Interner Bericht

DLR-IB-RM-OP-2024-119

Workspace Analysis of a Highly Coupled Quadruped Robot with Six Actuated Degrees of Freedom

Semesterarbeit

Pirmin Gaißer

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Robotik und Mechatronik
Standort Oberpfaffenhofen

Deutsches Zentrum
für Luft- und Raumfahrt

Chair of Sensor Based Robotic Systems and Intelligent Assistance Systems
School of Computation, Information & Technology
Technical University of Munich

Workspace Analysis of a Highly Coupled Quadruped Robot with Six Actuated Degrees of Freedom

Semester Thesis

at the School of Computation, Information & Technology of the Technical University of Munich

Supervised by	Prof. Dr. Alin Albu-Schäffer Florian Christoph Loeffl TUM School of Computation, Information and Technology (CIT)
Submitted by	Pirmin Gaißer Schanzenbachstraße 13 81371 München
Submitted on	July 19, 2024 in Garching

Institute of Robotics and Mechatronics
German Aerospace Center (DLR)

Task Description

Workspace Analysis of a Highly Coupled Quadruped Robot with Six Actuated Degrees of Freedom

Author:	Pirmin Gaißer	**Supervisor:**	Florian Christoph Loeffl
Registration:	February 8 2024	**Submission:**	Juli 19 2024

Introduction:
While legged robots offer advantages in dynamic capabilities and mobility over wheeled counterparts, they often face challenges in energy efficiency and reliability due to using multiple actuators. Traditional quadruped designs with 12 actuated degrees of freedom permit free foot placement relative to the body. Additionally, the body can reach many different positions and rotations inside its body workspace without moving the feet. This project explores a design with only six actuated degrees of freedom, aimed at combining and reducing functions while maintaining effective locomotion. This means the robot can still freely place its feet independently of each other within the respective workspace. To achieve this, each motor must be connected to multiple legs. Due to the coupled actuation, the reachable space of the feet is restricted compared to a version with 12 actuated degrees of freedom. Also, the body workspace is reduced to a single point. To allow the robot to later move freely in various terrains an optimization of the foot workspace is necessary. A sufficient parameter space must be defined, which is indifferent to the total size of the robot and includes some relation to the motor force necessary. Also, singularities must be avoided in the optimized geometry. Without sufficient movement of the feet and with singularities in the workspace, the crawler is not able to navigate stairs, slopes, and similar uneven terrain. A workspace analysis is, therefore, a vital step in designing a highly coupled quadruped robot.

Objective and Expected Outcome:
The project aims to analyze and optimize the workspace of a quadruped robot geometry with six actuated degrees of freedom. The successful completion of this project is expected to contribute to the field of robotics by demonstrating how quadrupeds with reduced actuator count can have a sufficient workspace to navigate uneven terrain. The findings will be applicable in designing more efficient and cost-effective robotic solutions for various applications.

Task:

- Comparison of Quadrupeds: Investigate the previous and current quadruped
- Kinematic Analysis: Analyze the kinematics of the quadruped, understanding its movement patterns and limitations.
- Workspace Optimization: Optimize the workspace of the end effectors in relation to the motor force and crawler size.
- Optionally Update Prototype: If the time allows, modify the existing prototype to reflect a symmetrical and optimized geometry.

Garching, July 19, 2024

Prof. Dr.-Ing Alin Albu-Schäffer	Dr.-Ing. Florian C. Loeffl	(B.Sc.) Pirmin Gaißer

Abstract

This thesis presents an optimization strategy for the parallel geometry of a coupled quadruped robot. The primary objective is to enhance the reachable workspace of the robot's feet by refining its characteristic lengths. The study introduces an innovative approach to defining and calculating the workspace of coupled quadrupeds efficiently. An analytic solution to the forward kinematics is developed to expedite the calculation process. The workspace evaluation ensures the optimized workspace is not only increased in volume but also possesses functional attributes suitable for a walking device. A genetic algorithm is employed for the optimization, balancing robustness and computational efficiency. The results demonstrate significant improvements in the workspace parameters, underscoring the effectiveness of the proposed methods.

Zusammenfassung

Diese Arbeit stellt eine Optimierungsstrategie für die parallele Geometrie eines gekoppelten vierbeinigen Roboters vor. Das Hauptziel ist es, den erreichbaren Arbeitsraum der Roboterfüße durch Verbesserung der charakteristischen Längen zu erweitern. Die Arbeit führt einen innovativen Ansatz zur Definition und effizienten Berechnung des Arbeitsraums von gekoppelten Vierbeinern ein. Eine analytische Lösung für die Vorwärtskinematik wurde entwickelt, um den Berechnungsprozess zu beschleunigen. Die Bewertung des Arbeitsraums stellt sicher, dass der optimierte Arbeitsraum nicht nur an Volumen gewinnt, sondern auch funktionale Eigenschaften besitzt, die für ein Laufroboter geeignet sind. Ein genetischer Algorithmus wird für die Optimierung eingesetzt, um Robustheit und Recheneffizienz zu vereinen. Die Ergebnisse zeigen signifikante Verbesserungen der Arbeitsraumparameter und unterstreichen die Wirksamkeit der vorgeschlagenen Methoden.

Contents

List of Abbreviations **ix**

1 Introduction **1**
- 1.1 Goals and Boundaries 1
- 1.2 Design of the Robot 2
- 1.3 State of the Art 3

2 Method Selection **5**
- 2.1 Spatial four-bar linkage 5
- 2.2 Calculating the Kinematics 6
- 2.3 Calculating the Jacobian 7
- 2.4 Defining the Workspace 8
- 2.5 Calculating the Workspace 10
- 2.6 Evaluating the Workspace 11
- 2.7 Choosing Discretization Parameters 12
- 2.8 Reducing Design Parameters 12
- 2.9 Choosing a Solver 12
- 2.10 Genetic Algorithm 14

3 Implementation **17**
- 3.1 Goals 17
- 3.2 Kinematics 18
  - 3.2.1 Analytic Forward Kinematic 18
  - 3.2.2 Kinematic Jacobian 23
- 3.3 Workspace 24
  - 3.3.1 Evaluation 24
  - 3.3.2 Discretization 28
- 3.4 Optimization 30
  - 3.4.1 Design Parameters 30
  - 3.4.2 Approach 32

4 Results **35**
- 4.1 Interpretation 36
- 4.2 Discussion 38

5 Conclusion and Outlook **41**

A Analytic Solution to the Forward Kinematics **43**

B Foot Distance Manifold **47**

Bibliography **55**

List of Abbreviations

DoF	Degree of Freedom
FK	Forward Kinemtics
GA	Genetic Algorithm
GCI	Global Conditioning Index
GD	Gradient Descent
GS	Grid Search
IK	Inverse Kinematics
MC	Monte Carlo Approach
PM	Parallel Manipulator
PSO	Particle Swarm Optimization
RI	Redundant Input
RO	Redundant Output
RPM	Redundant Passive Motion
RSD	Relative Standard Deviation
SA	Simulated Annealing
SGD	Stochastic Gradient Descent
WVI	Workspace Volume Index

Chapter 1

Introduction

Quadruped robots have emerged as a versatile solution for navigating complex terrains and uneven surfaces, making them ideal for applications ranging from search and rescue operations to planetary exploration. Compared to alternatives like wheeled robots, the applicability is still small though. This stems from the fact that conventional quadrupeds, like Bert with eight Degree of Freedom (DoF)(SEIDEL et al. 2020) or Mini Cheetah with twelve DoF (BENJAMIN KATZ et al. 2019), have many redundancies in their design. Especially the high motor count leads to high costs and inefficiencies, which prevent widespread use. The presented concept of a quadruped robot reduces the number of motors to six. Despite its reduction in motor count, the robot can control the motion of all four feet in all three directions. Six DoF is the minimum possible number to achieve this, as explained in section 1.2. The robot has the potential to acquire versatile gaits while reducing redundancies and costs to an optimum. Its main drawbacks are the necessary leg coupling and the compromised workspace of each leg due to the parallel nature of the leg design. Moving one motor does affect the motion of multiple legs. A small leg reach hinders the robot from traversing uneven terrain and leads to slow and jittery gaits. This motivates increasing the useful workspace of the robot's legs while considering their dependencies, which will be addressed in this work.

First, a short overview of the goals and boundaries, the design of the robot, and the current state of workspace optimizations are given before explaining the selection of methods (chapter 1, chapter 2). Afterwards, the chosen methods are applied to the optimizations of the proposed robot in chapter 3. The results are presented and discussed in chapter 4. In the end, the findings are summarized, and directions for future improvements are proposed (chapter 5).

1.1 Goals and Boundaries

This work addresses the problem of optimizing the characteristic lengths of the quadruped design (Figure 1.1), conceived by Florian Loeffl, to improve the reachable workspace of the feet under certain conditions. This includes the goal of finding an appropriate definition of a workspace for coupled quadrupeds, finding a way to calculate this workspace in a reasonable amount of time, and evaluating this workspace so a notion of optimality makes sense. Calculating the workspace efficiently includes the sub-goal of finding an analytic solution to the Kinematics. The workspace evaluation ensures the workspace is not only improved in volume but also has useful properties for a walking device. Beyond the scope of this work is finding stable walking patterns, which will not be considered for the optimization. Finally, the problem of finding a viable optimization strategy and how to implement it in software is addressed.

This research utilizes the Matlab (MathWorks 2022b) optimization toolbox, which offers numerous tools for optimization tasks. Nonetheless, there remains a lack of tailored optimization software for general multi-body systems, necessitating the development of an analytic kinematic solution to manage the optimization's complexity. This study introduces a novel perspective involving coupled spatial four-bar linkages to describe the proposed kinematic design.

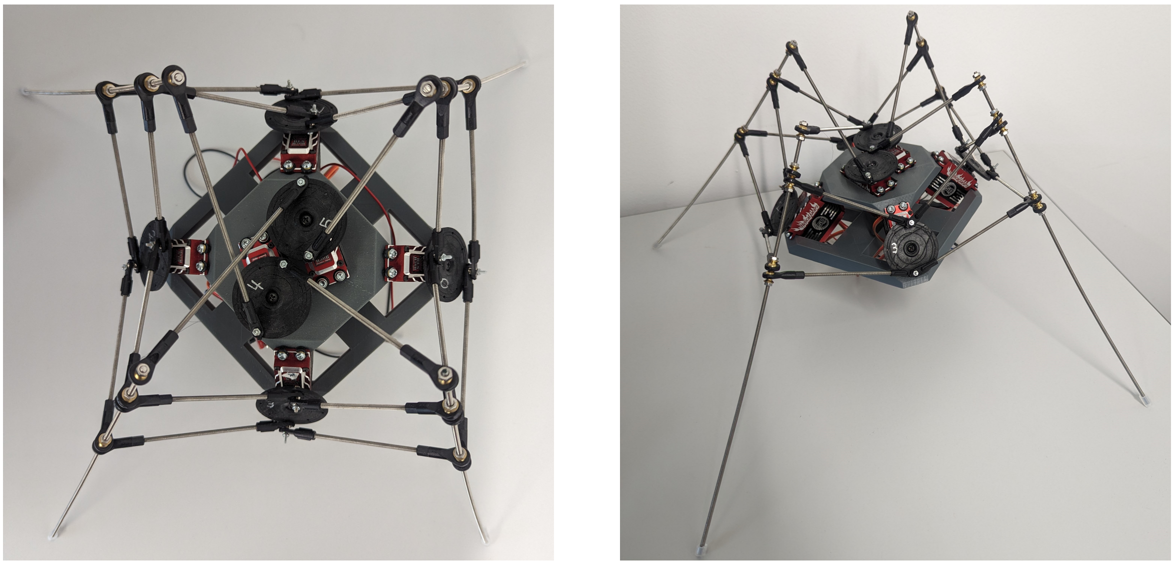


Figure 1.1: The unoptimized prototype in top view (left) and isometric view (right). The geometry is not optimized, and the upper motors do not retain rotational symmetry.

1.2 Design of the Robot

The quadruped is constructed so that each motor influences the motion of two distinct legs. The binomial coefficient $\binom{4}{2} = 6$ shows that with six actuated DoF, all possible combinations are addressed. A geometric counterpart is the tetrahedron, which has four corners and six edges connecting two corners each. By defining the lengths of all six edges, the corners' relative positions can be restricted to a single configuration and its chiral reflection. A design is proposed that can control the four foot points in a similar way while maintaining the shape of a quadruped.

The type synthesis of the kinematic structure is mostly given. It aspires to have four-way rotational symmetry around the vertical center line, allowing several simplifications. The previous prototype (Figure 1.1) violates the symmetry by placing two of the actuated joints in the same horizontal plane next to each other. To retain symmetry, in the improved geometry both actuated joints lie on the symmetry axis. The other four motors are placed symmetrically between two adjacent feet.

The proposed quadruped has a parallel tree structure consisting of 31 bodies connected by six revolute (1 DoF), 24 universal (2 DoF), and 16 ball joints (3 DoF). Applying Grübler's formula results in $6(31-1-46)+(6*1+24*2+16*3) = 6$ DoF, which are constrained by six motors actuating the six revolute joints. The body, where the revolute joints are attached, is considered the floating base of the robot. Its origin is placed so that the x-axis points toward active joint 1, the y-axis towards active joint 2, and the z-axis towards active joints 5 and 6. The robot has four-way rotational symmetry along the z-axis. A "leg assembly" is the sub-assembly depicted in blue in Figure 1.2 (left). The six bodies that are directly attached to the base via the active joints are called "motor links". Each motor link is part of two distinct leg assemblies. The four bodies of the robot touching the ground are called "legs", with the "feet" being the end effector points touching the floor. Five "passive links" connect each leg with three of the motor links, representing the remaining 20 bodies of the robot. The links are differentiated into four "top links" and 16 "leg links", as shown in Figure 1.2 (right). While the leg links connect the leg with a universal joint and the motor links with a spherical joint, the top links use universal joints for both ends. The design parameters for the optimization include the body's dimensions, the legs' lengths, and the lengths of the connecting links, as derived in section 3.4.1.

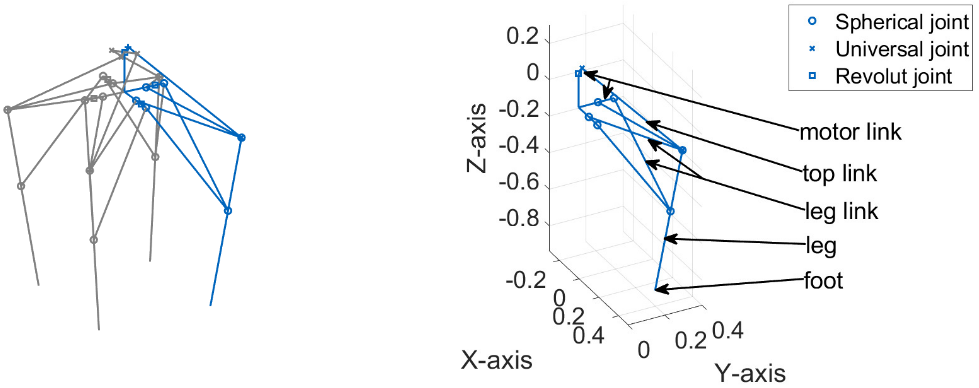


Figure 1.2: The structure of the robot (left). One of the four leg assemblies is marked in blue. The structure of a leg assembly (right). The joints are drawn as circles, crosses, and squares, and the links as lines. The scale is non-dimensionalized and shows the relative proportions.

1.3 State of the Art

Design optimization is crucial in enhancing the overall walking capabilities of quadruped robots. Genetic Algorithms (GAs) and performance indices are commonly used for this purpose. Gülhan and Erbatur (Gülhan and Erbatur 2018) employed a GA to improve the stability, straightness, and speed of trotting in a twelve DoF robot. Fedorov and Birglen (Fedorov and Birglen 2016) utilized GA to optimize the design of a pantograph leg with two DoF, while Birglen and Ruella (Birglen and Ruella 2014) optimized the design of a one DoF leg for efficiency and robustness.
While the proposed quadruped uses similar approaches, namely GA and performance indices, and is also concerned with reduced DoF, the focus will be improving the robot's workspace.

Concerning the optimization of leg workspace for quadrupeds with uncoupled legs, Hao et al. (Hao et al. 2020) found that the reach of each foot is primarily restricted by joint angle limits. Ren et al. (Ren et al. 2018) proposed a quadruped leg with an elastic linkage, necessitating design optimization to improve the workspace and reduce tracking errors, and Ratolikar et al. (Ratolikar and R 2020) optimized a five-bar linkage used as a quadruped leg using a GA.

The optimization of uncoupled quadruped legs has similarities to that of coupled ones. Still, this work requires additional concepts to address the coupling.

King's (KING 2022) work on a coupled six DoF robot preceded this work. It employed pantograph legs coupled with linear actuators to achieve motion in all three directions for each leg. The primary limitations on the workspace were due to actuator limits and the pantograph leg design. The proposed design (Figure 1.1) replaces the linear actuators with rotating motors and introduces a physical body to house the motors and battery. This redesign led to a parallel manipulator structure as described in section 1.2. An earlier quadruped design with six DoF was proposed by Kaneko et al. (KANEKO et al. 1986). It featured uncoupled legs with one DoF each and a rotating mass to position the center of mass.
Two concepts with six DoF were proposed before this work. They did not optimize the workspace, though, and used a different structure.

While workspace optimization for coupled quadrupeds with reduced DoF has not been extensively explored, similar methodologies have been applied to Parallel Manipulators (PMs). Various studies have optimized the workspace of planar and spatial manipulators. Some examples include the work of Badescu (BADESCU et al. 2002), who optimized a stack of 3-UPS platforms, Deng et al. (DENG et al. 2021), who optimized a Steward-Gough platform and Silva et al. (SILVA et al. 2014a), who optimized a Delta robot. All of them optimized for a large dexterous workspace, using the Global Conditioning Index (GCI) introduced by Gosling and Angeles (GOSSELIN and ANGELES 1990) and used a GA.
PMs have a single end effector. Since the quadruped has four, its structure is coined a "parallel tree structure" in this work and, therefore, requires additional methods, which are specified among others in the next chapter.

Chapter 2

Method Selection

The goal of this work is to improve the robot's workspace. This requires an optimization. Design parameters, boundary conditions, and a solver need to be chosen accordingly. Additionally, the objective function must be defined. It should evaluate the workspace's size and quality for the respective design parameters. Defining, calculating, and evaluating the workspace requires additional methods. Finally, the optimization requires calculating the kinematics and Jacobian matrices for different configurations, to obtain the workspace. A method using spatial four-bar linkages was found for that.

