre test machine. In *Proc. FISITA 9th Int. Technical Congress*, pp. 117–137, 1962.

[95] K.W. Grace, J. E. Colgate, M. R. Glucksberg, and J. H. Chun. A six degree of freedom micromanipulator for ophthalmic surgery. In *IEEE International Conference on Robotics and Automation*, pp. 630–635. IEEE, 1993.

[96] R. Guenther and D. Martins. Screw-based relative Jacobian for manipulators cooperating in a task. In *ABCM Symposium Series in Mechatronics. v3*, pp. 276–285, 2008.

[97] W.Z. Guo, F. Gao, and S. Mekid. A new analysis of workspace performances and orientation capability for 3-DOF planar manipulators. *International Journal of Robotics and Automation*, 25(6), 2010.

[98] K.C. Gupta and B. Roth. Design considerations for manipulator workspace. *Journal of Mechanical Design*, 104:704, 1982.

[99] F. Hao and J.M. McCarthy. Conditions for line-based singularities in spatial platform manipulators. *Journal of Robotic Systems*, 15(1):43–55, 1998.

[100] F. Hao and J.P. Merlet. Multi-criteria optimal design of parallel manipulators based on interval analysis. *Mechanism and Machine Theory*, 40(2): 157–171, 2005.

[101] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Regeneration homotopies for solving systems of polynomials. *Mathematics of Computation*, 80 (273):345, 2010.

[102] J.D. Hauenstein, A.J. Sommese, and C.W. Wampler. Regenerative cascade homotopies for solving polynomial systems. *Applied Mathematics and Computation*, 218(4):1240–1246, 2011.

[103] J.D. Hauenstein and F. Sottile. Algorithm 921: alphaCertified: Certifying solutions to polynomial systems. *ACM Transactions on Mathematical Software (TOMS)*, 38(4):28, 2012.

[104] A.M. Hay and J.A. Snyman. The optimal synthesis of parallel manipulators for desired workspace. *Advances in Robot Kinematics*, pp. 337–346, 2002.

[105] A.M. Hay and J.A. Snyman. A multi-level optimization methodology for determining the dextrous workspaces of planar parallel manipulators. *Structural and Multidisciplinary Optimization*, 30(6):422–427, 2005.

[106] A. Hernández, O. Altuzarra, O. Salgado, C. Pinto, and V. Petuya. Designing parallel manipulators: from specifications to a real prototype. *Industrial Robot: An International Journal*, 39(5):500–512, 2012.

[107] A. Hernández and V. Petuya. Position analysis of planar mechanisms with R-pairs using a geometrical–iterative method. *Mechanism and Machine Theory*, 39(2):133–152, 2004.

[108] A. Hernández, V. Petuya, and E. Amezua. A method for the solution of the forward position problems of planar mechanisms with prismatic and revolute joints. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 216(4):395–407, 2001.

[109] J. Hesselbach, C. Bier, I. Pietsch, N. Plitea, S. Büttgenbach, A. Wogersien, and J. Güttler. Passive-joint sensors for parallel robots. *Mechatronics*, 15 (1):43–65, 2005.

[110] T. Ho, C.G. Kang, and S. Lee. Efficient closed-form solution of inverse kinematics for a specific six-DOF arm. *International Journal of Control, Automation and Systems*, 10(3):567–573, 2012.

[111] M. Honegger, A. Codourey, and E. Burdet. Adaptive control of the hexaglide, a 6 dof parallel manipulator. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1, pp. 543–548. IEEE, 1997.

[112] M.B. Hong. On the robot singularity: A novel geometric approach. *Int J Adv Robotic Sy*, 9(197), 2012.

[113] J. Huang, X. Wang, D. Liu, and Y. Cui. A new method for solving inverse kinematics of an industrial robot. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, vol. 3, pp. 53–56. IEEE, 2012.

[114] T. Huang, H.T. Liu, and D.G. Chetwynd. Generalized Jacobian analysis of lower mobility manipulators. *Mechanism and Machine Theory*, 46(6): 831–844, 2011.

[115] X. Huang and G. He. New and efficient method for the direct kinematic solution of the general planar Stewart platform. In *Automation and Logistics, 2009. ICAL'09. IEEE International Conference on*, pp. 1979–1983. IEEE, 2009.

[116] X. Huang, Q. Liao, and S. Wei. Closed-form forward kinematics for a symmetrical 6-6 Stewart platform using algebraic elimination. *Mechanism and Machine Theory*, 45(2):327–334, 2010.

[117] X. Huang, Q. Liao, S. Wei, X. Qiang, and S. Huang. Forward kinematics of the 6-6 Stewart platform with planar base and platform using algebraic elimination. In *Automation and Logistics, 2007 IEEE International Conference on*, pp. 2655–2659. IEEE, 2007.

[118] K.H. Hunt. Review: Don't cross-thread the screw! *Journal of Robotic Systems*, 20(7):317–339, 2003.

[119] K.H. Hunt and P.R. McAree. The octahedral manipulator: geometry and mobility. *The International Journal of Robotics Research*, 17(8):868–885, 1998.

[120] M.L. Husty. An algorithm for solving the direct kinematics of general Stewart-Gough platforms. *Mechanism and Machine Theory*, 31(4):365–379, 1996.

[121] M.L. Husty. Non-singular assembly mode change in 3-RPR-parallel manipulators. In *Computational Kinematics: Proceedings of the 5th International Workshop on Computational Kinematics*, p. 51. Springer Verlag, 2009.

[122] M.L. Husty and A. Karger. Self-motions of Griffis-Duffy type parallel manipulators. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 7–12. IEEE, 2000.

[123] M.L. Husty, A. Karger, H. Sachs, and W. Steinhilper. *Kinematik und Robotik*. Springer Verlag, 1997.

[124] M.L. Husty, M. Pfurner, and H.P. Schröcker. A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mechanism and Machine Theory*, 42(1):66–81, 2007.

[125] P. Huynh and T. Arai. Maximum velocity analysis of parallel manipulators. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 3268–3273. IEEE, 1997.

[126] C. Innocenti. Forward kinematics in polynomial form of the general Stewart platform. *Journal of Mechanical Design(Transactions of the ASME)*, 123 (2):254–260, 2001.

