technische universität dortmund

Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

Kinematic Design and Optimization of a Quadruped Robot with Six Actuated DoF

Master Thesis

submitted to

Institute of Control Theory and Systems Engineering

Faculty of Electrical Engineering and Information Technology

Technische Universität Dortmund

by

Bryce King

Date of Submission: May 30, 2022

DLR Supervisor:
Dipl.-Ing. Florian Loeffl

University Supervisors:
Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram
apl. Prof. Dr. rer. nat. Frank Hoffmann

Eidesstattliche Versicherung

(Affidavit)

King, Bryce

Name, Vorname
(surname, first name)

222200

Matrikelnummer
(student ID number)

☐ Bachelorarbeit
(Bachelor's thesis)

■ Masterarbeit
(Master's thesis)

Titel
(Title)

Kinematic Design and Optimization of a Quadruped Robot with Six Actuated DoF

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem oben genannten Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present thesis with the above-mentioned title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution before.

Bamberg, 30.05.2022

Ort, Datum
(place, date)

Unterschrift
(signature)

Belehrung:
Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin") zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:
Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to EUR 50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the Chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, Section 63 (5) North Rhine-Westphalia Higher Education Act (*Hochschulgesetz, HG*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund University will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:*

Bamberg, 30.05.2022

Ort, Datum
(place, date)

Unterschrift
(signature)

***Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**

Acknowledgement

I would like to give a big thank you to Florian Loeffl for all the time he dedicated to assisting me in my work, answering my many questions, and patiently helping me through any problem that arose. It was truly a pleasure to work together and see this project come to fruition.

A huge thanks also to Prof. Frank Hoffmann and Prof. Torsten Bertram for agreeing to this collaboration with DLR, and especially to Frank Hoffmann for the all the valuable input provided through our periodic meetings.

To my fellow students and friends at both TU Dortmund and in the RMC, thank you for helping to make my time here very pleasant despite the annoying effects the pandemic had on the study and work experience.

To my parents, thank you for always encouraging me to do what makes me happy, no matter how far away it takes me. Without you I wouldn't be where I am today.

Finally, a huge thanks to my wonderful girlfriend Mireia, who has been my rock and has always supported me emotionally through this stressful time.

Abstract

While legged robots hold many advantages over wheeled robots, especially regarding dynamic capabilities and mobility, they often suffer from lower energy efficiency and reliability due to the use of more actuators. Many classic approaches to quadruped design rely on designs with 12 actuated degrees of freedom - three in each leg - which allows the feet to be freely placed with respect to the body. One obvious solution to this problem is a reduction in the number of actuators, but this usually comes at the cost of functionality reduction. In terms of simple locomotion, however, the robot's center of mass can be sufficiently moved by freely placing each foot with respect to the world, thus changing the robot's contact points. It should then be possible to achieve locomotion with free foot placement using only six actuated degrees of freedom if the existing functions are appropriately combined and/or reduced. The aim of this thesis is therefore to design a full quadruped kinematic structure with only six actuators which is capable of simple locomotion through free placement of its feet. A review of existing literature regarding legged locomotion and reduced-DoF quadrupeds is performed to form a basis for new concepts, and a novel kinematic structure is proposed which relies on two types of leg couplings to reduce the degrees of freedom. A kinematic analysis then provides representations of the model in terms of forward, inverse and differential kinematics, and a control algorithm with position error feedback is proposed for task-space trajectory following. The proposed model is implemented in Creo Parametric and simulated with the help of the LucaDynamics library in MATLAB. A few tests are performed in the simulated environment which show that the proposed robot is indeed capable of stable static walking with free placement of all four feet, with the task-space position errors remaining very low for all tested trajectories and no indications of singular or near-singular poses.

Contents

List of Tables

List of Figures

Nomenclature

Abbreviations and acronyms

CoM	Center of Mass
DH	Denavit-Hartenberg
DoF	Degree of Freedom
w.r.t	With respect to

Symbols

$\dot{\mathbf{q}}$	Joint velocity vector
$\mathbf{A}(\mathbf{q})$	Homogeneous transformation matrix
$\mathbf{I}_n$	$n \times n$ identity matrix
$\mathbf{J}(\mathbf{q})$	Jacobian matrix
$\mathbf{p}_k^i$	Position of frame k with respect to frame i
$\mathbf{q}$	Joint state vector
$\mathbf{R}_k^i$	Rotation of frame k with respect to frame i
$\mathbf{T}_k^i(\mathbf{q})$	Forward kinematic transformation from frame i to k

1

Introduction

Recent developments in mobile robotics have led to an increasing interest in their real-world applications, ranging from factory logistics to planetary exploration. While a majority of traditional research in the field focused on locomotion using wheels or tracks, the topic of legged locomotion has gained significant traction in recent years due to its advantages in terms of mobility and dynamic performance in unstructured environments. Quadrupeds in particular are an area of high interest due to their ability to statically balance and perform a range of different static and dynamic locomotion gaits. Nevertheless, they generally suffer from lower energy efficiency compared to their wheeled counterparts, a problem which is compounded by their use of more actuators to achieve their highly dynamic capabilities. From a locomotion standpoint, the number of actuated degrees of freedom used in many modern quadrupeds exceeds what is necessary for simple movement of the center of mass, with many utilizing twelve motors. Essentially, quadrupeds with a reduced number of actuators should still be capable of basic locomotion, albeit with fewer dynamic capabilities. Aside from savings in weight and energy, a reduction in complexity is also a common approach for increasing reliability in legged robots, since the addition of redundancies is clearly counterproductive. This has led to a wide range of research aiming to answer the many questions regarding design of efficient, reliable and robust legged robots which can be practically implemented in real-world scenarios.

Overall, the field of legged robotics is still relatively young, with some of the first large-scale research being performed by Marc Raibert and his team at MIT's Leg Laboratory in the 1980s. Their work formed a scientific basis for the design and analysis of legged machines which are capable of balancing, walking, jumping and running, and has since inspired an ever-growing interest in dynamic legged robot locomotion. In Raibert's book *Legged Robots That Balance*, which outlined the results of this research, two serious reasons for studying legged locomotion were presented. The first was mobility, namely the ability to navigate difficult terrains where existing wheeled vehicles cannot go, and the second was the decoupling of body path from feet paths, which allows a payload attached to the body to traverse a smooth path despite variations in the terrain (Raibert 1986). These two reasons remain important today, with many of the proposed use cases for legged robots involving environments with unstructured or unknown terrains, for example in planetary exploration and terrestrial disaster recovery.

Many researchers have further explored the principles of legged locomotion, with a particular focus on quadrupeds. Shigeo Hirose developed a series of quadruped robots at the Tokyo Institute of Technology, know as TITAN, which were used to explore various principles of quadruped locomotion and make developments such as fusion gait control for stable motion (Yoneda and Hirose 1992) and locomotion on steep slopes and staircases (Hirose et al. 1991; Hirose, Yoneda, and Tsukagoshi 1997). Hirose and his colleagues also proposed some higher efficiency leg actuation techniques such as the gravitationally decoupled actuator (Hirose 1984) and the coupled drive (Hirose and Sato 1989), and defined more efficient walking postures for quadrupeds (Arikawa and Hirose 2007).

Most of the quadrupeds developed over the years, especially those in more recent times, have been based on variations of a kinematic structure which consists of 12 fully-actuated degrees of freedom, and allows for a wide range of dynamic capabilities. The standard configuration for such robots is biologically inspired, with each leg consisting of a joint for hip abduction/adduction (HAA), a joint for hip flexion/extension (HFE) and a joint for knee flexion/extension (KFE), giving each foot the capability to move freely in three dimensional space relative to the main body. Quadrupeds with such "dog-like" designs, which can be referred to as *classical quadrupeds*, have been used prolifically in the past decade to research various techniques of improving energy efficiency and robustness, especially through the introduction of series elasticities. One example of such robots is StarlETH, designed at the Swiss Federal Institute of Technology to study "fast, efficient and versatile motion" (Hutter et al. 2012). The design features torque-controlled highly compliant series elastic actuators in which the springs decouple the motors from the joints, providing robustness against impact forces and energy storage for improved efficiency and making the system suitable for highly dynamic movements. Similar features can be seen in ANYmal, the successor to StarlETH, with higher joint mobility being its biggest advantage (Hutter et al. 2016), and in the MIT Cheetah 3, which features high-bandwidth proprioceptive actuators to allow it to handle and react to unexpected disturbances "without the need for external sensors or prior environment knowledge" (Bledt et al. 2018). At the Instituto Italiano di Tecnologia (IIT), a series of hydraulically actuated quadrupeds was designed using the same classical quadruped morphology, showing progressive improvements in robustness and efficiency through optimization of parameters in the hydraulic actuators (Semini et al. 2017).

Although these types of fully-actuated 12 degree of freedom quadrupeds have many advantages in terms of dynamic capabilities and mobility, limitations in power storage technology mean that the usage of many heavy actuators can be quite impractical. Research on the topic of reduced-DoF quadrupeds has produced a wide variety of interesting contributions and results, with many different structures being proposed over the years. A first important consideration when exploring such reductions is the obvious impact on system mobility and function. Kaneko, Abe, and Tachi (1986) defined some basic quadruped walking functions ranging from level 1 (one-dimensional walking on flat planes) to level 5 (free selection of foot placement within the movable space) and investigated the number of degrees of freedom necessary to achieve these different levels, showing that the highest level can theoretically be attained with a

minimum of six degrees of freedom. Yoneda et al. (2001) defined similar but simpler function levels, leaving out the body leveling function since it is not necessary to achieve walking. They proposed a quadruped design featuring only four actuated degrees of freedom which is capable of function level 4 behavior (omnidirectional body movement with arbitrary swing leg selection and placement), as well as a 3-active DoF mechanism capable of level 2 walking (movement along a curved path). The number of degrees of freedom necessary to achieve various different quadruped functions was further investigated by Yoneda (2007) along with various lower-energy actuation techniques, which were demonstrated in the design of a minimal 3-actuator quadruped, an omnidirectional 5-actuator model and a 9-actuator model capable of dynamic walking and trotting.

While much of this research has aimed to improve energy efficiency through either system compliance or kinematic reductions, some has combined the two strategies by investigating reduced-DoF models with elastic actuation. The quadruped BERT from the German Aerospace Center (DLR) is one example where elastic actuation is utilized in a reduced-DoF kinematic model to investigate walking and balancing on uneven terrains (Seidel et al. 2020). The quadruped consists of only 8 actuated DoF, with the structure resembling the standard 12 DoF model but with the hip adduction/abduction joints removed, leaving each leg only free to move within the saggital plane. As with all other reduced-DoF models, this reduction restricts the mobility of the robot, making changes in the body's yaw angle impossible without slipping of the feet.

Although many solutions have been proposed to the questions of quadruped efficiency, dynamic mobility and complexity, there remains room for more contributions in the form of novel kinematic structures which reduce the degrees of freedom of quadrupeds while maintaining useful functionalities. The work of this thesis therefore focuses on the design of a kinematic structure for a reduced degree of freedom quadruped crawler which utilizes only six actuators to attain free foot placement in three dimensions. With these six actuators, the robot is able to fully define the positions of its feet and body to achieve a static walking gait, which is verified with a detailed kinematic analysis and a handful of simulated walking tests.

2

System Design and Analysis

In this chapter, the kinematic design for a novel 6-actuated DoF quadruped is presented with a discussion of the background and underlying concepts which inspired the different components. An analysis of the forward and inverse kinematics as well as the differential kinematics is then performed in order to derive a suitable inverse kinematic algorithm for controlling the robot in simulation and testing its ability to achieve the desired goals.

2.1. Goals & Requirements

As with any system design, an important step in the development of a novel kinematic structure for a reduced degree of freedom quadruped is the formalization of the goals and requirements that should be met to consider the solution a success. Specifically, the work presented in this thesis is based on one primary goal, subject to a few additional goals and requirements, which are given as follows:

- **Degrees of Freedom** - The primary design goal relates to the relationship between the number of actuated degrees of freedom and the total number of available degrees of freedom in the system during certain motion tasks. Unlike some reduced-DoF quadrupeds which have their foot motions restricted to a particular plane or hyperplane, the new quadruped has the goal of being capable of free placement of each of its feet relative to the ground in 3-dimensional space, subject to defined joint limitations. Looking back to the classical quadruped design approach, this capability is easily achieved with its 12 actuated degrees of freedom, however some important insight comes from analyzing the total available degrees of freedom in a given pose. This can be computed by the difference between the number of system degrees of freedom and the number of constraints introduced on the system. Feet which make contact with the ground are commonly represented as virtual ball joints with assumed no-slip conditions, which introduces three constraints on the positional degrees of freedom of each contact foot. Assuming a single foot is lifted from the ground to perform a step, the available degrees of freedom for the classical quadruped can be computed as follows:

$$12\ \textit{Actuated DoF} + 6\ \textit{Floating Base DoF} - 3\ \textit{DoF} \times 3\ \textit{Contact Points} = 9\ \textit{DoF}$$

The presence of 9 available degrees of freedom shows that the classical quadruped can not only achieve the 3 DoF motion in its lifted foot, but also has 6 additional degrees of freedom which are commonly associated with the floating base, i.e. three positional and three orientational degrees of freedom. In other words, a lifted foot can be arbitrarily placed relative to the body while the additional degrees of freedom are exploited to change the position and orientation of the floating base body. However, since fully defining a foot's position only requires three degrees of freedom, any additional degrees of freedom can be viewed as redundant and unnecessary for the task of foot placement, therefore the number of required actuated degrees of freedom for a system without the redundant degrees of freedom can be computed as follows:

$$3\ \textit{DoF} + 3\ \textit{DoF} \times 3\ \textit{Contact Points} - 6\ \textit{Floating Base DoF} = 6\ \textit{Actuated DoF}$$

The goal therefore is to design a reduced kinematic structure for a quadruped which consists of only six actuated degrees of freedom that are able to fully define the system's motions and achieve free placement of each foot relative to the ground.