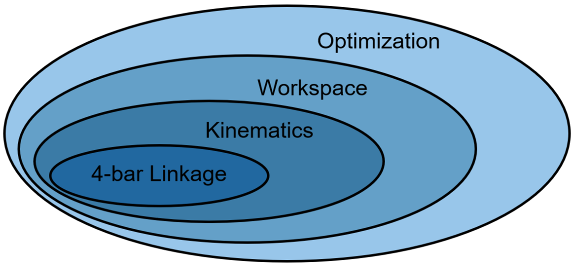


Figure 2.1: The four main parts of the work, where the spatial four-bar linkage is part of the kinematics, which are a sub-function of the workspace calculation - a sub-function of the optimization algorithm.

The core forms theory on spatial four-bar linkages, which are used to calculate the kinematics, required for determining the workspace. Increasing the workspace is the objective of the optimization. Figure 2.1 shows this structure. Different methods are presented and the chosen methods are highlighted in grey in the tables below. The reasoning behind choosing one over its alternatives is depicted in this chapter. Methods are ordered from the bottom up, starting with the spatial four-bar linkage and ending with the optimization (Figure 2.1).

2.1 Spatial four-bar linkage

The forward kinematics used in the optimization can be calculated using spatial four-bar linkages (section 3.2).

Chyi (DAVID PERNG CHYI 1969) found an analytic solution to the mechanism, shown in Figure 2.2. It consists of four links, one of which is the base. Two levers are connected to the base via revolute joints and the coupler connects the levers with two spherical joints. The rotation axes of the revolute joints, in general, are not parallel. It has one DoF and two

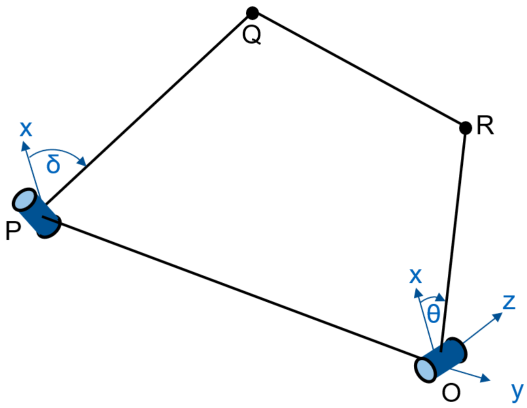


Figure 2.2: The spatial four-bar linkage as defined by Chyi (David Perng Chyi 1969), with the unknown angle θ at the origin O. The angle δ is given.

assembly modes. The equation 2.1 calculates the two possible locations of spherical joint R when given point Q. The points are defined in a reference frame O whose origin lies on the base point of the unknown lever OR. The z-axis coincides with the rotation axis of the revolut joint O, and the angle θ is defined relative to the x-axis, whose orientation can be defined arbitrarily. The position of Q is a function of the known angle δ.

$$_{O}R = \begin{pmatrix} cos(\theta)\overline{OR} \\ sin(\theta)\overline{OR} \\ 0 \end{pmatrix} \tag{2.1}$$

where

$$sin(\theta) = -\frac{_{O}Q(2)K \mp {}_{O}Q(1) * \sqrt{_{O}Q(1)^2 + {}_{O}Q(2)^2 - K^2}}{_{O}Q(1)^2 + {}_{O}Q(2)^2} \tag{2.2}$$

$$cos(\theta) = -\frac{_{O}Q(1)K \pm {}_{O}Q(2) * \sqrt{_{O}Q(1)^2 + {}_{O}Q(2)^2 - K^2}}{_{O}Q(1)^2 + {}_{O}Q(2)^2} \tag{2.3}$$

with the auxiliary variable

$$K = \frac{1}{2\overline{OR}}(l_1^2 - \overline{OR}^2 - {}_{O}Q(1)^2 - {}_{O}Q(2)^2 - {}_{O}Q(3)^2) \tag{2.4}$$

2.2 Calculating the Kinematics

The kinematics are used to calculate leg configurations to evaluate the workspace (Section 1.1). Two ways of viewing the robot are discussed. Either fixing the base and looking at how the floor has to be oriented for a given set of joint angles or fixing the floor and calculating the orientation of the floating base when standing on the floor. The first is related to Forward Kinemtics (FK) while the latter requires Inverse Kinematics (IK) to derive the joint angles from the given foot positions.

Method		Advantage	Disadvantage
IK	Numerical	More efficient for workspace discretization	Slow calculation time Whole robot must be considered Solution depends on initial guess
FK	Numerical	Faster calculation time compared to IK	Requires sampling for workspace estimation Solution depends on initial guess
	Analytical	Fastest calculation time No initial guess needed Reduced design parameters	Needs further assumptions Hard to build Requires sampling for workspace estimation

Table 2.1: Advantages and disadvantages of the IK and FK using analytical and numerical calculation methods.

Although the IK is more efficient for the workspace discretization, as explained by Porges (PORGES 2013), the long calculation time and the dependency on the initial guess do not lead to a faster workspace calculation compared to the FK. The mean calculation time using the robotics toolbox from Matlab (MATHWORKS 2022a) is significantly longer than the FK because the whole robot must be considered, leading to a more complex problem to solve. The FK can be split into four sub-assemblies, reducing the complexity and speeding up the mean calculation time. This can be achieved numerically using dual quaternions as proposed by Zsombor-Murray and Gfrerer (ZSOMBOR-MURRAY and GFRERER 2010). An analytic solution is found to further improve the calculation time and be independent of an initial guess. It assumes that the five passive links meet at two distinct positions on the leg. This allows a representation of the leg assembly as two connected spatial four-bar linkages, which is discussed in more detail in section 3.2. Building a physical robot with two or more spatial joints in the same spot is not trivial (HAMLIN and SANDERSON 1994, NASA 2022). Deviating from these assumptions reduces the accuracy of the model. Nevertheless, the simplified geometry reduces the number of required design parameters, which is beneficial, as discussed in section 2.8. The calculation time of around $5ms$ and the independence on an initial guess led to the choice of the analytical FK.

2.3 Calculating the Jacobian

Calculating the kinematic Jacobian ${}_gJ_{k,i}$ for each leg i in the ground-related frame g, defined in section 2.4, is required for calculating the performance index p_{RO}, described in section 2.6.

Method	Advantage	Disadvantage
Analytical Jacobian	Exact solution	Large expression which cannot be handled by MATLAB
Numerical Jacobian	Fast numeric solution	Requires multiple coordinate transformations

Table 2.2: Advantages and disadvantages of the analytical and numerical calculation methods for the kinematic Jacobian.

Since an analytic solution to the FK exists (Appendix A), the kinematic Jacobian can be calculated by differentiating the symbolic expression by the active joint angle vector. It results in a large analytical expression, which MATLAB (MathWorks 2022a) cannot handle and would demand alternative software. Instead, it was opted for numerically integrating the kinematic Jacobian by Newton's forward differentiation. It requires calculating one additional evaluation of the FK for each DoF and transformations to the ground-related frame, but it is still a fast and accurate calculation method.

2.4 Defining the Workspace

As introduced in the section 1.1, the goal is to find a workspace definition that allows optimizing the robot so it achieves the greatest variety of walking motions under certain constraints. Considering the position of the center of mass and, therefore, the position and orientation of the floating base, is beyond the scope of this work. The workspace only considers the position of the feet relative to each other. This reduced the dimensions of the workspace from twelve to six, effectively neglecting the rigid body motion of the robot. As will be shown in section 3.3, a further reduction to a 3D space is beneficial for meeting the memory limitations posed by the discretization method. Due to the symmetry of the robot as well as the constraints, all four feet will have the same workspace volume, so optimizing the relative position of one of the legs will be sufficient, which makes this further reduction possible. Three different workspaces are discussed.

Method	Advantages	Disadvantages
W_c	Useful workspace for path planning	Depends on configuration of stance feet
W_g	Intuitive workspace representation	Geometry leads to tall and thin workspace
W_d	Retains information from 6D space Improved shape	Non-Euclidian space

Table 2.3: Advantages and disadvantages of the constrained foot workspace W_c, the ground-related workspace W_g, and the distance workspace W_d.

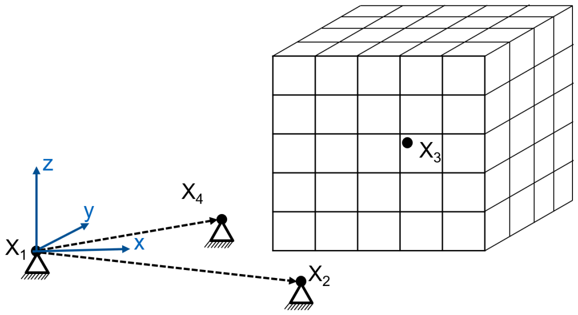


Figure 2.3: Visualization of the constrained foot workspace. The coordinate system is defined by the three feet on the ground, relative to which the position of the fourth foot is examined. The stance feet are at a fixed location relative to the ground.

The constrained foot workspace W_c, as shown in Figure 2.3, describes the possible positions of one foot while the others remain on a fixed point on the ground. It is a useful representation of the workspace for path planning of step trajectories, but due to the leg coupling, the workspace depends on the positions of the stance feet, which vary during crawling. Therefore, it is not used for the optimization.

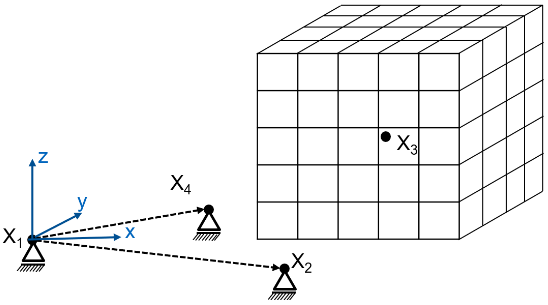


Figure 2.4: Visualization of the ground-related workspace. The coordinate system is defined by the three feet on the ground, relative to which the position of the fourth foot is examined. The stance feet are not fixed but can slide freely along the ground.

Many variations of the stance feet are included in the ground-related workspace W_g. Three feet define a ground plane relative to which the reachable positions of the fourth foot are examined, as shown in Figure 2.4. The stance legs are not constrained. The 3D Cartesian system gives an intuitive representation of the workspace. The robot's geometry favors tall and thin workspaces, an undesired shape for walking. For the optimization, the non-Euclidean space defined below is used to mitigate this tendency. Still, the ground-related workspace is suitable for visualizing the results in section 4 and calculating the kinematic Jacobian (Section 3.2.2).

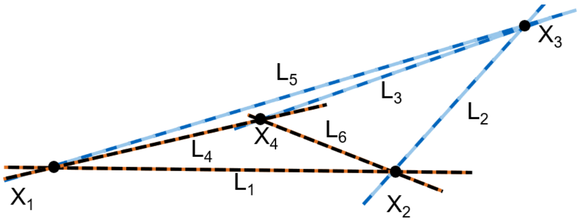


Figure 2.5: Visualization of the distance workspace. The three distances marked in blue make up the three-dimensional workspace used for the optimization.

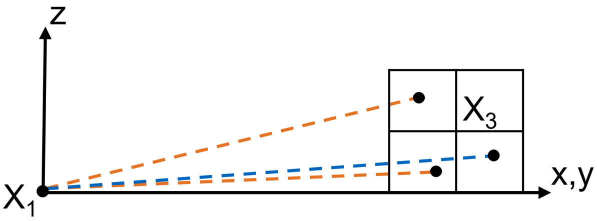


Figure 2.6: Visualization of the height bias. The distance space (dotted lines) does not reach a new length state when moved in the z-direction, while the Euclidean space does.

The relative position of the feet can be described by the six distances between the four feet (Figure 2.5). All reachable distances compose the 6D distance workspace W_d. As described in section 1.2), two different possibilities exist for defining four points by six lengths, so additional information on the configuration is needed for a unique representation. This makes the space non-Euclidean and hinders the visualization and definition of a physical vector norm required to evaluate the distance from singularities (Section 3.3). It still can be described as a manifold with boundaries as defined by Tu (Tu 2011 §22). These boundaries stem from the constraints imposed by the coupled kinematics of the robot. Due to symmetry, each leg has the same workspace influenced by parameters in the same way, so the six dimensions can be further reduced to three focusing on the three distances from one leg to the others. They are marked in blue in Figure 2.5. The space prefers distance changes in the ground plane, as shown in Figure 2.6, resulting in a more useful workspace shape which is why it is used in the optimization. The optimized workspace using the ground-related and distance space are compared in Figure 2.7. They are both visualized as reached voxels looking from the origin of the ground-related frame. The distance space is converted to the ground-related frame for comparison. The workspace in the right image, created using the distance space optimization, shows a wider projection than the ground-related optimization (left).

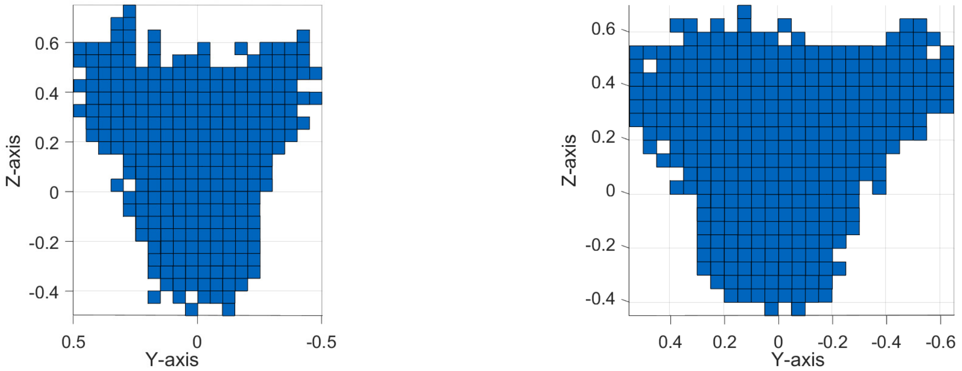


Figure 2.7: Visualization of an optimized workspace using the ground-related frame (left) and distance space (right). They are both shown looking from the origin of the ground-related frame, to see the difference in width.

2.5 Calculating the Workspace

The optimization of the quadruped design regarding the workspace requires calculating it in each iteration of the optimization, as shown in Figure 3.8. Three methods are commonly used (Merlet 2006).

Method	Advantages	Disadvantages
Analytic	Exact solution	Solution only for simple geometries Not all constraint can be included
Numerical	Potentially more efficient	Complex implementation Inclusion of constraints often not trivial
Discretization	All constraints can be included Easy implementation Widespread adoption	Computationally demanding

Table 2.4: Advantages and disadvantages of the analytic, numerical, and discretization method.

The most intuitive approach is integrating the FK with respect to the active joint angles. Gallant and Boudreau (Boudreau and Gallant 2002) used the method for a planar PM and Aboulissane et al. (Aboulissane et al. 2019) for the Delta robot. For the proposed design this method was unfeasible considering the lengthy analytical solution of the FK, shown in Appendix A.

The alternative is to approximate the analytic solution. The two main approaches are separated into discretization methods and numerical methods (Merlet 2006). Latter includes inferring volumetric properties of the workspace by approximating its boundary BOHIGAS et al. 2012, using interval analysis to iteratively reduce the size of a bounding box encompassing the workspace volume (MERLET 2006), and using the fact that the outward facing velocity vector normal to the surface of the workspace has to be zero (AGRAWAL 1991). Recent attempts also try to estimate the workspace using a neuronal network (BOANTA and BRIȘAN 2022).

The discretization method divides space into discrete voxels and determines which lie inside the robot's workspace. This can be done knowing either the FK or IK. When using the IK, one point of each voxel is tested to see whether it belongs to the workspace (MERLET 2006). Since the IK is much slower to calculate than the FK, as pointed out in section 2.2, the workspace is approximated by sampling in the joint space. Usually, in each sample, a random configuration of joint positions is generated and used to calculate the position and orientation of the end effector. The corresponding voxel is marked, and after N_s samples, all marked voxels together approximate the reachable workspace (PORGES 2013).

The analytic solution would require large simplifications of the model, which cannot capture the intricacies of the leg design. More sophisticated numerical methods may improve convergence speed but were not considered in this work. Despite its computational demands, the widespread adoption of the discretization method (MERLET 2006), the possibility of including arbitrary constraints, and the straightforward implementation led to the implementation of the discretization method for the workspace approximation. Since the FK is much faster than the IK, the sampling method described by Porges (PORGES 2013) is used.

2.6 Evaluating the Workspace

Performance indices are a way to evaluate the manipulator's capability and are widely used for performance analysis and optimal design (ROSYID et al. 2017). Many different indices exist, which are summarised by Li et al.(LI et al. 2023).

Since this work focuses on maximizing the workspace's volume, the volume index p_{VI} is the main performance index used. It tracks the geometrically reachable points. Additionally, the two metrics are introduced, which punish undesirable workspace configurations. They are chosen to fit into the performance index framework and evaluate if a foot position is not useful for walking (p_F) and if the force amplification from the feet to the active joints is above a certain threshold (p_{RO}). This indicates a closeness to a Redundant Output (RO) singularity, defined by Müller and Zlatanov (MÜLLER and ZLATANOV 2019), and prevents overloaded motors and control issues. Other singularities are not addressed for the performance. The Redundant Input (RI) singularity often lies on the border of the workspace because the velocity component normal to the surface has to be zero and the robot has a good accuracy close to it. Redundant Passive Motion (RPM) would occur when motion is possible even though both input and output are fixed. This should be avoided, but similar to RI it does not affect the reachable workspace (MÜLLER and ZLATANOV 2019).

The approximation of the Workspace Volume Index (WVI), defined in section 3.3.1, is used to evaluate the workspace's relative size compared to the robot's size while including the limitations of the robot. It considers the local performance indices p_{VI}, p_F, and p_{RO}. The conditioning index p_{LCI}, which is used in many PM optimizations as hinted at in section 1.3, is not considered for the optimization. The incentive to stay away from parallel singularities is captured by p_{RO}, so the index p_{LCI} is not necessary as a performance index.

2.7 Choosing Discretization Parameters

The use of the discretization method by Porges (PORGES 2013) requires choosing the voxel size and a break criterion to stop the sampling process.
The method for choosing the parameters revolves around estimating the calculation time for a workspace calculation. Samples outside the workspace require less calculation time, as they can leave the loop earlier, as shown in the section 3.4. Because the calculation time increases almost linearly with the number of samples, the calculation time depends on the size of the workspace. The parameters are selected with respect to the properties close to the optimal solution since the demand for accuracy and calculation speed increases for increasing absolute values of the WVI and decreasing improvements from one generation to the next.
These considerations make the choice of discretization parameters an iterative process, as the right choices depend on the solution of the optimization, in particular the WVI of the solution.

2.8 Reducing Design Parameters

The geometry of the robot is varied to find the best possible workspace. In general, the geometry of each body can be arbitrarily complex, resulting in many design parameters to be optimized. This reduces the convergence speed of the GA as stated in (LI et al. 2023), so a reduction of parameters is aimed for, which still captures the main geometrical properties of the system. This is done by exploiting the four-way rotational symmetry of the robot and using simple geometric shapes, which is explained in the section 3.4.1.