[127] C. Innocenti and V. Parenti-Castelli. Forward kinematics of the general 6–6 fully parallel mechanism: An exhaustive numerical approach via a mono-dimensional-search algorithm. *Journal of Mechanical Design*, 115(4):932–937, 1993.

[128] C. Innocenti and V. Parenti-Castelli. Exhaustive enumeration of fully-parallel kinematic chains. In *Proc. of the 1994 ASME Int. Winter Annual Meeting*, vol. 55, p. 2, 1994.

[129] C. Innocenti and V. Parenti-Castelli. Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators. *Journal of Mechanical design*, 120(1):73–79, 1998.

[130] L. Jaillet and J. Porta. Path planning with loop closure constraints using an atlas-based RRT. *15th International Symposium on Robotics Research, Flagstaff, USA*, 2011.

[131] L. Jelenković and L. Budin. Error analysis of a Stewart platform based manipulators. In *6th Int. Conf. on Intelligent Engineering Systems (INES 2002)*, 2002.

[132] Q. Jiang and C.M. Gosselin. Determination of the maximal singularity-free orientation workspace for the Gough-Stewart platform. *Mechanism and Machine Theory*, 44(6):1281–1293, 2009.

[133] Q. Jiang and C.M. Gosselin. Evaluation and representation of the theoretical orientation workspace of the Gough-Stewart platform. *Journal of Mechanisms and Robotics*, 1(2), 2009.

[134] Y. Jin, I. Chen, G. Yang, et al. Workspace evaluation of manipulators through finite-partition of $SE(3)$. *Robotics and Computer-Integrated Manufacturing*, 27(4):850–859, 2011.

[135] Y. Jin, I.M. Chen, and G. Yang. Finite-partition of $SE(3)$ and its applications on workspace optimization of parallel manipulators. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 2133–2138. IEEE, 2006.

[136] D.Y. Jo and E.J. Haug. Workspace analysis of multibody mechanical systems using continuation methods. *Journal of mechanisms, transmissions, and automation in design*, 111(4):581–589, 1989.

[137] T.P. Jones and G.R. Dunlop. Analysis of rigid-body dynamics for closed-loop mechanisms–its application to a novel satellite tracking device. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 217(4):285–298, 2003.

[138] K. Kamali and A. Akbarzadeh. A novel method for direct kinematics solution of fully parallel manipulators using basic regions theory. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 225(5):683–701, 2011.

[139] K. Kamali and A. A. Tootoonchi. Applications of workspace categorization for parallel manipulators in identification of desired direct kinematics solution. In *17th Annual (International) Conference on Mechanical Engineering - ISME 2009*, 2009.

[140] A. Karger. Singularity analysis of serial robot-manipulators. *Journal of mechanical design*, 118(4):520–525, 1996.

[141] C.T. Kelley. *Solving nonlinear equations with Newton's method*, vol. 1. Society for Industrial Mathematics, 1987.

[142] A.B. Kempe. On a general method of describing plane curves of the nth degree by linkwork. *Proceedings of the London Mathematical Society*, 1(1): 213, 1875.

[143] W.A. Khan and J. Angeles. The kinetostatic optimization of robotic manipulators: the inverse and the direct problems. *Journal of mechanical design*, 128(1):168–178, 2006.

[144] S. Khatami and F. Sassani. Isotropic design optimization of robotic manipulators using a genetic algorithm method. In *Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on*, pp. 562–567. IEEE, 2002.

[145] H.S. Kim and Y.J. Choi. The kinematic error bound analysis of the Stewart platform. *Journal of Robotic Systems*, 17(1):63–73, 2000.

[146] S.G. Kim and J. Ryu. New dimensionally homogeneous Jacobian matrix formulation by three end-effector points for optimal design of parallel manipulators. *Robotics and Automation, IEEE Transactions on*, 19(4):731–736, 2003.

[147] R. Köker. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. *Information Sciences*, 222(0):528 – 543, 2013.

[148] J. Kotlarski, H. Abdellatif, T. Ortmaier, and B. Heimann. Enlarging the useable workspace of planar parallel robots using mechanisms of variable geometry. In *Reconfigurable Mechanisms and Robots, 2009. ReMAR 2009. ASME/IFToMM International Conference on*, pp. 63–72. IEEE, 2009.

[149] J. Kotlarski, R. De Nijs, H. Abdellatif, and B. Heimann. New interval-based approach to determine the guaranteed singularity-free workspace of parallel robots. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 1256–1261. IEEE, 2009.

[150] R. Krawczyk and A. Neumaier. Interval slopes for rational functions and associated centered forms. *SIAM Journal on Numerical Analysis*, 22(3): 604–616, 1985.

[151] S. Kucuk and Z. Bingul. Robot workspace optimization based on a novel local and global performance indices. *IEEE ISIE 2005*, pp. 20–23, 2005.

[152] V. Kumar. Introduction to robot geometry and kinematics. Lecture Notes, 2002. URL `http://www.seas.upenn.edu/~meam520/notes02/IntroRobotKinematics5.pdf`.

[153] M. Kunzinger. Differential geometry 1. Lecture Notes, 2008. URL `http://www.mat.univie.ac.at/~mike/teaching/ss08/dg.pdf`.

[154] D. Lazard. Stewart platform and Gröbner basis. *Proc. ARK*, pp. 136–142, 1992.

[155] D. Lazard et al. On the representation of rigid-body motions and its application to generalized platform manipulators. *Computational Kinematics*, 1 (1):175–181, 1993.

[156] D. Lazard and J.P. Merlet. The (true) Stewart platform has 12 configurations. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 2160–2165. IEEE, 1994.

[157] E. Lee, C. Mavroidis, and J.P. Merlet. Five precision point synthesis of spatial RRR manipulators using interval analysis. *Journal of Mechanical Design*, 126:842, 2004.

[158] H.Y. Lee and C.G. Liang. Displacement analysis of the general spatial 7-link 7R mechanism. *Mechanism and Machine Theory*, 23(3):219–226, 1988.

[159] T.L. Lee, T.Y. Li, and C.H. Tsai. HOM4PS-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83(2):109–133, 2008.