- **Locomotion capabilities** - The ability to lift and freely place a single foot is of course an important aspect to getting a quadruped moving, but successful locomotion requires the robot to follow a defined gait, or pattern of feet placement, which shifts the center of mass relative to the world in a desired direction. One of the most simple defined gaits is the static walking or crawling gait, which involves moving only one foot at a time, maintaining at least three ground contact points at all times. In order to maintain static stability, the center of mass must always remain within the support polygon formed by the three ground contact points. If one defines locomotion as the repeated changing of the robot's ground contact points to shift the center of mass from one position to another, the proposed quadruped should be capable of achieving at least a static walking gait with its six actuated degrees of freedom.

- **Representation of Feet** - The four feet should be represented as points with no area, meaning that at any given time, each foot can only make a single contact with the ground, regardless of leg orientation. The point-like foot representation plays an important role in modeling of the system's constraints, since contacts with the ground can be expressed as a restriction of the three translational degrees of freedom of the corresponding foot point, with the orientational degrees of freedom left unconstrained. This allows for the virtual ball-joint representation, which is a common approach used in virtually all notable quadruped designs.

2.2. Design Concept

Keeping the system goals and requirements in mind, several ideas were explored which ultimately led to a formalized design concept for a new quadruped kinematic

structure. The proposed design concept features classic pantograph legs, with their parallel geometry exploited to obtain good range of motion in the feet using smaller motions at the driving joints. Two types of novel coupling mechanisms are introduced, one between the adjacent legs and the other between opposite legs, which are based on the *common drive* and *coupled drive* concepts and serve to reduce the required number of actuators while preserving the desired motion capabilities. This section provides a look into each of the concepts on which the proposed design is based as well as the procedural process utilized to combine the different concepts and thus reduce the kinematic structure to the desired number of actuated degrees of freedom.

2.2.1. Pantograph Leg

In order to come up with ideas for a new quadruped which is capable of achieving the stated goals, it was useful to first analyze and take inspiration from existing solutions to similar problems. One feature which has appeared in several legged robots is the pantograph, a mechanical structure which consists of four rigid linkages connected by pin joints to form a parallelogram. Kinematically the pantograph behaves like a simple two-bar linkage with two planar degrees of freedom, but the addition of a parallel branch facilitates flexibility in how the actuation of the two degrees of freedom can be achieved. In the classic two-bar linkage example, each joint is typically directly actuated, which can introduce a large amount of inertia due to the mass of the motor mounted at the second joint. In contrast, the two degrees of freedom in the pantograph structure can be easily controlled by placing both motors at the base and directly driving the two parallel branches, making the mechanism generally more energy efficient, a highly appealing feature for legged robots. Another interesting feature provided by the pantograph mechanism is best seen in its original use as a copying device for drawings in the 17th century (Stewart 1965). As shown in Figure 2.1, if the point A represents a rotational joint about a fixed body, then the linear motions at point F directly imitate the linear motions at point D, with a certain degree of amplification. A small range of motion at point D could therefore be used for driving the point F with a much larger range of motion.

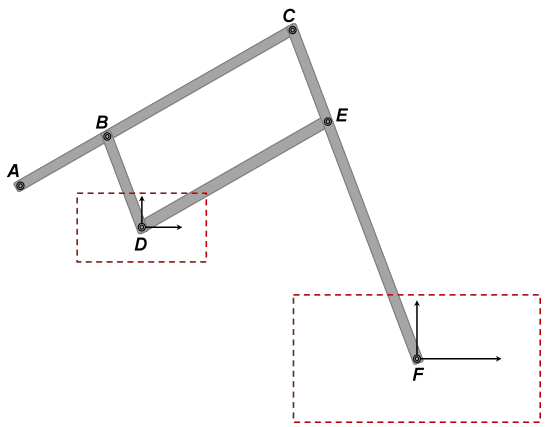


Figure 2.1.: A pantograph, where motions at point D are amplified into motions at point F

Although the classic pantograph is a planar mechanism, the addition of rotation about a vertical axis allows the pantograph mechanism's mobility to be transferred from planar motion to three-dimensional motion, as originally introduced by Hirose (1984) and shown in Figure 2.2. A leg designed using this structure is therefore capable of free placement of its foot point in space as well as a typically large range of motion, both of which are desired in legged locomotion.

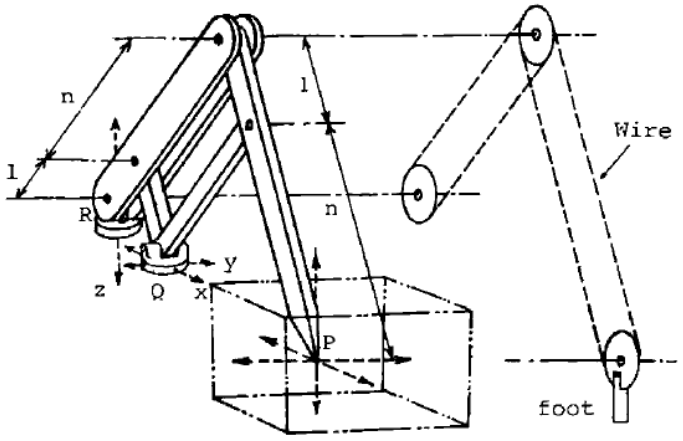


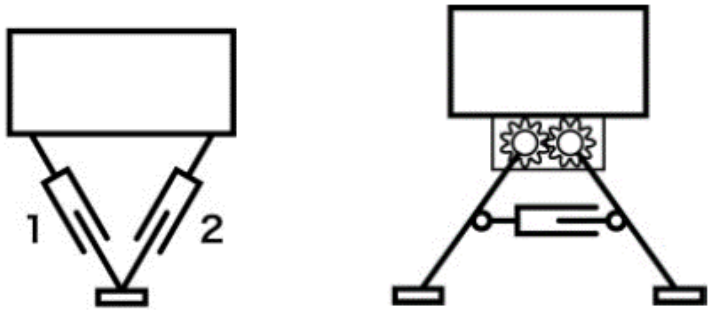
Figure 2.2.: The three-dimensional Cartesian motion PANTOMEC, introduced by Hirose (1984). Vertical motions at **R** result in vertical motions of **P**, while horizontal motions at **Q** result in horizontal motions at **P**.

2.2.2. Kinematic Couplings

Many approaches to quadruped design feature legs which are kinematically independent from one another, whereby each leg requires its own independent actuators to control its motion relative to the robot body. Referring again to the classical quadruped model, each of the four legs features three independently actuated degrees of freedom, giving the entire system 12 actuated degrees of freedom. In order to achieve the goal of independent foot motion in three degrees of freedom using only six actuators, it is clear that each of the legs cannot be driven by independent actuators, but rather must rely on some form of kinematic couplings, allowing the actuated degrees of freedom to be shared between the legs. The design concept proposed in this thesis is based primarily on the *coupled drive* and *common drive* actuator arrangements, which were discussed in detail in terms of their application in quadrupeds by Yoneda (2007). A brief look into each of these concepts is provided here with discussion of their usefulness in the proposed design.

Coupled Drive

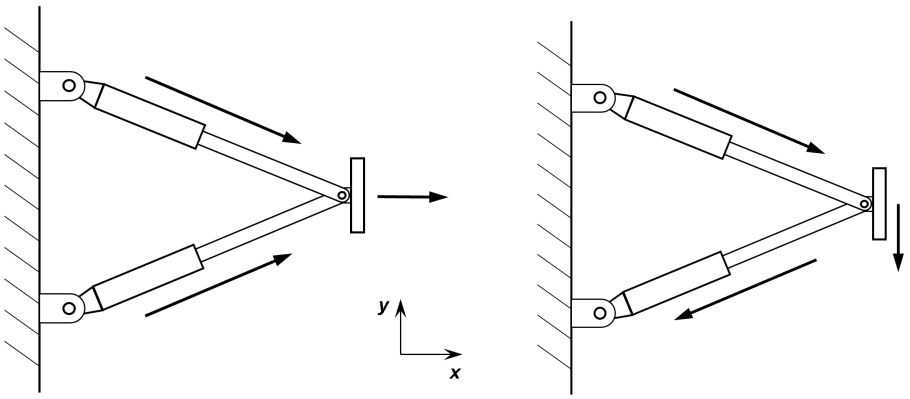
The coupled drive actuator arrangement, shown in Fig. 2.3a, is a concept which was originally proposed by Hirose and Sato (1989) as a method for reducing the total weight of multi-DoF walking robots. The concept involves a single point on a mechanism being driven jointly by two different actuators such that the total force requirements



(a) Coupled drive (b) Common drive

Figure 2.3.: Actuator arrangements introduced by Yoneda (2007)

of the individual actuators is reduced. This allows for smaller, lighter weight actuators to be used for producing the same total forces on the point being driven. The method was implemented, for example, by Hirose et al. (2009) in a wall climbing robot in order to increase its *actuation index*, a measure of how much of the total installed actuator power is available for use for a particular motion. Besides its proposed use in legged robots, this actuator arrangement can also be seen in the widely-known Stewart-Gough platform, where three pairs of coupled linear drives control the six degrees of freedom of a rigid platform (Stewart 1965). An interesting feature of this actuation concept can be seen by analyzing the simple planar case with two linear actuators. If both linear actuators are driven in sync with one another, the end effector point moves linearly along the x-axis as seen in Fig. 2.4a. On the other hand, a differential motion between the actuators causes the end effector point to move along the y-axis, shown in Fig. 2.4b. It is therefore clear that this sort of actuator arrangement is capable of producing the same two translational degrees of freedom at the end effector as a simple planar two-bar linkage, but using linear actuation as opposed to rotational, which allows for the possibility of higher force output and better energy efficiency due to lower inertia at the end effector.

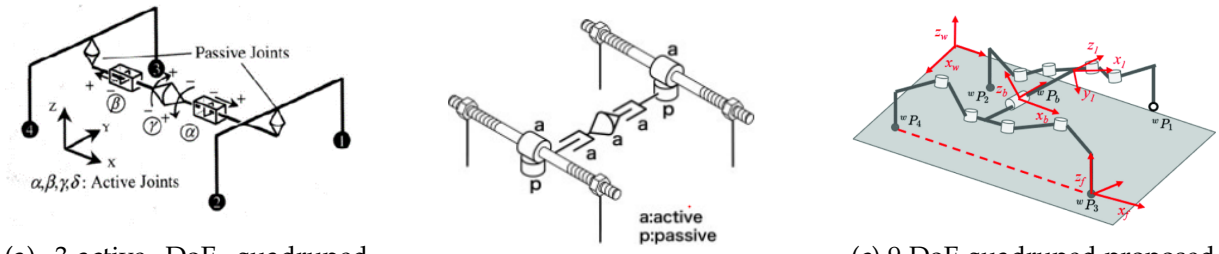


(a) Uniform motion (b) Differential motion

Figure 2.4.: Motion behavior of a planar coupled drive mechanism

Common Drive

In classical quadrupeds, the motion of each leg during swing stage is controlled using an independent set of actuators which are decoupled from those of the grounded legs. In this case, even the non-swing legs have their own available degrees of freedom which can be exploited to manipulate the position and orientation of the floating base body. In a static walking locomotion task, only the swing leg requires actuated degrees of freedom, namely three, to achieve free motion in the world, therefore the actuated degrees of freedom in the non-swing legs can be seen as redundancies. This idea inspired the so-called *common drive* concept, shown in Fig. 2.3b and discussed in detail by Yoneda et al. (2001). Instead of producing the motions of a swing leg relative to the main body, the motions can instead be produced relative to one or more of the non-swing legs by means of kinematic coupling between the respective legs. This allows for a reduction in the total number of required actuators, but usually at the expense of some functionality. A few examples of proposed quadrupeds featuring the common drive technique can be seen in Fig. 2.5, where a common approach between them is the placement of an active joint in the center body that produces a differential twisting motion between the front and back leg pairs. This single active degree of freedom is capable of producing a lifting motion for any of the four legs, with the swing leg selection being dictated by the location of the robot's center of mass. In kinematic designs such as these, the ground contacts play a critical role in defining the motions of a swing leg relative to the world, and removal of one or more of the contacts can result in an under-constrained system which is no longer capable of the desired functionalities.



