TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DISSERTATION

Singularity Distance Computations of Parallel Manipulators of Stewart-Gough Type

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Georg NAWRATIL

E104 Institut für Diskrete Mathematik und Geometrie

eingereicht an der Technischen Universität Wien

Fakultät für Mathematik und Geoinformation

von

Aditya KAPILAVAI
11916484

Wien, am 05. September 2024

REVIEWERS

Prof. Dr. Jonathan Hauenstein
201E Crowley Hall
Department of Applied and Computational Mathematics and Statistics
University of Notre Dame
Notre Dame, IN 46556
USA

Prof. Dr. Philippe Wenger
Directeur de recherche CNRS
Laboratoire des Sciences du Numérique de Nantes
BP 92101 44321, Nantes Cedex 3
France

Declaration of Authorship

I, Aditya Kapilavai, declare that this thesis titled, " Singularity Distance Computations of Parallel Manipulators of Stewart-Gough Type" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

In this cumulative PhD dissertation, we address the problems related to singularity distance computations for the Stewart-Gough type parallel robots, primarily focusing on theoretical analysis and computational algorithms.

Under the term parallel manipulators of the Stewart-Gough type, we summarize mechanisms, where the moving platform is connected to the base by a certain number of prismatic (P) legs according to the robot's degree of freedom. For planar mechanisms, the legs are anchored by passive revolute (R) joints, and for spatial ones by passive spherical (S) joints.

In so-called *singular* (also known as *shaky*) *configurations* these manipulators gain at least one instantaneous degree of freedom. Therefore, minor variations in the manipulator geometry (e.g. backlash in passive joints or uncertainties in the actuation of the P joints) can significantly affect the realized configuration. Near singular configurations, forces in prismatic actuators can become excessively large, potentially leading to the breakdown of the mechanism. Therefore, singular configurations and their vicinity should be avoided.

In this context, we consider 3-RPR manipulators and present a comparison of singular distances with respect to extrinsic and intrinsic metrics along a 1-parametric motion. Note that different metrics can be used depending on the chosen interpretations of the platform/base; e.g. as a triangular plate or as a pin-jointed triangular bar structure.

There also exist so-called *architecture singularities* referring to robot designs, which are shaky in every configuration. These designs have to be avoided but also their vicinity, as every anchor point can be associated with a space of uncertainties (e.g. tolerances in the passive joints or deviations of the platform/base from the geometric model). In this context, we consider linear pentapods (5-SPS manipulators with a linear platform) and present an approach to measure the distance of a given design from being architectural singular.

For both kinds of singularities, the distances are computed as the global minima of optimization problems. Their critical points are found through a generic computational pipeline that relies on algorithms from symbolic and numerical algebraic geometry implemented in `Maple`, `Bertini`, and `Paramotopy`. Note that we do not only obtain the singularity distance but also the corresponding closest singular configuration and architecture singularity, respectively.

This PhD thesis was funded by Grant No. P 30855-N32, from the Austrian Science Fund (FWF), devoted to evaluating the closeness of Stewart-Gough platforms to singularities.

Keywords: Singularity distance, Parallel manipulator, Architectural singularity, `Bertini`.

Kurzfassung

In dieser kumulativen Dissertation befassen wir uns mit den Problemen im Zusammenhang mit der Berechnung der Singularitätsdistanz für parallele Roboter vom Typ Stewart-Gough, wobei wir uns in erster Linie auf die theoretische Analyse und die Berechnungsalgorithmen konzentrieren.

Unter dem Begriff Parallelmanipulatoren des Stewart-Gough-Typs fassen wir Mechanismen zusammen, bei denen die bewegte Plattform über eine bestimmte Anzahl prismatischer (P) Beine entsprechend dem Freiheitsgrad des Roboters mit der Basis verbunden ist. Bei planaren Mechanismen sind die Beine durch passive Drehgelenke (R) verankert, bei räumlichen Mechanismen durch passive Kugelgelenke (S).

In sogenannten *singulären* (auch als *wackelig bezeichneten*) *Konfigurationen* erhalten diese Manipulatoren mindestens einen momentanen Freiheitsgrad. Daher können geringfügige Abweichungen in der Manipulatorgeometrie (z. B. Spiel in passiven Gelenken oder Ungenauigkeiten bei der Betätigung der P-Gelenke) die realisierte Konfiguration erheblich beeinflussen. In der Nähe singulärer Konfigurationen können die Kräfte in prismatischen Aktuatoren zu groß werden, was zum Versagen des Mechanismus führen kann. Daher sollten singuläre Konfigurationen und deren Umgebung vermieden werden.

In diesem Zusammenhang betrachten wir 3-RPR-Manipulatoren und präsentieren einen Vergleich der singulären Abstände in Bezug auf extrinsische und intrinsische Metriken entlang einer 1-parametrischen Bewegung. Man beachte, dass verschiedene Metriken verwendet werden können, je nach der gewählten Interpretationen der Plattform/Basis, z. B. als dreieckige Platte oder als gelenkig verbundene dreieckige Stabstruktur.

Es gibt auch sogenannte *Architektonische-Singularitäten*, die sich auf Roboterdesigns beziehen, die in jeder Konfiguration wackelig sind. Diese Konstruktionen müssen vermieden werden, aber auch ihre Umgebung, da jeder Ankerpunkt mit einem Raum von Ungenauigkeiten verbunden sein kann (z. B. Toleranzen in den passiven Gelenken oder Abweichungen der Plattform/Basis vom geometrischen Modell). In diesem Zusammenhang betrachten wir lineare Pentapoden und präsentieren einen Ansatz zur Messung des Abstands eines gegebenen Designs von der architektonischen Singularität.

Für beide Arten von Singularitäten werden die Abstände als globale Minima von Optimierungsproblemen berechnet. Ihre kritischen Punkte werden durch eine generische Berechnungspipeline gefunden, die sich auf Algorithmen der symbolischen und numerischen algebraischen Geometrie stützt, die in `Maple`, `Bertini` und `Paramotopy` implementiert sind. Es sei bemeht, dass wir nicht nur die Singularitätsdistanz, sondern auch die entsprechende nächstgelegene singuläre Konfiguration bzw. Architektursingularität erhalten.

Schlüsselwörter: Singularitätsabstand, Parallelmanipulator, Architektonische Singularität, Bertini.

"To me, mathematics, computer science, and the arts are insanely related.
They're all creative expressions."

– *Sebastian Thrun*

To my mother (Valli), father (Kumar), sister (Aparna), Jevgeni Ignasov, who always believed I could do anything, without whom this dissertation would not be possible.

Acknowledgements

I want to thank all who helped me throughout my life – those who helped me get to where I am today. While not a complete list, I hope to acknowledge some highlights.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, **Dr. Georg Nawratil**. Coming from an experimental robotics background, I had little experience with computational algebra, numerical algebraic geometry, and software tools such as `Maple` and `Bertini`. Many questions weighed on my mind as I began my Ph.D. journey: *Can I progress in this field? Can I write theoretical and computational research papers?* **Dr. Nawratil's** unwavering support has profoundly impacted my belief in myself. His encouragement, coupled with my determination, patience, persistence, and hard work, has been crucial to my growth and achievements.

During my Ph.D., I learned a lot from him, including how to write publications, conduct research, answer reviewer's questions, and comprehend and visualize complex geometric concepts. He spent his valuable time helping me prepare for conference presentations. He has consistently been there for me whenever I needed assistance, readily available to discuss and provide input, and sometimes even helped me debug my code in his busy schedule. His patience and willingness to address all my questions, even those related to high school-level geometry, were invaluable. He has been both a professional and personal inspiration, demonstrating that success is attainable even within the busy life of a scientist.

I would like to express my gratitude to **Prof. Jonathan Hauestein**, **Prof. Silviana Amethyst**, **Prof. Paul Breiding**, and **Aravind Baskar** for responding to my emails whenever I had questions regarding computational software `Bertini`, `HC.jl and Paramotopy` and technical concepts from numerical algebraic geometry, and for their expert advice in advancing my research.