[160] T.Y. Lee and J.K. Shim. Forward kinematics of the general 6–6 Stewart platform using algebraic elimination. *Mechanism and Machine Theory*, 36 (9):1073–1085, 2001.

[161] T.Y. Lee and J.K. Shim. Improved dialytic elimination algorithm for the forward kinematics of the general Stewart–Gough platform. *Mechanism and Machine Theory*, 38(6):563–577, 2003.

[162] J. Lenarcic. An efficient numerical approach for calculating the inverse kinematics for robot manipulators. *Robotica*, 3(1):21–26, 1985.

[163] H. Li, C. Gosselin, and M. Richard. Determination of the maximal singularity-free zones in the six-dimensional workspace of the general Gough–Stewart platform. *Mechanism and Machine Theory*, 42(4):497–511, 2007.

[164] TY Li. Numerical solution of polynomial systems by homotopy continuation methods. *Handbook of numerical analysis*, 11:209–304, 2003.

[165] T.Y. Li and C.H. Tsai. HOM4PS-2.0 para: Parallelization of HOM4PS-2.0 for solving polynomial systems. *Parallel Computing*, 35(4):226–238, 2009.

[166] Y. Li and Q. Xu. Kinematic analysis of a 3-PRS parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 23(4):395–408, 2007.

[167] Z.M. Li, C.G. Li, and S.J. Lv. A method for solving inverse kinematics of PUMA560 manipulator based on PSO-RBF network. In *Natural Computation (ICNC), 2012 Eighth International Conference on*, pp. 298–301. IEEE, 2012.

[168] Q. Lin, J.W. Burdick, and E. Rimon. Constructing minimum deflection fixture arrangements using frame invariant norms. *Automation Science and Engineering, IEEE Transactions on*, 3(3):272–286, 2006.

[169] H. Lipkin and J. Duffy. Hybrid twist and wrench control for a robotic manipulator. In *Trans. ASME J. Mech. Transm. Automation Design*, 1988.

[170] H. Liu, T. Huang, and D.G. Chetwynd. A method to formulate a dimensionally homogeneous Jacobian of parallel manipulators. *IEEE Transactions on Robotics*, 27(1):150–156, 2011.

[171] X. Liu, W. Chao, and W. Jinsong. A new approach for singularity analysis and closeness measurement to singularities of parallel manipulators. *Journal of Mechanisms and Robotics*, 4(4), 2012.

[172] X.J. Liu, Z.L. Jin, and F. Gao. Optimum design of 3-DOF spherical parallel manipulators with respect to the conditioning and stiffness indices. *Mechanism and Machine Theory*, 35(9):1257–1267, 2000.

[173] Y. Lu and B. Hu. Unification and simplification of velocity/acceleration of limited-dof parallel manipulators with linear active legs. *Mechanism and Machine Theory*, 43(9):1112–1128, 2008.

[174] E. Macho, C. Pinto, E. Amezua, and A. Hernandez. Software tool to compute, analyze and visualize workspaces of parallel kinematics robots. *Advanced Robotics*, 25(6-7):675–698, 2011.

[175] G. Malajovich and K. Meer. Computing minimal multi-homogeneous Bézout numbers is hard. *Theory of Computing Systems*, 40(4):553–570, 2007.

[176] D. Manocha and J.F. Canny. Efficient inverse kinematics for general 6R manipulators. *Robotics and Automation, IEEE Transactions on*, 10(5):648 –657, 1994.

[177] I. Mansouri and M. Ouali. The power manipulability–a new homogeneous performance index of robot manipulators. *Robotics and Computer-Integrated Manufacturing*, 27(2):434–449, 2011.

[178] J.E. Marsden and T.S. Ratiu. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, vol. 17. Springer, 1999.

[179] E.W. Mayr and A.R. Meyer. The complexity of the word problems for commutative semigroups and polynomial ideals. *Advances in mathematics*, 46(3):305–329, 1982.

[180] J.M. McCarthy and G.S. Soh. *Geometric design of linkages*, vol. 11. Springer, 2010.

[181] S. Mekid and T. Ogedengbe. A review of machine tool accuracy enhancement through error compensation in serial and parallel kinematic machines. *International Journal of Precision Technology*, 1(3):251–286, 2010.

[182] J.R. Mendoza-Vázquez, E. Tlelo-Cuautle, J.L. Vázquez-Gonzalez, and A.Z. Escudero-Uribe. Simulation of a parallel mechanical elbow with 3 DOF. *Journal of Applied Research and Technology*, 7(02), 2009.

[183] J. Meng, D. Zhang, and Z. Li. Accuracy analysis of parallel manipulators with joint clearance. *Journal of Mechanical Design*, 131(1), 2009.

[184] J. Meng, D. Zhang, T. Zhang, H. Wang, and Z. Li. Accuracy analysis of general parallel manipulators with joint clearance. In *Robotics and Automation, 2007 IEEE International Conference on*, pp. 889–894. IEEE, 2007.

[185] J.P. Merlet. Singular configurations of parallel manipulators and Grassmann geometry. *The International Journal of Robotics Research*, 8(5):45–56, 1989.

[186] J.P. Merlet. Geometrical determination of the workspace of a constrained parallel manipulator. *ARK: Ferrare*, pp. 326–329, September 7–9 1992.

[187] J.P. Merlet. Closed-form resolution of the direct kinematics of parallel manipulators using extra sensors data. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pp. 200–204. IEEE, 1993.

[188] J.P. Merlet. Determination of the orientation workspace of parallel manipulators. *Journal of Intelligent & Robotic Systems*, 13(2):143–160, 1995.

[189] J.P. Merlet. Articular velocities of parallel manipulators. II. Finding all the robots with fixed extremal articular velocity for performing a fixed cartesian velocity over a whole workspace. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, pp. 3262–3267. IEEE, 1997.

[190] J.P. Merlet. Estimation efficace des caractéristiques de robots parallèles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et singularités dans un espace de travail en translation. INRIA report, 1997. URL `http://hal.inria.fr/docs/00/07/34/46/PDF/RR-3243.pdf`.

[191] J.P. Merlet. Determination of 6D workspaces of a Gough-type 6 DOF parallel manipulator. *12th RoManSy, Paris*, pp. 261–268, July 6–9 1998.