2.9 Choosing a Solver

The workspace optimization needs a solver that can deal with constrained problems with discrete nonlinear noisy objective functions. The FK introduces non-linearities and the discretization method introduces noise. This restricts the considered solver categories to the following selection (Table 2.5). Excluded are 2nd-order solvers, which are not suitable for discrete problems. For 1-st order solvers, a representative algorithm is selected for each. Many additional variants exist. They require finite difference methods due to the discrete objective fuction to estimate the derivatives.

Purely sampling-based approaches disregard all information from past samples which makes it quite inefficient for workspace optimization. Since no information is drawn from the previous samples in the discretization method a sampling-based approach is used for calculating the workspace. It uses the MC approach, to have improved high-dimensional coverage compared to GS. Sampling plans can be used in conjunction with surrogate models when the

Method	Deterministic	Stochastic
Sampling-based	Grid Search (GS)	Monte Carlo Approach (MC)
0-th order	Patternsearch	General Algorithm (GA) Particle Swarm Optimization (PSO) Simulated Annealing (SA)
1-st order	Gradient Descent (GD)	Stochastic Gradient Descent (SGD)

Table 2.5: Different solver categories that can handle constrained problems with discrete nonlinear noisy objective functions, including deterministic and stochastic as well as sampling-base, 0-th, and 1-st order solvers.

objective function takes a long time to calculate. This path was not chosen, as the calculating time was manageable without, though it might be a direction for future improvements (Kochenderfer and Wheeler 2018).

In practice, it was found that the large search space led to a slow convergence speed of deterministic solvers like the Matlab implemented Patternsearch solver compared to stochastic meta-heuristic solvers. These do not mathematically guarantee convergence to a global optimum, but they often find them faster. They use a non-greedy approach meaning they do not locally make the best choice, which allows for exploration and good global optimization capabilities, which concludes the choice of stochastic solvers in the optimization.

Both GD and SGD lack the ability to explore the optimization landscape to reach a global optimum and struggle with noisy objective functions, but they can still be useful as a hybrid approach. Combining the GA with SGD might be a way to mitigate the slow convergence at a later stage. Due to the additional complexity of the approach, this path was not chosen for the optimization.

The advantages and disadvantages of three different meta-heuristic algorithms are discussed in Table 2.6.

Method	Advantage	Disadvantage
PSO	Easy implementation Fast convergence at early stage	Low convergence speed at later stage Prone to premature convergence poor performance for discrete optimization
SA	Global convergence with slow cooling schedule	Slow convergence
GA	Distributed calculations Robustness Wide-spread adoption	Poor local search ability Prone to premature convergence Many hyper-parameters

Table 2.6: Advantages and disadvantages of the PSO, SA, and GA.

The PSO is easy to implement and has a fast convergence in the early stage, but is not very efficient for discrete problems as stated by Li et al. 2023. SA is used by Aboulissane et al. 2019 to optimize the workspace of a Delta robot. It can guarantee global convergence, but only with a logarithmic cooling schedule, making convergence very time-consuming (Kochenderfer and Wheeler 2018). GA is inefficient for local search, is not guaranteed to converge globally, and has many tunable parameters. Still, it is used extensively for geometric design optimizations, as shown in the section 1.3. It offers robustness to noise, can be run in parallel, and is non-greedy. Additionally, it explores the design space efficiently and finds good solutions comparatively fast, which is why it is used in this work. Some tuning was done to find good parameters as explained in section 2.10, but the default setting of the Matlab solver already gave a good starting point.

2.10 Genetic Algorithm

The GA is an optimization technique inspired by the process of natural selection. It was first applied to optimization problems by De Jong (JONG 1975) and belongs to the larger class of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover (EIBEN and SMITH 2015).

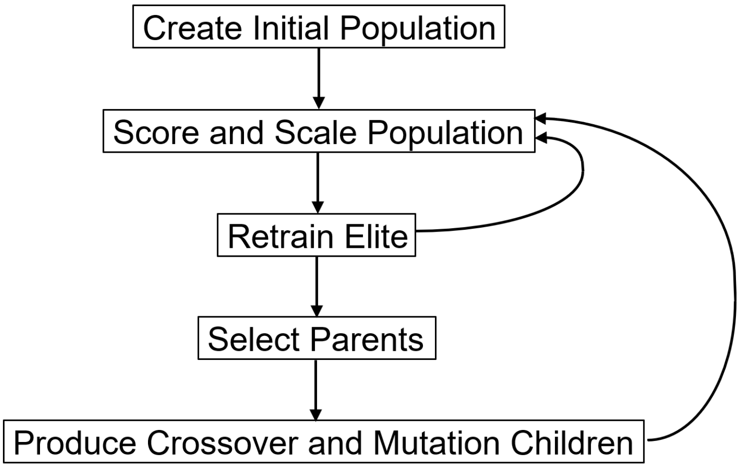

Create Initial Population
Score and Scale Population
Retrain Elite
Select Parents
Produce Crossover and Mutation Children

Figure 2.8: The general structure of the GA. After creating an initial population it is ranked and split into elite, crossover and mutation children, which make up the next generation (MATHWORKS 2022b).

A genetic algorithm mimics the process of natural evolution, a concept based on the Darwinian principle of reproduction and survival of the fittest. Matlab offers an implementation in their Global Optimization Toolbox (MATHWORKS 2022b), which is used for this problem. The default values of the Matlab solver were mostly kept as they produced good results. Its main steps are shown in Figure 2.8.

Parameters	Values	Matlab Function
Number of individuals	100	gacreationuniform
Percentage selected for reproduction	20%	selectionstochunif
Number of elites	2	
Crossover fraction	0.8	crossoverscattered
Mutation rate	0.1	mutationgaussian
Fitness function tolerance	10^{-6}	
Maximum stall generations	50	

Table 2.7: The values and function names of the GA concerning the initialization, reproduction, and break condition

First, 100 individuals are initialized from a random uniform distribution. The default value of 50 is increased here to be able to better explore the search space. After scoring and scaling the population, 20% are selected for reproduction. The stochastic uniform selection allows for multiple selections of the same individual. The two lowest-scoring individuals, the "elites", are fed back without changing their vector representation. 80% of the new population is created using crossover, combining the vector representations of two parents, and the rest is created by adding a Gaussian distributed random value to each component of a single parent. The mutation rate is inversely proportional to the number of design parameters. The three groups of children make up the next generation (Figure 2.9). The GA is terminated when the fitness improves less than the default value of 10^{-6} for 50 successive generations. The used parameters and functions are summarized in Table 2.7.

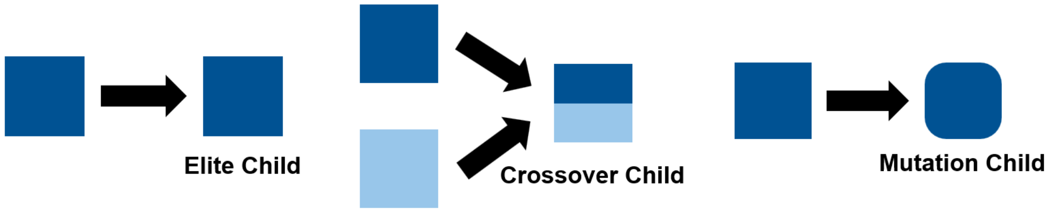


Figure 2.9: The three groups of children - elite, crossover and mutation - and their basic concept, as implemented in Matlab (MathWorks 2022b).

Chapter 3

Implementation

This chapter explains how the selected methods from the chapter 2 are implemented into the algorithm. The implementation is split up into four parts. The goals ,depicted in section 3.1; The kinematics, including the FK and the calculation of the kinematic Jacobian (section 3.2); the workspace, including its evaluation and discretization methods (section 3.3); and the optimization, where the design parameters are selected, the solver is described and the approach is summarized (section 3.4).

3.1 Goals

Primary goals	Secondary goals	Disregarded goals
Increased workspace volume	Optimized workspace shape	Walking patterns
Retained rotational symmetry	Avoidance of self-collisions	Stability
Distance to parallel singularities		Compliance with joint force limits of passive joints
Compliance with joint angle limits of active joints		Compliance with joint angle limits of passive joints

Table 3.1: Primary, secondary, and disregarded goals of this work.

The goal of this work is to optimize the characteristic lengths of the robot to maximize the workspace volume under certain conditions. The rotational symmetry should be retained, which influences the choice of the design parameters (Section 3.4.1). Furthermore, the distance from singularities (Section 3.3.1) and the compliance with joint angle limits of the active joints are integrated (Section 3.4.2). A useful shape of the workspace and avoiding self-collisions of the robot influence some decisions but are not explicitly addressed in this work. Additional requirements regarding stability when walking or crawling are not included in the optimization as they exceed the scope of this work. This specifically excludes the consideration of walking patterns or properties of the center of mass for the cost function. Especially since the robot has no additional DoF to control the center of mass, disregarding it might significantly reduce the usable workspace of the optimized robot due to it tipping over while stepping. Therefore, future work might be necessary to address this problem. Also, the joint force and joint angle limits of the passive joints are disregarded, since the properties of the physical joints are unknown.

3.2 Kinematics

The chapter includes the analysis of the analytic FK (section 3.2.1) and the kinematic Jacobian (section 3.2.2), which are necessary for evaluating points in the workspace. Methodes 2.1, 2.2, and 2.3 are used for implementing the kinematics.

3.2.1 Analytic Forward Kinematic

The robot has a four-way rotational symmetry, and the legs are only coupled at the active joints, so studying one leg assembly allows for a description of the full FK. Under the assumption that the passive links join to the leg at exactly 2 points and all joints are ideal, it allows for a novel analytic description of the kinematics. The leg assembly is depicted in Figure 3.1

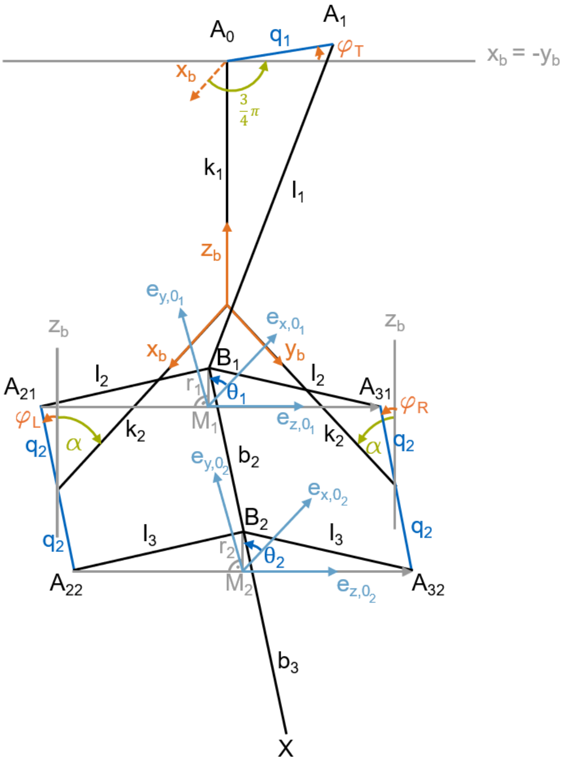


Figure 3.1: Visualization of a leg assembly. The base frame is depicted in orange, the leg assembly in black, besides the active links being blue. Two additional frames are introduced in light blue, used to describe the motion of two four-bar linkages.

Each leg assembly reminds of a push rod suspension in cars. Each of the six passive links connects the fixed and the moving frame so that the connection point on the moving frame moves on a sphere, with the center of rotation being the universal joint at the fixed frame. The movement is thereby restricted to the motion on a sphere, restricting one DoF in the radial direction. Four links are positioned similarly to the leg links of the proposed robot, while the fifth link, the steering rod, restricts the wheel's steering angle. The remaining DoF is mostly along the vertical axis and constrained by the suspension system. In the quadruped leg, the vertical motion is restricted by the top link, similar to a push rod suspension, and the rotation is mitigated by choosing two U-Joints instead of ball joints for the top link, effectively transferring the rotational moment from the upper links to the feet. This makes the steering rod obsolete. The selection of the top link for constraining the rotational DoF is arbitrary since the foot is modeled as a point, and only the translations of the foot are of interest. Unlike the suspension system, the leg link's attachment points can move, making for a more complex motion range.

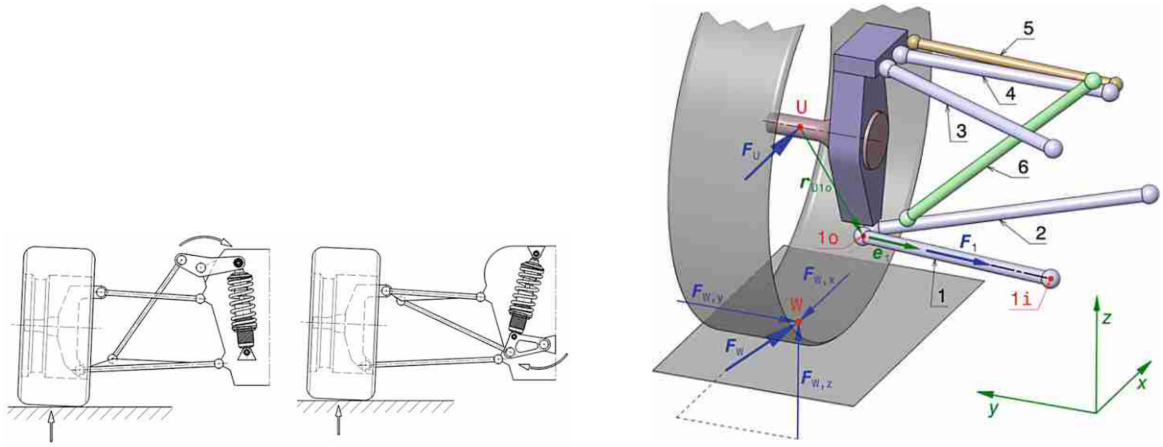


Figure 3.2: Push-Rod system (left) and Pull-Rod system (middle) in the side view of a Formula 1 car. On the right is the isometric view of the Push-Rod system, showing the six links that constrain the motion of the wheel. 1-4 correspond to the leg links, 5 is the steering rod, not present in the robot, and 6 is the push-rod, equivalent to the top link of each leg assembly (Trzesniowski 2019).

The upper and lower leg links are each modeled as connecting to the leg at the same point. This requires points B1 and B2 to lie on the circular intersection of the two spheres defined by the passive links, as shown in Figure 3.3. The top link can be connected to B1 or B2 to solve the FK analytically. Unlike the push rod system in cars, the top link should not be connected to the leg close to point B_2, because of potential intersections with the leg links. B_1 was chosen as the connection point to avoid self-intersections and ensure ground clearance.

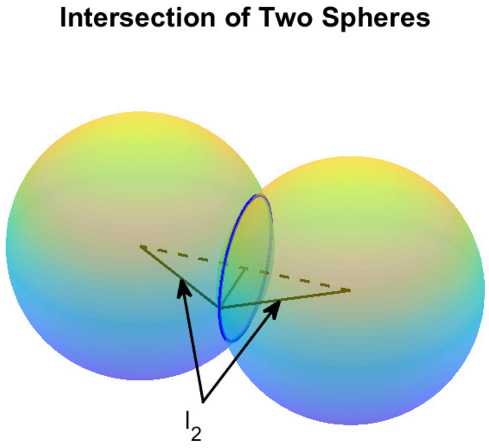


Figure 3.3: A point restricted to two spheres has to move along the circular intersection.

The angles ϕ_t, ϕ_l, and ϕ_r of the three active joints affecting the leg assembly are known. This defines the position of the five centers of the spheres through trigonometric relations relative to the base frame:

$$A_1 = \begin{pmatrix} q_1 \cos\left(\phi_t + \frac{3\pi}{4}\right) \\ q_1 \sin\left(\phi_t + \frac{3\pi}{4}\right) \\ k_1 \end{pmatrix} \tag{3.1}$$

$$A_{21} = \begin{pmatrix} k_2 - q_2 \cos(\phi_l) \sin(\alpha) \\ -q_2 \sin(\phi_l) \\ q_2 \cos(\phi_l) \cos(\alpha) \end{pmatrix} \qquad A_{31} = \begin{pmatrix} q_2 \sin(\phi_r) \\ k_2 - q_2 \cos(\phi_r) \sin(\alpha) \\ q_2 \cos(\phi_r) \cos(\alpha) \end{pmatrix} \tag{3.2}$$

$$A_{22} = \begin{pmatrix} k_2 + q_2 \cos(\phi_l) \sin(\alpha) \\ q_2 \sin(\phi_l) \\ -q_2 \cos(\phi_l) \cos(\alpha) \end{pmatrix} \qquad A_{32} = \begin{pmatrix} -q_2 \sin(\phi_r) \\ k_2 + q_2 \cos(\phi_r) \sin(\alpha) \\ -q_2 \cos(\phi_r) \cos(\alpha) \end{pmatrix} \tag{3.3}$$

B_1 and B_2 are defined on a circle as long the spheres intersect and are not identical. The center of the upper circle with radius r_1 lies on the midpoint M_1 of the line $A_{21}A_{31}$. This line also coincides with the rotation axis of B_1. The sphere's radius is the height of a triangle and a function of A_{21}, A_{31}, and the length l_2 (Figure 3.3). The same is true for the lower circle, which is defined by A_{22}, A_{32}, and the length l_3. The leg section $\overline{B_1B_2}$ acts as the coupler of a spatial four-bar linkage. Note that the active joint angles ϕ_l and ϕ_r only affect the position of the points A_{ij} and, therefore, the geometry of the spatial four-bar linkage. They influence the skew angle of axes $A_{21}A_{31}$ and $A_{22}A_{32}$, the position of M_1 and M_2, and the lengths M_1B_1 and M_2B_2. Only the length $\overline{B_1B_2} = b_1$ remains constant. The actual DoF of the spatial four-bar linkage is actuated by the angle ϕ_t. It influences the motion of the spatial four-bar linkage via the top link, which acts as a coupler to a second spatial four-bar linkage with points A_1 and B_1, moving on a circular trajectory. Since an analytic solution exists for the spatial four-bar linkage, the forward kinematic is solvable analytically.