I also extend my gratitude to **Prof. Ivan Izmestiev** for allowing me to remain in the workplace even after my contract had expired at TU Wien. I am also thankful to **Ronald Haidvogl** for supporting me in running my computations on the `Bertini` server and for always being available during the challenges posed by COVID-19 to fix the server when it experienced issues.

Next, I would like to extend my heartfelt gratitude to my special friends: **Jevegni Igansov**, **Paul Pospichal**, and **Roman Prosanov**, who provided not only emotional but also financial support, uplifting my spirits when I needed it most. During a challenging period when I fell ill and was hospitalized, they stood steadfastly by my side—an experience I will forever cherish in my life.

I am grateful to **Felix Dellinger**, a colleague in the research group. Despite not working in kinematics and robot geometry, he consistently shows enthusiasm for discussing mathematical concepts, particularly when I am reading publications to grasp complex ideas.

PhD life wouldn't have been so much fun without my special friends. I want to thank my beloved and kind friends in Wien, including **Adriano Spalveri**, **Davide Pellis**, **Luca San Mario**, **Paolo Carpeggiani**, **Fabian Doná**, **Giulio Coccia**, **Alessandra Bellisamo**, **Heinz Rossbacher**, **Thomas Kowatsch**, **Alberto Corella**, **Anya Nazarova** and **Alessandro Maddaloni** who made my time in Wien enjoyable and allowed me to explore new places.

I want to express my sincere gratitude to **Morenike Farag**, **Illa Mainali**, **Patrice Senard** for bringing me to wonderful places and introducing me to lovely and kind people in Wien. I also thank them for helping me with my visa extension and AMS issues in the past few months before graduation.

Lastly, I want to express my gratitude to my friend **Abhilash Uday Nayak** for discussions and guidance whenever it is required, from life to kinematics problems.

Contents

Chapter 1

Introduction

1.1 Topic and outline

A robot is an actuated mechanism composed of a series of (generally rigid) **links** connected by **joints** that allow relative motion between different parts. Actuation of some or all of the joints, typically using electric motors, enables the robot to move. The **End-Effector** (EE) is the part of the robot where the tool is attached, designed to interact with the environment, such as the gripper at the end of a robotic arm. A mechanical device that consists of two or more kinematic chains that connect the base to the end-effector in a closed loop, is also known as **Parallel Manipulator (PM).** The joints that can be controlled are called the **active joints**, while the **passive joints** are allowed to move freely.

Parallel manipulators of the Stewart-Gough (SG) type can be classified into three types based on their degrees of freedom and geometric design [33]: the hexapod and linear pentapod, which are spatial manipulators with six and five degrees of freedom (DoFs), respectively, and the 3-RPR planar parallel mechanism, which has three DoFs, as shown in Figure 1.1.

In the geometric design of hexapods, the moving platform is connected to the base by six spherical-prismatic-spherical (SPS) legs (see Figure 1.1(a)). Linear pentapods have a linear moving platform (lmp) connected to the base through five SPS legs (see Figure 1.1(b)). In the 3-RPR manipulator, the moving platform is connected to the base via three rotational-prismatic-rotational (RPR) legs (see Figure 1.1(c)). The prismatic joints are active in these manipulator designs, while the spherical and rotational joints are passive.

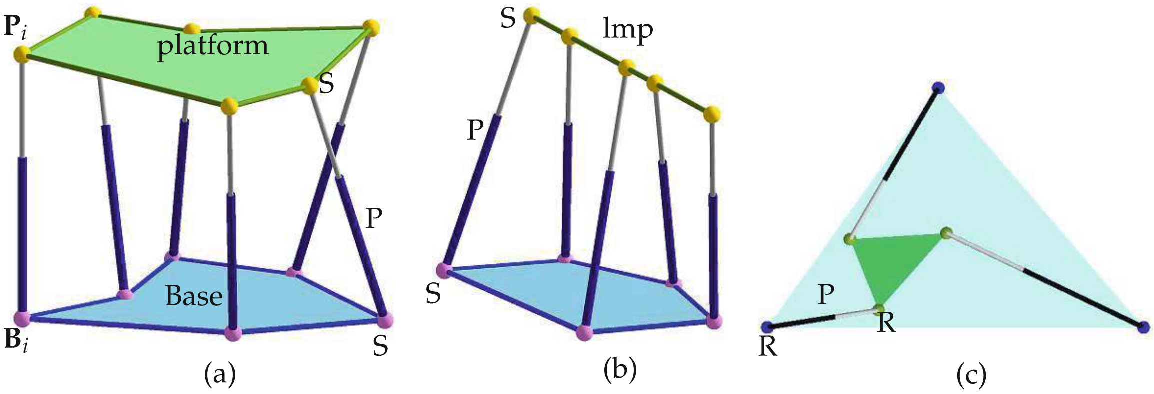

FIGURE 1.1: Sketch of a (a) hexapod [33] (b) linear pentapod [33] and (c) 3-RPR manipulator. For the planar mechanism as well as the spatial mechanical devices the anchor points of the legs are denoted by $\mathbf{B}_i$ (at the base) and $\mathbf{P}_i$ (at the platform).

The number of applications of parallel robots, ranging from medical surgery to astronomy, has increased enormously during the last decades due to their advantages of high speed, stiffness, large payload capacity, high precision, and load-to-weight ratio, among others [33].

1.1.1 Motivation and problem statement

Parallel manipulators (PMs) have several drawbacks, including a smaller workspace, high coupling in kinematic relationships, and the presence of singularities.

Singularities of PMs are classified into two types: Type I (serial singularities), where the manipulator loses an instantaneous DoF, and Type II (parallel singularities), where the manipulator gains an instantaneous DoF [27].

Note that PMs of SG type are free of Type I singularities due to the design of their legs. Therefore, this thesis only deals with Type II singularities, which can further be distinguished into **configuration-dependent** and **configuration-independent ones.** The latter are also known as architectural singularities [26] as they are related to robot designs, which are shaky in every configuration. This thesis addresses the problems of evaluating singularity distance computations related to Type II singularities.

Problem I: configuration-dependent singularities

When the manipulator is in a singular configuration, it gains infinitesimal degrees of freedom and can move slightly in certain directions, a phenomenon thus referred to as shakiness in the literature. Mathematically, this corresponds to the Jacobian matrix becoming rank-deficient, which relates the velocities of the actuators (in joint space) to the end-effector velocities (in configuration space).

Type II singularities can cause unpredictable movements and loss of control over the end effector, even if all prismatic joints are locked. Additionally, in the vicinity of these singularities, minor variations in the manipulator's geometry—such as backlash in passive joints or uncertainties in prismatic joint actuation—can significantly affect its configuration. Near singular configurations, forces in prismatic actuators can become excessively large, potentially leading to the breakdown of the mechanism. This case can be seen as an "ordinary" type II configuration-dependent singularity. In extreme cases, this can result in self-motion of the manipulator. Therefore, avoiding these singular configurations and their vicinity is crucial. Current methods for evaluating the closeness to a singularity in the literature are referred to as indices. A crucial issue is that these indices are not distance functions. Consequently, no conclusions can be drawn about the shape and size of a singularity-free region in the workspace and joint space around the given configuration based on these index values. A detailed literature survey on singularity closeness indices is presented in Section 1.1 of Chapter 3.

There are only a few approaches in the literature that use distance functions to determine the shape and size of such regions, but these approaches also have limitations. For a review on this topic, readers are referred to Sections 1.2–1.3 of Chapter 3 and Section 1.2 of Chapter 4 of this thesis.

Problem II: configuration-independent singularities

Architectural singularities are inherent to the design of the manipulator and are independent of its pose. These are manipulator designs that are shaky in every configuration of the workspace. Such designs should be avoided, along with their vicinity, as every anchor point can be associated with a space of uncertainties (e.g., tolerances in the passive joints or deviations of the platform/base from the geometric model). Therefore, there is a need for a metric allowing the evaluation of the distance of a given design to the closest architecture singularity. For a detailed review of the literature on this topic, readers are referred to Section 1.1 of Chapter 5 of this thesis.