[192] J.P. Merlet. Efficient computation of the extremum of the articular velocities of a parallel manipulator in a translation workspace. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, pp. 1976–1981. IEEE, 1998.

[193] J.P. Merlet. Determination of 6D workspaces of Gough-type parallel manipulator and comparison between different geometries. *The International Journal of Robotics Research*, 18(9):902–916, 1999.

[194] J.P. Merlet. Guaranteed in-the-workspace improved trajectory/ surface/ volume verification for parallel robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4, pp. 4103–4108. IEEE, 2004.

[195] J.P. Merlet. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *The International Journal of robotics research*, 23(3):221–235, 2004.

[196] J.P. Merlet. Computing the worst case accuracy of a PKM over a workspace or a trajectory. In *The 5th Chemnitz Parallel Kinematics Seminar, Chemnitz, Germany*, pp. 83–96, 2006.

[197] J.P. Merlet. Jacobian, manipulability, condition number, and accuracy of parallel robots. *Journal of Mechanical Design*, 128:199, 2006.

[198] J.P. Merlet. *Parallel Robots*. Springer, 2nd ed., 2006.

[199] J.P. Merlet. A formal-numerical approach for robust in-workspace singularity detection. *IEEE Transactions on Robotics*, 23(3):393–402, 2007.

[200] J.P. Merlet. Interval analysis and reliability in robotics. *International Journal of Reliability and Safety*, 3(1):104–130, 2009.

[201] J.P. Merlet. Interval analysis for certified numerical solution of problems in robotics. *International Journal of Applied Mathematics and Computer Science*, 19(3):399–412, 2009.

[202] J.P. Merlet and D. Daney. A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot. *Computational Kinematics*, pp. 167–176, 2001.

[203] J.P. Merlet and D. Daney. Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pp. 942–947. IEEE, 2005.

[204] J.P. Merlet and D. Daney. Legs interference checking of parallel robots over a given workspace or trajectory. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 757–762. IEEE, 2006.

[205] J.P. Merlet and D. Daney. *Appropriate design of parallel manipulators*, chapter 1, pp. 1–25. Springer, 2008.

[206] J.P. Merlet and P. Donelan. On the regularity of the inverse Jacobian of parallel robots. *Advances in Robot Kinematics*, pp. 41–48, 2006.

[207] B. Mishra. *Algorithmic algebra*. Springer, 1993.

[208] R.E. Moore, R.B. Kearfott, and M.J. Cloud. *Introduction to interval analysis*. Society for Industrial Mathematics, 2009.

[209] A.P. Morgan. *Solving polynominal systems using continuation for engineering and scientific problems*, vol. 57. Society for Industrial Mathematics, 2009.

[210] A.P. Morgan and A.J. Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Applied Mathematics and Computation*, 24(2):101–113, 1987.

[211] A.P. Morgan and A.J. Sommese. Computing all solutions to polynomial systems using homotopy continuation. *Applied Mathematics and Computation*, 24(2):115–138, 1987.

[212] A.P. Morgan, A.J. Sommese, and C.W. Wampler. A product-decomposition bound for Bezout numbers. *SIAM journal on numerical analysis*, 32(4): 1308–1325, 1995.

[213] B. Mourrain. The 40 "generic" positions of a parallel robot. In *Proceedings of the 1993 international symposium on Symbolic and algebraic computation*, pp. 173–182. ACM, 1993.

[214] B. Mourrain and P. Trébuchet. Generalized normal forms and polynomial system solving. In *Proceedings of the 2005 international symposium on Symbolic and algebraic computation*, pp. 253–260. ACM, 2005.

[215] B. Mourrain and P. Trébuchet. Stable normal forms for polynomial system solving. *Theoretical Computer Science*, 409(2):229–240, 2008.

[216] R.M. Murray, Z. Li, and S.S. Sastry. A mathematical introduction to robotic manipulation. Manuscript, 1994. URL `http://www.cds.caltech.edu/~murray/mlswiki/index.php/First_edition`.

[217] R. Nair and J.H. Maddocks. On the forward kinematics of parallel manipulators. *The International Journal of Robotics Research*, 13(2):171–188, 1994.

[218] A. Neumaier. *Interval methods for systems of equations*, vol. 37. Cambridge university press, 1991.

[219] A. Neumaier. Constraint satisfaction and optimization in robotics. Manuscript (unpublished), 2003. URL `http://www.mat.univie.ac.at/~neum/ms/robslides.pdf`.

[220] A. Neumaier. All about 3-dimensional rotations. Manuscript, 2008.

[221] L. Nie and Q. Huang. Inverse kinematics for 6-DOF manipulator by the method of sequential retrieval. *Proceedings of 2012 International Conference of Mechanical Engineering and Material Science (MEMS 2012)*, pp. 255 – 258, 2012.

[222] J. Nielsen and B. Roth. On the kinematic analysis of robotic mechanisms. *The International Journal of Robotics Research*, 18(12):1147–1160, 1999.

[223] J. Nielsen and B. Roth. Solving the input/output problem for planar mechanisms. *Journal of Mechanical Design*, 121(2):206–211, 1999.

[224] S. Oren and D. Luenberger. Self-Scaling Variable Metric (SSVM) Algorithms Part I: Criteria and Sufficient Conditions for Scaling a Class of Algorithms. *Management Science*, 20(5):845–862, 1974.

[225] E. Ottaviano, M. Ceccarelli, and M. Husty. Workspace topologies of industrial 3R manipulators. *International Journal of Advanced Robotic Systems*, 4(3):355–364, 2007.

[226] S. Panda, D. Mishra, and B. Biswal. Revolute manipulator workspace optimization: A comparative study. *Applied Soft Computing*, 2012.

[227] V. Parenti-Castelli and R. Di Gregorio. Determination of the actual configuration of the general Stewart platform using only one additional sensor. *Journal of Mechanical Design*, 121(1):21–25, 1999.

[228] V. Parenti-Castelli and R. Di Gregorio. A new algorithm based on two extra-sensors for real-time computation of the actual configuration of the generalized Stewart-Gough manipulator. *Journal of Mechanical Design*, 122 (3):294–298, 2000.