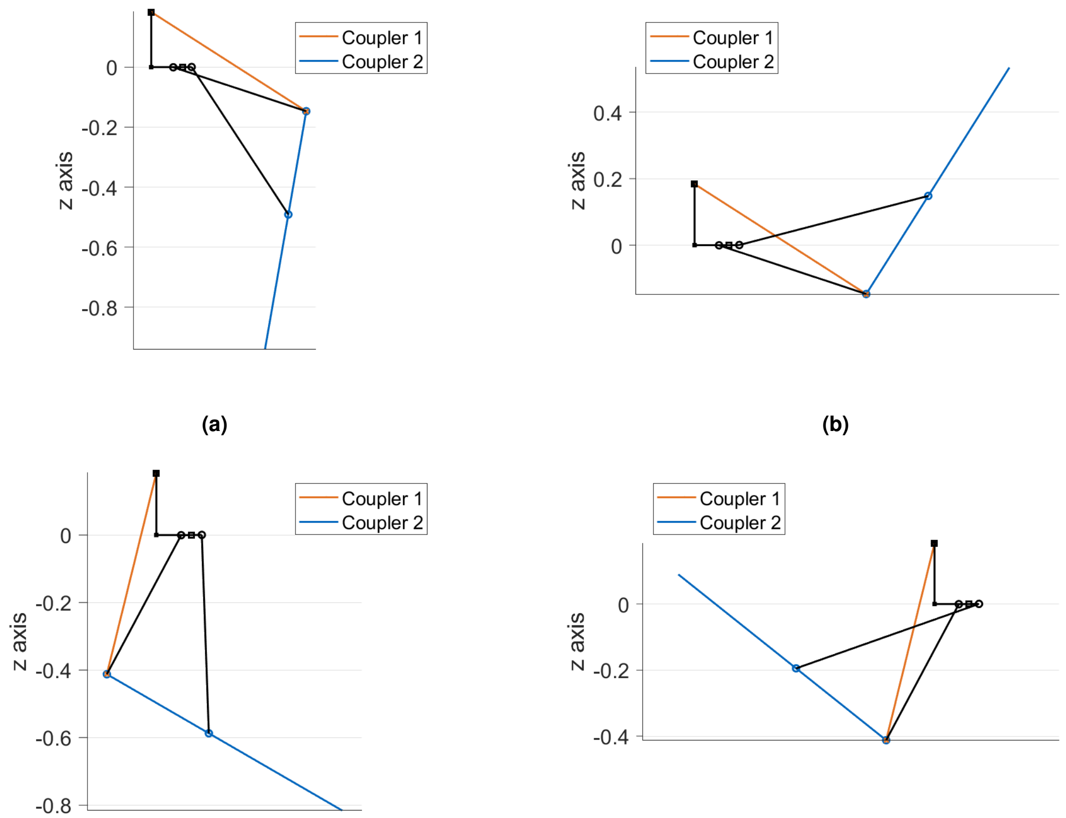
The four possible solutions of the leg configuration for a given set of active joint angles are shown in Figure3.4. In this case, all angles are set to 0. Each analytic solution corresponds to one aspect of the workspace. The robot should not be able to reach another aspect as it would either have to pass an RO singularity configuration or the geometry does not allow the transition between aspects (Merlet 2006). The solution corresponding to the blue configuration is the one used for this robot because the leg points in the right direction for walking. This constrains the solution space to the right aspect. The upper spatial four-bar linkage uses the first solution of equation 2.2 and 2.3 (section 2.1), while the lower uses the second.

$$_{O_1}B_1 = \begin{pmatrix} cos(\theta_1)\overline{M_1B_1} \\ sin(\theta_1)\overline{M_1B_1} \\ 0 \end{pmatrix} \tag{3.4}$$

where

$$sin(\theta_1) = -\frac{_{O_1}A_1(2)K_1 - {}_{O_1}A_1(1)\sqrt{_{O_1}A_1(1)^2 + {}_{O_1}A_1(2)^2 - K_1^2}}{_{O_1}A_1(1)^2 + {}_{O_1}A_1(2)^2} \tag{3.5}$$

$$cos(\theta_1) = -\frac{_{O_1}A_1(1)K_1 + {}_{O_1}A_1(2)\sqrt{_{O_1}A_1(1)^2 + {}_{O_1}A_1(2)^2 - K_1^2}}{_{O_1}A_1(1)^2 + {}_{O_1}A_1(2)^2} \tag{3.6}$$



(c) (d)

Figure 3.4: The four different assembly modes of a leg assembly, where (a) is the used configuration.

with

$$K_1 = \frac{1}{2\overline{M_1B_1}}(l_1^2 - \overline{M_1B_1}^2 - {}_{O_1}A_1(1)^2 - {}_{O_1}A_1(2)^2 - {}_{O_1}A_1(3)^2) \tag{3.7}$$

The length of the virtual lever $\overline{M_1B_1}$ of the spatial four-bar linkage can be calculated from the distance of points A_{21} and A_{31} (equation 3.2) and length l_2:

$$\overline{M_1B_1} = \sqrt{l_2^2 - ||(A_{21} - A_{31})||^2} = r_1 \tag{3.8}$$

With equation 3.4 point ${}_{O_1}B_1$ can be calculated, knowing ${}_{O_1}A_1$. The points are defined in the coordinate frame O_1, so ${}_bA_1$, defined in equation 3.1, has to be transformed to the coordinate frame of the first spatial four-bar linkage. $\hat{A}_1$ is the homogeneous representation of A_1.

$${}_{O_1}\hat{A}_1 = {}_{O_1}T_b\,{}_b\hat{A}_1 \in \mathbb{R}^{1\times 4} \tag{3.9}$$

The origin of O_1 lies at M_1. The z-axis points towards A_{31} and the x-axis is defined as $e_{x,O_1} = e_{z,O_1} \times e_{z,b}$. The orientation of the x-axis was chosen so it avoids possible configurations where r_1 and e_{x,O_1} are parallel. To form a right-handed coordinate system, the unit y-axis is defined as $e_{y,O_1} = e_{z,O_1} \times e_{x,O_1}$. Equation 3.10 shows how the transformation matrix ${}_bT_{O_1}$ is constructed. Its inverse ${}_{O_1}T_b$ is defined in equation 3.13 . The rotation matrix ${}_bR_{O_1}$

represented in the body frame is defined by the unit vectors of O_1. The translation vector to the origin of O_1 is ${}_bM_1$.

$$ {}_bT_{O_1} = \begin{pmatrix} {}_bR_{O_1} & {}_bM_1 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{4\times4} \tag{3.10} $$

with

$$ {}_bM_1 = \frac{A_{21} + A_{31}}{2} \tag{3.11} $$

and

$$ {}_bR_{O_1} = \begin{pmatrix} e_{x,O_1} & e_{y,O_1} & e_{z,O_1} \end{pmatrix} \in \mathbb{R}^{3\times3} \tag{3.12} $$

$$ {}_{O_1}T_b = {}_bT_{O_1}^{-1} = \begin{pmatrix} {}_bR_{O_1}^T & -{}_bR_{O_1}^T\,{}_bM_1 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{4\times4} \tag{3.13} $$

Next, ${}_{O_1}B_1$ is converted to the frame of the second spatial four-bar linkage using two transformations:

$$ {}_{O_2}\hat{B}_1 = {}_{O_2}T_{b\,b}T_{O_1\,O_1}\hat{B}_1 \tag{3.14} $$

The origin of base O_2 is located at M_2, and the coordinate system is again oriented relative to the rotation axis and the z-axis of the base as shown in Figure 3.1. Similar to equations 3.10 and 3.13 the transformation matrices ${}_bT_{O_2}$ and ${}_{O_2}T_b$ can be constructed. ${}_{O_2}B_2$ can be calculated using equation 3.15 now using the second solution of the spatial four-bar linkage equation (Equation 2.1).

$$ {}_{O_2}B_2 = \begin{pmatrix} cos(\theta_2)\overline{M_2B_2} \\ sin(\theta_2)\overline{M_2B_2} \\ 0 \end{pmatrix} \tag{3.15} $$

where

$$ sin(\theta_2) = -\frac{{}_{O_2}B_1(2)K_2 + {}_{O_2}B_1(1)\sqrt{{}_{O_2}B_1(1)^2 + {}_{O_2}B_1(2)^2 - K_2^2}}{{}_{O_2}B_1(1)^2 + {}_{O_2}B_1(2)^2} \tag{3.16} $$

$$ cos(\theta_2) = -\frac{{}_{O_2}B_1(1)K_2 - {}_{O_2}B_1(2)\sqrt{{}_{O_2}B_1(1)^2 + {}_{O_2}B_1(2)^2 - K_2^2}}{{}_{O_2}B_1(1)^2 + {}_{O_2}B_1(2)^2} \tag{3.17} $$

with

$$ K_2 = \frac{1}{2\overline{M_2B_2}}(l_1^2 - \overline{M_2B_2}^2 - {}_{O_2}B_1(1)^2 - {}_{O_2}B_1(2)^2 - {}_{O_2}B_1(3)^2) \tag{3.18} $$

and

$$ \overline{M_2B_2} = \sqrt{l_3^2 - ||(A_{22} - A_{32})||^2} = r_2 \tag{3.19} $$

	Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	β_z
X_1	ϕ_l	ϕ_r			ϕ_t		0
X_2		ϕ_l	ϕ_r			ϕ_t	$\pi/2$
X_3			ϕ_l	ϕ_r	ϕ_t		π
X_4	ϕ_r			ϕ_l		ϕ_t	$3\pi/2$

Table 3.2: Table showing which motor angles are used for which leg assembly.

With equation 3.20 ${}_{O_2}X$ is calculated, which then needs to be transformed back to the base frame by applying ${}_{b}T_{O_2}$ to its homogeneous representation.

$$
{}_{O_2}X = B_1 + \frac{b_1 + b_2}{b_1}({}_{O_2}B_2 - {}_{O_2}B_1) \tag{3.20}
$$

Lastly, the calculation must be performed once for each leg, using the corresponding active joint angles and then rotated by β_z along the z-axis of the base frame to find all four foot positions. Table 3.2 summarizes, which active joint angles need to be used to calculate each end effector's position and by which β_z the solution needs to be rotated.

The analytic solution of X_1 corresponding to the right aspect was determined with the symbolic toolbox in Matlab and can be found in the Appendix A. It describes the body-related forward kinematics of a single leg.

3.2.2 Kinematic Jacobian

The ground-related kinematic Jacobian is required to calculate the distance from the RO singularity with the local performance index p_{RO}. It is calculated by defining the forward kinematics with respect to the ground and using the numeric forward differentiation.

The ground-related forward kinematics of foot X_1 can be found by transforming the solution to the ground frame. The ground frame is defined as described in the section 2.4. The unit z-vector $e_{z,g}$ is calculated by the cross product of the unit vector $\hat{u}_{34}$ and $\hat{u}_{32}$, shown in equation 3.21 so that it is perpendicular to the ground and points away from it. the unit x-vector $e_{x,g}$ is defined as the unit angular bisector of $\hat{u}_{34}$ and $\hat{u}_{32}$ as shown in equation 3.22. To form a right-handed coordinate system, the unit y-axis is defined as $e_{y,g} = e_{z,g} \times e_{x,g}$.

$$
e_{z,g} = \hat{u}_{34} \times \hat{u}_{32} \tag{3.21}
$$

$$
e_{x,g} = \frac{\hat{u}_{34} + \hat{u}_{32}}{||\hat{u}_{34} + \hat{u}_{32}||} \tag{3.22}
$$

A transformation matrix ${}_{g}T_{b}$ is defined similarly to equation 3.13 to transform end effector X_1 to the ground frame. The translation vector is ${}_{b}X_3$, as the origin lies in the opposing foot.

$$
{}_{g}T_{b} = {}_{b}T_{g}^{-1} = \begin{pmatrix} {}_{b}R_{g}^{T} & -{}_{b}R_{g}^{T}\,{}_{b}X_3 \\ 0 & 1 \end{pmatrix} \in \mathbb{R}^{4\times4} \tag{3.23}
$$

To approximate the kinematic Jacobian of X_1, the same steps of calculating the FK of all four feet are done six times. In each iteration, one of the active joint angles is increased by a small delta of 10^{-6} to calculate the finite differences with the Forward Euler scheme. For all joint deviation, the transformation to the corresponding ground frame is performed. The matrix $J_{k,X_1} \in \mathbb{R}^{3\times 6}$ relates all six angular velocities of the joints to the three velocities of the end effector X_1.

The performance index p_{RO} requires the kinematic Jacobian of all four legs. The deviated FK needs to be performed a total of six times, once for each active joint angle. For each leg and each joint deviation, the ground must be defined accordingly, requiring a total of 28 transformations - four to transform the end effector position of each leg without deviation and 24 for the deviated configurations.

3.3 Workspace

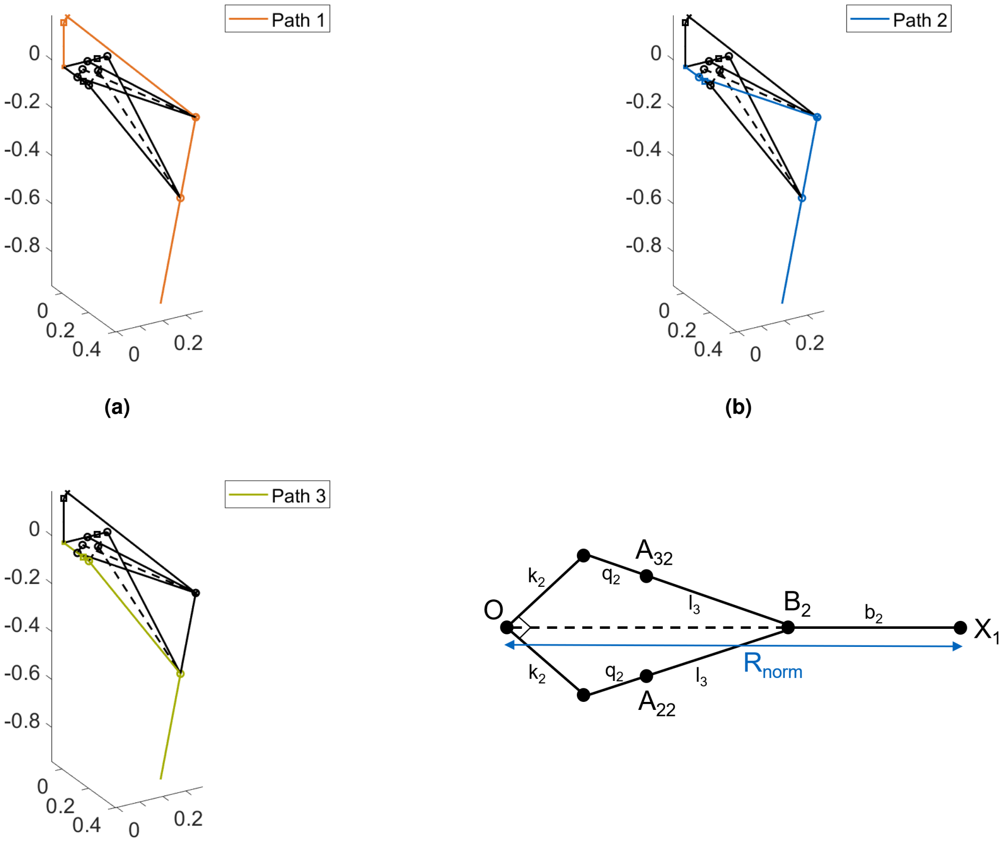
The workspace is calculated for every member of the population by random sampling in the active joint space. Each sample is evaluated (section 3.3.1) to determine whether it belongs to the workspace. If it does, the sample is allocated to a three-dimensional voxel. This voxel is then marked as inside the workspace. The sum of all marked voxels defines the workspace (section 3.3.2).

3.3.1 Evaluation

As introduced in section 2.6 the workspace is evaluated by three local performance indices p_j. They are multiplied to determine the local performance P_i, corresponding to a specific configuration i. Summed over all configurations inside the volume V, which should include all potentially reachable end effector points, and normalized by the volume itself, they form the WVI. The integral can be approximated using the discretization method by evaluating P on N voxels inside the discretized space. The negation is introduced to use it as the objective function to a minimization problem in section 3.4.

$$WVI = -\frac{\int_V P dV}{\int_W dV} \approx -\frac{1}{N}\sum_{i=1}^{N} P_i, \; where \; P_i = \prod_{j=1}^{3} p_j \tag{3.24}$$

To guarantee that the whole reachable workspace lies inside V, a common approach is to use the sum of all link lengths as the radius R_{norm} of a sphere with the center defined relative to the base. This has also been done, for example, by Silva et al. (SILVA et al. 2014b) for the Delta PM. For parallel tree structures, each end effector has its own normalization volume, and the radius should be the sum of one of the chains connecting the respective end effector to the base. To simplify the calculations, the normalization sphere is defined identically for all end effectors. Its center is located at the origin of the base frame. Each end effector can be reached through five different paths, three of which have unique length compositions. From looking at Figure 3.5, paths 1 and 2 are longer than path 3 in most designs. To minimize the size of V, path 3 was selected to determine the radius. A smaller volume leads to smaller



(a)

(b)

(c)

(d)

Figure 3.5: The three possible paths from the base's origin to the end effector, where path 3 (c) is used for determining the maximum reach. It is used to calculate R_{norm}, shown in (d).

memory requirements as shown in section 3.3.2. A slight improvement to the path length can be done by noting that each leg has a second instance of path 3, on which k_2 is perpendicular to the first. Considering the largest possible distance of B_2 from the base yields a lower bound of the maximum reach as shown in Figure 3.5 (d).

$$R_{norm} = \frac{1}{\sqrt{2}}k_2 + \sqrt{(q_2 + l_3)^2 - \frac{1}{2}k_2^2} + b_2 := 1 \tag{3.25}$$

Since the WVI is independent of scale, the scale can be defined by non-dimensionalizing all lengths by the radius R_{norm} (Equation 3.25). Volume V, therefore, has the size $4\pi/3$.

The performance index p_{VI} is defined as 1 if the point lies inside the reachable workspace and 0 if the point lies outside. A point is reachable if the solution of the analytic FK is real and finite.

$$p_{VI} = \begin{cases} 1, & X(\Phi) \in \mathbb{R} \\ 0, & X(\Phi) \notin \mathbb{R} \end{cases} \tag{3.26}$$

Performance index p_F penalizes undesired positions of an end effector. These might be areas where the robot intersects itself or does not provide a useful task space for the considered end effector. If areas in the normalization volume should be avoided, they can be included as follows:

$$p_F = \begin{cases} 1, & X(\Phi) \in W_d \\ 0, & X(\Phi) \notin W_d \end{cases} \tag{3.27}$$

where W_d are the desired areas of the workspace.
When the feet reach a configuration where the ground intersects with the robot, the solution should be excluded from the useful workspace. Also, only points should be counted for the workspace, which can be reached without the robot intersecting itself. Checking for intersections requires analyzing multiple bodies for every sample. The shape of the ground is unknown, and the decision if there exists a path that does not include intersections requires information on past and future samples. These unknowns led to the choice of labeling areas as "undesired" where intersections are likely to occur. Defining the areas with upper bounds that guarantee no ground or self-intersection restricts the optimization unjustly while setting the bounds too generously includes many unreachable points in the workspace. The balance was found through trial and error. A simple bound to reduce ground intersection is the xy-plane of the body. If any foot is above that plane, the configuration is penalized by p_F.
Self-intersection was found to primarily occur at the legs and the leg links. The latter is addressed in section 3.4.2. For the foot X_1, the plane which fulfills $x = -y$ was chosen as a boundary. The other feet have symmetric restrictions to avoid intersections with the opposing leg but allow adjacent legs to reach the same space. Walking patterns might benefit from feet reaching the same space, so a compromise must be found to avoid restricting the workspace too much. The geometry of the adjacent legs prevents large relative motion, which makes self-intersections in these areas unlikely. The leg intersection of opposing legs had to be accounted for, as without, the algorithm found the configuration shown in Figure 3.6 (left) to be optimal, which clearly includes self-intersections.