To address **Problem I** and **Problem II**, we build upon the extensive prior work of Nawratil [31, 32, 33, 34, 35]. Based on his research, two distinct distance metrics can be identified for evaluating closeness to singularities.

1. Extrinsic metrics: The distance to the singularity is measured based on the metric in the embedding space, which is the space in which the robot is situated.
2. Intrinsic metrics: The distance to the singularity is measured based on the inner metric of the manipulator (e.g. lengths of the prismatic legs).

The two problems outlined above serve as the motivation for this thesis. In the context of **Problem I**, we consider 3-RPR manipulators and present a comparison of singular distances with respect to extrinsic and intrinsic metrics along a 1-parameter motion. Note that different metrics can be used depending on the chosen interpretations of the platform/base, such as a triangular plate or a pin-jointed triangular bar structure. In the context of **Problem II**, this thesis considers linear pentapods and presents an algorithmic approach to measure the extrinsic distance of a given design from being architecturally singular.

1.1.2 Key research objectives

The primary aim of this research is to address the challenges associated with Type II singularities in parallel manipulators by developing and applying novel methods for evaluating singularity distances. The key objectives of this thesis are:

- **Propose Distance Functions**: Formulate distance functions to evaluate the closeness to both configuration-dependent and independent type II singularities.
- **Formulate Optimization Problems**: Use these distance functions as the objective functions of the minimization problem to determine the closest singularity. In the case of constrained optimization problems, we use the Lagrangian approach. Our rigorous mathematical framework also incorporates the singular points of constraint varieties.
- **Develop Computational Algorithms**: For finding the global minimizer, we have to solve the system of polynomial equations resulting from taking the optimization function's partial derivatives. We focus on developing efficient and novel computational algorithms for determining this zero-set (critical points). For this purpose, we will use computational tools from algebraic geometry, computer algebra, and numerical algebraic geometry. Our primary focus will be on utilizing homotopy continuation algorithms implemented in the freeware packages `Bertini` [5] and `HC.jl` [10] to solve these optimization problems. Additionally, we aim to explore the functionalities of these two software packages with regard to their computational efficiency and their ability to handle our specific issues.
- **Demonstration**: Present numerical examples to demonstrate the computational procedure. For instance, we apply our method to the planar 3-RPR manipulator for configuration-dependent type II singularities and to a linear pentapod for architectural singularities.

- **Compare Metrics**: Evaluate and compare the proposed metrics for singularity assessment with the existing relevant performance indicies in the literature, highlighting our approach's advantages and potential improvements.

- **Application**: Demonstrate the application of the developed methods to more complex manipulators, and manipulator design optimization, showcasing their effectiveness and versatility in addressing real-world challenges.

The computation of singularity distances based on a well-defined metric is significant for quantifying how close the robot is to a singular configuration. This measure is critical for ensuring that the robot operates within safe and controllable regions, thereby preventing sudden failures. The singularity-free spheres and the associated closest singular configurations can be used for path optimization [40]. Additionally, it plays a key role in optimizing the design of the manipulator and in developing robust control algorithms that can handle near-singularity conditions.

1.1.3 Structure of the thesis

The motivation as mentioned above and the problem statement connect all four papers, which are the main contributions of this thesis and are published in peer-reviewed journals and conferences. The chronological order of the publications is as follows:

[A] Aditya Kapilavai and Georg Nawratil: On Homotopy Continuation Based Singularity Distance Computations for 3-RPR Manipulator. New Trends in Mechanism Science (D. Pisla, B. Corves eds.), pp. 56–64, 2020.

[B] Aditya Kapilavai and Georg Nawratil: Singularity distance computations for 3-RPR manipulators using extrinsic Metrics. Mechanism and Machine Theory, Vol. 195, pp. 105595, 2024.

[C] Aditya Kapilavai and Georg Nawratil: Singularity distance computations for 3-RPR manipulators using intrinsic Metrics. Computer Aided Geometric Design, Vol. 111, pp. 102343, 2024.

[D] Aditya Kapilavai and Georg Nawratil: Architecture Singularity Distance Computations for Linear Pentapods. ASME Journal of Mechanisms and Robotics, Vol. 17(2), pp. 021008, 2025.

These publications [A, B, C, D] are reprinted in Chapters 2–5, respectively.

Summary of the scientific publications and their contributions

- Chapter 2: To gain experience for the treatment of the mentioned spatial manipulators, paper [A] attempts to find minimal multi-homogeneous Bézout numbers for the homotopy continuation-based singularity distance computation with respect to various algebraic motion representations of planar Euclidean/equiform kinematics. We investigated algebraic motion representations by considering the 3-RPR manipulator as a case study. We divided these representations into two classes: non-homogeneous and homogeneous, to determine which representation results in the minimal Bézout bound. This study suggests that the point-based representation resulted in the minimal Bézout bound.

- Chapter 3: We proposed extrinsic metrics by taking the combinatorial structure of the 3-RPR manipulator into account as well as different design options. Utilizing these extrinsic metrics, we formulated constrained optimization problems. These problems are aimed at identifying the closest singular configurations for a given non-singular configuration. The solution to the associated system of polynomial equations relies on algorithms from numerical algebraic geometry implemented in the software package `Bertini`. A limitation of the approach presented in [33] concerns the computational efficiency, caused by Gröbner basis computations. As we aim to compute the singularity distance (incl. the closest singularity) along a one-parametric motion of the manipulator, we use the numerical algebraic geometry tool of homotopy continuation, which is implemented in the freeware `Bertini` and `paramotopy`. Moreover, the publication [B] also fills the gap that singular points of the singular variety are excluded from the Lagrangian approach. We identify these points, which are considered separately within the presented computational pipeline. The effectiveness of the presented approach is demonstrated based on numerical examples and compared with existing indices evaluating the singularity closeness.

  Finally, it should be noted that the presented method can also be applied to more complicated mechanisms, which can be abstracted into a joint composition of bars, triangular plates, and tetrahedral bodies. We demonstrated this for 3-RRR mechanisms. In addition, this example showed a further advantage of using the proposed metrics, namely that serial and parallel singularities can be treated uniformly.

  The supplementary materials, which contain animations and proofs of the theorems, can be downloaded from [23].

- Chapter 4: We proposed a novel algorithm for computing the singularity distance for the same design options as in Chapter 3 of the 3-RPR manipulator using intrinsic metrics. The algorithm utilizes intrinsic metrics based on the framework's total elastic strain energy density, employing the physical concept of Green-Lagrange strain. The constrained optimization problem for detecting the closest singular configuration with respect to these metrics is solved globally using tools from numerical algebraic geometry implemented in the software package `Bertini`. Additionally, we compared our method with the existing intrinsic singularity distances mentioned in the literature, highlighting its application in design optimization. We compared our intrinsic singularity distances with the corresponding extrinsic ones presented in Chapter 3.

  Note that the presented methods and applications are not restricted to 3-RPR manipulators used for the proof of concept in the publication [C], but can be applied to any mechanism/robot that can be considered as a pin-jointed body-bar framework according to [31], where e.g. Stewart Gough platforms were discussed in Section 6.

  The supplementary materials, which contain animations and proofs of the theorems, can be downloaded from [24].

- Chapter 5: We studied the architecture singularity distance computations of linear pentapods. We present the architecture singularity distance function, later we discuss how the corresponding optimization task can be managed by breaking it up into several minimization problems, which are related to the different classes of architectural singularity known in the literature [35]. This fragmentation of the computation reduces the maximal number of needed unknowns, which allows us to compute the global minimizer; i.e., the closest architecturally singular design. For some of the classes, we can give a geometric characterization of the corresponding minimizes. For the more involved classes, we resort to homotopy continuation computational algorithms implemented in the software `Beritini` and `HomotopyContinuation.jl` respectively. We discussed the developed computational pipeline for computing the architectural singularity distance. We presented two numerical examples, where the first design has a planar base and the second manipulator has a non-planar one. The first example is also compared with the existing approach.