[229] V. Parenti-Castelli and R. Di Gregorio. Real-time actual pose determination of the general fully parallel spherical wrist, using only one extra sensor. *Journal of Robotic Systems*, 18(12):723–729, 2001.

[230] P.J. Parikh and S.S.Y. Lam. A hybrid strategy to solve the forward kinematics problem in parallel manipulators. *Robotics, IEEE Transactions on*, 21(1):18–25, 2005.

[231] F.C. Park and J.W. Kim. Singularity analysis of closed kinematic chains. *Journal of Mechanical Design*, 121(1):32–38, 1999.

[232] A.J. Patel and K.F. Ehmann. Volumetric error analysis of a Stewart platform-based machine tool. *CIRP Annals-Manufacturing Technology*, 46 (1):287–290, 1997.

[233] Y.D. Patel, P.M. George, et al. Parallel manipulators applications—A survey. *Modern Mechanical Engineering*, 2(3):57–64, 2012.

[234] P. Pedersen and B. Sturmfels. Product formulas for resultants and Chow forms. *Mathematische Zeitschrift*, 214(1):377–396, 1993.

[235] A. Perez and J.M. McCarthy. Dual quaternion synthesis of constrained robotic systems. *J. Mechanical Design*, 126:425–435, 2004.

[236] F. Pernkopf and M.L. Husty. Workspace analysis of Stewart-Gough-type parallel manipulators. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 220(7):1019–1032, 2006.

[237] P.B. Petrović and V.R. Milačić. Closed-form resolution scheme of the direct kinematics of parallel link systems based on redundant sensory information. *CIRP Annals-Manufacturing Technology*, 48(1):341–344, 1999.

[238] V. Petuya, A. Alonso, O. Altuzarra, and A. Hernández. Resolution of the direct position problem of parallel kinematic platforms using the geometrical-iterative method. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3244 – 3249, 2005.

[239] V. Petuya, J.M. Gutierrez, A. Alonso, O. Altuzarra, and A. Hernández. A numerical procedure to solve non-linear kinematic problems in spatial mechanisms. *International Journal for Numerical Methods in Engineering*, 73(6):825–843, 2008.

[240] M. Pfurner. *Analysis of spatial serial manipulators using kinematic mapping.* PhD thesis, University of Innsbruck, Austria, 2006.

[241] M. Pfurner and M.L. Husty. Implementation of a new and efficient algorithm for the inverse kinematics of serial 6R chains. *New Trends in Mechanism Science: Analysis and Design*, 5:91, 2010.

[242] D.L. Pieper. *The Kinematics of manipulators under Computer control.* PhD thesis, Standford University, 1968.

[243] F. Pierrot, P. Dauchez, and A. Fournier. HEXA: a fast six-DOF fully-parallel robot. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, vol. 2, pp. 1158 –1163, 1991.

[244] G. Pond and J.A. Carretero. Formulating Jacobian matrices for the dexterity analysis of parallel manipulators. *Mechanism and Machine Theory*, 41(12): 1505–1519, 2006.

[245] G. Pond and J.A. Carretero. Quantitative dexterous workspace comparison of parallel manipulators. *Mechanism and Machine Theory*, 42(10):1388–1400, 2007.

[246] G. Pond and J.A. Carretero. Dexterity measures and their use in quantitative dexterity comparisons. *Meccanica*, 46(1):51–64, 2011.

[247] J.M. Porta, L. Ros, and F. Thomas. A linear relaxation technique for the position analysis of multiloop linkages. *Robotics, IEEE Transactions on*, 25 (2):225–239, 2009.

[248] A. Pott and M. Hiller. *Kinematic modeling, linearization and first-order error analysis*, chapter 8. InTech, 2008.

[249] E.J.F. Primrose. On the input-output equation of the general 7R-mechanism. *Mechanism and Machine Theory*, 21(6):509–510, 1986.

[250] S. Qiao, Q. Liao, S. Wei, and H.J. Su. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions. *Mechanism and Machine Theory*, 45(2):193–199, 2010.

[251] M. Raghavan. The Stewart platform of general geometry has 40 configurations. *Journal of Mechanical Design*, 115(2):277–282, 1993.

[252] M. Raghavan and B. Roth. Kinematic analysis of the 6R manipulator of general geometry. In *The Fifth International Symposium on Robotics Research*, pp. 263–269. MIT Press, 1991.

[253] M. Raghavan and B. Roth. Solving polynomial systems for the kinematic analysis and synthesis of mechanisms and robot manipulators. *Journal of Mechanical Design*, 117(B):71–79, 1995.

[254] R. Ramamoorthi and A.H. Barr. Fast construction of accurate quaternion splines. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 287–292. ACM Press/Addison-Wesley Publishing Co., 1997.

[255] A. C. Rao. Parallelism in planar kinematic chains (manipulators). *Mechanism and Machine Theory*, 39(10):1111 – 1122, 2004.

[256] L. Ren, Z. Feng, and J.K. Mills. A self-tuning iterative calculation approach for the forward kinematics of a Stewart-Gough platform. In *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, pp. 2018–2023. IEEE, 2006.

[257] C.R. Rocha, C.P. Tonetto, and A. Dias. A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. *Robotics and Computer-Integrated Manufacturing*, 27 (4):723–728, 2011.

[258] J. Rodriguez and M. Ruggiu. Analytical solution of the forward displacement problem for spherical parallel manipulators. *Arxiv preprint arXiv:1203.6023*, 2012.

[259] N. Rojas and F. Thomas. Closed-form solution to the position analysis of Watt–Baranov trusses using the bilateration method. *Journal of Mechanisms and Robotics*, 3(3):031001, 2011.

[260] N. Rojas and F. Thomas. Distance-based position analysis of the three seven-link Assur kinematic chains. *Mechanism and Machine Theory*, 46(2): 112–126, 2011.

[261] N. Rojas and F. Thomas. On closed-form solutions to the position analysis of Baranov trusses. *Mechanism and Machine Theory*, 50:179–196, 2012.

[262] L. Rolland. Certified solving of the forward kinematics problem with an exact algebraic method for the general parallel manipulator. *Advanced Robotics*, 19(9):995–1025, 2005.