Each foot has a quarter of the volume V as a useful workspace to prevent intersection with itself and the ground, which is visualized in Figure 3.6 (right). The simple shape makes it easy to define boundaries for the discretized voxel space described below.

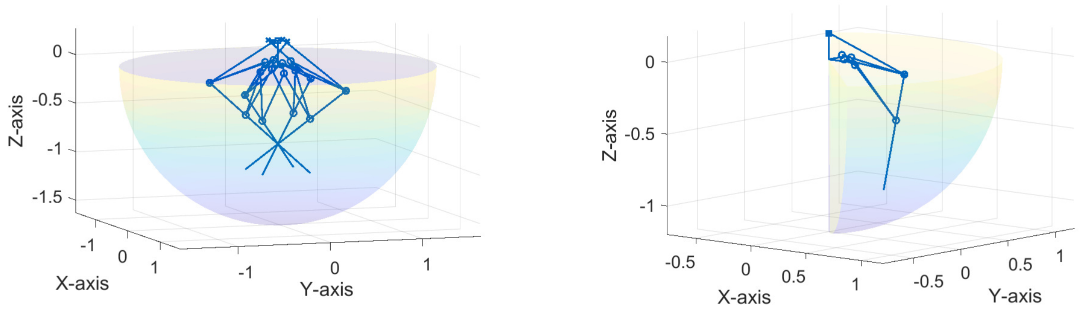


Figure 3.6: Shown is the volume W_d, restricted only by the xy-plane (left), leading to an undesired solution with crossed legs and the restriction to a quarter of the sphere, which is used in the optimization (right).

The performance index p_{RO} is introduced to address the distance from the RO singularity. It is set to one if the largest singular value of the kinematic Jacobian is larger than the largest allowed amplification factor W_{max} and zero otherwise (Equation 3.28). σ_{max} describes the largest error amplifications from the input to the output and the largest wrench amplification from the output to the input. A value close to infinity indicates a singularity. The approach, described by Li and Ye (LI and YE 2003) as absolute error sensitivity, is simple compared to alternatives like screw theory (YUAN et al. 2023), which is the reason for using it in this work. The main drawback of the approach is the dimensional inconsistency of a general Jacobi matrix, as mixing rotation and translation does not allow for finding a physically sensible distance from the RO singularity. By studying only the translational velocities via the kinematic Jacobian, rotational singularities can not be detected, but they can be neglected, assuming that no large moments act on the feet. Additionally, gaining a rotational DoF at one of the end effector points does not matter for the workspace size nor the controllability, so studying the kinematic Jacobian is sufficient for detecting singularities. This reduced set of singularities is called parallel singularities by Gosseling (GOSSELIN and ANGELES 1990).

$$p_{RO} = \begin{cases} 1, & \sigma_{max} < W_{max} \\ 0, & \sigma_{max} \geq W_{max} \end{cases} \tag{3.28}$$

where $\sigma_{max} = ||J_k||$ and W_{max} is the largest allowed amplification factor, which is determined by approximating the torque to weight ratio of the servo motors. This allows to give a broad estimation of the force acting on the feet due to the weight of the motors and, from that, the maximally allowed force to torque amplification from the feet to the motors. The output forces and the input torque are related by equation 3.29.

$$\tau = {J_k}^T F \tag{3.29}$$

Assume the size of the output force vector is bounded to one via the 2-norm: $||\hat{F}|| < 1$
Further assume that $||J_k^T|| = ||J_k|| < W_{max}$. It follows that the 2-norm of the torque is bounded as well. A higher value of σ_{max} therefore implies a larger maximum torque-to-force ratio in the studied configuration. In any direction, a unitary force is amplified by at most W_{max} in the torque space.

$$\frac{||\tau||}{||F||} = ||J_k|| < W_{max} = \frac{||\tau_{max}||}{||\hat{F}||} \tag{3.30}$$

The order of magnitude for the torque-to-force ratio W_{max} is estimated with an exemplary servo motor. The servo BLS-H50B from BlueBird (LIN 1978) is currently used for the prototype, shown in Figure 1.1. At 7.4V, it produces a torque of 4.91 Nm and a speed of 9.52 rad/s, while weighing 83g. This amounts to a power-to-weight ratio of 563,53 W/kg, which is a sensible value for a mid-range servo motor used for robotics applications. It is assumed that a force equal to the gravitational force generated by all six motors acts on one of the feet while the others remain on the ground. The weight of the rest of the components is neglected. All six motors should be able to generate the resulting torques at the active joints for the robot to be in a useful configuration independent on which leg the force acts on. In a gravity field of 9.81 N/kg, and with $||J_k|| \leq 1$ at most 4.89 Nm is generated in the joint space, which can be handled by a single motor (4.89 Nm<4.91 Nm). The norm of the kinematic Jacobian should, therefore, not exceed 1 for any leg. The distance workspace is not Euclidian, so the 2-norm has less physical meaning in this space. Therefore, the FK for each leg defined in the corresponding ground-related Cartesian frame is differentiated to form the kinematic Jacobian, as explained in the section 3.2. Ratolikar et al. (RATOLIKAR and R 2020) used a similar approach for deriving torque requirements, but had more knowledge on the load cases during walking.

3.3.2 Discretization

As the discretization method is chosen in section 2.5 to calculate the workspace, several discretization parameters must be defined through an iterative process. This includes the size of the discretized space, the voxel size, and the break criterion for exiting the sampling loop.

The optimization is limited by computation time and available memory. The following hardware specifications are used for the optimization:

Property	Specification
Processor	Intel Core i7-1165G7
Processor Speed	2.8 GHz
Number of parallel workers	4
RAM	16 GB
Matlab version	2022b

Table 3.3: Hardware specifications including the processor, its speed, the number of parallel workers, the memory, and the used version of Matlab.

The size of the distance workspace is constructed from information on the maximum reach. As the feet must lie inside the non-dimensionalized volume V, the largest possible distance between two feet is $2R = 2$. The smallest distance is zero. Therefore, the range D_L of the space is two in all three dimensions: $D_L \in [0, 2]$

The voxel size v_s of 0.05 with the upper bound of $N_{samples}$ set to 10^5 was found to give good results. An early break of the sampling loop is possible if none of the last 100 samples that have fallen inside the workspace have visited new samples. This was done to speed up calculation in the late stage, where samples take longer to calculate on average due to the increased WVI. The optimized geometry takes, on average, 30,795 samples before exiting. A lower bound on the percentage of new samples found in a sliding window larger than 100 as a breaking criterion was not implemented (PORGES 2013). Besides the window size, it introduces the break percentage as another hyper-parameter to be tuned.

The range D_L in each direction is divided by the voxel size v_s and rounded up to the nearest integer. The product of all three dimensions reveals the number of voxels N in the workspace, shown in equation 3.31. As the foot points can be in two configurations for a set of distances, as shown in Figure 3.7, the number of voxels is doubled to address both. The discretized space, therefore, includes 128.000 Voxels

$$N = \left\lceil \frac{D_L}{v_s} \right\rceil^3 * 2 = d_L^3 * 2 = 128.000 \tag{3.31}$$

The position in the continuous distance space is represented by $L \in \mathbb{R}^{1\times3}$. With equation 3.32 the location of L is mapped to the smallest valued corner L_{grid} of the corresponding voxel in a three-dimensional Euclidean grid space.

$$L_{grid} = \left\lfloor \frac{L}{v_s} \right\rfloor = \begin{bmatrix} L_{grid,1} \\ L_{grid,2} \\ L_{grid,3} \end{bmatrix}, \quad \text{where} \quad L_{grid,i} \in \{0, 1, 2, \ldots, d_L\} \quad \text{for} \quad i = 1, 2, 3. \tag{3.32}$$

Since L_{grid} is a positive zero-based subset of $\mathbb{Z}$, equation 3.33 can be used to uniquely map the location in the grid space to an index i of a vector, which stores the information, whether a voxel is reached or not. A value of one is added to conform with Matlab's vector notation.

$$i = (L_{grid,1} + L_{grid,2} * d_L + L_{grid,3} * d_L^2) * C + 1 \in \{1, 2, 3, \ldots, N + 1\} \tag{3.33}$$

with

$$C = \begin{cases} 1, & X_1X_2 \cdot (X_1X_3 \times X_2X_4) < 0 \\ 2, & X_1X_2 \cdot (X_1X_3 \times X_2X_4) \geq 0 \end{cases} \tag{3.34}$$

The cross product between the two lengths L_5 and L_6 points in the direction of the distance vector between them. The sign of the dot product between the L_1 and the distance vector defines the assembly mode of the tetrahedron, as shown in Figure 3.7. This holds as long as the shortest distance between L_5 and L_6 is in between the foot points X_i and is used to determine the value of C (Equation 3.34).

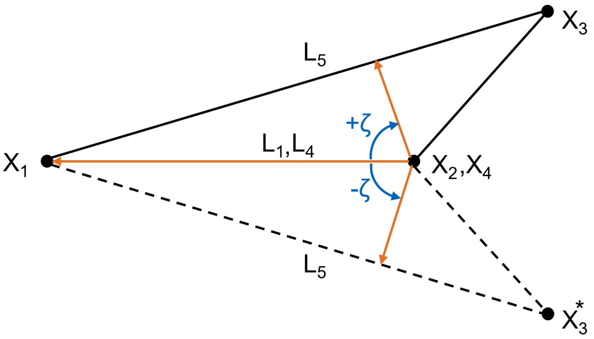


Figure 3.7: The two possible assembly modes of the distance space, which can be differentiated by the sign of angle ξ.

The optimal solution requires around 25s to calculate all 30,795 samples for the workspace estimation. The GA calculated 96 generations with 100 individuals using 4 cores, which would take 16.6h if only the optimal solution is repeatedly calculated. Smaller workspaces in the early optimization stage take less time, as explained in section 2.7, so it represents an upper bound for the calculation time used for the iterative process.

The ideal voxel size is influenced by the size of the distance workspace, as a larger space increases memory demands. Storing all $40^3 * 2 = 128,000$ voxels requires 16 kB of memory, which is significantly lower than the available RAM per worker (table 3.3). With a 6D workspace representation, around 1.02 GB would be required, which cannot be handled efficiently by the hardware and led to reducing the dimensions of the workspace to three (section 2.4). While a smaller voxel size can approximate the shape and size of the workspace more accurately, at some point, the limited number of samples is unlikely to find all voxels inside the workspace, which creates holes and fuzzy boundaries and significantly underestimates the volume. The probability of covering all states with 100,000 uniformly distributed samples is more than 95% if less than 33,389 states exist, resembling the coupon collector's problem (Erdös and Renyi 1961). Due to the non-uniformity of the sampling distribution, even fewer states can be covered by the sampling. Although the optimization cares about the relative change in workspace size, at some point, a saturation effect occurs when almost all samples reach new voxels. The voxel size of 0.05 is chosen to ensure that the workspace is approximated well by the smallest possible voxel size, which allows for proper sampling coverage.

3.4 Optimization

The optimization is initialized by selecting a set of design parameters. It then picks a random joint angle and calculates the end effector points through the FKs, which are evaluated regarding the performance criteria, and the corresponding voxel is marked accordingly. This is repeated until the sampling loop terminates. The design parameters are varied, and the workspace calculation is repeated until the GA is terminated.

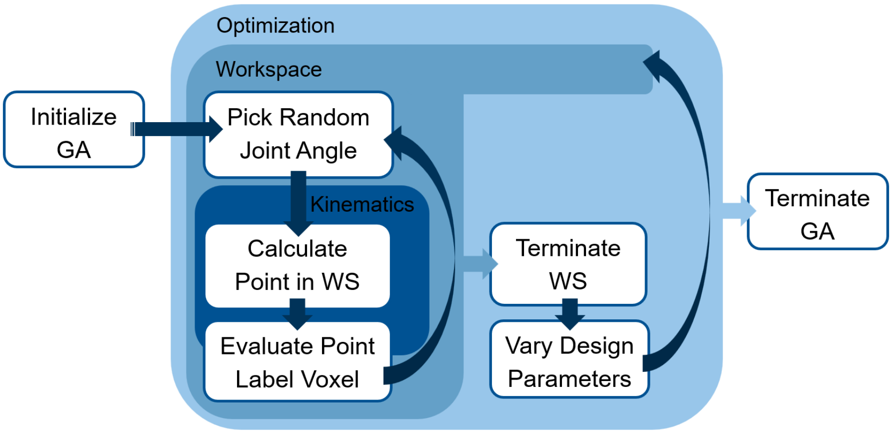


Figure 3.8: The general structure of the optimization algorithm. The workspace calculation is done by sampling in the joint space. After the sampling is terminated, the design parameters are varied, and the workspace calculation is repeated until the GA terminates.

The chapter first set the goals for the optimization. It then discusses the selection of design parameters. Thereafter, the main properties and parameters of the used genetic algorithm are explained. Finally, the optimization approach is outlined, and the boundary conditions are defined.

3.4.1 Design Parameters

As pointed out in the section 2, the optimization benefits from a reduction in design parameters. The symmetry and the simple geometric shapes of the robot allow for applying the design space reduction.

The 31 bodies of the robot include the passive links, the legs, the base, and the motor links. The 20 passive links are represented as one-dimensional beams whose lengths are parameterized. Retaining rotational symmetry, these 20 parameters can be reduced to five. Further,

each leg should be in the middle when the active joint angles are zero, reducing the parameters to three, which is assumed to lead to a more symmetric workspace.
Each leg is identical and modeled as a straight beam with two parameters. Using the analytical method, as described in section 2, the passive links connect to two points on the leg. A third point is considered the end effector of the leg assembly touching the ground. The distance between these three points is defined by two parameters: b_1 and b_2.
The base can be described with two parameters k_1 and k_2 due to symmetry, as shown in Figure 3.9 (right).

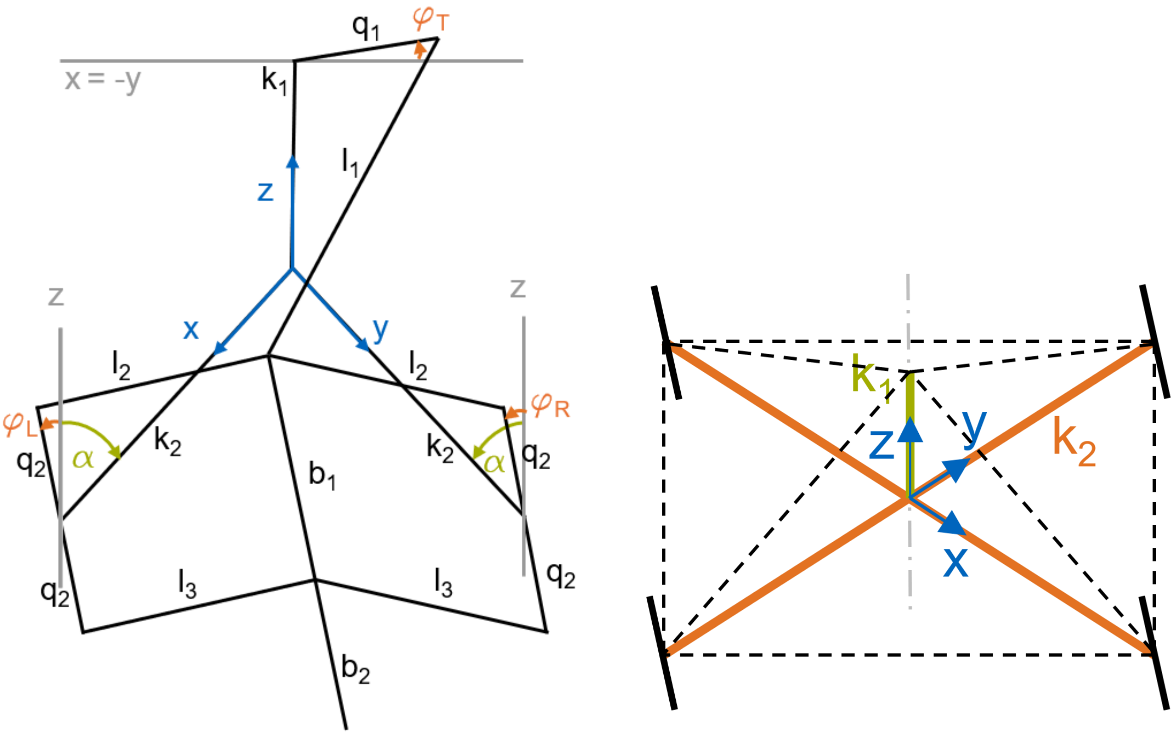


Figure 3.9: The parameters of the leg assembly (left) and the base (right) showing how the geometry can be reduced to ten parameters.

When viewed from the base frame, the yaw and roll angle of the joint axis is set so the leg assemblies remain symmetric to the leg plane. The pitch of the joint axis is defined by parameter α.
The four lower motor links and the two upper motor links are each identical and modeled as a straight beam with three connection points. The four passive links connect to the two endpoints of each motor link, and the base is connected to the middle of the link, requiring one parameter that defines the distance from the middle to each endpoint. This is done to reduce the number of design parameters and equally distribute the forces to the upper and lower links. Two motor parameters q_1 and q_2 are required, one each for the upper and lower motor links.

In summary, a reduction of the design space to 10 parameters is proposed. Two lengths and one angle define the body, two length the motor links, three the passive links and two the geometry of the legs. These are represented by the vector $k = [k_1, k_2, \alpha, q_1, q_2, l_1, l_2, l_3, b_1, b_2]$.

3.4.2 Approach

The Optimization can be formulated as follows:

$$
\begin{array}{lllll}
\min\limits_{k} & WVI = -\frac{1}{N}\sum_{n=1}^{N} P_n(k,\Phi) & & & \\
\text{s.t.} & k_i & \leq & k_{max,i} & \\
 & k_i & \geq & k_{min,i} & i = 1,2,\dots,10 \\
 & \Phi_j & \leq & \frac{\pi}{4} & \\
 & \Phi_j & \geq & -\frac{\pi}{4} & j = 1,2,\dots,6
\end{array}
\tag{3.35}
$$

It uses the WVI as the objective function to maximize the workspace by varying the geometrical design parameters of the robot. The WVI evaluates the size of the useful workspace compared to the size of the robot while respecting the constraints depicted in equation (3.35). A genetic algorithm, implemented in MATLAB, is used to minimize the objective function. Reducing the objective function leads to an increased workspace.

Both the active joint angles Φ_j and the design parameters k_i are restricted to sensible sizes. As the joint angles Φ_5 and Φ_6 share the same rotation axis, the two motor joints can only swipe half of the rotation plane to avoid intersections. The angle is defined to be zero centered, which results in a lower bound of $-\pi/4$ and an upper bound of $\pi/4$. The other four motors would allow for a larger range. Still, the same bounds were set. The reasoning follows the cost reduction aspect of reducing the DoF of the quadruped. Besides reducing the number of motors, all motors have similar requirements, which makes them easier to produce and install.