(a) 3-active DoF quadruped proposed by Yoneda et al. (2001)

(b) 5-active DoF quadruped proposed by Yoneda (2007)

(c) 9 DoF quadruped proposed by Zhang, Shen, and Hong (2020)

Figure 2.5.: Examples of quadruped designs featuring common drives for DoF reduction [1]

2.2.3. Proposed Design Concept

The proposed design concept is based largely on a combination of the previously discussed components into a unique structure for which the defined system goals and requirements from Chapter 2.1 are met. The four legs are modeled as identical pantograph structures to allow free placement of the feet, and two types of leg couplings

[1]Enlarged figures are provided in Appendix A

are used to integrate a form of the common drive concept for achieving the desired motions with a reduced number of actuators.
To aid in the explanation of the features and functionalities of the pantograph leg mechanism, it is first convenient to provide some naming conventions for some of its parts. Referring to Fig. 2.6, names are given to each of the links in the leg, based roughly on biological leg part names (i.e. thigh & shank). Continuing the biological-based naming convention, the joint at the base of the thigh can be called the hip and the joint connecting the thigh and shank can be called the knee. Finally, the joint connecting the thighlink and shanklink is named the driving joint, since it can be used to directly control the foot motion using the aforementioned principle.

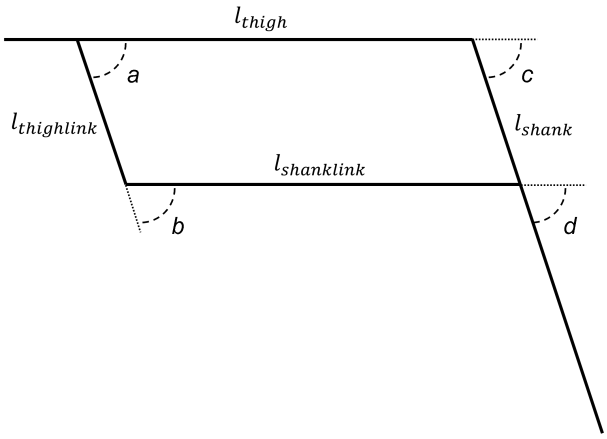


Figure 2.6.: A simple parallelogram seen in the pantograph mechanism

System Constraints

Throughout the design process, the number of available degrees of freedom, defined by Eq. 2.2.1, had to remain an important consideration since the number of actuated degrees of freedom should always be greater or equal to the number of available degrees of freedom to prevent having an under-constrained system. This of course requires explicit knowledge of the constraints which are present in a kinematic structure and is of particular importance when dealing with a mixture of active and passive joints, since all the passive joints should be constrained to the active ones to have a fully-defined system.

$$\text{System DoF} - \text{System Constraints} = \text{Available DoF} \tag{2.2.1}$$

In simpler kinematic structures, the constraints can often be easily recognized with some basic knowledge of geometry. Take for example the pantograph mechanism, which is used for the quadruped's legs. Knowing that a parallelogram consists of four joint angles which are always equal, as seen in Fig. 2.6, the constraint equations are simply defined by $a = b = c = d$. Since there are a total of four 1-DoF joints in the parallelogram, the number of available degrees of freedom can be calculated as 4 system DoF $-$ 3 constraints $=$ 1 DoF, meaning that arbitrary selection of one of the parallelogram joints as an active joint would fully define the system. Unfortunately

in more complex systems, not all constraints are so easily recognized based on the geometry, which means a more general approach to defining certain types of constraints can prove useful. Specifically, the constraints that appear in the mechanisms discussed here are introduced through closed-chains. An approach for representing these constraints in kinematic terms is presented in detail in Chapter 2.3, so for now it is assumed that the number of constraints is explicitly known when analyzing the following kinematic structures.

Single Pantograph Leg

In order to explain the employed kinematic couplings in a concise manner, a useful approach is to start by analyzing a basic single-leg model and build up to a four-leg model, then introduce the couplings and their role in kinematic reduction. Imagine a single pantograph leg, with the axis of the hip joint rotated 90 degrees such that it is orthogonal to the ground plane and to the other joint axes. This rotated hip joint can then be renamed as the pivot joint. If a coupled drive actuation technique is employed with this pantograph as shown in Fig. 2.7, with joints j_7 and j_{10} being actuated, the driving point at j_8 can be moved with two degrees of freedom corresponding to side-to-side motion about joint j_1 and forward-backward motion about joint j_4. The number of available degrees of freedom can be confirmed as 2 using Eq. 2.2.1, where the total number of system DoF is 17 and the number of system constraints is 15 ($17 - 15 = 2$ DoF).[2] As previously discussed, the parallel geometry of the pantograph mechanism exhibits the trait that any motions at the driving point within the plane of the pantograph are linearly mapped and amplified at the foot point. This mapping is described by Eq. 2.2.2, where $v_{d,x}$ and $v_{f,x}$ represent the linear velocities of the driving point and the foot point in the x_w direction. The mapping between the driving point and foot point for motions in the y_w direction is described by Eq. 2.2.3, where θ denotes the angle between the thigh and shank.

$$v_{f,x} = \frac{l_{shank}}{l_{thighlink}} v_{d,x} \tag{2.2.2}$$

$$v_{f,y} = \frac{l_{thigh} + l_{shank}\cos\theta}{l_{thigh} - l_{shanklink} + l_{thighlink}\cos\theta} v_{d,y} \tag{2.2.3}$$

While this configuration allows the foot to move side-to-side and forward/backward, the up-down motion is still needed to achieve free foot placement in 3 dimensions. This degree of freedom can be provided by keeping the new pivot joint j_1 and reintroducing the original hip joint as j_2, shown in Fig. 2.8. By actuating the hip joint, the up/down motion can be directly controlled. Combined with the two prismatic coupled drive joints j_8 and j_{11}, the foot point can then be moved in 3-dimensional space. The available degrees of freedom can again be confirmed using Eq. 2.2.1, where the number of system degrees of freedom is equal to 18 and the number of system constraints is equal to 15, yielding $18 - 15 = 3$ DoF.[3]

[2] Joint degrees of freedom provided in Table B.1 of Appendix B
[3] Joint degrees of freedom provided in Table B.2 of Appendix B

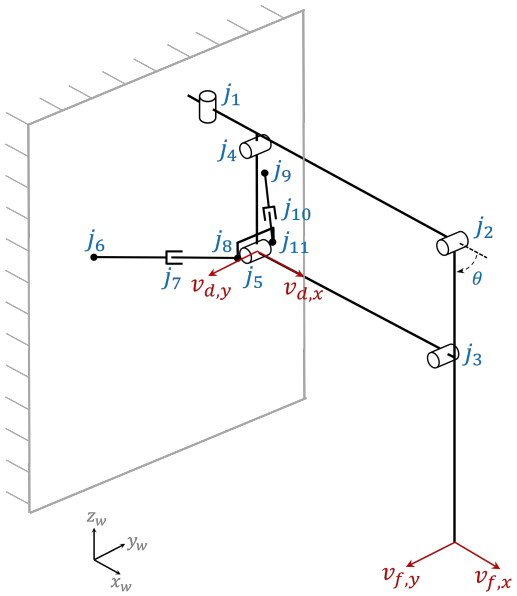


Figure 2.7.: Pantograph leg with 2 DoF foot motion

The kinematic structure in Fig. 2.8 only represents a single pantograph leg with the assumption that both the pantograph pivot joint and the two coupled drive prismatic actuators are connected to a fixed ground. This is obviously not the case in quadruped robots, where the legs are mounted to a floating base body and the only direct contacts with the fixed ground are through the feet. This manipulator structure, however, provides a good framework for expansion to the case of four identical legs connected to a floating base body, which is discussed in the following section.

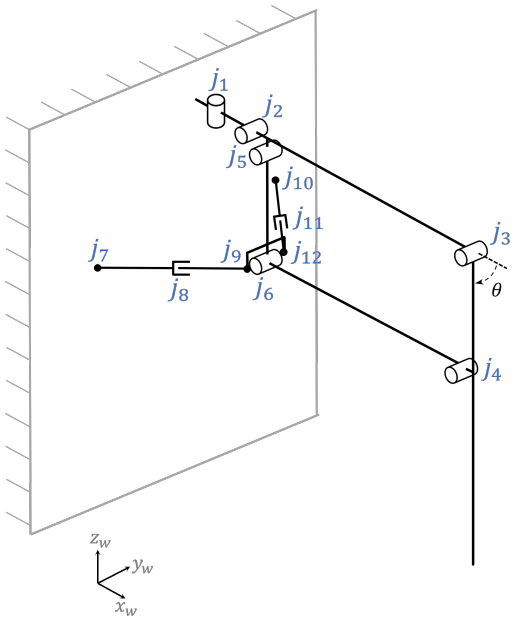


Figure 2.8.: Pantograph leg with 3 DoF foot motion

Four Pantograph Legs

Looking again to the single leg example in Fig. 2.8, the pivot joint j_1 is connected to a fixed ground, as are the two coupled drive ball joints j_7 and j_{10}, allowing control of the driving point's position relative to the fixed ground with the actuated joints j_8 and j_{11}. Shifting to the four-legged case, the fixed ground can be replaced by a floating base body to which all four leg pivot joints as well as the coupled drive ball joints are connected, as shown in Fig. 2.9. This model represents a basic construction of a quadruped using the described pantograph leg mechanisms before any kinematic reductions are applied. Each leg has three independent actuators, one at the hip joint and two in the coupled drive prismatic joints, bringing the total number of actuators in the system to 12. In models containing a floating base, the standard method of representation involves placing six "virtual" joints between the fixed ground frame and a reference frame on the robot's body, in this case represented by frames w and b, respectively. Since contact occurs solely through the feet points, these contacts are typically modeled as virtual ball joints, which introduce 3 constraints on the translational degrees of freedom. With this knowledge in mind, it can be shown that the current kinematic structure exhibits the same number of available degrees of freedom as classical quadrupeds. Assuming three feet are grounded during static walking, the total number of system constraints can be computed as 69 while there are 78 total system degrees of freedom[4], so the available degrees of freedom can be computed as 9 using Eq. 2.2.1. This was seen previously to also be the case in classical quadruped models. Considering the goal of reducing the actuated degrees of freedom to six, this model is obviously not a solution since at least 9 actuators would be needed to control the available degrees of freedom, but it presents a starting point for the introduction of kinematic couplings which can reduce the degrees of freedom.

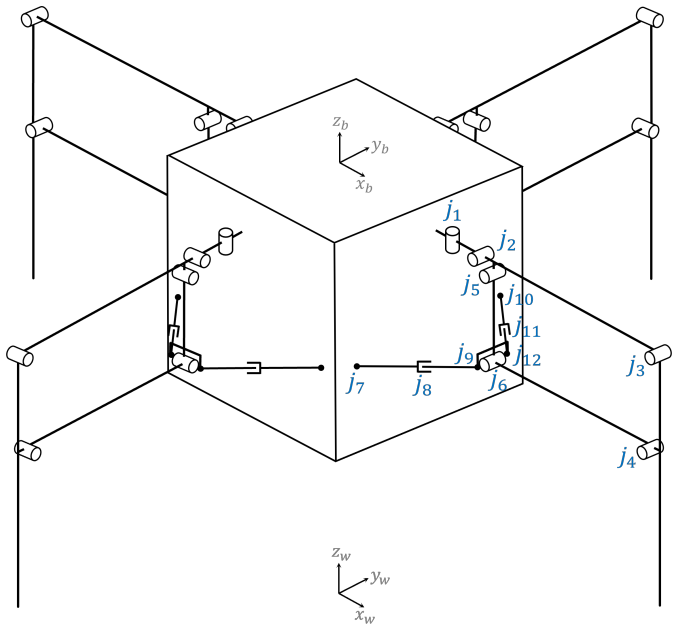


Figure 2.9.: Four pantograph legs mounted to a floating base body

[4]Joint degrees of freedom provided in Table B.3 of Appendix B

Adjacent Leg Couplings

The first of the aforementioned couplings comes from the realization that the two prismatic coupled drive actuators are only needed by a leg when it is in swing stage, as they simply control the side-to-side and forward/backward motion of the respective foot. These actuators can therefore be shared between the neighboring legs in such a way that they control the linear distance between the two legs' driving points. For each leg, one of the coupled drive branches can therefore be removed completely, and the remaining branch can be connected to the adjacent leg's driving point, as seen in Fig. 2.10. These couplings can be referred to as the adjacent leg couplings. In order to implement such couplings in a mechanically feasible manner, the degrees of freedom between the coupled driving points must be noted. It is known that the position of each driving point is capable of moving with three degrees of freedom relative to the floating base, so there are a total of six degrees of freedom between the two points. These six degrees of freedom can be practically compensated in the coupling branch by using ball joints to connect each end with its corresponding leg, but together with the actuated prismatic joint, the branch then has a total of 7 degrees of freedom. Each ball joint has a degree of freedom corresponding to a twisting motion about aligned axes, so one joint can be effectively replaced with a universal joint to eliminate this non-unique degree of freedom.