The supplementary materials, which contain animations and proofs of the theorems, can be downloaded from [22].

Based on the contributions of this thesis, we can conclude that substantial progress has been made in evaluating the closeness of Type II singularities in parallel manipulators of the Stewart-Gough type. This advancement enhances real-time computational efficiency and global optimization techniques. Now, we will highlight a few interesting topics for future work and open problems related to the Chapters 2–5 of the thesis.

Future work and open problems

- The usage of paired intrinsic and extrinsic metrics, which were compared at the level of velocities, is also expected to contribute to sensitivity analysis [11, 16], as in this way one can quantify the change in the shape of the manipulator implied by variations in the inner geometry.
- In the Chapters 4 and 5, there also remain open computational issues, as we cannot guarantee the completeness of the solution set for some of the optimization problems in the ab-initio phase. They may be resolved by using isotropic coordinates, which facilitate the numerical solution of problems in planar kinematics [47].
- The relationship between the extrinsic and intrinsic singularity distances for 3-RPR configurations observed in Chapter 4 is not yet fully understood.
- In Chapter 5, the geometric interpretations for the minimizers of some classes of architectural singularity remain open. The development of an algorithm for design optimization using the information of the closest architectural singularity is devoted to future research.
- In the Chapters 3 and 5, the critical points obtained by the ab-initio phase using `Bertini` are validated heuristically but are not yet certified solutions. They can be verified using tools like alphaCertified [18] and interval arithmetic [9].
- Further open problems are the computation of extrinsic and intrinsic singularity distances for linear pentapods [38, 40] and hexapods. Moreover, one can attack the determination of architectural singularity distances for the design of hexapods.
- It is desirable to implement the proposed singularity distance computation algorithms in more sophisticated programming languages (e.g. C++, Python, or Julia) and to develop a graphical user interface for a given manipulator design, allowing the user to select the suitable metric to visualize the closest singular configuration.

In subsequent sections, we review the fundamentals and theoretical background of parallel robot kinematics, the properties and definitions of metrics, and tools from algebraic and numerical algebraic geometry that are rigorously used in the publications forming this cumulative PhD thesis.

1.2 Fundamentals and theoretical background

1.2.1 Fundamental tools from kinematics

In this subsection, we provide some fundamental concepts from kinematics. Kinematics is used to describe the motion of a system's links and joints, and kinematic analysis is the process of measuring the kinematic quantities used to describe this motion. For PMs, it implies establishing relations between the joint coordinates and the moving platform pose (position and orientation) [27]. We start with the hierarchy of transformations that are extensively used in the publications [A, B, C].

Transformations

Euclidean transformations are transformations of the n-dimensional space $\mathbb{R}^n$ preserve distances. They include translations, rotations, and reflections. A Euclidean transformation in n-dimensional space can be given as follows:

$$f : \mathbb{R}^n \to \mathbb{R}^n; \mathbf{x} \mapsto A\mathbf{x} + \mathbf{b} \tag{1.1}$$

where A is an orthogonal matrix from the orthogonal group $O(n)$, which represents either a rotation (if $\det(A) = 1$) or a reflection (if $\det(A) = -1$). The matrix A satisfies $AA^T = I$, and $\mathbf{b}$ is a translation vector $\in \mathbb{R}^n$.

Equiform/Similarity transformations combine scaling (resize), translation, and rotation. In n-dimensional space, a similarity transformation f is represented as:

$$f : \mathbb{R}^n \to \mathbb{R}^n, \mathbf{x} \mapsto sA\mathbf{x} + \mathbf{b}, \tag{1.2}$$

where s is a scalar factor.

Affine transformations of n-dimensional space are represented by an affine mapping f as:

$$f : \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbf{x} \mapsto A\mathbf{x} + \mathbf{b} \tag{1.3}$$

where A is an $n \times n$ matrix that can represent any linear transformation. Affine transformations preserve parallelism and ratios of distances along lines but not necessarily angles or lengths.

Inverse kinematics

Inverse kinematics establishes the values of actuated joint coordinates given the moving platform pose [27]. It is essential for the position control of PMs. For a given moving platform pose, the limbs of a PM can have different postures resulting in more than one solution to the inverse kinematics problem. These solutions are called working modes.

Parallel manipulators of SG type have only one working mode.

Forward/Direct kinematics

Direct kinematics involves determining the moving platform pose from the actuated joint coordinate values [27]. In general, this problem is computationally more expensive to solve than the inverse kinematic problem for parallel architectures. There are usually many solutions to this problem, meaning several ways exist to assemble the PMs. Therefore, they are also called assembly modes [27].

The number of assembly modes for a 3-RPR manipulator may be up to 6 [20, 21], whereas, for a hexapod, it is up to 40 [13]. For a pentapod, it is up to 8 solutions [49, 8, 35].

Plücker coordinates

Following H. Grassman and J. Plücker, the coordinates of a line $\mathcal{L}$ spanned by two points A and B with homogeneous coordinates $(a_0, \mathbf{a}) = (a_0, a_1, a_2, a_3)$ and $(b_0, \mathbf{b}) = (b_0, b_1, b_2, b_3)$, respectively, are given by their exterior product as follows [37]:

$$(a_0, a_1, a_2, a_3) \wedge (b_0, b_1, b_2, b_3) = (\mathbf{l}, \hat{\mathbf{l}}) = (p_{01}, p_{02}, p_{03}, p_{23}, p_{31}, p_{12}), \quad p_{ij} = a_i b_j - a_j b_i \quad (1.4)$$

where $\wedge$ denotes the wedge product. The six-tuple = $(p_{01}, p_{02}, p_{03}, p_{23}, p_{31}, p_{12})$ contains the Plücker coordinates of a line and they must satisfy the following Plücker identity:

$$p_{01}p_{23} + p_{02}p_{31} + p_{03}p_{12} = 0 \quad (1.5)$$

The vector $\mathbf{l}$ denotes the direction vector of $\mathcal{L}$ and the vector $\hat{\mathbf{l}}$ denotes the moment vector of the line with respect to the origin.

Using the line-geometric characterization, parallel manipulators of SG-type are in a singular configuration if and only if the Plücker coordinates of the involved legs are linearly dependent. For the specific manipulators mentioned, the singular configurations can be characterized as follows:

- **Hexapod**: The hexapod is in a singular configuration if and only if the six lines $l_1, \ldots, l_6$ belong to a linear line complex [33].
- **Linear Pentapod**: The linear pentapod is in a singular configuration if and only if the five lines $l_1, \ldots, l_5$ belong to a congruence of lines [40].
- **3-RPR Manipulator**: It is known that this planar analog of the hexapod is infinitesimally movable if and only if the three lines l_1, l_2, l_3 belong to a pencil of lines [33].

1.2.2 Distance metric

One of the primary focuses of this thesis is the development and application of a distance metric for evaluating the singularity closeness of parallel manipulators. In this section, we will review the concept of metrics, which is fundamental to our approach. The following concepts are derived from [42].