Parameters	Boundaries
k_1	[0,1]
k_2	[0,1]
α	$[-\pi/2,\pi/2]$
q_1	[0,1]
q_2	[0,1]
l_1	[0,1]
l_2	[0,1]
l_3	[0,1]
b_1	[0,1]
b_2	[0,1]
Φ_j	$[-\pi/4,\pi/4]$

Table 3.4: The boundaries of the design parameters and the active joint angle limits used in the optimization.

All non-dimensionalized design parameters are bounded between zero and one. Negative values are not useful in this context, as they relate to negative lengths. Values larger than one cannot be reached by k_2, q_2, l_3, and b_2 as they are summed up in R_{norm}. The values k_1, q_1, l_1, l_2, and b_1 are also not allowed to exceed the maximum range of the robot. This is a reasonable boundary, which is supported by the fact the optimal solution, depicted in the section 4, does stay away from the upper bound. Angle α is not divided by R_{norm}, since it is not a length. It is defined between $-\pi/2$ and $\pi/2$. At an angle of zero, the motor link is perpendicular to k_2 while with $\alpha = \pm\pi/2$, the motor link rotates inside the XY-plane of the base. Large absolute values of α may result in links l_2 and l_3 crossing over each other. As for the index p_F, self-intersection is a secondary goal (Table 3.1) and is only addressed implicitly through sensible boundaries. The sensibility is discussed in section 4.

Algorithm 1 Objective Function

```
Require: $\Phi \in \{-\pi/2, \pi/2\}$
  Normalize $k$
  for $samples = 1, \dots, 10^5$ do
      Generate random $\Phi$
      $p_{VI}$ = isConditions1Met($\Phi, k$)
      $p_F$ = isConditions2Met($\Phi, k$)
      if $p_{VI}$ and $p_F$ then
          $L$ = calculateDistances($\Phi, k$)
          $i$ = determineVoxelIndex($L$)
          if $Bin(i)$ is unvisited then
              $p_{RO}$ = isCondition3Met($\Phi, k$)
              if $p_{RO}$ then
                  mark $Bin(i)$ as visited
          else
              Increment $BinRevisited$
      if $BinRevisited > 100$ then
          Break Loop
  $WVI$ = calculateWVI($Bin, N$)
  return $WVI$
```

Algorithm 1 shows the pseudo-code for the objective function. It includes the calculation of the performance indices and the sampling iteration. Several methods to improve calculation speed were implemented. The FK and Jacobi matrix calculations are converted to MEX files, the performance index p_{RO}, which takes the most time, is only calculated once for each voxel inside the workspace, and an early break condition is implemented.

Chapter 4

Results

The implementation, including the analytical solution detailed in Appendix A, along with the workspace and optimization, successfully achieved the objectives outlined in Chapter 3.1. The main goal of increasing the workspace volume was achieved with the parameter set depicted in Table 4.1. They are listed both non-dimensionalized and rescaled to the size of the unoptimized prototype. Rescaling was done using the prototype's characteristic length, which is 27.8 cm. Also, the unoptimized parameter values are presented.

Parameters	Non-dimensionalized optimized values	Rescaled optimized values (cm)	Values of unoptimized prototype (cm)
k_1	0.1532	5.78	5
k_2	0.1229	4.63	9
α	-	1.5608	$\pi/6$
q_1	0.0465	1.75	2
q_2	0.0361	1.36	2
l_1	0.5140	19.38	12
l_2	0.3943	14.87	10
l_3	0.5031	18.97	11
b_0	0	0	2
b_1	0.2913	10.98	8
b_2	0.3810	14.36	20
Workspace W_g	1.34	51,288 cm^3	6,361 cm^3

Table 4.1: The values of the optimized parameters, both non-dimensionalized and rescaled and the prototype's parameters values as well as the corresponding sizes of the ground-related workspace.

In the ground-related frame, a eight-fold volume increase relative to the robot's size was attained (Figure 4.1). It was estimated by sampling with 10^6 points and using a hull surface as the boundary. The workspace is both wider and closer to the center. The two-dimensional projections of the upper and lower assembly mode of the 3D distance manifold are visualized in Appendix B. The optimal workspace volume index (WVI) turned out to be 0.02426 in the discretized lengths space. In the ground-related and re-scaled frame, a workspace volume of $57,342cm^3$ is reachable and complies with the constraints.

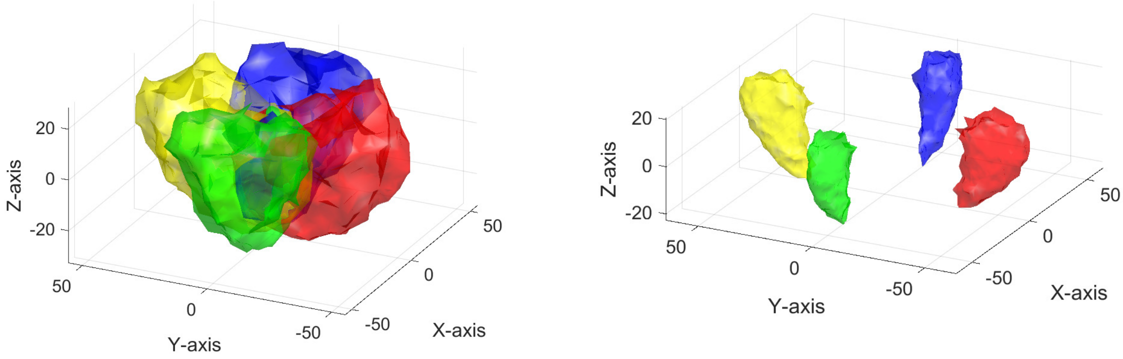


Figure 4.1: The rescaled workspaces of all four feet in ground frame for the optimized robot (left) and unoptimized robot (right) using hull surface around points.

4.1 Interpretation

The found parameters seem reasonable, with long legs and passive links, and short motor links to be able to reach a large workspace with small force amplifications (Figure 1.2). Compared to the prototype (Figure 1.1), the passive links were extended further to improve the reach. In the initial guess, the variable α was set so that the motor link and the leg are parallel when all active joint angles are set to zero. This design choice was rejected, as the optimization found a value of 1.5608, which lets the motor link almost move inside the xy-plane of the base, as shown in Figure 4.2. One reason might be that it attempts to increase the distance of the instantaneous center of rotation of the lower spatial four-bar linkage to the upper motor link. This might improve the force propagation because the lever arm is larger in some configurations compared to a geometry with a lower value of α. If the lever arm is zero, a RO singularity occurs, which the algorithm avoids. Variable b_0 represents the length between the upper leg links and the top link. Due to the assumptions of the kinematic model (section 3.2.1), it is set to zero in the optimized geometry. Increasing this length might also improve the force propagation through the top link (Figure 4.2). Concerning the shape

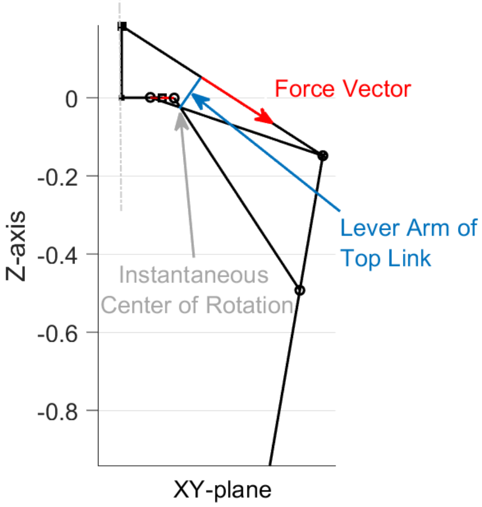


Figure 4.2: Visualization of the lever arm of the top link with respect to the instantaneous center of rotation represented in the base's coordinate frame

of the workspace, the robot mostly gains mobility toward the center and in the y-direction of the ground-related frame. The z-direction is not improved by much. This results in a more uniform shape compared to the tall and slim workspace of the prototype, which should be beneficial for achieving a variety of walking patterns.

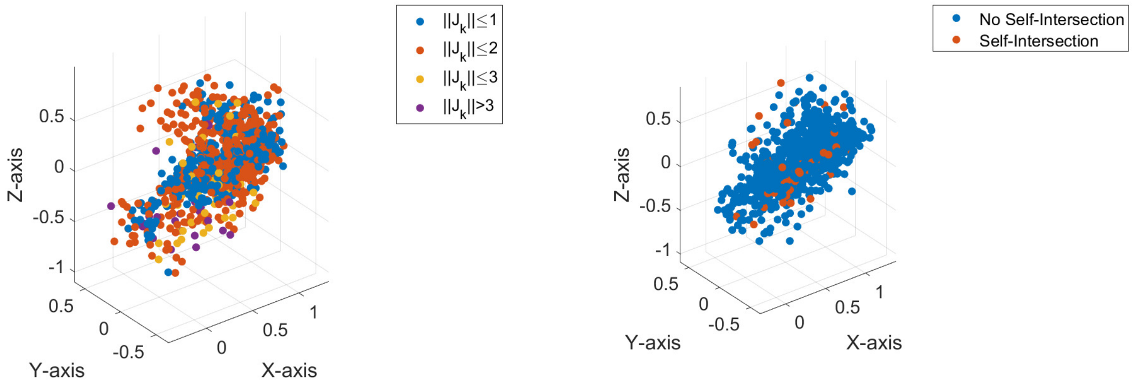


Figure 4.3: Visualization of 1000 sampling points in the ground-related workspace, color-coded by the distance from singularity (left) and whether a sample is unreachable due to self-intersection (right).

When looking beyond the boundary of one for the maximum singular value of the kinematic Jacobian (Figure 4.3), the sampling finds many configurations with singular values in the range of 1-3 inside the workspace and even some above three, as shown in Table 4.2. This is possible since the ground-related workspace does not fix the stance feet. Multiple configurations can correspond to the same voxel. Some of these will exceed the bound. The knowledge, which corresponds to the singular values compliant with the constraints, is lost by the reduction from six to three-dimensional space. For each voxel inside the ground-related workspace, the definition only guarantees that any configuration exists that fulfills the constraints. Nevertheless, as the upper bound of the singular values was only a rough estimation, many other configurations will still be useful, as the motors will likely have enough force to maneuver the robot. Also, the vertical bound set to avoid self-intersections of the legs (p_F) can be violated for specific active joint angle sets. These are not included in the workspace volume estimation and potentially further increase the reachable volume. All sets of active joint angles are valid by metric p_{VI} and around 2% of the samples cannot be reached due to self-intersection of the leg links.

	Percentage of Samples
Compliant with all constraints	54%
Violating p_{VI}	0%
Violating p_F	1.6%
Violating p_{RO}	45%
Violating p_{RO} by more than a Factor of 3	0.7%
Unreachable due to self-intersection	2%

Table 4.2: The percentages of samples, which fulfill and violate certain constraints.

In conclusion, the volume and shape of the optimized robot's workspace are significantly improved compared to the initial guess. The algorithm has found a good optimum. Although further improvements cannot be disproved, it is assumed to be the global optimum. The robot can reach additional configurations that do not fulfill the constraints, which likely allows for an ever larger motion range.

4.2 Discussion

The optimization found a sensible solution after converging at generation 96. It achieved the set goals, as shown in section 4. Still, the accuracy and parameter sensitivity need to be discussed to evaluate the optimality of the result. Further improvements regarding the accuracy and other aspects of the implementation will be proposed in this section. As only 2.4% of the discretized space was reached, a smaller volume could have been chosen. The spherical boundaries defined by the maximum reach were still a valid assumption and guaranteed that none of the explored solutions reached a bound. Since the memory requirement is not a limiting factor, this boundary definition should not have hurt the performance. The small fluctuation in the best fitness (Figure 4.4) likely stems from the noise due to the sampling. This makes a slightly increased fitness value possible in later generations despite retraining the elite. The Relative Standard Deviation (RSD) of the optimal parameter set due to sampling and discretization errors is shown in Figure 4.4 (right). It was evaluated by recalculating the best solution, shown in table 4.1 30 times, and shows the variability in the WVI. The median workspace, displayed as a red line in the box plot, is around -0.0223.

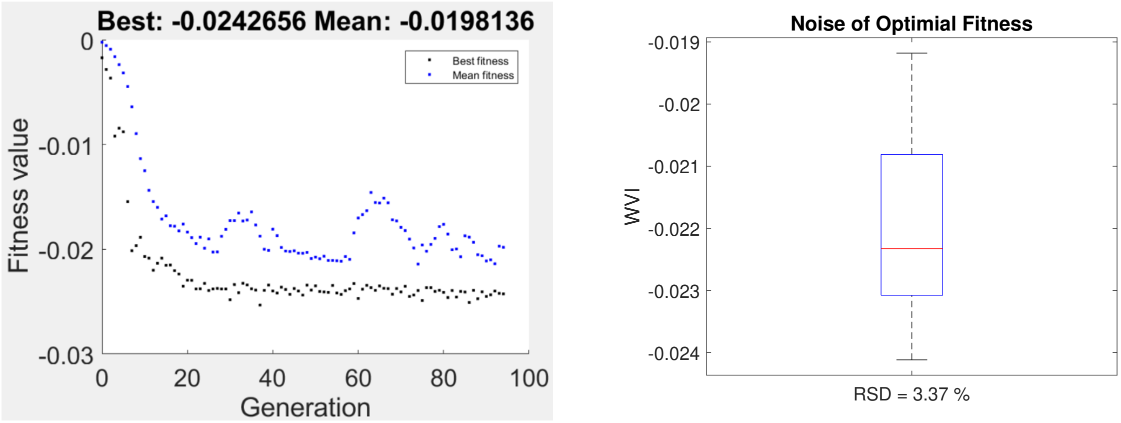
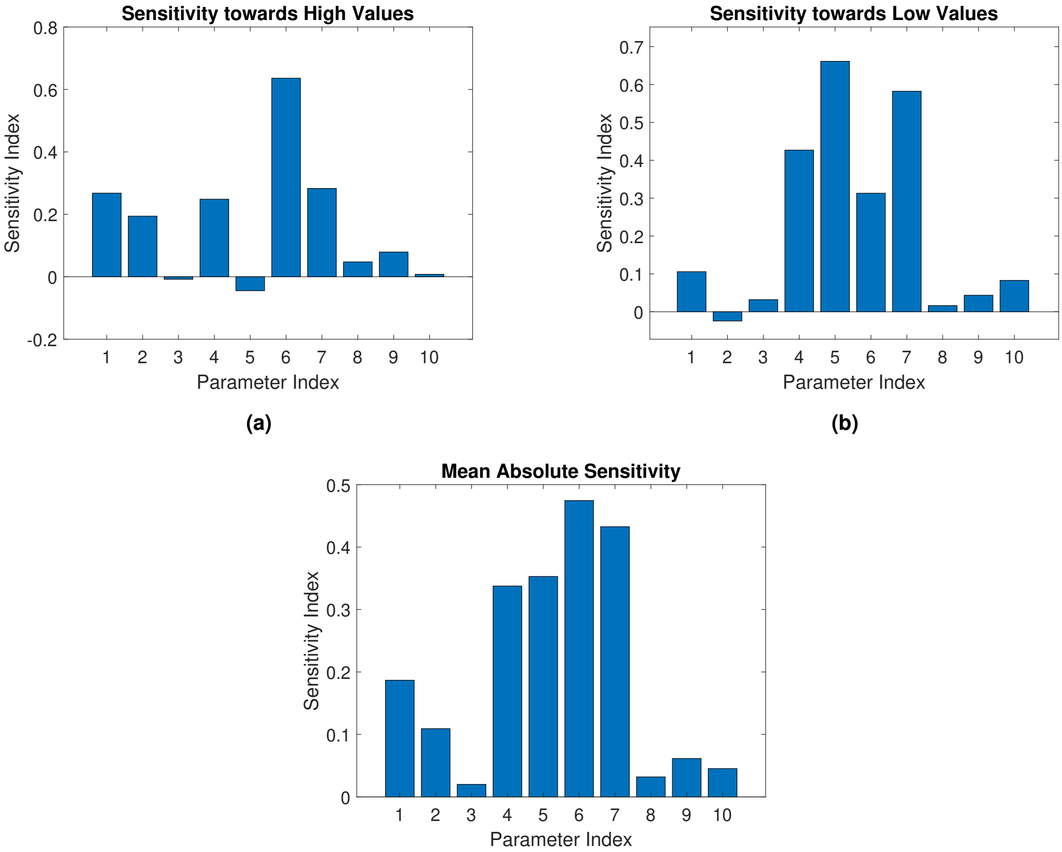


Figure 4.4: The best and average value of the GA for each generation with 100 individuals each (left) and the noise of the optimal solution regarding variations of the WVI (right).

Despite the small diversity of 0.0614, there are still some outliers in the last generation, suggesting that the design space is sensitive in some dimensions. A sensitivity analysis was performed to study the effects of a deviation from the optimal values. It is derived using the following equation, were I_s is the sensitivity index, $WVI_{p,i}$ the workspace index when perturbating the i-th parameter, p the perturbation factor, and k_i^* the optimal parameter value:

$$I_{s,i} = \frac{WVI^* - WVI_{p,i}}{p * k_i^*} \tag{4.1}$$

The sensitivity index in Figure 4.5 (a) shows how much the volume increases relative to a perturbation towards high values. Figure 4.5 (b) shows the effect when perturbating towards low values. The mean of the absolute values of both are shown in (c). An index of 0.5 indicates that a perturbation of 1% leads to an increase in workspace volume of 0.5%, assuming a linear relation for small perturbations. Since the WVI is optimized, it is expected that all sensitivity indices are positive, indicating that changing the parameter decreases the workspace volume. The fact that this is not the case for all parameters is likely a consequence of the noise, which might have led to a slightly suboptimal solution or influenced the sensitivity analysis. The latter is addressed by averaging over five samples and using a perturbation factor of 3%. The analysis reveals that the mean absolute sensitivity of q_1 and q_2, as well as l_1 and l_2 are comparatively sensitive to variations, while α, l_3, b_1 and b_2 are not. The outliers present in the last generation likely have small variations in some of the sensitive dimensions of the design space, which results in the RSD of 29% in the last generation despite the small diversity of 6% in the parameter space. When building the robot, the lengths of the sensitive parameters must be adjusted precisely. If necessary in construction, small deviations should only be applied to the less sensitive parameters.



(a) **(b)**

(c)

Figure 4.5: Sensitivity analysis showing the effects on the WVI when perturbing parameters towards smaller values (a), larger values (b) ,and the average effect (c).

Due to the chosen methods, the noise, and the sensitivity of optimization, the result can not guarantee a global optimum, but it nevertheless greatly improves the usable workspace of the robot's legs. Due to self-intersections, a slightly suboptimal kinematic might be necessary for the physical robot. Additionally, the workspace is not the only metric to rely on when trying to assess the walking capabilities of the robot. Further studies must be performed to ensure the robot is stable while walking. These considerations might require reviewing the geometry to specifically enhance the feasible walking patterns by considering the center of mass in the objective function. Still, the improved robot design can be utilized to further explore the coupled geometry and its implications for walking.