[263] F. Ronga and T. Vust. Stewart platforms without computer? In *Real Analytic and Algebraic Geometry: Proceedings of the International Conference, Trento (Italy), September 21-25th, 1992*, p. 197. De Gruyter, 1995.

[264] B. Roth and F. Freudenstein. Synthesis of path-generating mechanisms by numerical methods. *Journal of Engineering for Industry*, 85:298, 1963.

[265] B. Roth, J. Rastegar, and V. Scheinmann. On the design of computer controlled manipulators. In *On theory and practice of robots and manipulators: first CISM-IFToMM Symposium, 5-8 September, 1973*, vol. 1, p. 93. Springer, 1974.

[266] F. Rouillier. Real roots counting for some robotics problems. In *Computational Kinematics' 95*, vol. 40, pp. 73–82, 1995.

[267] S. Russell and P. Norvig. *Artificial intelligence: A Modern appraoch*. Prentice-Hall, 2010.

[268] M.H. Saadatzi, M.T. Masouleh, H.D. Taghirad, C. Gosselin, and P. Cardou. On the optimum design of 3-RPR parallel mechanisms. In *Electrical Engineering (ICEE), 2011 19th Iranian Conference on*, pp. 1–6. IEEE, 2011.

[269] H. Sadjadian and H.D. Taghirad. Numerical methods for computing the forward kinematics of a redundant parallel manipulator. In *Proceedings of the IEEE Conference on Mechatronics and Robotics, Aachen, Germany*, pp. 557–562, 2004.

[270] H. Sadjadian, H.D. Taghirad, and A. Fatehi. Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator. *International Journal of Computational Intelligence*, 2(1):40–47, 2005.

[271] J.K. Salisbury and J.J. Craig. Articulated hands force control and kinematic issues. *The International Journal of Robotics Research*, 1(1):4–17, 1982.

[272] S. Saramago, E. Ottaviano, and M. Ceccarelli. A characterization of the workspace boundary of three-revolute manipulators. In *Design Engineering Technical Conferences (DETC'02)*, pp. 34342–34352, 2002.

[273] H. Schichl and M.C. Márkót. Algorithmic differentiation techniques for global optimization in the COCONUT environment. *Optimization Methods and Software*, 27(2):359–372, 2012.

[274] H. Schichl, M.C. Márkót, and A. Neumaier. Exclusion regions for optimization problems. In *7th International Conference on Computational Management Science*, p. 30, 2010.

[275] J.M. Selig. Geometric fundamentals of robotics. *Monographs in Computer Science*, 2005.

[276] H. Shah, M.S. Narayanan, and V.N. Krovi. CAD-enhanced workspace optimization for parallel manipulators: A case study. In *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pp. 816–821. IEEE, 2010.

[277] L. Shi, M.H. Xu, and B. Liu. A fast algorithm for workspace of a robotic manipulator. *Advanced Materials Research*, 538:2614–2617, 2012.

[278] X. Shi and R.G. Fenton. Solution to the forward instantaneous kinematics for a general 6-DOF Stewart platform. *Mechanism and Machine Theory*, 27 (3):251–259, 1992.

[279] K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH computer graphics*, 19(3):245–254, 1985.

[280] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2008.

[281] N. Sima'an, D. Glozman, and M. Shoham. Design considerations of new six degrees-of-freedom parallel robots. In *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1327–1333. IEEE, 1998.

[282] A.J. Sommese, J. Verschelde, and C.W. Wampler. Numerical decomposition of the solution sets of polynomial systems into irreducible components. *SIAM Journal on Numerical Analysis*, 38(6):2022–2046, 2001.

[283] A.J. Sommese, J. Verschelde, C.W. Wampler, et al. Advances in polynomial continuation for solving problems in kinematics. *Trans. ASME Journal of Mechanical Design*, pp. 262–268, 2004.

[284] A.J. Sommese and C.W. Wampler. Numerical algebraic geometry. *Lectures in Applied Mathematics - American Mathematical Society*, 32:749–764, 1996.

[285] A.J. Sommese and C.W. Wampler. *The numerical solution of systems of polynomials arising in engineering and science*. World Scientific Publishing Company Incorporated, 2005.

[286] M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Dynamics and Control*. 2nd ed., 2004.

[287] R.S. Stoughton and T. Arai. A modified Stewart platform manipulator with improved dexterity. *Robotics and Automation, IEEE Transactions on*, 9(2): 166–173, 1993.

[288] E. Study. *Geometrie der Dynamen*. Teubner: Leipzig, 1903.

[289] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6:422–430, 1964.

[290] H.J. Su, J.M. McCarthy, M. Sosonkina, and L.T. Watson. Algorithm 857: POLSYS_GLP–a parallel general linear product homotopy code for solving polynomial systems of equations. *ACM Transactions on Mathematical Software (TOMS)*, 32(4):561–579, 2006.

[291] M. Tandirci, J. Angeles, and F. Ranjbaran. The characteristic point and the characteristic length of robotic manipulators. *Proceedings of ASME 22nd Biennial Conf. Robotics, Spatial Mechanisms & Mech. Sys., Scotsdale, Arizona*, 45:203–208, 1992.

[292] T.K. Tanev. Kinematics of a hybrid (parallel–serial) robot manipulator. *Mechanism and Machine Theory*, 35(9):1183–1196, 2000.

[293] H. Tari, H.J. Su, and T.Y. Li. A constrained homotopy technique for excluding unwanted solutions from polynomial equations arising in kinematics problems. *Mechanism and Machine Theory*, 45(6):898–910, 2010.

[294] M. Tarokh. Real time forward kinematics solutions for general Stewart platforms. In *Robotics and Automation, 2007 IEEE International Conference on*, pp. 901–906. IEEE, 2007.

[295] K.L. Ting, J. Zhu, and D. Watkins. The effects of joint clearance on position and orientation deviation of linkages and manipulators. *Mechanism and Machine Theory*, 35(3):391–401, 2000.

[296] K.Y. Tsai and J.C. Lin. Determining the compatible orientation workspace of Stewart-Gough parallel manipulators. *Mechanism and Machine Theory*, 41(10):1168–1184, 2006.