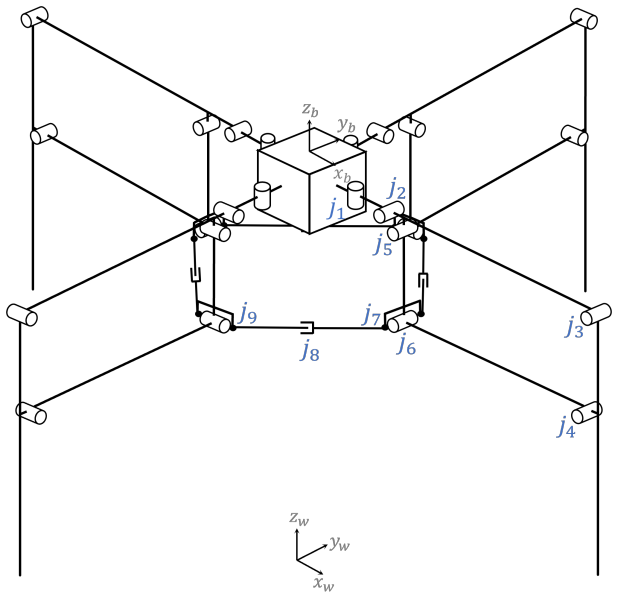


Figure 2.10.: Pantograph leg quadruped after first kinematic reduction

Imagine then that a particular leg is in swing stage and both of its neighbor legs are in contact with the ground. The two adjacent leg couplings connected to either side of the swing leg's driving point can be used to control its position relative to the two neighbor legs' driving points. This reduced model has 54 system degrees of freedom and 45 constraints during static walking, so using Eq. 2.2.1 it can be seen that the total available degrees of freedom is 9.[5] Since the model has only 8 actuated degrees

[5]Joint degrees of freedom provided in Table B.4 of Appendix B

of freedom, the system is under-constrained and an additional joint would need to be actuated for it to become fully-constrained and controllable. It is therefore clear that the integration of further constraints and/or kinematic reductions is necessary to achieve only 6 actuated DoF.

Opposite Leg Couplings

With the adjacent leg couplings implemented, the new kinematic model has a total of 8 active joints, which means that two more must be removed in order to achieve the desired six. Since the coupled drive actuators have already been reduced, it makes sense that the remaining four active hip joints could be a point of interest for the next kinematic reduction or introduction of constraints.
Notice first the specific functions provided by the four hip joints. With each of the hip joints independently actuated, they control the lifting motions of their respective feet relative to the floating base. The hip joint opposite of the swing leg can therefore be kept stationary such that the floating base maintains its position and orientation during the lifting of the swing leg. Enumerating the legs in a clockwise direction with $i, i+1, i+2$ and $i+3$, it can be seen that the hip joints of opposing legs i and $i+2$ are aligned such that the lifting of foot i or $i+2$ results in a seesaw-like pose about an axis formed between feet $i+1$ and $i+3$. In this pose, it is possible for either foot i or $i+2$ to be the swing foot, depending on the position of the quadruped's center of mass. Since only the leg in swing stage requires a lifting function and its opposite leg remains grounded, the two actuated hip joints of the opposite legs can be reduced into a single actuated degree of freedom. This degree of freedom then produces a differential motion between the opposite legs, with the swing leg's lifting defined w.r.t. its opposite leg instead of the floating base.
This reduction can be applied to both pairs of opposite legs, bringing the total degrees of freedom in the hips from 4 to 2. The elimination of independent degrees of freedom leads to a reduction of the available independent functions, which in this case happens to be the floating base orientation about x_b and y_b, since the coupling of the opposite hips prevents keeping the floating base independently stationary. The function for swing leg selection, which is nothing more than position selection of the center of mass, can then be achieved with a combination of other functions, as originally discussed by Yoneda (2007).
Before introducing a practical way of implementing the opposite leg couplings, it is important to again use Eq. 2.2.1 to determine the number of available degrees of freedom and ensure that it does not exceed the desired number of actuators, namely six. The new couplings simply introduce 2 more constraints to the system, so the number of constraints becomes 47 during static walking while the number of system degrees of freedom remains unchanged at 54. The total number of available degrees of freedom is therefore 7, making the system under-constrained when only 6 degrees of freedom are actuated. To make the system fully-constrained, it is therefore necessary to either introduce one more constraint or actuate an additional joint. In this case, the former option is the clear answer since the system should remain at six actuated joints. Viewing the current model from the top, as shown in Fig. 2.11, one feature can be seen

which yields itself perfectly to further constraining. Assuming that the positions of all four feet are grounded in a particular configuration and the six actuated joints are kept stationary, it can be seen that each of the adjacent leg couplings forms a quasi five-bar linkage with its respective legs and the floating base body, denoted by $l_1 \dots l_5$. Each of these five-bar linkages consists of four passive joints j_1, j_2, j_3 and j_5, and one actuated joint, j_4. Since there are only three constraints due to planar loop closure, the number of available degrees of freedom is 2 and the loop can be considered under-constrained when only one joint is actuated. This unconstrained degree of freedom allows the floating base body to rotate freely about the floating base frame axis z_b, resulting in an infinite number of possible body configurations for a given set of actuated joint states.

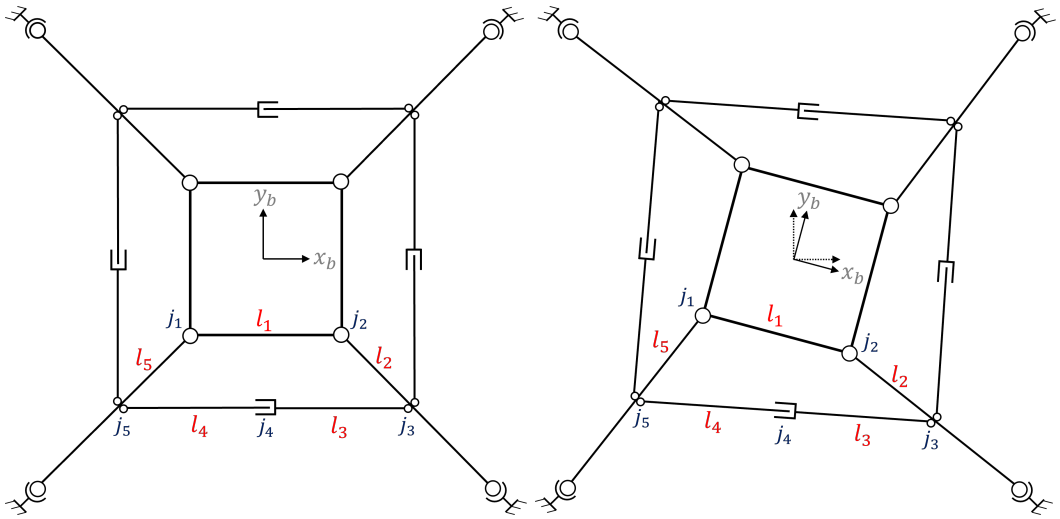


Figure 2.11.: Unconstrained body rotation about z_b

A simple solution for making these loops fully-constrained is to reduce them into four-bar linkages where only one available degree of freedom exists and is actuated by the prismatic joint. Since the four pivot joints are parallel at all times, their axes can be shifted to a coinciding central point, as shown in Fig. 2.12. The adjacent leg couplings together with the legs now form quasi four-bar linkages represented by links $l_1 \dots l_4$ which are fully constrained since 4 system DoF $-$ 3 constraints $=$ 1availableDoF which is actuated by the prismatic joint. With the original floating base body replaced by a single central pivot axis at j_1, the floating base frame b can be fixed instead to the first leg, with the other three legs capable of pivoting relative to the first leg. In other words, the first leg's pivot joint is removed and the other three legs' pivot joints are aligned with z_b. This kinematic structure preserves the original motion capabilities of the pivot joints since they are not eliminated or reoriented, but rather just shifted, so no reduction in functionality occurs. Checking once again the available degrees of freedom in the system during static walking with Eq. 2.2.1, the result is 6 degrees of freedom since the total number of system degrees of freedom is now 53 and the number of constraints is 47.[6] The system is therefore now fully constrained with 6 actuated degrees of freedom.

[6]Joint degrees of freedom provided in Table B.5 of Appendix B

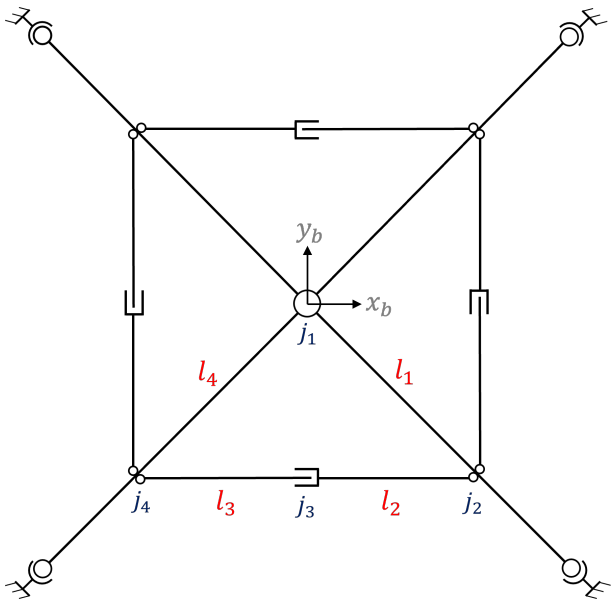


Figure 2.12.: Pivot axes shifted to coincide at center point

With the system fully constrained, a mechanically feasible solution for implementing the opposite leg couplings must be introduced, where the goal is for each pair of opposite hip joints to have their motions linearly coupled. The central pivot joint between the hip joints changes the angle between their axes of rotation, so the coupling of these joints is not as simple as it would be if the hip joint axes were always parallel. However, the linear distance between each hip joint and the central pivot axis remains constant, meaning that the central axis can be used as the actuation point for the hip joints. By fixing a linear actuator vertically along the central axis and connecting rigid links between each of the opposite thighs and the end of the actuator, its linear motion can produce an equal lifting effect in both legs, as shown in Fig. 2.13 It is then only needed to add an additional passive rotational joint between the linear actuator's end and the connected rigid link, to allow the legs to still rotate about the central pivot axis.