Chapter 5

Conclusion and Outlook

The goal of this work was to analyze and optimize the geometry of a highly coupled quadruped robot with six actuated DoF. Several steps were performed to achieve the optimal kinematic. First, an analytic solution to the forward kinematics had to be found, which relied on the concept of virtual four-bar linkages, expanding on a concept used in multi-link suspension systems in cars. All positions that can be reached while fulfilling several constraints sum up the workspace, which is improved by varying ten geometrical parameters of the robot. These constraints include the avoidance of self-intersection and the distance from singularities. A special workspace definition had to be considered to ensure the coupled nature of the robot is grasped by the optimization. It is inspired by the tetrahedron. Its six side lengths can characterize the position of the four corners. These coincide with the position of the robot's feet. The novel workspace definition through distances between the feet allows for capturing their dependencies.

The deployed methods have proven effective in significantly improving the robot's movement capabilities. However, several areas for future research and potential enhancements have been identified. The non-Euclidian foot distance manifold heuristically offered a better-shaped workspace compared to the ground-related method. The exact effects of the inherent lengths bias have not been studied sufficiently yet. Also, the shape could be influenced more precisely by defining a desired shape and position, which the optimization tries to match as done by Boudreau and Gallant (BOUDREAU and GALLANT 2002).
Also, exact calculations of the boundaries posed by self-intersections might allow for a larger usable workspace. Bonev and Gosselin provided an approach using branch set theory (BONEV and GOSSELIN 2002).
The index p_{RO} evaluating the distance from the parallel singularities could benefit from a better understanding of the load cases (RATOLIKAR and R 2020). Additionally, other performance indices addressing stiffness, dexterity, or joint forces could be included for further insights.
The noise from the discretization method may be reduced by increasing the window size, as done by Porges (PORGES 2013). To further reduce the noise and potentially improve the efficiency of the workspace calculation a numerical method like the boundary search method could be utilized, instead of the discretization approach (MERLET 2006).
Concerning the solver, the use of surrogate models or hybrid approaches might allow for improved performance.
More capable hardware, like GPU clusters, could improve the accuracy of the results. Also, more tailored optimization and symbolic software like Maple (MAPLE 2024) may improve the results.
The next steps involve building the physical robot. Here, the motor placements and non-zero

thickness of the links might cause additional collisions, which need to be considered.
Finally, the findings indicate that a small footprint benefits the workspace size. Stable walking might require a wider stance, though. Conditions for a desired height-to-width ratio of the robot might be a way to address the stability concerns. Additionally, since the position of the center of mass is coupled with the leg positions, due to the reduced DoF, conditions on the location of the center of mass in the x- and y-directions might be necessary for statically stable walking. Assessing feasible gait dynamics will require knowledge of the center of mass and may demand reiterating the design if no stable patterns exist.

With the found results, this research sets a foundation for further advancements in the design and optimization of quadruped robots, paving the way for more efficient and capable robotic systems.

Appendix A

Analytic Solution to the Forward Kinematics

$$X_1 = \begin{pmatrix} \frac{k_2}{2} + \sigma_{54} - \frac{\sigma_{59}}{2} - \frac{(b_1+b_2)\left(q_2\,\sin(\Phi_r) - \sigma_{59} + \sigma_{25} - \frac{\sigma_2\,\sigma_{26}\,\sigma_4}{\sigma_{17}\,\sigma_3} + \sigma_{24} - \frac{\sigma_1\,(\sigma_{56}-\sigma_{55})\,\sigma_{27}\,\sigma_4}{\sigma_3\,\sigma_{12}}\right)}{b_1} + \sigma_{25} + \sigma_{24} \\ \frac{k_2}{2} - \sigma_{53} - \frac{\sigma_{60}}{2} + \frac{(b_1+b_2)\left(q_2\,\sin(\Phi_l) + \sigma_{60} + \sigma_{23} + \frac{\sigma_2\,\sigma_{27}\,\sigma_4}{\sigma_{17}\,\sigma_3} - \sigma_{22} - \frac{\sigma_1\,(\sigma_{56}-\sigma_{55})\,\sigma_{26}\,\sigma_4}{\sigma_3\,\sigma_{12}}\right)}{b_1} - \sigma_{23} + \sigma_{22} \\ -\frac{(b_1+b_2)\left(\sigma_{56} + \sigma_{55} + \sigma_{21} - \frac{\sigma_{20}\,\sigma_1\,\sigma_4}{\sigma_3\,\sigma_{12}}\right)}{b_1} + \frac{\sigma_{56}}{2} + \frac{\sigma_{55}}{2} + \sigma_{21} \end{pmatrix}$$

where

$$\sigma_1 = \sigma_5\,\sigma_8 + \frac{\sigma_7\,\sigma_6}{2\,\sigma_4}$$

$$\sigma_2 = \sigma_5\,\sigma_7 - \frac{\sigma_8\,\sigma_6}{2\,\sigma_4}$$

$$\sigma_3 = {\sigma_7}^2 + {\sigma_8}^2$$

$$\sigma_4 = \sqrt{{l_3}^2 - \sigma_{10} - \sigma_{11} - \frac{\sigma_{47}}{4}}$$

$$\sigma_5 = \sqrt{{\sigma_7}^2 + \frac{{\sigma_6}^2}{4\left(-{l_3}^2 + \sigma_{10} + \sigma_{11} + \frac{\sigma_{47}}{4}\right)} + {\sigma_8}^2}$$

$$\sigma_6 = {\sigma_7}^2 - \sigma_{10} - \sigma_{11} - \frac{\sigma_{47}}{4} - {b_1}^2 + {l_3}^2 + {\sigma_8}^2 + \\ \left(\frac{\left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)\sigma_{16}}{\sigma_9} + \frac{\left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)\sigma_{15}}{\sigma_9} - \frac{\sigma_{40}\,\sigma_{13}}{\sigma_9} - \frac{\left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)\sigma_{18}}{\sigma_9} + \frac{\left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)\sigma_{19}}{\sigma_9} - \frac{\sigma_{40}\,\sigma_{41}}{\sigma_9}\right)^2$$

$$\sigma_7 = \frac{\sigma_{14}\,\sigma_{41}}{\sigma_{12}} + \frac{\sigma_{14}\,\sigma_{13}}{\sigma_{12}} - \frac{\sigma_{40}\left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)\sigma_{18}}{\sigma_{12}} + \frac{\sigma_{40}\left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)\sigma_{19}}{\sigma_{12}} + \frac{\sigma_{40}\left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)\sigma_{16}}{\sigma_{12}} + \frac{\sigma_{40}\left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)\sigma_{15}}{\sigma_{12}}$$

$$\sigma_8 = \frac{\left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)\sigma_{19}}{\sigma_{17}} + \frac{\left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)\sigma_{18}}{\sigma_{17}} + \frac{\left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)\sigma_{15}}{\sigma_{17}} - \frac{\left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)\sigma_{16}}{\sigma_{17}}$$

$$\sigma_9 = \sqrt{{\left|\sigma_{27}\right|}^2 + {|\sigma_{26}|}^2 + \sigma_{47}}$$

$$\sigma_{10} = \frac{{\left|\sigma_{27}\right|}^2}{4}$$

$$\sigma_{11} = \frac{{|\sigma_{26}|}^2}{4}$$

$$\sigma_{12} = \sqrt{\left|(\sigma_{56} - \sigma_{55})\,\sigma_{27}\right|^2 + |(\sigma_{56} - \sigma_{55})\,\sigma_{26}|^2 + |\sigma_{20}|^2}$$

$$\sigma_{13} = \frac{\sigma_{56}}{2} + \frac{\sigma_{55}}{2} + \sigma_{21}$$

$$\sigma_{14} = \left(\overline{k_2} + \sigma_{51} + \sigma_{50}\right)^2 + \left(\overline{k_2} - \sigma_{52} + \sigma_{49}\right)^2$$

$$\sigma_{15} = \sigma_{53} - \frac{k_2}{2} + \frac{\sigma_{60}}{2} + \sigma_{23} - \sigma_{22}$$

$$\sigma_{16} = \frac{k_2}{2} + \sigma_{54} - \frac{\sigma_{59}}{2} + \sigma_{25} + \sigma_{24}$$

$$\sigma_{17} = \sqrt{\left|\sigma_{27}\right|^2 + |\sigma_{26}|^2}$$

$$\sigma_{18} = \frac{k_2}{2} - \sigma_{54} + \frac{\sigma_{59}}{2}$$

$$\sigma_{19} = \frac{k_2}{2} + \sigma_{53} + \frac{\sigma_{60}}{2}$$

$$\sigma_{20} = {\sigma_{27}}^2 + {\sigma_{26}}^2$$

$$\sigma_{21} = \frac{\sigma_{28}\,\sigma_{48}\,\sigma_{31}}{\sigma_{30}\,\sigma_{39}}$$

$$\sigma_{22} = \frac{\sigma_{28}\,(\sigma_{56} - \sigma_{55})\,\sigma_{58}\,\sigma_{31}}{\sigma_{30}\,\sigma_{39}}$$

$$\sigma_{23} = \frac{\sigma_{29}\,\sigma_{57}\,\sigma_{31}}{\sigma_{43}\,\sigma_{30}}$$

$$\sigma_{24} = \frac{\sigma_{28}\,(\sigma_{56} - \sigma_{55})\,\sigma_{57}\,\sigma_{31}}{\sigma_{30}\,\sigma_{39}}$$

$$\sigma_{25} = \frac{\sigma_{29}\,\sigma_{58}\,\sigma_{31}}{\sigma_{43}\,\sigma_{30}}$$

$$\sigma_{26} = k_2 - q_2\,\sin(\Phi_l) + \sigma_{60}$$

$$\sigma_{27} = k_2 + q_2\,\sin(\Phi_r) + \sigma_{59}$$

$$\sigma_{28} = \sigma_{32}\,\sigma_{35} + \frac{\sigma_{34}\,\sigma_{33}}{2\,\sigma_{31}}$$

$$\sigma_{29} = \sigma_{32}\,\sigma_{34} - \frac{\sigma_{35}\,\sigma_{33}}{2\,\sigma_{31}}$$

$$\sigma_{30} = {\sigma_{35}}^2 + {\sigma_{34}}^2$$

$$\sigma_{31} = \sqrt{{l_2}^2 - \sigma_{37} - \frac{\sigma_{47}}{4} - \sigma_{38}}$$

$$\sigma_{32} = \sqrt{{\sigma_{35}}^2 + {\sigma_{34}}^2 + \frac{{\sigma_{33}}^2}{4\left(-{l_2}^2 + \sigma_{37} + \frac{\sigma_{47}}{4} + \sigma_{38}\right)}}$$

$$\sigma_{33} = \left(\frac{\left(\overline{k_2} + \sigma_{52} - \sigma_{49}\right)\sigma_{44}}{\sigma_{36}} + \frac{\sigma_{40}\,\sigma_{41}}{\sigma_{36}} - \frac{\sigma_{45}\left(\sigma_{51} - \overline{k_2} + \sigma_{50}\right)}{\sigma_{36}} - \frac{k_1\,\sigma_{40}}{\sigma_{36}} + \frac{q_1\,\sin\left(\sigma_{46}\right)\left(\overline{k_2} + \sigma_{52} - \sigma_{49}\right)}{\sigma_{36}} + \frac{q_1\,\cos\left(\sigma_{46}\right)\left(\sigma_{51} - \overline{k_2} + \sigma_{50}\right)}{\sigma_{36}}\right)^2 + {\sigma_{35}}^2 + {\sigma_{34}}^2 - \sigma_{37} - \frac{\sigma_{47}}{4} - {l_1}^2 + {l_2}^2 - \sigma_{38}$$

$$\sigma_{34} = \frac{k_1\,\sigma_{42}}{\sigma_{39}} - \frac{\sigma_{42}\,\sigma_{41}}{\sigma_{39}} + \frac{\sigma_{40}\left(\overline{k_2} + \sigma_{52} - \sigma_{49}\right)\sigma_{44}}{\sigma_{39}} - \frac{\sigma_{40}\,\sigma_{45}\left(\sigma_{51} - \overline{k_2} + \sigma_{50}\right)}{\sigma_{39}} + \frac{q_1\,\sin\left(\sigma_{46}\right)\sigma_{40}\left(\overline{k_2} + \sigma_{52} - \sigma_{49}\right)}{\sigma_{39}} + \frac{q_1\,\cos\left(\sigma_{46}\right)\sigma_{40}\left(\sigma_{51} - \overline{k_2} + \sigma_{50}\right)}{\sigma_{39}}$$

$$\sigma_{35} = \frac{\left(\overline{k_2}+\sigma_{52}-\sigma_{49}\right)\sigma_{45}}{\sigma_{43}} + \frac{\sigma_{44}\left(\sigma_{51}-\overline{k_2}+\sigma_{50}\right)}{\sigma_{43}} + \frac{q_1\,\sin\left(\sigma_{46}\right)\left(\sigma_{51}-\overline{k_2}+\sigma_{50}\right)}{\sigma_{43}} - \frac{q_1\,\cos\left(\sigma_{46}\right)\left(\overline{k_2}+\sigma_{52}-\sigma_{49}\right)}{\sigma_{43}}$$

$$\sigma_{36} = \sqrt{|\sigma_{58}|^2 + \sigma_{47} + \left|\sigma_{57}\right|^2}$$

$$\sigma_{37} = \frac{|\sigma_{58}|^2}{4}$$

$$\sigma_{38} = \frac{\left|\sigma_{57}\right|^2}{4}$$

$$\sigma_{39} = \sqrt{\left|\sigma_{48}\right|^2 + |(\sigma_{56}-\sigma_{55})\,\sigma_{58}|^2 + \left|(\sigma_{56}-\sigma_{55})\,\sigma_{57}\right|^2}$$

$$\sigma_{40} = \cos\left(\overline{\Phi_l}\right)\,\cos\left(\overline{\alpha}\right)\,\overline{q_2} - \cos\left(\overline{\Phi_r}\right)\,\cos\left(\overline{\alpha}\right)\,\overline{q_2}$$

$$\sigma_{41} = \frac{\sigma_{56}}{2} + \frac{\sigma_{55}}{2}$$

$$\sigma_{42} = \left(\overline{k_2} + \sigma_{52} - \sigma_{49}\right)^2 + \left(\sigma_{51} - \overline{k_2} + \sigma_{50}\right)^2$$

$$\sigma_{43} = \sqrt{|\sigma_{58}|^2 + \left|\sigma_{57}\right|^2}$$

$$\sigma_{44} = \sigma_{53} - \frac{k_2}{2} + \frac{\sigma_{60}}{2}$$

$$\sigma_{45} = \frac{k_2}{2} + \sigma_{54} - \frac{\sigma_{59}}{2}$$

$$\sigma_{46} = \Phi_t + \frac{3\,\pi}{4}$$

$$\sigma_{47} = |\sigma_{56} - \sigma_{55}|^2$$

$$\sigma_{48} = {\sigma_{58}}^2 + {\sigma_{57}}^2$$

$$\sigma_{49} = \cos\left(\overline{\Phi_r}\right)\,\sin\left(\overline{\alpha}\right)\,\overline{q_2}$$

$$\sigma_{50} = \cos\left(\overline{\Phi_l}\right)\,\sin\left(\overline{\alpha}\right)\,\overline{q_2}$$

$$\sigma_{51} = \sin\left(\overline{\Phi_r}\right)\,\overline{q_2}$$

$$\sigma_{52} = \sin\left(\overline{\Phi_l}\right)\,\overline{q_2}$$

$$\sigma_{53} = \frac{q_2\,\sin(\Phi_l)}{2}$$

$$\sigma_{54} = \frac{q_2\,\sin(\Phi_r)}{2}$$

$$\sigma_{55} = q_2\,\cos\left(\Phi_r\right)\,\cos\left(\alpha\right)$$

$$\sigma_{56} = q_2\,\cos\left(\Phi_l\right)\,\cos\left(\alpha\right)$$

$$\sigma_{57} = q_2\,\sin\left(\Phi_r\right) - k_2 + \sigma_{59}$$

$$\sigma_{58} = k_2 + q_2\,\sin\left(\Phi_l\right) - \sigma_{60}$$

$$\sigma_{59} = q_2\,\cos\left(\Phi_l\right)\,\sin\left(\alpha\right)$$

$$\sigma_{60} = q_2\,\cos\left(\Phi_r\right)\,\sin\left(\alpha\right)$$

Appendix B

Foot Distance Manifold

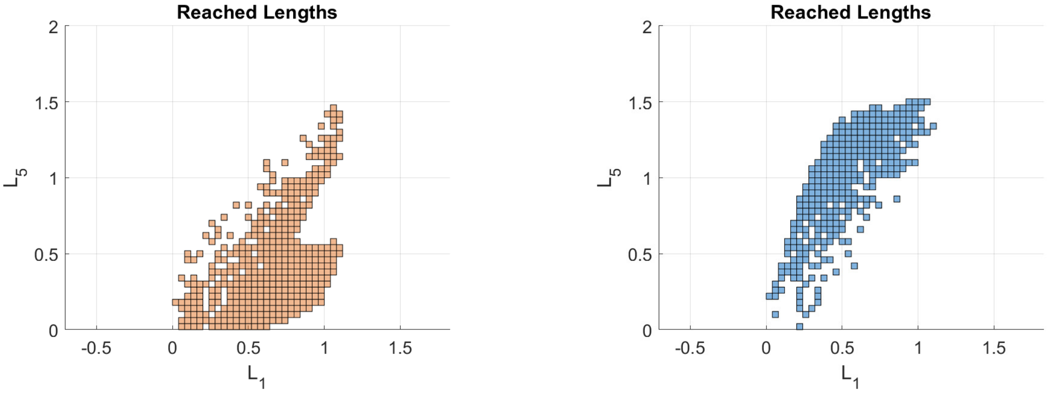


Figure B.1: The projection of coordinate charts of the distance space, showing the relation of lengths L1 ad L5 for the upper (left) and lower (right) assembly mode.

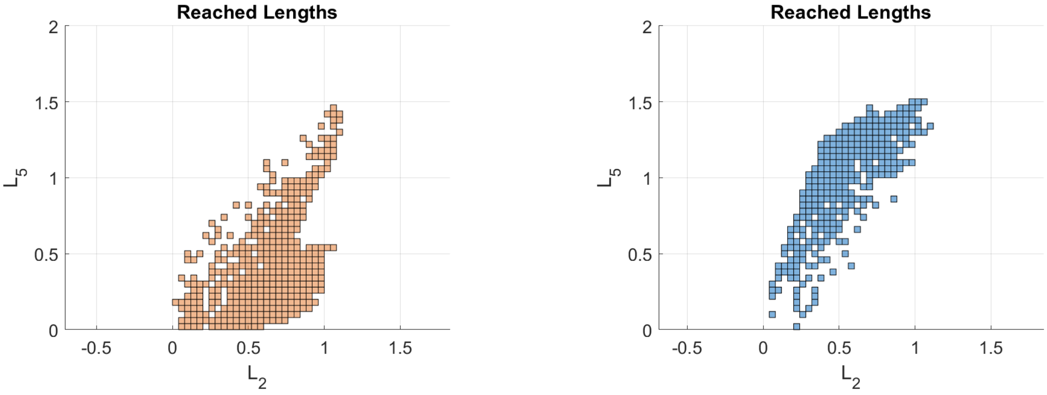


Figure B.2: A projection of coordinate charts of the distance space, showing the relation of lengths L2 ad L5 for the upper (left) and lower (right) assembly mode.