Scalar product

Given is an n- dimensional vector space $\mathbb{R}^n$ containing the vectors

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad \mathbf{0} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (1.6)$$

Then the standard scalar product (also called dot product) on $\mathbb{R}^n$ equals

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + \ldots u_nv_n \quad (1.7)$$

which can also be rewritten as

$$\mathbf{u}^T\mathbf{v} = \mathbf{u}^T \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & \ddots \\ & & & & 1 \end{bmatrix} \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle_I \tag{1.8}$$

One can replace the identity matrix I with any symmetric matrix G, which is a positive definite

$$\mathbf{u}^T G\mathbf{u} > 0 \quad \text{for all} \quad \mathbf{u} \in \mathbb{R}^n \setminus \{\mathbf{0}\}. \tag{1.9}$$

Remark 1. *A symmetric matrix G is positive definite if and only if all the eigenvalues of G (roots of det (G-λI)=0) are positive.* ◇

Then we write the scalar product induced by G as follows:

$$\langle \cdot, \cdot \rangle_G : \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R} \quad \text{with} \quad (\mathbf{u}, \mathbf{v}) \longmapsto \langle \mathbf{u}, \mathbf{v} \rangle_G := \mathbf{u}^T G\mathbf{v} \tag{1.10}$$

Norm and metric

A scalar product induces the following norm on the vector space $\mathbb{R}^n$:

$$|| \cdot || : \mathbb{R}^n \longrightarrow \mathbb{R}_{\geq 0} \quad \text{with} \quad \mathbf{u} \longmapsto ||\mathbf{u}||_G := \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_G} \tag{1.11}$$

A vector space on which a norm is defined is called a normed vector space. Furthermore, a norm induces the following metric (distance function) on the vector space $\mathbb{R}^n$:

$$d_G(.,.) : \quad \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}_{\geq 0} \quad \text{with} \quad (\mathbf{u}, \mathbf{v}) \longmapsto d_G(\mathbf{u}, \mathbf{v}) := ||\mathbf{u} - \mathbf{v}||_G \tag{1.12}$$

Remark 2. *A vector space equipped with a metric is a metric vector space.* ◇

More generally a metric (distance function) [42] can be defined on a set S as follows:

$$d : S \times S \mapsto \mathbb{R}_{\geq 0} \quad \text{with} \tag{1.13}$$

1. $d(u, v) \geq 0$ (non-negativity)
2. $d(u, v) = 0 \iff u = v$ (non-degeneracy)
3. $d(u, v) = d(v, u)$ (symmetry)
4. $d(u, v) \leq d(u, v) + d(v, w)$ (triangle inequality)

If a function only fulfills the properties 1, 3, and 4 then it is called a pseudometric. Therefore, a pseudometric $d(u, v)$ can be zero even if u and v are not identical.

1.2.3 Distance between poses

It is widely recognized [30, 36], that a bi-invariant (positive-definite) metric cannot be defined on the Special Euclidean group $SE(n)$. A metric d on $SE(n)$ is termed *bi-invariant* if it remains invariant under changes in both the fixed frame (left invariant) and the moving frame (right invariant). As a result, defining a geometrically meaningful distance between two poses is not feasible. This issue is further emphasized in [28], which states: *"Measuring closeness between a pose and a singular configuration is a difficult problem: there exists no mathematical metric defining the distance between a prescribed pose and a given singular pose. Hence, a certain level of arbitrariness must be accepted in the definition of the distance to a singularity . . . "*

In [36], it is mentioned that there is an approach to come up with a geometrically meaningful distance function, presenting an alternative to distance metrics on $SE(n)$ by changing the perspective as follows: One can consider the distance between two poses of the same rigid body, leading to so-called object-dependent metrics first studied by Kazerounian and Rastegar [25].

Concept of extrinsic metric

As the moving platform has m exceptional points (i.e., platform anchor points), it suggests measuring the distance between two poses of the moving platform (given pose $\mathbf{P}_i$ and transformed pose $\mathbf{P}_i^\alpha$) by the distance measure

$$d_m := \sqrt{\frac{1}{m}\sum_{i=1}^{m}\langle \mathbf{P}_i^\alpha - \mathbf{P}_i, \mathbf{P}_i^\alpha - \mathbf{P}_i\rangle_I} \tag{1.14}$$

Eq. (1.14) was used in [39] to compute the distance of a linear pentapod to the next singularity. However, this measure has a limitation: when we change our perspective by considering the platform as fixed and the base as the moving part (or vice versa), the calculated distance to the singularity generally changes, which is less desirable from a geometric perspective.

A more sophisticated approach based on the idea of transforming the base and platform anchor points simultaneously was first introduced in [33]. The new distance measure is given by:

$$D_m := \sqrt{\frac{1}{2m}\sum_{i=1}^{m}\left[\left\langle \mathbf{P}_i^\alpha - \mathbf{P}_i, \mathbf{P}_i^\alpha - \mathbf{P}_i\right\rangle_I + \left\langle \mathbf{B}_i^\beta - \mathbf{B}_i, \mathbf{B}_i^\beta - \mathbf{B}_i\right\rangle_I\right]} \tag{1.15}$$

where $\mathbf{B}_i^\beta$ denotes the transformed base points by the base transformation $\beta \in SE(n)$. Note that Eqs. (1.14) and (1.15) can be seen as metrics induced by a scalar product (cf. Section 1.2.2). Moreover, following the idea presented in Eq. (1.15), Chapter 5 of the thesis introduces the definition of an architectural singularity distance metric.

In the following subsection, we present the concept of intrinsic metrics by restricting to changes of the leg lengths only; i.e., using this metric, the distance is measured in the joint space instead of the configuration space.

Concept of intrinsic metric

Intrinsic metrics, particularly those based on total elastic strain energy density, are defined in [34, Section 3] based on the physical concept of Green-Lagrange strain. It can be defined as follows:

$$d\colon \mathbb{R}^b \times \mathbb{R}^b \to \mathbb{R}_{\geq 0} \quad \text{with} \quad (\mathbf{L}, \mathbf{L}') \mapsto d(\mathbf{L}, \mathbf{L}') := \frac{|u(\mathbf{L}) - u(\mathbf{L}')|}{E} \tag{1.16}$$

where $\mathbf{L} = (L_1, \ldots, L_m)$ and $\mathbf{L}' = (L_1', \ldots, L_m')$ represent undeformed and deformed leg lengths, respectively, with $L_i = \|\mathbf{P}_i - \mathbf{B}_i\|$ and E is the Young modulus. Eq. (1.16) defines a pseudometric on the m-dimensional joint space given by $\mathbf{L}$ and $\mathbf{L}'$. Notably, the pseudometric d is independent of the choice of E.

1.2.4 Preliminaries from algebraic geometry

In algebraic geometry, a system of polynomials is said to define a **polynomial ideal**, which consists of all the algebraic combinations of the generating polynomials. The geometric counterpart of an ideal is called an **algebraic variety**, consisting of the set of all the common (complex) roots of the polynomials in the ideal. The definitions of the fundamental concepts of an ideal and a algebraic variety can be found in the classical reference by Cox, Little, and O'Shea [12], so we will not repeat them here. There are other symbolic methods like resultants and excursions methods for this we refer to [44, Chapter 6]. In our research, we exclusively use Gröbner basis to analyze and solve polynomial systems. Here's a brief overview:

Gröbner basis

A Gröbner basis is a set of multivariate polynomials with desirable algorithmic properties. Every set of polynomials can be transformed into a Gröbner basis. This process generalizes three familiar techniques: Gaussian elimination for solving linear systems of equations, the Euclidean algorithm for computing the greatest common divisor of two univariate polynomials, and the simplex algorithm for linear programming [45]. In the following, we list the applied software packages that use the Gröbner basis algorithm for finding solutions to systems of polynomial equations.

- `Maple`: A mathematical software package for symbolic and numerical computing. In `Maple`, Gröbner bases are computed using the **Groebner[Basis]** function, which can handle various types of polynomial systems and monomial orders [12]. The process is flexible, allowing for the elimination of variables, solving systems, and performing advanced polynomial algebra.
- `msolve`: It uses advanced Gröbner basis algorithm to compute isolate all real solutions to polynomial systems with rational coefficients and finitely many complex solutions [7].

In this thesis, the author extensively used the computer algebraic software `Maple` to prove the theoretical arguments mentioned in Chapters 3, 4, and 5 of this thesis.

1.2.5 Optimization techniques

Our goal is to find the closest singular configuration to a given pose. This leads to a constrained optimization task, where the distance function serves as the objective, and the singularity condition acts as the constraint.

Iterative optimization techniques like Newton's method and gradient descent often struggle with these constraints. However, the method of Lagrange multipliers is particularly powerful for problems with explicit equality constraints, as it integrates these constraints c_i directly into the optimization process. The problem of minimizing a distance function using Lagrange multipliers λ_i can be defined as follows:

$$L = d + \sum_i \lambda_i c_i \tag{1.17}$$

By finding the partial derivatives of L with respect to the unknowns, we obtain a system of polynomial equations. Its solutions are also known as critical points. This method ensures that the constraints are satisfied exactly at these points.