[297] K.Y. Tsai and S.R. Zhou. The optimum design of 6-DOF isotropic parallel manipulators. *Journal of Robotic Systems*, 22(6):333–340, 2005.

[298] L.W. Tsai. *Robot analysis: the mechanics of serial and parallel manipulators*. Wiley-Interscience, 1999.

[299] L.W. Tsai and A.P. Morgan. Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods. *Journal of Mechanisms Transmissions and Automation in Design*, 107(2):189–200, 1985.

[300] Y.C. Tsai and A.H. Soni. Workspace synthesis of 3R, 4R, 5R and 6R robots. *Mechanism and Machine Theory*, 20(6):555–563, 1985.

[301] B.L. Van der Waerden. *Modern Algebra*, vol. 2. Frederick Ungar, 1950.

[302] S.M. Varedi, H.M. Daniali, and D.D. Ganji. Kinematics of an offset 3-UPU translational parallel manipulator by the homotopy continuation method. *Nonlinear Analysis: Real World Applications*, 10(3):1767–1774, 2009.

[303] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. *ACM Transactions on Mathematical Software (TOMS)*, 25(2):251–276, 1999.

[304] J. Verschelde. Polynomial homotopy continuation with PHCpack. *ACM Communications in Computer Algebra*, 44(3/4):217–220, 2011.

[305] R. Vertechy and V. Parenti-Castelli. Real-time direct position analysis of parallel spherical wrists by using extra sensors. *Trans. ASME Journal of Mechanical Design*, 128(1):288, 2006.

[306] R. Vertechy and V. Parenti-Castelli. Accurate and fast body pose estimation by three point position data. *Mechanism and Machine Theory*, 42(9):1170 – 1183, 2007.

[307] R. Vertechy and V. Parenti-Castelli. Robust, fast and accurate solution of the direct position analysis of parallel manipulators by using extra-sensor. In *Parallel Manipulators: Towards New Applications*. Huapeng Wu, 2008.

[308] P. Voglewede and I. Ebert-Uphoff. Overarching framework for measuring closeness to singularities of parallel manipulators. *Robotics, IEEE Transactions on*, 21(6):1037–1045, 2005.

[309] C.W. Wampler. Bezout number calculations for multi-homogeneous polynomial systems. *Applied Mathematics and Computation*, 51(2):143–157, 1992.

[310] C.W. Wampler. Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates. *Mechanism and Machine Theory*, 31(3):331–337, 1996.

[311] C.W. Wampler. Solving the kinematics of planar mechanisms by Dixon determinant and a complex-plane formulation. *Journal of Mechanical Design*, 123(3):382–387, 2001.

[312] C.W. Wampler. Displacement analysis of spherical mechanisms having three or fewer loops. *ASME Journal of Mechanical Design*, 126(1):93–100, 2004.

[313] C.W. Wampler and A.J. Sommese. Numerical algebraic geometry and algebraic kinematics. *Acta Numerica*, 20(1):469–567, 2011.

[314] H. Wang, G. Chen, Y. Zhao, and Z. Lin. Output error bound prediction of parallel manipulators based on the level set method. *Mechanism and Machine Theory*, 45(8):1153–1170, 2010.

[315] S. Wang and Y. Wan. A mixed real-time algorithm for the forward kinematics of Stewart parallel manipulator. *Journal of Electronic Science and Technology of China Vol*, 4:173 – 180, 2006.

[316] S.L. Wang and K.J. Waldron. A study of the singular configurations of serial manipulators. *Journal of Mechanisms, Transmissions, and Automation in Design*, 109(1):14–20, 1987.

[317] S.M. Wang and K.F. Ehmann. Error model and accuracy analysis of a six-DOF Stewart platform. *Trans. ASME Journal of Manufactoring Science and Engineering*, 124(2):286–295, 2002.

[318] X.S. Wang, M.L. Hao, and Y.H. Cheng. On the use of differential evolution for forward kinematics of parallel manipulators. *Applied Mathematics and Computation*, 205(2):760–769, 2008.

[319] Y. Wang. A direct numerical solution to forward kinematics of general Stewart-Gough platforms. *Robotica*, 25(1):121, 2007.

[320] L.T. Watson, M. Sosonkina, R.C. Melville, A.P. Morgan, and H.F. Walker. Algorithm 777: HOMPACK90: A suite of Fortran 90 codes for globally convergent homotopy algorithms. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):514–549, 1997.

[321] M. Weck and D. Staimer. Parallel kinematic machine tools–current state and future potentials. *CIRP Annals-Manufacturing Technology*, 51(2):671–683, 2002.

[322] C.E. Wee and R.N. Goldman. Elimination and resultants. 1. Elimination and bivariate resultants. *Computer Graphics and Applications, IEEE*, 15 (1):69–77, 1995.

[323] C.E. Wee and R.N. Goldman. Elimination and resultants. 2. Multivariate resultants. *Computer Graphics and Applications, IEEE*, 15(2):60–69, 1995.

[324] P. Wenger, D. Chablat, et al. Uniqueness domains in the workspace of parallel manipulators. *Syroco'97*, pp. 1–6, 1997.

[325] P. Wenger, D. Chablat, et al. Workspace and assembly modes in fully-parallel manipulators: A descriptive study. *Advances in Robot Kinematics and Computational Geometry*, pp. 117–126, 1998.

[326] N.L. White. Grassmann–Cayley algebra and robotics. *Journal of Intelligent and Robotic Systems*, 11(1-2):91–107, 1994.

[327] J.H. Wilkinson. Modern error analysis. *SIAM review*, 13(4):548–568, 1971.

[328] J.H. Wilkinson. *Rounding errors in algebraic processes.* Dover Publications, 1994.

[329] S.M. Wise, A.J. Sommese, and L.T. Watson. Algorithm 801: POLSYS_PLP: A partitioned linear product homotopy code for solving polynomial systems of equations. *ACM Transactions on Mathematical Software (TOMS)*, 26(1): 176–200, 2000.

[330] T.M. Wu. A study of convergence on the Newton-homotopy continuation method. *Applied Mathematics and Computation*, 168(2):1169–1174, 2005.