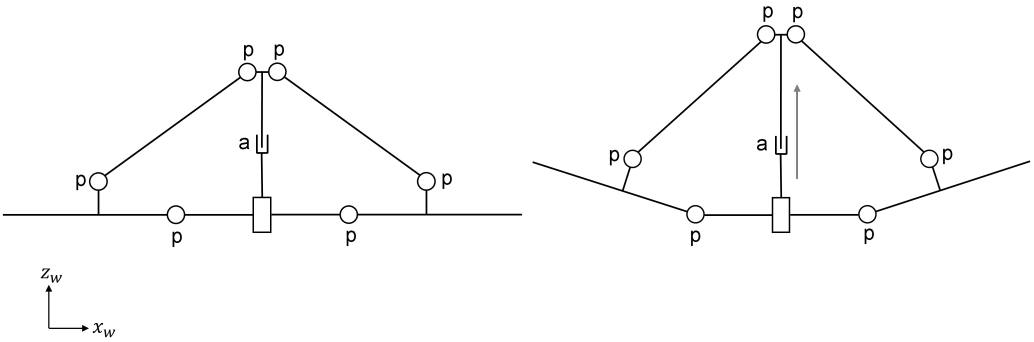


Figure 2.13.: Side view of the proposed opposite leg coupling mechanism (a: active joint, p: passive joint)

Assuming the length of the linear actuator is known, algebraic derivation of the joint angles for all the passive joints within the two symmetrical loops is possible with a

basic geometrical analysis of the coupling mechanism. Referring to Fig. 2.14, the values of a and b can first be determined by Eq. 2.2.4 and Eq. 2.2.5.

$$a = \sqrt{l_2^2 + l_3^2} \tag{2.2.4}$$

$$b = \sqrt{(l_1 - l_6)^2 + l_a^2} \tag{2.2.5}$$

Using the law of cosines, it is then possible to determine both ϑ_{2a} and ϑ_{2b}, which are summed together to get ϑ_2.

$$\vartheta_{2a} = \cos^{-1}\left(\frac{b^2 + (l_1 - l_6)^2 - l_a^2}{2b(l_1 - l_6)}\right) \tag{2.2.6}$$

$$\vartheta_{2b} = \cos^{-1}\left(\frac{a^2 + b^2 - l_5^2}{2ab}\right) \tag{2.2.7}$$

$$\vartheta_2 = \vartheta_{2a} + \vartheta_{2b} \tag{2.2.8}$$

The value of ϑ_1 can then be determined using the law of sines.

$$\vartheta_1 = \sin^{-1}\left(\frac{b \sin \vartheta_{2b}}{l_5}\right) \tag{2.2.9}$$

Finally, the values of ϑ_{3a} and ϑ_{3b} can be determined by the values calculated in Eq. 2.2.6, 2.2.7 and 2.2.8, then summed together to get ϑ_3.

$$\vartheta_{3a} = \pi - \vartheta_{2a} \tag{2.2.10}$$

$$\vartheta_{3b} = \pi - \vartheta_1 - \vartheta_{2b} \tag{2.2.11}$$

$$\vartheta_3 = \vartheta_{3a} + \vartheta_{3b} \tag{2.2.12}$$

Since the two sides of the coupling mechanism are symmetrical, the passive joint angles on both sides are equal and no further calculations are necessary to determine their values. Since the rotational joint axes on each side of the symmetrical mechanism are aligned, each side can be viewed as a planar mechanism. The central pivot axis then allows rotation between the two planes, meaning that the rotational joint added to the end of the linear actuator is geometrically constrained to the pivot axis. With the inclusion of this additional constraint, the entire coupling mechanism has 8 degrees of freedom and 7 constraints, therefore it can be considered fully-constrained when the single linear degree of freedom is actuated. The opposite coupling mechanism is then integrated by placing one on the top side of the pivot axis to couple legs 1 and 3, and the other on the bottom side to couple legs 2 and 4. The final kinematic structure can be seen in Fig. 2.15. Since the opposite couplings introduce additional degrees of freedom and constraints, the total available degrees of freedom can again be check with Eq. 2.2.1. With 65 system degrees of freedom and 59 constraints during

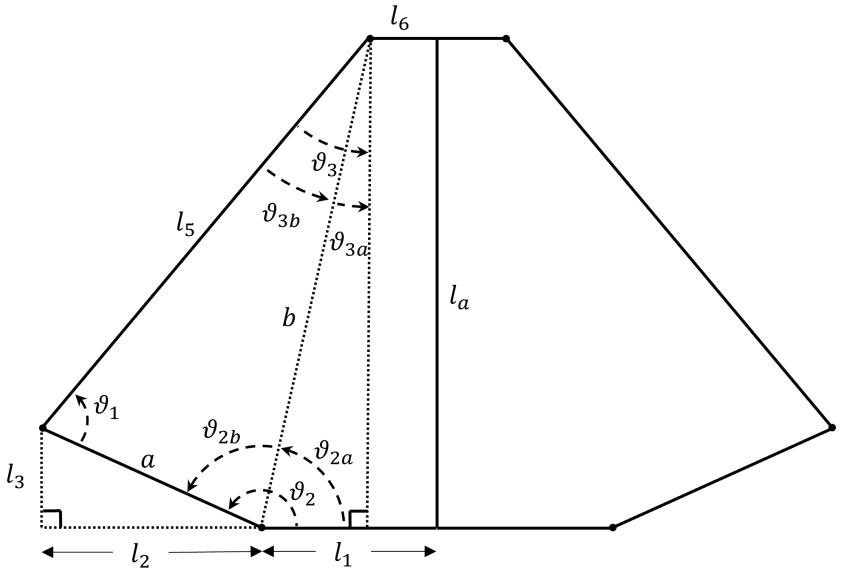


Figure 2.14.: Geometric analysis of the opposite leg coupling mechanism

static walking, the total available degrees of freedom still stands at 6, meaning that the system is fully-defined with the current number of actuators.[7]

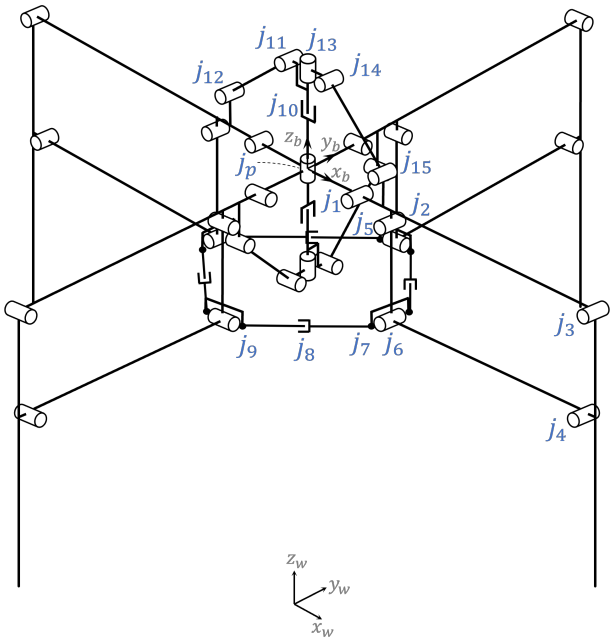


Figure 2.15.: Fully reduced kinematic model with 6 actuated degrees of freedom

2.3. Kinematic Analysis

While the previous section outlined the general design process which led to the proposed kinematic model, a more concrete kinematic analysis is necessary to confirm the

[7]Joint degrees of freedom provided in Table B.6 of Appendix B

expected motion behaviors of the system and to design an adequate control scheme for testing the robot in a simulation environment. This section provides a schematic representation of the model using coordinate frames which is then used to derive representations of the forward and inverse kinematics problems. The model is then further represented in terms of differential kinematics so that an inverse kinematics algorithm can be implemented for executing closed-loop kinematic control, with desired task-space trajectories given as input and actuator velocities computed as output.

2.3.1. Forward Kinematics

The first step in kinematic modeling and analysis of the proposed design is to define a formal representation of its kinematic structure, which can be viewed as a chain of rigid bodies connected to one another by different types of joints. The position and orientation of each of these rigid bodies in space can be described using standard x-y-z coordinate frames, where each rigid body, or link, is assigned a single coordinate frame and referenced with respect to its parent link's frame. It is convenient to place the coordinate frames on each of the rigid bodies using a predefined convention such that the relevant parameters are easy to define and understand, therefore a modified version of the *Denavit-Hartenberg (DH) convention* used by Craig (2005) is adopted.
As proposed by Denavit and Hartenberg (1955), the original DH convention involves a few key rules regarding where the origin of a coordinate frame should be located and how the x- and z-axes should be oriented. Namely, z_i should be placed along the axis of joint $i+1$ and the origin of frame i, denoted by O_i, should be located at the the intersection of axis z_i with the common normal to z_{i-1} and z_i. Four parameters known as the *DH parameters* are then used to represent the transformation between frames $i-1$ and i. The angle between axes x_{i-1} and x_i about z_{i-1} is denoted by parameter ϑ_i, the distance between O_{i-1} and O_i along z_{i-1} by parameter d_i, the angle between axes z_{i-1} and z_i about x_i by parameter α_i, and the distance between O_{i-1} and O_i along x_i by parameter a_i. The modified version simply places the coordinates of frame O_i along joint axis i instead of $i+1$ such that a transformation between coordinate frames O_{i-1} and O_i is represented by the homogeneous transformation matrix in Eq. 2.3.1, where c_ϑ represents $\cos(\vartheta)$ and s_ϑ represents $\sin(\vartheta)$. The parameters ϑ_i and d_i remain the same as in the original convention, but α_i represents the angle between axes z_{i-1} and z_i about x_{i-1} and a_i represents the distance between O_{i-1} and O_i along x_{i-1}. This modified version of DH convention is chosen over the classic DH convention because it places the coordinate frames in such as way that more closely resembles the RPY coordinate frame transformations used by URDF, which is eventually utilized to simulate the proposed model.

$$\mathbf{A}_i^{i-1} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i} & 0 & a_{i-1} \\ s_{\vartheta_i} c_{\alpha_{i-1}} & c_{\vartheta_i} c_{\alpha_{i-1}} & -s_{\alpha_{i-1}} & -d_i s_{\alpha_{i-1}} \\ s_{\vartheta_i} s_{\alpha_{i-1}} & c_{\vartheta_i} s_{\alpha_{i-1}} & c_{\alpha_{i-1}} & d_i c_{\alpha_{i-1}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3.1}$$

The forward kinematics problem involves computation of the pose of a coordinate frame, usually the last frame in a kinematic chain, with respect to a reference frame

when all of the joint states are known. The forward kinematics equation for a particular kinematic chain can be defined by simply post-multiplying the transformation matrices corresponding to each coordinate frame along the chain, as follows:

$$\mathbf{A}_n^i = \mathbf{A}_{i+1}^i \mathbf{A}_{i+2}^{i+1} \dots \mathbf{A}_n^{n-1} \tag{2.3.2}$$

Looking again at the final proposed kinematic model in Fig. 2.15, the representation of the kinematic structure with coordinate frames and derivation of the forward kinematics equations are broken down into smaller subsections of the system for simplicity. Specifically, the representation and forward kinematics derivation starts with the floating base and legs, followed by the adjacent and opposite leg couplings. It is important to note that the forward kinematics of the system is only derived to the extent necessary for representation of the inverse kinematics problem, since the derivation of joint configurations from given task-space configurations is of more interest in motion planning.

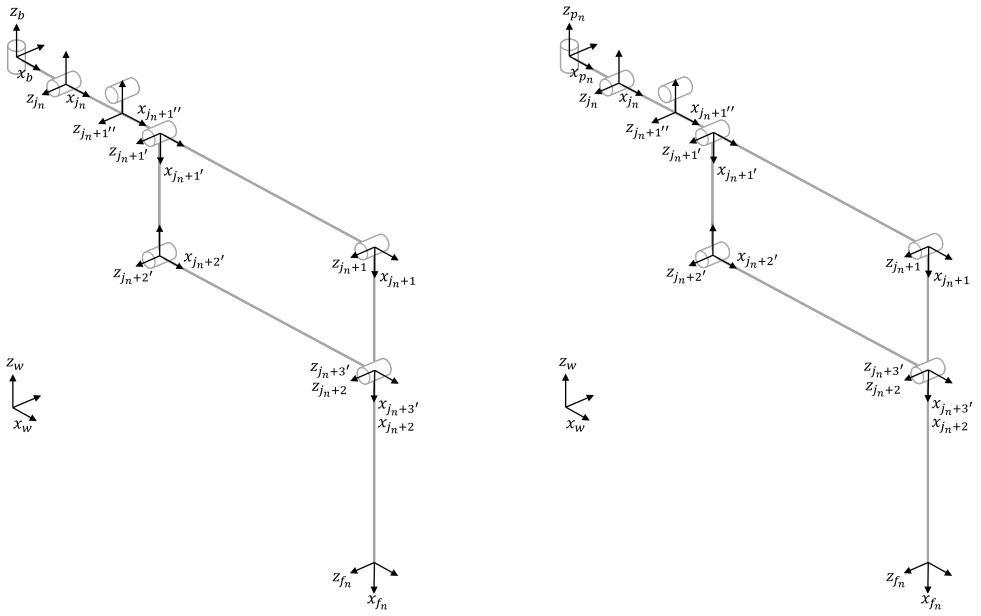
Floating Base & Legs

The coordinate frame representation starts with defining a fixed reference frame, often called the inertial frame or world frame, which is placed arbitrarily in the world. In the case of walking robots, motion is defined by movement relative to the ground, so the world frame O_w can logically be placed somewhere on the surface of the ground plane, with z_w pointing upwards, orthogonal to the ground. As opposed to the standard fixed-base manipulator representation, legged robots are usually modeled with a floating base which has six degrees of freedom relative to the world frame. To represent this, a coordinate frame is fixed to the floating base and six virtual passive joints are placed between the world frame and the base frame frame. The transformation between world frame w and floating base frame b is then represented by a single transformation matrix which defines orientation using RPY angles and translation in all three dimensions relative to the world frame. This matrix is given by Eq. 2.3.3, where ϕ, ϑ and ψ denote the rotations of frame b about axes z_w, y_w and x_w, respectively, and x, y and z denote the translations of frame b along axes x_w, y_w and z_w.

$$\mathbf{T}_b^w = \begin{bmatrix} c_\phi c_\vartheta & c_\phi s_\vartheta s_\psi - s_\phi c_\psi & c_\phi s_\vartheta c_\psi + s_\phi s_\psi & x \\ s_\phi c_\vartheta & s_\phi s_\vartheta s_\psi + c_\phi c_\psi & s_\phi s_\vartheta c_\psi - c_\phi s_\psi & y \\ -s_\vartheta & c_\vartheta s_\psi & c_\vartheta c_\psi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3.3}$$

Most quadrupeds have a clearly defined central body to which all the legs are attached, and the base coordinate frame is typically attached to the center of this body. The proposed quadruped differs in that no clearly defined central body exists, and instead the base frame is connected to the central pivot shaft, which is rigidly fixed to the first leg's pivot link. The axis z_b is aligned with the central pivot axis, and x_b is aligned so that it points in the direction of the first leg's hip joint. Fig. 2.16a shows the placement of all the coordinate frames for the first leg and Fig. 2.16b shows the placement for the other three legs, where $n \in 1, 2, 3, 4$ denotes the leg number. The representation for all

the legs is therefore the same, except that the first frame p_n of legs 2, 3 and 4 is the pivot frame, which is a child of the base frame b on leg 1.