However, an important limitation is that the method of Lagrange multipliers fails at singular points of the constraint variety [40, section 3.2.2]. Therefore, these points are treated separately in the Chapters 3 – 5 of this thesis.

Reason for global optimization

Global optimization is essential in our research to ensure that we identify the true minimum of our objective functions, rather than settling on local minima. This necessity becomes even more apparent when dealing with the high degree and complexity of the polynomial optimization problems we address in this thesis.

While symbolic methods such as Gröbner basis and dialytic elimination theoretically offer ways to find critical points for polynomial systems, these methods become computationally infeasible given the complexity of our problems. This limitation highlights the need for robust global optimization techniques.

Traditional optimization methods like gradient descent are prone to getting stuck in local minima, making them insufficient for our purposes. To overcome these challenges, this thesis primarily utilizes numerical continuation techniques based on homotopy continuation. We now provide a brief overview of the relevant concepts from numerical algebraic geometry to support our approach.

1.2.6 Preliminaries from numerical algebraic geometry

Numerical algebraic geometry's primary focus is on applying numerical techniques to algebraic geometry problems. At the heart of nonlinear algebra is algebraic geometry but there are also links to many other branches of mathematics. These include combinatorics, algebraic topology, commutative algebra, and discrete geometry. Nonlinear algebra connects these different branches of mathematics with a strong focus on computations and applications [44].

Homotopy continuation

Numerical homotopy continuation works based on a *homotopy*. This method solves a polynomial system of interest by creating a related system with known solutions, and then gradually transforming that system and its solutions into the system and solutions of interest. For the reader's understanding, we discuss a brief overview of this method. For more information on homotopy continuation and other numerical methods for solving polynomial systems, we refer to [6, 44].

Let us consider the system $F(\mathbf{x}) = 0$ of N polynomials in variables $\mathbf{x} \in \mathbb{C}^N$, given by:

$$F(\mathbf{x}) = \begin{pmatrix} f_1(x_1, \dots, x_N) \\ \vdots \\ f_N(x_1, \dots, x_N) \end{pmatrix} \tag{1.18}$$

Our goal is to compute the solutions to this system, our target system. Homotopy continuation utilizes a system $G(\mathbf{x})$, called the start system, that is similar to $F(\mathbf{x})$. The simplest start system is such that the degree of $g_i(x)$ is equal to the degree of $f_i(x)$, this is a total-degree homotopy. Homotopy continuation deforms $G(\mathbf{x})$ and its solution is set into $F(\mathbf{x})$. One a common type of homotopy used in literature is the *straight-line homotopy equation:*

$$H(\mathbf{x}, t) = (1 - t)F(\mathbf{x}) + tG(\mathbf{x}) = 0 \tag{1.19}$$

In this simple construction, t is a parameter that goes from 1 to 0, because floating point numbers are more accurate near the origin.

At $t = 1$, we know the solutions of $H(\mathbf{x}, 1) = G(\mathbf{x}) = 0$. As the function deforms continuously, the behavior of the features of this function, such as its zeros, is tracked. The zeros of the start system and the target system are referred to as starting points and target/end points, respectively.

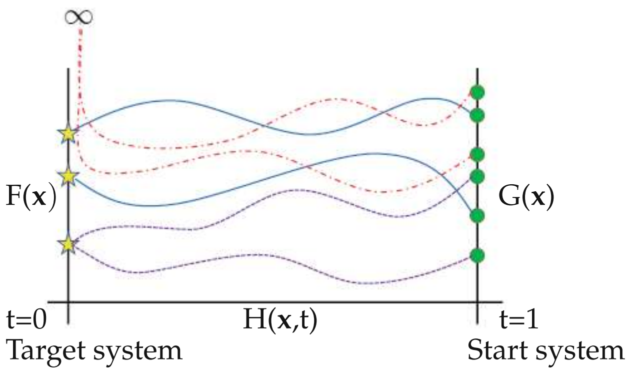

FIGURE 1.2: Pictorial representation of homotopy continuation [43].

Figure 1.2 shows the basic idea of the numerical homotopy continuation, the red dot-dash line indicates paths that diverge to infinity, the dashed purple line indicates paths that merge to a so-called singular solution as its multiplicity is greater than 1, and solid blue ends at distinct non-singular solutions.

Euler-Newton predictor-corrector is used to follow the paths for $t \in [1, 0]$ ending at the solutions to $H(\mathbf{x}, 0) = F(\mathbf{x}) = 0$. For more detailed basics on path tracking, we refer to [15, 4].

Various established techniques exist in polynomial homotopy continuation, including multi-homogeneous homotopy [29] (Bézout bound), spare polyhedral homotopy (BKK bound) [46], monodromy [14], and regeneration [17], among others can be used for finding the solutions to a system of polynomial equations. The Bézout bound and the BKK bound provide upper bounds on the number of common solutions (roots) for a system of polynomial equations. For a detailed discussion of these approaches, we refer to [6, 44].

Parameter homotopy and its advantages

Parameter homotopy is a technique used to solve families of polynomial systems that depend on parameters. This approach allows for efficient resolution of the polynomial system across different parameter sets (see [6, Chapter 6]). Consider the parameterized polynomial system given by:

$$F(\mathbf{x}; \mathbf{p}) = \begin{pmatrix} f_1(x_1, \ldots, x_N; p_1, \ldots, p_P) \\ f_2(x_1, \ldots, x_N; p_1, \ldots, p_P) \\ \vdots \\ f_N(x_1, \ldots, x_N; p_1, \ldots, p_P) \end{pmatrix} \tag{1.20}$$

where $\mathbf{x} = (x_1, \ldots, x_N) \in \mathbb{C}^N$ are the variables, and $\mathbf{p} = (p_1, \ldots, p_P) \in \mathbb{C}^P$ are the analytical parameters. Parameter homotopy comes in a two-step process:

Step 1 The first step in parameter homotopy is the *ab-initio phase*, which involves computing all the generic finite solutions of $F(\mathbf{x}; \mathbf{p}^\star) = 0$, where $\mathbf{p}^\star \in \mathbb{C}^P$ are randomly chosen values for the parameters. Although this initial computation requires significant effort, it is performed only once. Typically, multi-homogeneous homotopy and sparse polyhedral homotopy are employed in this phase to track both non-singular and singular roots. It is crucial to ensure that no path failures occur, as they can disrupt the tracking process.

Step 2 In this step, we construct a straight-line homotopy between the start and final parameters. The generic roots obtained from the *ab-initio phase* are used as starting points to track solutions for the actual system of interest. Note that Step 2 is just a special case of a user-defined homotopy.

Remark 3. *Theoretically, one could use random real numbers for the ab-initio phase. For `Bertini`, the main reason that random complex numbers are used is that in the parameter homotopy stage (ParameterHomotopy: 2), `Bertini` uses a straight-line homotopy between the start parameters and the final parameters. Over the real numbers, such a straight line homotopy often runs into singularities: e.g., $f(x;p) = x^2 - p = 0$ moving along a straight line from $p = 1$ to $p = -1$ hits a singularity at 0. However, if we move along a straight line from p = 1+i to p = -1, then we do not hit a singularity along the way [19].* ◇

1.2.7 Software for numerical algebraic geometry

There are several open-source software packages available for polynomial continuation. Some notable examples include `PHCpack`, `HOM4PS-2.0`, and `Macaulay2`. The most recent open-source numerical continuation software packages are `HC.jl` and `Bertini`. A comparison of these and other available software packages, along with their performance, can be found in [10]. We have chosen `Bertini 1.6v` for computing the singularity distance of parallel manipulators due to (a) its exceptional features highlighted in [48] and (b) the longevity of the `Bertini` software. We will now provide a brief overview of the functionality of this software.