[331] T.M. Wu. Searching all the roots of inverse kinematics problem of robot by homotopy continuation method. *Applied Sciences*, 5:666–673, 2005.

[332] T.M. Wu. Solving the nonlinear equations by the Newton-homotopy continuation method with adjustable auxiliary homotopy function. *Applied Mathematics and Computation*, 173(1):383–388, 2006.

[333] W. Wu. *Mechanical theorem proving in geometries: Basic principles.* Springer Verlag, 1994.

[334] W. Wu and S.S. Rao. Uncertainty analysis and allocation of joint tolerances in robot manipulators based on interval analysis. *Reliability Engineering & System Safety*, 92(1):54–64, 2007.

[335] D.C.H. Yang and T.W. Lee. Heuristic combinatorial optimization in the design of manipulator workspace. *Systems, Man and Cybernetics, IEEE Transactions on*, (4):571–580, 1984.

[336] G. Yang and I.M. Chen. Equivolumetric partition of solid spheres with applications to orientation workspace analysis of robot manipulators. *Robotics, IEEE Transactions on*, 22(5):869–879, 2006.

[337] L. Yang, X. Hou, and Z. Zeng. A complete discrimination system for polynomials. *Science in China Series Technological Sciences-English Edition*, 39: 628–646, 1996.

[338] T. Yoshikawa. Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, 1985.

[339] A. Yu, I.A. Bonev, and P. Zsombor-Murray. Geometric approach to the accuracy analysis of a class of 3-DOF planar parallel robots. *Mechanism and Machine Theory*, 43(3):364–375, 2008.

[340] W.I. Zangwill and C.B. Garcia. *Pathways to solutions, fixed points, and equilibria*. Prentice-Hall, Englewood Cliffs, N.J., 1981.

[341] S. Zarkandi. Kinematics of a star-triangle planar parallel manipulator. *Journal of Mechanical Science and Technology*, 25(12):3223–3230, 2011.

[342] M. Zefran and V. Kumar. Planning of smooth motions on $SE(3)$. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1, pp. 121–126. IEEE, 1996.

[343] M. Zein, P. Wenger, and D. Chablat. An exhaustive study of the workspace topologies of all 3R orthogonal manipulators with geometric simplifications. *Mechanism and Machine Theory*, 41(8):971–986, 2006.

[344] D. Zhang and J. Lei. Kinematic analysis of a novel 3-DOF actuation redundant parallel manipulator using artificial intelligence approach. *Robotics and Computer-Integrated Manufacturing*, 27(1):157–163, 2011.

[345] G.F. Zhang. Classification of direct kinematics to planar generalized Stewart platforms. *Computational Geometry*, 45(8):458–473, 2012.

[346] J.S. Zhao, M. Chen, K. Zhou, J.X. Dong, and Z.J. Feng. Workspace of parallel manipulators with symmetric identical kinematic chains. *Mechanism and Machine Theory*, 41(6):632–645, 2006.

[347] J.S. Zhao, F. Chu, and Z.J. Feng. Symmetrical characteristics of the workspace for spatial parallel mechanisms with symmetric structure. *Mechanism and Machine Theory*, 43(4):427–444, 2008.

[348] D. Zlatanov, R.G. Fenton, and B. Benhabib. A unifying framework for classification and interpretation of mechanism singularities. *Journal of Mechanical Design*, 117:566, 1995.

[349] D. Zlatanov, R.G. Fenton, and B. Benhabib. Classification and interpretation of singularities of redundant mechanisms. In *Proceedings of the 24th ASME Annual Design Automation Conference (DETC98)*, pp. 1–11, 1998.

[350] P.J. Zsombor-Murray. An improved approach to the kinematics of Clavel's DELTA robot. Online, 2009. URL `http://www.cim.mcgill.ca/~paul/Delta9Af.pdf`.

Curriculum Vitae

Personal Data

Name	**Bettina Ponleitner**
Academic Title	**Bakk. rer. nat.**
Date of Birth	**March 12, 1984**
Place of Birth	**Mödling**, Lower Austria
Contact	`bettina.ponleitner@gmx.at`

Education

1990–1994	**Primary School**, Hinterbrühl, Lower Austria
1994–1998	**Junior High School**, BG/BRG Bachgasse, Mödling
1998–2002	**Senior High School**, BG Perchtoldsdorf, Lower Austria
06/2002	**Matura** (with distinction)
2002–2003	**Study of Mathematics/Chemistry** (Teacher Training Program)
2004–2009	**Study of Mathematics/Sport Science** (Teacher Training Program)
2002–2013	**Study of Mathematics**, University of Vienna Diploma Thesis: *Fundamental Mathematical Concepts for Solving Problems Arising in Robotics* (advisor: Hermann Schichl)
2004–2008	**Bachelor study of Sport Science** (Bakk. rer. nat.), University of Vienna Bachelor Theses in Applied Computer Science (advisor: Arnold Baca) and Sport Ethics (advisor: Konrad Kleiner)
2009–2013	**Study of Sport Science**, University of Vienna Diploma Thesis: *LESSi: Lernmodul zur Einführung in die Statistk und Sportinformatik–Entwicklung eines E-learning-Tools als Unterstützung zur Prüfungsvorbereitung* (advisor: Arnold Baca)

Scientific Talk/Publication

09/2010	**8. Symposium der dvs-Sektion Sportinformatik**, Darmstadt, Germany
2011	*Modellierung von Sportspielen mittels Random Walk am Beispiel der Rückschlagsportart Badminton.* In: Link & Wiemayer (Hrsg.) Sportinformatik trifft Sporttechnologie, Schriften der Dt. Vereinigung für Sportwissenschaft, Bd. 217. S. 191–195.

Employment

2012	**Tutor**, Faculty of Mathematics, University of Vienna
2012–2013	**Development of Multiple Choice Exams**, Institute of Sport Science, Department for Biomechanics, Kinesiology, and Applied Computer Science
2010–2011	**Teacher for Mathematics**, Schulzentrum für ganzheitliches Lernen, Enzesfeld, Lower Austria
2010	**Study Assistent (Studienassistentin)**, Institute of Sport Science, Department for Biomechanics, Kinesiology, and Applied Computer Science
Since 2002	**Private Tutor for Mathematics and Latin**