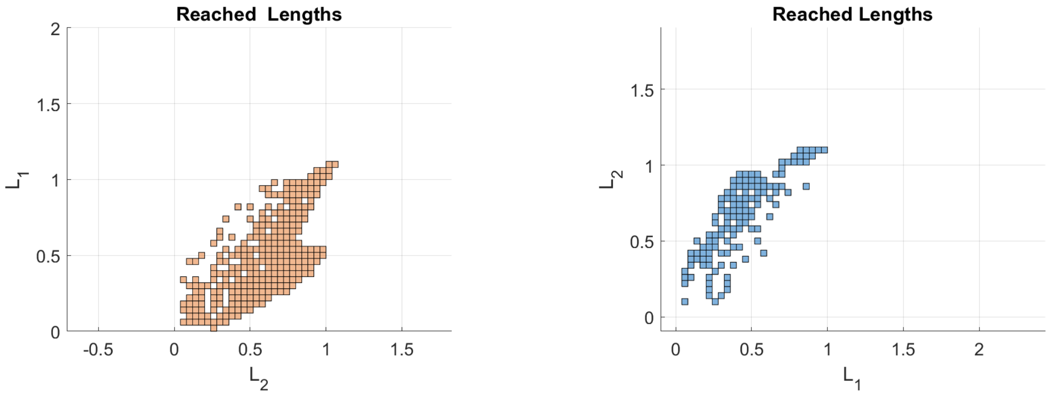


Figure B.3: A projection of coordinate charts of the distance space, showing the relation of lengths L1 ad L2 for the upper (left) and lower (right) assembly mode.

List of Figures

1.1 The unoptimized prototype in top view (left) and isometric view (right). The geometry is not optimized, and the upper motors do not retain rotational symmetry. 2

1.2 The structure of the robot (left). One of the four leg assemblies is marked in blue. The structure of a leg assembly (right). The joints are drawn as circles, crosses, and squares, and the links as lines. The scale is non-dimensionalized and shows the relative proportions. 3

2.1 The four main parts of the work, where the spatial four-bar linkage is part of the kinematics, which are a sub-function of the workspace calculation - a sub-function of the optimization algorithm. 5

2.2 The spatial four-bar linkage as defined by Chyi (DAVID PERNG CHYI 1969), with the unknown angle θ at the origin O. The angle δ is given. 6

2.3 Visualization of the constrained foot workspace. The coordinate system is defined by the three feet on the ground, relative to which the position of the fourth foot is examined. The stance feet are at a fixed location relative to the ground. 8

2.4 Visualization of the ground-related workspace. The coordinate system is defined by the three feet on the ground, relative to which the position of the fourth foot is examined. The stance feet are not fixed but can slide freely along the ground. 9

2.5 Visualization of the distance workspace. The three distances marked in blue make up the three-dimensional workspace used for the optimization. 9

2.6 Visualization of the height bias. The distance space (dotted lines) does not reach a new length state when moved in the z-direction, while the Euclidean space does. 9

2.7 Visualization of an optimized workspace using the ground-related frame (left) and distance space (right). They are both shown looking from the origin of the ground-related frame, to see the difference in width. 10

2.8 The general structure of the GA. After creating an initial population it is ranked and split into elite, crossover and mutation children, which make up the next generation (MATHWORKS 2022b). 14

2.9 The three groups of children - elite, crossover and mutation - and their basic concept, as implemented in Matlab (MATHWORKS 2022b). 15

3.1 Visualization of a leg assembly. The base frame is depicted in orange, the leg assembly in black, besides the active links being blue. Two additional frames are introduced in light blue, used to describe the motion of two four-bar linkages. 18

3.2 Push-Rod system (left) and Pull-Rod system (middle) in the side view of a Formula 1 car. On the right is the isometric view of the Push-Rod system, showing the six links that constrain the motion of the wheel. 1-4 correspond to the leg links, 5 is the steering rod, not present in the robot, and 6 is the push-rod, equivalent to the top link of each leg assembly (TRZESNIOWSKI 2019). . 19

3.3 A point restricted to two spheres has to move along the circular intersection. 19

3.4 The four different assembly modes of a leg assembly, where (a) is the used configuration. 21

3.5 The three possible paths from the base's origin to the end effector, where path 3 (c) is used for determining the maximum reach. It is used to calculate R_{norm}, shown in (d). 25

3.6 Shown is the volume W_d, restricted only by the xy-plane (left), leading to an undesired solution with crossed legs and the restriction to a quarter of the sphere, which is used in the optimization (right). 26

3.7 The two possible assembly modes of the distance space, which can be differentiated by the sign of angle ξ. 29

3.8 The general structure of the optimization algorithm. The workspace calculation is done by sampling in the joint space. After the sampling is terminated, the design parameters are varied, and the workspace calculation is repeated until the GA terminates. 30

3.9 The parameters of the leg assembly (left) and the base (right) showing how the geometry can be reduced to ten parameters. 31

4.1 The rescaled workspaces of all four feet in ground frame for the optimized robot (left) and unoptimized robot (right) using hull surface around points. . 36

4.2 Visualization of the lever arm of the top link with respect to the instantaneous center of rotation represented in the base's coordinate frame 36

4.3 Visualization of 1000 sampling points in the ground-related workspace, color-coded by the distance from singularity (left) and whether a sample is unreachable due to self-intersection (right). 37

4.4 The best and average value of the GA for each generation with 100 individuals each (left) and the noise of the optimal solution regarding variations of the WVI (right). 38

4.5 Sensitivity analysis showing the effects on the WVI when perturbing parameters towards smaller values (a), larger values (b) ,and the average effect (c). . 39

B.1 The projection of coordinate charts of the distance space, showing the relation of lengths L1 ad L5 for the upper (left) and lower (right) assembly mode. . . 47

B.2 A projection of coordinate charts of the distance space, showing the relation of lengths L2 ad L5 for the upper (left) and lower (right) assembly mode. . . 47

B.3 A projection of coordinate charts of the distance space, showing the relation of lengths L1 ad L2 for the upper (left) and lower (right) assembly mode. . . 48

List of Tables

2.1 Advantages and disadvantages of the IK and FK using analytical and numerical calculation methods. 7

2.2 Advantages and disadvantages of the analytical and numerical calculation methods for the kinematic Jacobian. 7

2.3 Advantages and disadvantages of the constrained foot workspace W_c, the ground-related workspace W_g, and the distance workspace W_d. 8

2.4 Advantages and disadvantages of the analytic, numerical, and discretization method. 10

2.5 Different solver categories that can handle constrained problems with discrete nonlinear noisy objective functions, including deterministic and stochastic as well as sampling-base, 0-th, and 1-st order solvers. 13

2.6 Advantages and disadvantages of the PSO, SA, and GA. 13

2.7 The values and function names of the GA concerning the initialization, reproduction, and break condition 14

3.1 Primary, secondary, and disregarded goals of this work. 17

3.2 Table showing which motor angles are used for which leg assembly. 23

3.3 Hardware specifications including the processor, its speed, the number of parallel workers, the memory, and the used version of Matlab. 28

3.4 The boundaries of the design parameters and the active joint angle limits used in the optimization. 32

4.1 The values of the optimized parameters, both non-dimensionalized and rescaled and the prototype's parameters values as well as the corresponding sizes of the ground-related workspace. 35

4.2 The percentages of samples, which fulfill and violate certain constraints. 37

Bibliography

ABOULISSANE, B., HAIEK, D. E., BAKKALI, L. E., and BAHAOUI, J. E., (2019). “On The Workspace Optimization of Parallel Robots Based on CAD Approach”. In: *Procedia Manufacturing* 32, pp. 1085–1092. ISSN: 23519789. DOI: 10.1016/j.promfg.2019.03.065.

AGRAWAL, S. K., (1991). “Workspace boundaries of in-parallel manipulator systems”. In: *Robots in unstructured environments*. New York N.Y. and Piscataway NJ: IEEE and Available from IEEE Service Center, 1147–1152 vol.2. ISBN: 0-7803-0078-5. DOI: 10.1109/ICAR.1991.240401.

BADESCU, M., MORMAN, J., and MAVROIDIS, C., (2002). “Workspace Optimization of Orientational 3-Legged UPS Parallel Platforms”. In: *Computers and Information in Engineering Conference*.

BENJAMIN KATZ, JARED DI CARLO, and SANGBAE KIM, (2019). *2019 International Conference on Robotics and Automation (ICRA)*. Piscataway, NJ: IEEE. ISBN: 9781538660270. URL: https://ieeexplore.ieee.org/servlet/opac?punumber=8780387.

BIRGLEN, L. and RUELLA, C., (2014). “Analysis and Optimization of One-Degree of Freedom Robotic Legs”. In: *Journal of Mechanisms and Robotics* 6.4. ISSN: 1942-4302. DOI: 10.1115/1.4027234.

BOANTA, C. and BRIȘAN, C., (2022). “Estimation of the Kinematics and Workspace of a Robot Using Artificial Neural Networks”. In: *Sensors (Basel, Switzerland)* 22.21. DOI: 10.3390/s22218356.

BOHIGAS, O., MANUBENS, M., and ROS, L., (2012). “A Complete Method for Workspace Boundary Determination on General Structure Manipulators”. In: *IEEE Transactions on Robotics* 28.5, pp. 993–1006. ISSN: 1552-3098. DOI: 10.1109/TRO.2012.2196311.

BONEV, I. A. and GOSSELIN, C. M., (2002). “Geometric Algorithms for the Computation of the Constant-Orientation Workspace and Singularity Surfaces of a Special 6-RUS Parallel Manipulator”. In: *Volume 5: 27th Biennial Mechanisms and Robotics Conference*. American Society of Mechanical Engineers, pp. 505–514. ISBN: 0-7918-3653-3. DOI: 10.1115/DETC2002/MECH-34257.

BOUDREAU, R. and GALLANT, M., (2002). “The Synthesis of Planar Parallel Manipulators with Prismatic Joints for an Optimal, Singularity-Free Workspace”. In: *Journal of Robotic Systems*.

DAVID PERNG CHYI, (1969). “The analysis of R-S-S-R spatial four-bar mechanism”. Master Thesis. Missouri: University of Missouri.

DENG, Z., HOU, X., and WANG, S., (2021). “Workspace Optimization of a 6-RSS Stewart-Gough Robotic Platform to Assist Ultrasound Diagnosis”. In: *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*. IEEE, pp. 230–233. ISBN: 978-1-6654-3337-2. DOI: 10.1109/DTPI52967.2021.9540100.

EIBEN, A. E. and SMITH, J. E., (2015). *Introduction to Evolutionary Computing*. 2nd ed. 2015. Natural Computing Series. Berlin Heidelberg: Springer Berlin Heidelberg and Imprint: Springer. ISBN: 9783662448731. DOI: 10.1007/978-3-662-44874-8.

ERDÖS, P. and RENYI, A., (1961). "On a Classical Problem of Probability Theory". In: *Americal Mathematical Society* 27.794, pp. 215–220.

FEDOROV, D. and BIRGLEN, L., (2016). "Analysis and Design of a Two Degree of Freedom Hoeckens-Pantograph Leg Mechanism". In: *Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference – 2015*. New York N.Y.: American Society of Mechanical Engineers. ISBN: 978-0-7918-5713-7. DOI: 10.1115/DETC2015-47330.

GOSSELIN, C. and ANGELES, J., (1990). "Singularity analysis of closed-loop kinematic chains". In: *IEEE Transactions on Robotics and Automation* 6.3, pp. 281–290. ISSN: 1042296X. DOI: 10.1109/70.56660.

GÜLHAN, M. M. and ERBATUR, K., (2018). "Kinematic arrangement optimization of a quadruped robot with genetic algorithms". In: *Measurement and Control* 51.9-10, pp. 406–416. ISSN: 0020-2940. DOI: 10.1177/0020294018795640.

HAMLIN, G. J. and SANDERSON, A. C., (1994). "A novel concentric multilink spherical joint with parallel robotics applications". In: *The institute of electrical and electronics engineers 1994 – 1994 IEEE International Conference*, pp. 1267–1272. DOI: 10.1109/ROBOT.1994.351313.

HAO, Q., WANG, Z., WANG, J., and CHEN, G., (2020). "Stability-Guaranteed and High Terrain Adaptability Static Gait for Quadruped Robots". In: *Sensors (Basel, Switzerland)* 20.17. DOI: 10.3390/s20174911.

JONG, K. A. de, (1975). "An Analysis of the Behavior of a Class of Genetic Adaptive Systems". PhD. Michigan: University of Michigan.

KANEKO, M., ABE, M., and TACHI, S., (1986). "Basic considerations of the degrees of freedom of multi-legged locomotion machines". In: *Advanced Robotics* 1.2, pp. 101–116. ISSN: 0169-1864. DOI: 10.1163/156855386X00021.

KING, B., (2022). "Kinematic Design and Optimization of a Quadruped Robot with Six Actuated DoF". Master Thesis. Dortmund: Technische Universität.

KOCHENDERFER, M. J. and WHEELER, T. A., (2018). *Algorithms for Optimization*. Stanford: The MIT Press.

LI, Q., YANG, C., XU, L., and YE, W., (2023). *Performance Analysis and Optimization of Parallel Manipulators*. Singapore: Springer Nature. ISBN: 978-981-99-0541-6. DOI: 10.1007/978-981-99-0542-3.

LI, T. and YE, P., (2003). "The measurement of kinematic accuracy for various configurations of parallel manipulators". In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*. IEEE, pp. 1122–1129. ISBN: 0-7803-7952-7. DOI: 10.1109/ICSMC.2003.1244562.

LIN, J., (1978). *Blue Bird Technology*. https://www.blue-bird-model.com/products/17_54.htm. Accessed: 2024-06-03.

MAPLE, (2024). *Maple Inc.* Version 2024. Maplesoft, a division of Waterloo Maple Inc. Waterloo, Ontario, Canada. URL: https://www.maplesoft.com/products/Maple/.

MATHWORKS, (2022a). *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States. URL: https://www.mathworks.com.

— (2022b). *Optimization Toolbox version: 9.4 (R2022b)*. Natick, Massachusetts, United States. URL: https://www.mathworks.com.

MERLET, J., (2006). *Parallel robots*. 2nd ed. Vol. 74. Solid mechanics and its applications. Dordrecht and Boston MA: Kluwer Academic Publishers. ISBN: 1402041322.

MÜLLER, A. and ZLATANOV, D., (2019). *Singular Configurations of Mechanisms and Manipulators*. Vol. 589. Cham: Springer International Publishing. ISBN: 978-3-030-05218-8. DOI: 10.1007/978-3-030-05219-5.

NASA, (2022). “Multi-Link Spherical Joint”. 11635107.

Porges, O., (2013). “Analysis and applications of reachability and capability maps for robotic manipulators”. Master Thesis. Würzburg: Julius Maximilian University of Würzburg.

Ratolikar, D. M. and R, P. K., (2020). “Optimal 5R parallel leg design for quadruped robot gait cycle”. In: *Vibroengineering Procedia* 35, pp. 94–98. issn: 2345-0533. doi: 10.21595/vp.2020.21806.

Ren, H., Li, Q., Liu, B., and Dou, Z., (2018). “Design and optimization of an elastic linkage quadruped robot based on workspace and tracking error”. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 232.22, pp. 4152–4166. issn: 0954-4062. doi: 10.1177/0954406217750186.

Rosyid, A., El-Khasawneh, B., and Alazzam, A., (2017). “Kinematic Performance Measures and Optimization of Parallel Kinematics Manipulators: A Brief Review”. In: *Kinematics*. Ed. by Hurtado, E. G. InTech. isbn: 978-953-51-3687-3. doi: 10.5772/intechopen.71406.

Seidel, D., Hermann, M., Gumpert, T., Loeffl, C. F., and Albu-Schäffer, A., (2020). *Using Elastically Actuated Legged Robots in Rough Terrain: Experiments with DLR Quadruped bert: Yellowstone Conference Center, Big Sky, Montana, March 7-14, 2020*. Piscataway, NJ: IEEE. isbn: 9781728127347. doi: 10.1109/AERO47225.2020. url: https://ieeexplore.ieee.org/servlet/opac?punumber=9161765.

Silva, V. G., Tavakoli, M., and Marques, L., (2014a). “Dexterity Optimization of a Three Degrees of Freedom DELTA Parallel Manipulator”. In: *ROBOT2013: First Iberian Robotics Conference*. Ed. by Armada, M. A., Sanfeliu, A., and Ferre, M. Vol. 253. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, pp. 719–726. isbn: 978-3-319-03652-6. doi: 10.1007/978-3-319-03653-3{\textunderscore}51.

— (2014b). “Dexterity Optimization of a Three Degrees of Freedom DELTA Parallel Manipulator”. In: *ROBOT2013: First Iberian Robotics Conference*. Ed. by Armada, M. A., Sanfeliu, A., and Ferre, M. Vol. 253. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing, pp. 719–726. isbn: 978-3-319-03652-6. doi: 10.1007/978-3-319-03653-3{\textunderscore}51.

Trzesniowski, M., (2019). *Fahrwerk*. Wiesbaden: Springer Fachmedien Wiesbaden. isbn: 978-3-658-26699-8. doi: 10.1007/978-3-658-26700-1.

Tu, L. W., (2011). *An Introduction to Manifolds*. New York, NY: Springer New York. isbn: 978-1-4419-7399-3. doi: 10.1007/978-1-4419-7400-6.

Yuan, X., Meng, Q., Xie, F., Liu, X.-J., and Wang, J., (2023). “Error modeling and accuracy evaluation of parallel manipulators with mixed DoFs based on motion/force transmissibility and constrainability”. In: *Mechanism and Machine Theory* 186, p. 105346. issn: 0094114X. doi: 10.1016/j.mechmachtheory.2023.105346.

Zsombor-Murray, P. J. and Gfrerrer, A., (2010). “A Unified Approach to Direct Kinematics of Some Reduced Motion Parallel Manipulators”. In: *Journal of Mechanisms and Robotics* 2.2. issn: 1942-4302. doi: 10.1115/1.4001095.

Disclaimer

I hereby declare that this thesis is entirely the result of my own work except where otherwise indicated. I have only used the resources given in the list of references.

Garching, July 19, 2024

(Signature)