In general, for a given system of polynomial equations as input, `Bertini` will automatically construct the start system $G(\mathbf{x})$ and the homotopy $H(\mathbf{x}, t)$. It offers multiple choices of predictor-corrector methods, with the classical 5^{th} order Runge-Kutta method typically used as the default configuration setting in `Bertini`. Depending on the complexity of the polynomial system, the next task is to consider tuning the configuration settings in the input file. This includes, among other things, `tracker settings`, `end game settings`, and `security settings`.

During tracking some paths will fail due to various numerical reasons. This can usually be fixed by adjusting settings and tolerances within `Bertini` input file configuration settings.

In our research, we are dealing with a parameterized system of polynomial equations that differ only in their coefficients, not in their monomials. This motivated us to use the advantage of parameter homotopy, as outlined in the previous section.

There is a special `Bertini` module called `Paramotopy` [3], which solves para–metrized polynomial systems in parallel, using `Bertini` as the underlying mathematical solver. It is a numeric menu-driven program, which runs both Step 1 and Step 2.

In Chapter 3, we proposed an alternative computational algorithm using the `paramotopy` for computing singularity distances using extrinsic metrics. In Chapter 4 of the thesis, in Section 4.1.2 we discussed alternative approaches involving `Paramotopy` for computing singularity distances using intrinsic metrics.

Comparison of `Bertini` and `HomotopyContinuation.jl`

We extensively used `Bertini` and `HomotopyContinuation.jl` throughout our research, depending on the problem at hand. We will now compare the features and limitations of these two software packages based on our experience and provide insights into potential improvements to enhance their computational efficiency.

Features of `Bertini`

- Offers a parallel (single machine multiple threads) version for the Linux operating system.
- Multiple choices of predictor-corrector methods.
- Equation-by-equation methods such as *regeneration homotopy*.
- Multiple configuration settings for secure path tracking.
- Adaptive multi-precision, various end games (tools to compute singular points), computing. Positive-dimensional sets are represented by witness sets.
- Parameter homotopies.

Limitations of `Bertini`

- Hassle for solving over-determined systems.
- Written in C, so have to rely on the input and output of the files. Generates a lot of intermediate files.
- Need a wrapper like Python to allow the user to interact with the core software.
- Currently there is no direct implementation of polyhedral or monodromy homotopies.
- In Bertini, the evaluation of the square root is consciously implemented only over $\mathbb{R}$ to avoid numerical instabilities near the branch cut [1].
- Solutions are validated heuristically, with no direct implementation to certify solutions. The program alpha-Certified [18], proving that candidates are approximate roots is intended to remedy this shortcoming.

Features of `HomotopyContinuation.jl`

- It is built in Julia, leveraging the language's performance and modern features.
- Polyhedral and monodromy solvers are implemented.
- Overdetermined systems can be solved.
- No separation between the computational core and a wrapper.
- Modular design allows users to extend and customize functionalities easily.
- Julia's just-in-time compilation provides the potential for high-performance computations, especially for large and complex systems.
- Wrapper is written so the user can interact with `Bertini`, `Macaulay 2` via `HC.jl`.
- Direct implementation of certifying zeros of polynomial systems using interval arithmetic [9].
- In Julia, the evaluation of the square root is also implemented over $\mathbb{C}$.

Potential improvements

Below is a list of the main features that are still missing, which could be considered for future research to enhance computational methods using homotopy continuation software `Bertini` and `HC.jl`.

- Hybrid algorithms that combine advantages of both symbolic and numerical approaches.
- Use a symbolic preprocessor to simplify the system that is then solved using numerical methods.
- Artificial Intelligence and machine learning modules for `Bertini`/`HC.jl` for tuning the configuration settings for successful path tracking.

The two software packages mentioned above, `Bertini` and `HC.jl`, can be used for various robotic applications. These include but are not limited to kinematic synthesis problems [2, 15, 43], and computer vision applications [41].

This concludes the basic prerequisites required to understand the rest of the chapters in this thesis.

Bibliography

[1] Silviana Amethyst and Jonathan D. Hauenstein. *private communication*. Mar. 2021.

[2] Aravind Baskar. “Optimal Kinematic Synthesis Via Saddle Graphs Computed by Random Monodromy Loops”. PhD thesis. University of Notre Dame, 2024.

[3] Dan Bates, Danielle Brake, and Matt Niemerg. “Paramotopy: Parameter homotopies in parallel”. In: *International Congress on Mathematical Software*. Springer. 2018, pp. 28–35.

[4] Daniel J Bates, Jonathan D Hauenstein, and Andrew J Sommese. “Efficient path tracking methods”. In: *Numerical Algorithms* 58.4 (2011), pp. 451–459.

[5] Daniel J. Bates et al. *Bertini: Software for Numerical Algebraic Geometry*. Available at bertini.nd.edu with permanent doi: dx.doi.org/10.7274/R0H41PB5. 2013.

[6] Daniel J. Bates et al. *Numerically Solving Polynomial Systems with Bertini*. SIAM, 2013.

[7] Jérémy Berthomieu, Christian Eder, and Mohab Safey El Din. “Msolve: A library for solving polynomial systems”. In: *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*. 2021, pp. 51–58.

[8] Júlia Borras et al. “A reconfigurable 5-DoF 5-SPU parallel platform”. In: *2009 ASME–IFToMM International Conference on Reconfigurable Mechanisms and Robots*. IEEE. 2009, pp. 617–623.

[9] Paul Breiding, Kemal Rose, and Sascha Timme. “Certifying Zeros of Polynomial Systems Using Interval Arithmetic”. In: 49.1 (2023).

[10] Paul Breiding and Sascha Timme. “HomotopyContinuation. jl: A package for homotopy continuation in Julia”. In: *Mathematical Software–ICMS 2018: 6th International Conference, South Bend, IN, USA, July 24-27, 2018, Proceedings 6*. Springer. 2018, pp. 458–465.

[11] Stéphane Caro, Nicolas Binaud, and Philippe Wenger. “Sensitivity Analysis of 3-RPR Planar Parallel Manipulators”. In: *Journal of Mechanical Design* 131.12 (2009), p. 121005.

[12] David Cox. “Ideals, Varieties, and Algorithms-An Introduction to Computational Algebraic Geometry and Commutative Algebra”. In: *Undergraduate Texts in Mathematics* (1991).

[13] P. Dietmaier. “The Stewart-Gough Platform of General Geometry can have 40 Real Postures”. In: *Advances in Robot Kinematics: Analysis and Control*. Ed. by Jadran Lenarčič and Manfred L. Husty. Dordrecht: Springer Netherlands, 1998, pp. 7–16.

[14] Timothy Duff. “Applications of monodromy in solving polynomial systems”. PhD thesis. Georgia Institute of Technology Atlanta, Georgia, USA, 2021.

[15] Jeffrey Glabe. “Numerical Continuation on a GPU for Kinematic Synthesis”. PhD thesis. University of California, Irvine, 2020.

[16] Alexandre Goldsztejn, Stéphane Caro, and Gilles Chabert. “A three-step methodology for dimensional tolerance synthesis of parallel manipulators”. In: *Mechanism and Machine Theory* 105 (2016), pp. 213–234.

[17] Jonathan Hauenstein, Andrew Sommese, and Charles Wampler. "Regeneration homotopies for solving systems of polynomials". In: *Mathematics of Computation* 80.273 (2011), pp. 345–377.

[18] Jonathan D Hauenstein and Frank Sottile. "Algorithm 921: alphaCertified: certifying solutions to polynomial systems". In: *ACM Transactions on Mathematical Software (TOMS)* 38.4 (2012), pp. 1–20.

[19] Jonathan D. Hauenstein. *private communication*. May 2024.

[20] KH Hunt and EJF Primrose. "Assembly configurations of some in-parallel-actuated manipulators". In: *Mechanism and Machine Theory* 28.1 (1993), pp. 31–42.

[21] Manfred Husty. "Multiple Solutions of Direct Kinematics of 3-RPR Parallel Manipulators". In: *IFToMM World Congress on Mechanism and Machine Science*. Springer. 2023, pp. 599–608.