(a) First leg (b) Other legs

Figure 2.16.: Coordinate frame placement for the pantograph legs

In contrast to the simple serial kinematic chains of classical robotic manipulators, the proposed quadruped has a complex kinematic chain which consists of many branches, making the representation of the forward kinematics more complicated. Classical serial manipulators typically have a single end-effector frame which is the target frame for derivation of the forward kinematics. In contrast, in quadrupeds such as the one proposed, the four feet are simultaneously considered as forward kinematic targets since it is the positions of the feet which are of interest in locomotion tasks. In addition, the floating base body is sometimes considered as a forward kinematic target when the pose of the body is of interest. In the proposed quadruped, the existence of several closed chains further complicates the representation with DH convention and the derivation of the forward kinematic equations due to the recursive nature of the method. Fortunately, accounting for these closed chains is quite simple. Each of the closed chains can be virtually opened so that the rigid body tree consists of purely open chains. These open chains can be easily represented with DH convention, then constraint equations can be introduced to represent the closing conditions for each loop. It is important to note that a closed-form solution to the constraints is not guaranteed and the forward kinematics solution for parallel mechanisms can often be quite ambiguous since some arbitrary set of joint variables could be infeasible or correspond to multiple possible poses.
The kinematic chain for the pantograph leg is shown in Fig. 2.17 where it can be seen that a closed loop exists between frames j_n and $j_n + 2$. The existence of this loop presents two possible paths from the base frame b to the foot frame $j_n + 3$ when deriving the forward kinematics equations, one following the upper branch denoted

by

$$\mathbf{T}_{j_n+3}^{b} = \mathbf{A}_{j_n}^{b}\mathbf{A}_{j_n+1}^{j_n}\mathbf{A}_{j_n+2}^{j_n+1}\mathbf{A}_{j_n+3}^{j_n+2} \tag{2.3.4}$$

and the other following the lower branch denoted by

$$\mathbf{T}_{j_n+3}^{b} = \mathbf{A}_{j_n}^{b}\mathbf{A}_{j_n+1'}^{j_n}\mathbf{A}_{j_n+2'}^{j_n+1'}\mathbf{A}_{j_n+2}^{j_n+2'}\mathbf{A}_{j_n+3}^{j_n+2} \tag{2.3.5}$$

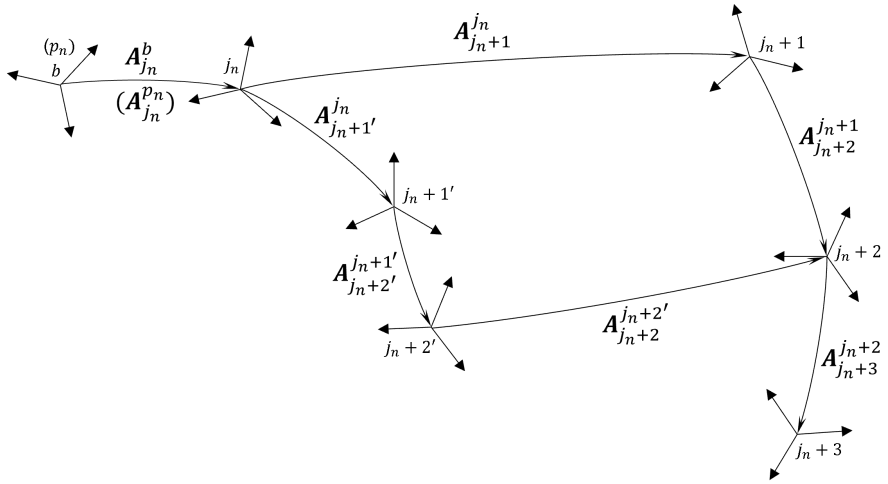


Figure 2.17.: Kinematic chain for the pantograph leg with closed loop present

The method for representing the closed chain as open chains begins by selecting the passive joint $j_n + 2$ to be the *cutting joint* since it is the point where the upper and lower branches rejoin into a single branch. The connection at joint $j_n + 2$ is removed and a new coordinate frame $j_n + 3'$ is attached to the end of link $j_n + 2'$ as shown in Fig. 2.18, resulting in an open chain with two branches. Since the position of frames $j_n + 2$ and $j_n + 3'$ and the orientation of their z-axes should be aligned at all times, the following constraints are imposed between them:

$$\begin{cases} \mathbf{R}_{j_n}^{j_n+2}(\mathbf{p}_{j_n+2}^{j_n} - \mathbf{p}_{j_n+3'}^{j_n}) = \begin{bmatrix} 0 & 0 & d_{j_n+2,j_n+3'} \end{bmatrix}^T \\ \mathbf{z}_{j_n+2}^{j_n} = \mathbf{z}_{j_n+3'}^{j_n} \end{cases} \tag{2.3.6}$$

where $\mathbf{R}_{j_n}^{j_n+2}$ denotes the orientation of frame j_n with respect to frame $j_n + 2$, $\mathbf{p}_{j_n+2}^{j_n}$ and $\mathbf{p}_{j_n+3'}^{j_n}$ denote the positions of frames $j_n + 2$ and j_n with respect to frame j_n and $d_{j_n+2,j_n+3'}$ is the fixed offset between frames $j_n + 2$ and $j_n + 3'$ along axis z_{j_n+2}. Since the loop is a planar parallelogram, it is also known that $d_{j_n+2,j_n+3'} = 0$. The kinematic transformation for the branch from j_n to $j_n + 2$ is then denoted as follows:

$$\mathbf{T}_{j_n+2}^{j_n}(\mathbf{q}') = \mathbf{A}_{j_n+1}^{j_n}\mathbf{A}_{j_n+2}^{j_n+1} \tag{2.3.7}$$

where the state vector is represented by $\mathbf{q}' = \{\vartheta j_n + 1\}$. The transformation for the branch from j_n to $j_n + 3'$ is then denoted as follows:

$$\mathbf{T}^{j_n}_{j_n+3'}(\mathbf{q}'') = \mathbf{A}^{j_n}_{j_n+1'}\mathbf{A}^{j_n+1'}_{j_n+2'}\mathbf{A}^{j_n+2'}_{j_n+3'} \tag{2.3.8}$$

where the state vector is represented by $\mathbf{q}'' = \{\vartheta_{j_n+1'}\ \vartheta_{j_n+2'}\}$. The DH parameters for both branches are provided in Table 2.1.

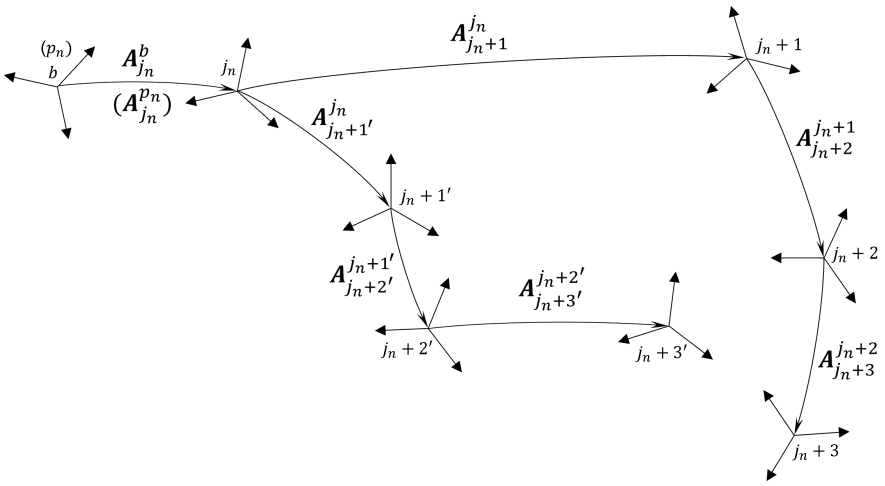


Figure 2.18.: Kinematic chain for the pantograph leg after cutting the loop

Link	a_i	α_i	d_i	ϑ_i
(p_n)	0	0	0	ϑ_{p_n}
j_n	a_{j_n}	$\pi/2$	0	ϑ_{j_n}
j_n+1	a_{j_n+1}	0	0	ϑ_{j_n+1}
j_n+2	a_{j_n+2}	0	0	0
j_n+1'	$a_{j_n+1'}$	0	0	$\vartheta_{j_n+1'}$
j_n+2'	$a_{j_n+2'}$	0	0	$\vartheta_{j_n+2'}$
j_n+3'	$a_{j_n+3'}$	0	0	$\vartheta_{j_n+3'}$
j_n+3	a_{j_n+3}	0	0	0

Table 2.1.: DH parameters for the pantograph legs

It should be noted that the constraints in Eq. 2.3.6 present six equalities which must hold true for any given joint state vector $\mathbf{q}$, but some of these equalities can become dependent for simple geometries, e.g. planar structures, reducing the number of constraint equations and making a solution easier to find. Since the closed chain in the proposed leg mechanism is planar, the orientation constraints and the z-positional constraint are satisfied independently from $\mathbf{q}$, reducing the number of independent constraint equations to the two corresponding to $\mathbf{p}_x$ and $\mathbf{p}_y$.
It should be noted that the kinematic chains for all four legs are identical aside from the exclusion of the pivot joint frame p_n in the first leg. For the three legs with the pivot frame, the transformation tree begins with the base frame b, then continues as normal with frame p_n.

Adjacent Leg Couplings

The next component of the model for which coordinate frames must be assigned and the forward kinematic equations derived is the adjacent leg coupling. Just as with the pantograph legs, the adjacent leg couplings form closed loops in the kinematic chain which must be virtually cut open to form an open chain tree. Constraint equations are then introduced to represent the loop closing conditions. As already discussed in Chapter 2.2.3, a single coupling exists for each adjacent leg pair with connections at the driving point on the thighlink of each leg. If leg n is an arbitrarily chosen leg and leg $n+1$ is the clockwise neighboring leg, the adjacent leg couplings can be represented identically in both directions (n to $n+1$ or $n+1$ to n), therefore a common approach is chosen for all couplings with their kinematic chains defined as starting from leg n and ending at leg $n+1$. The coordinate frames can then be assigned using the modified DH convention as shown in Fig. 2.19, with all four couplings being identical aside from the exclusion of the pivot joint frame p_n in the first leg.

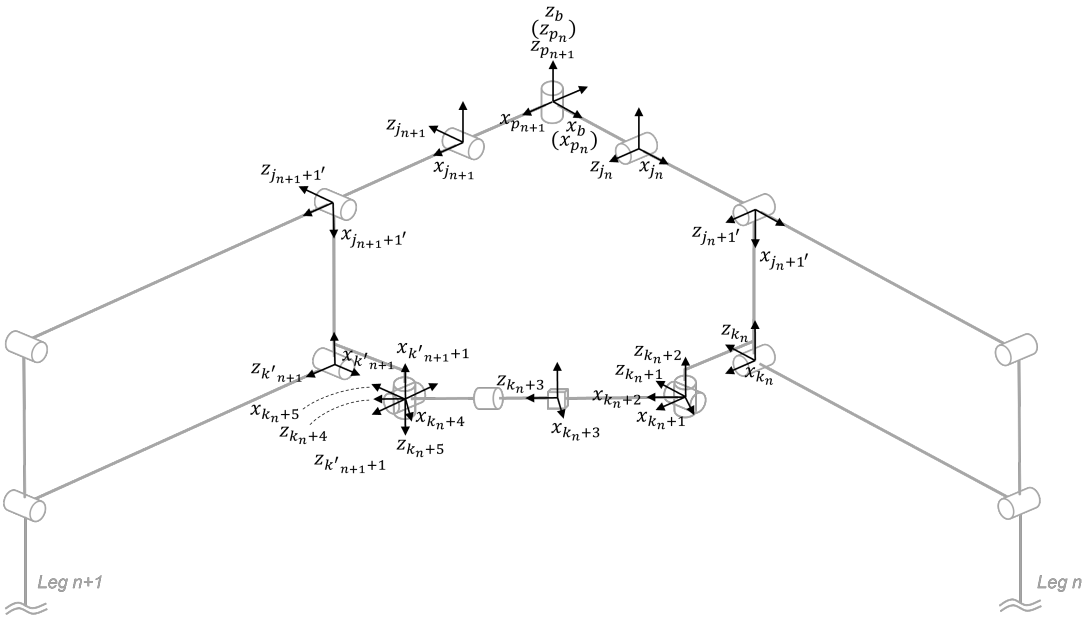


Figure 2.19.: Coordinate frame placement for the adjacent leg couplings

The closed kinematic chain for the adjacent leg couplings can then be represented as shown in Fig. 2.20. The passive joint $k'_{n+1}+1$ can be chosen to be the cutting joint, and a new frame k_n+6 is added such that the loop is transformed into two open-chain branches. Using the DH parameters provided in Table 2.2, the forward kinematic equations can be derived for the two resulting branches, with the first branch transformation derived as follows:

$$\mathbf{T}^{b}_{k_n+6}(\mathbf{q}') = \mathbf{A}^{b}_{p_n}\mathbf{A}^{p_n}_{j_n}\mathbf{A}^{j_n}_{j_n+1'}\mathbf{A}^{j_n+1'}_{k_n}\mathbf{A}^{k_n}_{k_n+1}\mathbf{A}^{k_n+1}_{k_n+2}\mathbf{A}^{k_n+2}_{k_n+3}\mathbf{A}^{k_n+3}_{k_n+4}\mathbf{A}^{k_n+4}_{k_n+5}\mathbf{A}^{k_n+5}_{k_n+6} \tag{2.3.9}$$

with the joint vector denoted by $\mathbf{q}' = \{\vartheta_{p_n}, \vartheta_{j_n}, \vartheta_{j_n+1'}, \vartheta_{k_n+1}, \vartheta_{k_n+2}, d_{k_n+3}, \vartheta_{k_n+4}, \vartheta_{k_n+5}, \vartheta_{k_n+6}\}$. The second branch transformation can then be derived as follows:

$$\mathbf{T}^{b}_{k'_{n+1}+1}(\mathbf{q}'') = \mathbf{A}^{b}_{p_{n+1}}\mathbf{A}^{p_{n+1}}_{j_{n+1}}\mathbf{A}^{j_{n+1}}_{j_{n+1}+1'}\mathbf{A}^{j_{n+1}+1'}_{k'_{n+1}}\mathbf{A}^{k'_{n+1}}_{k'_{n+1}+1} \tag{2.3.10}$$

where the joint vector is denoted by $\mathbf{q}'' = \{\vartheta_{p_{n+1}}, \vartheta_{j_{n+1}}, \vartheta_{j_{n+1}+1'}\}$.

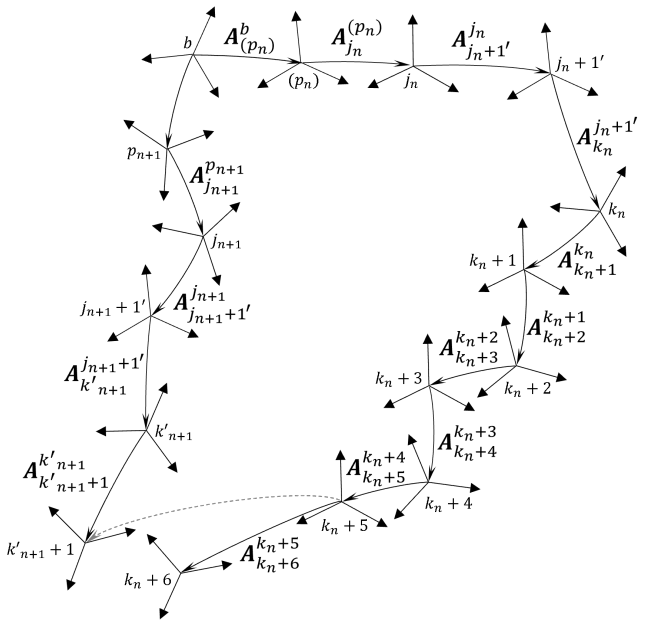


Figure 2.20.: Kinematic chain for the adjacent leg couplings before and after cutting the closed loop

It is then necessary to introduce constraint equations for the closing conditions of the loop, which are shown in Eq. 2.3.11. Just as before, there are six total equalities to be satisfied, however since the kinematic structure of this loop is not planar and contains many non-parallel joints, no clear reduction is possible and all six equalities can be seen as independent and necessary for solving the forward kinematics. There are therefore six independent constraint equations for each adjacent leg coupling.

$$\begin{cases} \mathbf{R}^{k_n+6}_{b}(\mathbf{p}^{b}_{k_n+6} - \mathbf{p}^{b}_{k'_{n+1}+1}) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ z^{b}_{k_n+6} = z^{b}_{k'_{n+1}+1} \end{cases} \tag{2.3.11}$$

Opposite Leg Couplings

The final component in the proposed design which needs to have coordinate frames assigned is the opposite leg coupling. There are two such couplings, one on the top of the pivot shaft connecting legs 1 and 3, and the other inverted on the bottom connecting legs 2 and 4. These couplings again form closed loops in the kinematic chain which must be cut open to derive the open chain forward kinematics for each branch, with constraint equations introduced to represent the closing conditions. Specifically, each of the couplings contains two separate closed loops, but the couplings are virtually identical except for the exclusion of the pivot frame in the first leg of the top coupling. The coordinate frames for the top coupling can be assigned based on the modified

Link	a_i	α_i	d_i	ϑ_i
(p_n)	0	0	0	ϑ_{p_n}
j_n	a_{j_n}	$\pi/2$	0	ϑ_{j_n}
$j_n + 1'$	$a_{j_n+1'}$	0	0	$\vartheta_{j_n+1'}$
k_n	a_{k_n}	$\pi/2$	0	$\pi/2$
$k_n + 1$	a_{k_n+1}	0	0	ϑ_{k_n+1}
$k_n + 2$	0	$-\pi/2$	0	ϑ_{k_n+2}
$k_n + 3$	0	$\pi/2$	d_{k_n+3}	0
$k_n + 4$	0	0	d_{k_n+4}	ϑ_{k_n+4}
$k_n + 5$	0	$\pi/2$	0	ϑ_{k_n+5}
$k_n + 6$	0	$\pi/2$	0	ϑ_{k_n+6}
p_{n+1}	0	0	0	$\vartheta_{p_{n+1}}$
j_{n+1}	$a_{j_{n+1}}$	$\pi/2$	0	$\vartheta_{j_{n+1}}$
$j_{n+1} + 1'$	$a_{j_{n+1}+1'}$	0	0	$\vartheta_{j_{n+1}+1'}$
k'_{n+1}	$a_{k'_{n+1}}$	$-\pi/2$	0	$\pi/2$
$k'_{n+1} + 1$	$a_{k'_{n+1}+1}$	0	0	$\pi/2$

Table 2.2.: DH parameters for the adjacent leg couplings

DH convention as shown in Fig. 2.21, while the coordinate frame assignment for the bottom coupling can be seen in Fig. 2.22.
For the opposite leg coupling between legs 1 and 3, the closed kinematic chain is then represented as shown in Fig. 2.23. It can be seen that two closed loops exist, so both must be virtually cut, resulting in four open chain branches. Choosing the cutting joints to be at the two passive joints $j_1 + 1''$ and $j_3 + 1''$, two new frames $f + 2$ and $f + 3'$ are added to the ends of the cut branches and the new open chain is represented as shown in Fig. 2.24.
The forward kinematic transformations for the four branches are then given as follows:

$$\mathbf{T}^b_{j_1+2''}(\mathbf{q}') = \mathbf{A}^b_{j_1}\mathbf{A}^{j_1}_{j_1+1''}\mathbf{A}^{j_1+1''}_{j_1+2''} \tag{2.3.12}$$

$$\mathbf{T}^b_{f+2}(\mathbf{q}'') = \mathbf{A}^b_f\mathbf{A}^f_{f+1}\mathbf{A}^{f+1}_{f+2} \tag{2.3.13}$$

$$\mathbf{T}^b_{j_3+2''}(\mathbf{q}''') = \mathbf{A}^b_{p_3}\mathbf{A}^{p_3}_{j_3}\mathbf{A}^{j_3}_{j_3+1''}\mathbf{A}^{j_3+1''}_{j_3+2''} \tag{2.3.14}$$

$$\mathbf{T}^b_{f+3'}(\mathbf{q}'''') = \mathbf{A}^b_f\mathbf{A}^f_{f+1'}\mathbf{A}^{f+1'}_{f+2'}\mathbf{A}^{f+2'}_{f+3'} \tag{2.3.15}$$

where the joint state vectors are given by $\mathbf{q}' = \{\vartheta_{j_1}\}$, $\mathbf{q}'' = \{d_f, \vartheta_{f+1}\}$, $\mathbf{q}''' = \{\vartheta_{p_3}, \vartheta_{j_3}\}$ and $\mathbf{q}'''' = \{d_f, \vartheta_{f+1'}, \vartheta_{f+2'}\}$. The DH parameters are provided in Table 2.3.
The positional and orientational constraint equations can then be introduced for the two loops as follows:

$$\begin{cases} \mathbf{R}^{f+2}_b(\mathbf{p}^b_{f+2} - \mathbf{p}^b_{j_1+2''}) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ z^b_{f+2} = z^b_{j_1+2''} \end{cases} \tag{2.3.16}$$

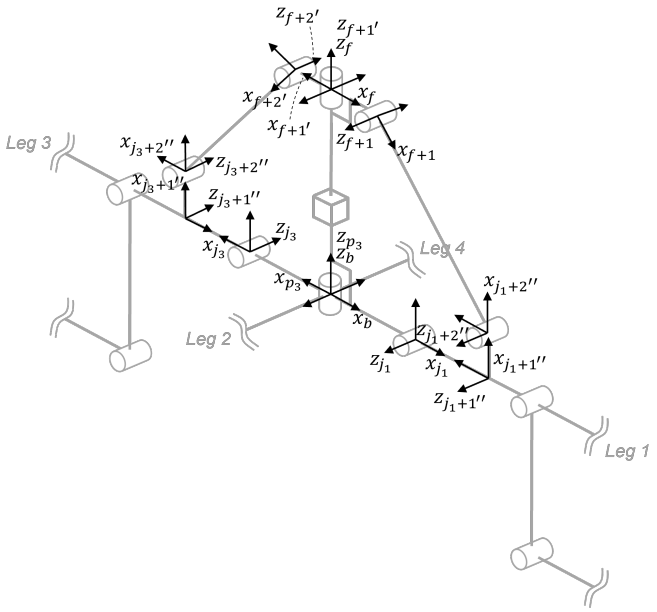


Figure 2.21.: Coordinate frame placement for the opposite leg coupling between legs 1 and 3

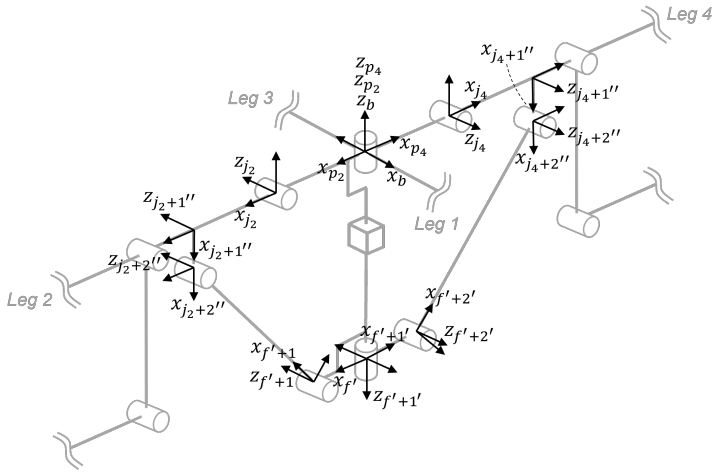


Figure 2.22.: Coordinate frame placement for the opposite leg coupling between legs 2 and 4