[22] Aditya Kapilavai. *Supplementary materials: Architecture Singularity Distance Computations for Linear Pentapods*. Mendeley Data, (2023), V1.

[23] Aditya Kapilavai. *Supplementary materials: Singularity Distance Computations for 3-RPR Manipulators using extrinsic metrics*. Mendeley Data, (2023), V3.

[24] Aditya Kapilavai. *Supplementary materials: Singularity Distance Computations for 3-RPR Manipulators using intrinsic metrics*. Mendeley Data, (2023), V2.

[25] Kazem Kazerounian and Jahangir Rastegar. "Object norms: A class of coordinate and metric independent norms for displacements". In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. 9419. American Society of Mechanical Engineers. 1992, pp. 271–275.

[26] Ou Ma and Jorge Angeles. "Architecture singularities of platform manipulators". In: *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE Computer Society. 1991, pp. 1542–1543.

[27] Jean-Pierre Merlet. *Parallel robots*. Vol. 128. Springer Science & Business Media, 2006.

[28] Jean-Pierre Merlet and Clément Gosselin. "Parallel mechanisms and robots". In: *Springer Handbook of Robotics* (2008), pp. 269–285.

[29] Alexander Morgan and Andrew Sommese. "A homotopy for solving general polynomial systems that respects m-homogeneous structures". In: *Applied Mathematics and Computation* 24.2 (1987), pp. 101–113.

[30] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[31] Georg Nawratil. "On the snappability and singularity-distance of frameworks with bars and triangular plates". In: *2nd IMA Conference on Mathematics of Robotics 2*. Springer. 2022, pp. 144–152.

[32] Georg Nawratil. "Point-models for the set of oriented line-elements – a survey". In: *Mechanism and Machine Theory* 111 (2017), pp. 118–134.

[33] Georg Nawratil. "Singularity Distance for Parallel Manipulators of Stewart Gough Type". In: *Advances in Mechanism and Machine Science*. Ed. by Tadeusz Uhl. Springer, 2019, pp. 259–268.

[34] Georg Nawratil. "Snappability and singularity-distance of pin-jointed body-bar frameworks". In: *Mechanism and Machine Theory* 167 (2022), p. 104510.

[35] Georg Nawratil and Josef Schicho. "Self-motions of pentapods with linear platform". In: *Robotica* 35.4 (2017), pp. 832–860.

[36] F. C. Park. “Distance Metrics on the Rigid-Body Motions with Applications to Mechanism Design”. In: *Journal of Mechanical Design* 117.1 (Mar. 1995), pp. 48–54.

[37] Helmut Pottmann and Johannes Wallner. *Computational line geometry*. Vol. 6. Springer, 2001.

[38] Arvin Rasoulzadeh. “Design and path optimization of linear pentapods based on the geometry of their singularity varieties”. PhD thesis. Technische Universität Wien, 2020.

[39] Arvin Rasoulzadeh and Georg Nawratil. “Rational parametrization of linear pentapod’s singularity variety and the distance to it”. In: *Computational Kinematics: Proceedings of the 7th International Workshop on Computational Kinematics that was held at Futuroscope-Poitiers, France, in May 2017*. Springer. 2017, pp. 516–524.

[40] Arvin Rasoulzadeh and Georg Nawratil. “Variational path optimization of linear pentapods with a simple singularity variety”. In: *Mechanism and Machine Theory* 153 (2020), p. 104002.

[41] Margaret Hayley Regan. “Parameterized Polynomial Systems and Their Applications”. PhD thesis. University of Notre Dame, 2020.

[42] Walter Rudin et al. *Principles of mathematical analysis*. Vol. 3. McGraw-hill New York, 1964.

[43] Samantha Nicole Sherman. “Mechanism Analysis and Synthesis Using Numerical Algebraic Geometry”. PhD thesis. University of Notre Dame, 2021.

[44] Andrew J Sommese and Charles W Wampler. *The Numerical solution of systems of polynomials arising in engineering and science*. World Scientific, 2005.

[45] Bernd Sturmfels. “WHAT IS... a Gröbner Basis?” In: *Notices-American Mathematical Society* 52.10 (2005), p. 1199.

[46] Jan Verschelde. “Polyhedral methods in numerical algebraic geometry”. In: *Contemporary Mathematics* 496 (2009), p. 243.

[47] Charles W. Wampler. “Isotropic Coordinates, Circularity, and Bezout Numbers: Planar Kinematics From a New Perspective”. In: vol. Volume 2A: 24th Biennial Mechanisms Conference. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Aug. 1996, V02AT02A073.

[48] Charles W. Wampler and Andrew J. Sommese. “Applying Numerical Algebraic Geometry to Kinematics”. In: *21st Century Kinematics*. Ed. by J Michael McCarthy. Springer, 2013, pp. 125–159.

[49] Chang-De Zhang and Shin-Min Song. “Forward kinematics of a class of parallel (Stewart) platforms with closed-form solutions”. In: *Journal of Robotic Systems* 9.1 (1992), pp. 93–112.

Chapter 2

On Homotopy Continuation Based Singularity Distance Computations for 3-RPR Manipulator

This chapter consists of the conference paper:

- Aditya Kapilavai and Georg Nawratil: On Homotopy Continuation Based Singularity Distance Computations for 3-RPR Manipulator.

It was published in the conference proceedings of New Trends in Mechanism Science (D. Pisla, B. Corves eds.), pp. 56–64, 2020.

Contributions

The author of this thesis contributed to performing groupings of unknown variables for all the presented nonhomogeneous and homogeneous representations, conducting computations, analyzing the results, and writing and editing the draft of the paper.

The results of this chapter are presented at the conference

- PARALLEL 2020, The 4^{th} International Workshop on Fundamental Issues, Applications and Future Research Directions for Parallel Mechanisms/Manipulators Machines, September 9-11 2020, online conference due to COVID-19.

Chapter 3

Singularity distance computations for 3-RPR manipulators using extrinsic metrics

This chapter consists of the journal paper:

- Aditya Kapilavai and Georg Nawratil: Singularity distance computations of 3-RPR manipulator using extrinsic metrics.

It was published in the Journal of Mechanism and Machine Theory, Vol. 195, pp. 105595, 2024 (Open Access).

Contributions

The thesis author was mainly involved in data curation, investigation, methodology, software development, conducting computations, validation, visualization, writing – the original draft, writing – the review and editing, and the main results of the paper.

The results of this chapter are presented at the conference

- IFToMM D-A-CH Konferenz, February 18-19, 2021. Online conference due to COVID-19.
- Effective Methods in Algebraic Geometry (MEGA 2021), June 7-11, 2021. Oral presentation in the software session held online due to COVID-19.

Chapter 4

Singularity distance computations for 3-RPR manipulators using intrinsic metrics

This chapter consists of the journal paper:

- Aditya Kapilavai and Georg Nawratil: Singularity distance computations of 3-RPR manipulator using intrinsic metrics.

It was published in Computer-Aided Geometric Design, Vol. 111, pp. 102343, 2024 (Open-Access).

Contributions

The thesis author was primarily involved in data curation, investigation, methodology, software development, validation, visualization, original draft writing, and reviewing and editing.

The results of this chapter are presented at the conference

- Conference on Geometry: Theory and Applications, Kefermarkt, June 19-23, 2023, Austria.

Chapter 5

Architectural Singularity distance computations of linear pentapods

This chapter consists of the journal paper:

- Aditya Kapilavai and Georg Nawratil: Architectural singularity distance computations of Linear pentapods.

It was published in the ASME Journal of Mechanisms and Robotics, Vol. 17(2), pp. 021008, July 2024.

Contributions

The author of this thesis has contributed to data acquisition, conducting computations, writing the original draft, and analysis of the results.

The initial idea of this chapter is presented at the conference

- ICRA 2022 Workshop "New Frontiers of Parallel Robotics," Philadelphia, May 27, 2022, USA. Oral online presentation in the session "My Work in 5 Minutes," held in a hybrid format due to COVID-19.