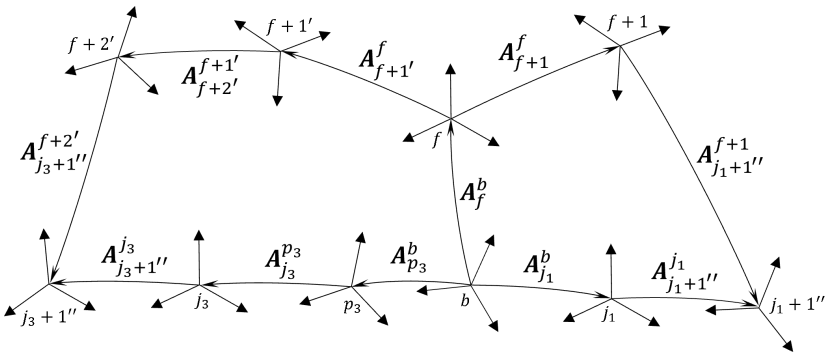


Figure 2.23.: Kinematic chain for the opposite leg coupling between legs 1 and 3

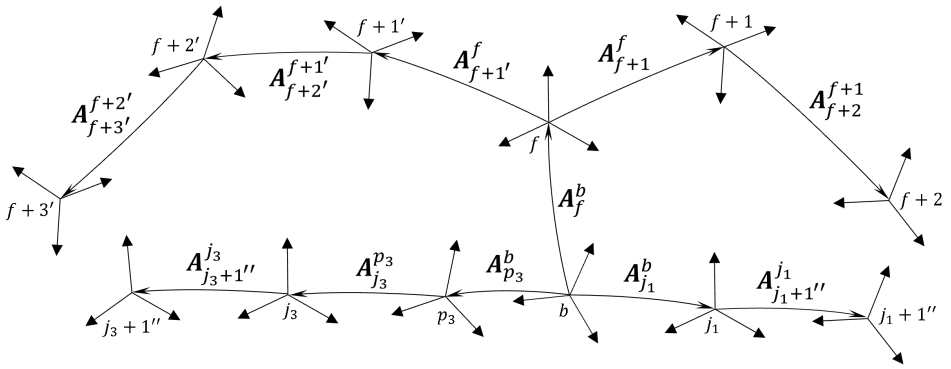


Figure 2.24.: Kinematic chain for the opposite leg coupling between legs 1 and 3 after cutting the loop

$$\begin{cases} \mathbf{R}^{f+3'}_{p_3}(\mathbf{p}^{p_3}_{f+3'} - \mathbf{p}^{p_3}_{j_3+2''}) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ z^{p_3}_{f+3'} = z^{p_3}_{j_3+2''} \end{cases} \tag{2.3.17}$$

Link	a_i	α_i	d_i	ϑ_i
j_1	a_{j_1}	$\pi/2$	0	ϑ_{j_1}
j_1+1''	$a_{j_1+1''}$	0	0	$\pi/2$
j_1+2''	$a_{j_1+2''}$	0	0	$-\pi/2$
f	0	0	d_f	0
$f+1$	a_{f+1}	$\pi/2$	0	ϑ_{f+1}
$f+2$	a_{f+2}	0	0	0
p_3	0	0	0	ϑ_{p_3}
j_3	0	$\pi/2$	0	ϑ_{j_3}
j_3+1''	$a_{j_3+1''}$	0	0	$\pi/2$
j_3+2''	$a_{j_3+2''}$	0	0	$-\pi/2$
$f+1'$	0	0	0	$\vartheta_{f+1'}$
$f+2'$	$a_{f+2'}$	$\pi/2$	0	$\vartheta_{f+2'}$
$f+3'$	$a_{f+3'}$	0	0	0

Table 2.3.: DH parameters for the opposite leg coupling between legs 1 and 3

For the opposite leg coupling between legs 2 and 4, the closed kinematic chain is shown in Fig. 2.25. The structure is almost identical to that of the coupling between legs 1 and 3 and also contains two closed loops which must be cut open, resulting in four kinematic branches. With the passive joints j_2+1'' and j_4+1'' chosen as the cutting joints, the frames $f'+2$ and $f'+3'$ are added to the ends of the cut branches and the resulting kinematic chain is shown in Fig. 2.26. The forward kinematic equations for the four branches are then represented as follows:

$$\mathbf{T}^{b}_{j_2+2''}(\mathbf{q}') = \mathbf{A}^{b}_{p_2}\mathbf{A}^{p_2}_{j_2}\mathbf{A}^{j_2}_{j_2+1''}\mathbf{A}^{j_2+1''}_{j_2+2''} \tag{2.3.18}$$

$$\mathbf{T}^{b}_{f'+2}(\mathbf{q}'') = \mathbf{A}^{b}_{p_2}\mathbf{A}^{p_2}_{f'}\mathbf{A}^{f'}_{f'+1}\mathbf{A}^{f'+1}_{f'+2} \tag{2.3.19}$$

$$\mathbf{T}^{b}_{j_4+2''}(\mathbf{q}''') = \mathbf{A}^{b}_{p_4}\mathbf{A}^{p_4}_{j_4}\mathbf{A}^{j_4}_{j_4+1''}\mathbf{A}^{j_4+1''}_{j_4+2''} \tag{2.3.20}$$

$$\mathbf{T}^{b}_{f'+3'}(\mathbf{q}'''') = \mathbf{A}^{b}_{p_2}\mathbf{A}^{p_2}_{f'}\mathbf{A}^{f'}_{f'+1'}\mathbf{A}^{f'+1'}_{f'+2'}\mathbf{A}^{f'+2'}_{f'+3'} \tag{2.3.21}$$

where the joint state vectors are denoted by $\mathbf{q}' = \{\vartheta_{p_2}\ \vartheta_{j_2}$, $\mathbf{q}'' = \{\vartheta_{p_2}\ d'_f\ \vartheta_{f'+1}\}$, $\mathbf{q}''' = \{\vartheta_{p_4}\ \vartheta_{j_4}\}$ and $\mathbf{q}'''' = \{\vartheta_{p_2}\ d'_f\ \vartheta_{f'+1'}\ \vartheta_{f'+2'}$. The DH parameters are provided in Table 2.4.

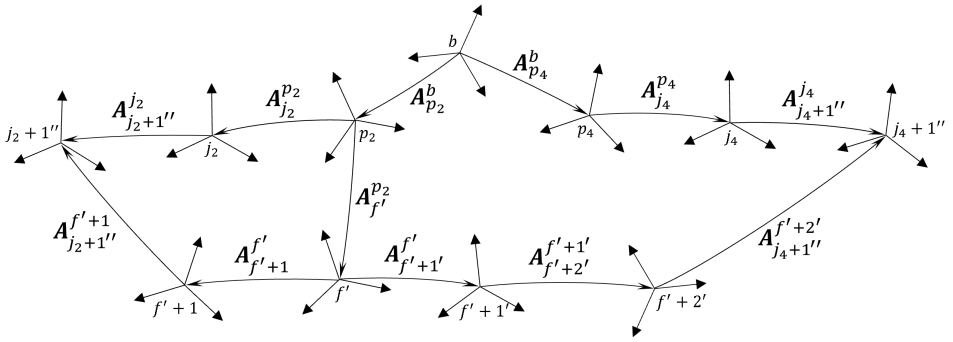


Figure 2.25.: Kinematic chain for the opposite leg coupling between legs 2 and 4

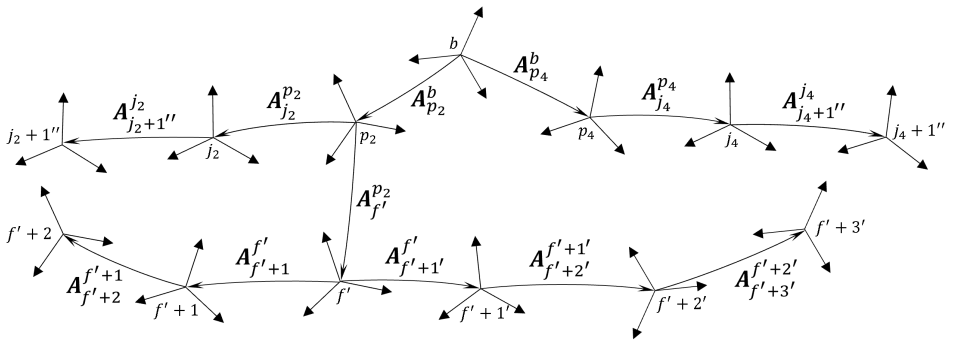


Figure 2.26.: Kinematic chain for the opposite leg coupling between legs 2 and 4 after cutting the loop

The constraint equations can then be introduced for the two loops in the same way as for the coupling between legs 1 and 3 as follows:

$$\begin{cases} \mathbf{R}^{f'+2}_{p_2}(\mathbf{p}^{p_2}_{f'+2} - \mathbf{p}^{p_2}_{j_2+2''}) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ z^{p_2}_{f'+2} = z^{p_2}_{j_2+2''} \end{cases} \tag{2.3.22}$$

$$\begin{cases} \mathbf{R}^{f'+3'}_{p_4}(\mathbf{p}^{p_4}_{f'+3'} - \mathbf{p}^{p_4}_{j_4+2''}) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ z^{p_4}_{f'+3'} = z^{p_4}_{j_4+2''} \end{cases} \tag{2.3.23}$$

Link	a_i	α_i	d_i	ϑ_i
p_2	0	0	0	ϑ_{p_2}
j_2	a_{j_2}	$\pi/2$	0	ϑ_{j_2}
j_2+1''	$a_{j_2+1''}$	0	0	$-\pi/2$
j_2+2''	$a_{j_2+2''}$	0	0	0
f'	0	π	$d_{f'}$	0
$f'+1$	$a_{f'+1}$	$-\pi/2$	0	$\vartheta_{f'+1}$
$f'+2$	$a_{f'+2}$	0	0	0
p_4	0	0	0	ϑ_{p_4}
j_4	0	$\pi/2$	0	ϑ_{j_4}
j_4+1''	$a_{j_4+1''}$	0	0	$-\pi/2$
j_4+2''	$a_{j_4+2''}$	0	0	0
$f'+1'$	0	0	0	$\vartheta_{f'+1'}$
$f'+2'$	$a_{f'+2'}$	$-\pi/2$	0	$\vartheta_{f'+2'}$
$f'+3'$	$a_{f'+3'}$	0	0	0

Table 2.4.: DH parameters for the opposite leg coupling between legs 2 and 4

For both opposite leg couplings, there exist some notable geometrical features which allow for the constraint equations to be reduced. First it can been seen that just as with the pantograph legs, all the joints within each coupling have parallel axes, aside from the central pivot joints p_n and the top/bottom pivot joints $f+1'$, $f'+1'$. Looking again at the coupling between legs 1 and 3 in Fig. 2.21, both the central and the top pivot frames, p_3 and $f+1'$, have their z-axes aligned and are referenced with respect to the base frame b. Due to the planar closed loop consisting of joints with axes orthogonal to the pivot axis, it is known that the joint values ϑ_{p_3} and $\vartheta_{f+1'}$ must be equal at all times, so the equality $\vartheta_{p_3} = \vartheta_{f+1'}$ can be introduced as a constraint and the closed loop can then be viewed as a planar mechanism which is decoupled from its rotation about the pivot axis. Viewing each loop as purely planar, the orientation constraints in Eqs. 2.3.16 and 2.3.17 are satisfied at all times, thus they are dependent and can be ignored. This leaves only the three positional constraints representing each loop, for which $\mathbf{p}_y$ is also dependent due to the planar geometry, leaving only two independent position constraints for each loop corresponding to $\mathbf{p}_x$ and $\mathbf{p}_\mathbf{z}$. Together with the pivot joint equality, the opposite leg coupling between legs 1 and 3 is therefore represented by five constraint equations.

Looking then at the coupling between legs 2 and 4 in Fig. 2.22, the same geometric principles hold for decoupling of the planar loop and the rotation about the pivot axis, with the pivot joint equality constraint represented by $\vartheta_{p_4} = \vartheta_{f'+1'}$. The orientation constraints as well as the positional constraints corresponding to $\mathbf{p}_y$ from Eq. 2.3.22 and 2.3.23 are again dependent and can be ignored, leaving only the positional constraints corresponding to $\mathbf{p}_x$ and $\mathbf{p}_z$. Together with the pivot axis equality, the total number of constraint equations for the coupling between legs 2 and 4 is therefore also five.

Forward Kinematics Summary

Each of the subsections of the system has now been represented by means of coordinate frames using a modified DH convention, and all closed chains have been virtually cut open, with constraint equations introduced for representing the closing conditions. This results in a tree which begins at the world frame w and moves along several different serial branches to their open chain ends. While normally the constraint equations would be solved for a reduced number of independent joint variables in order to obtain a full forward kinematics solution, the number of independent constraint equations and forward kinematic transformations present in the proposed design makes this process quite complicated. Additionally, the scope of this work does not require such a solution since the ultimate goal is to demonstrate the robot's ability to achieve desired task space trajectories, a problem which can be solved instead with inverse kinematics. It is however necessary for the forward kinematic transformations and constraint equations to be defined for each open chain branch since these are required for the inverse kinematics, which relies on numerical optimization of a system of equations, as is discussed in the next section.

2.3.2. Inverse Kinematics

While the forward kinematics equations presented in Chapter 2.3.1 provide mappings between the joint values and the position and orientation of each branch's end frame, the computation of joint values given the pose of a particular end frame is of more interest. Motion control schemes for legged locomotion typically involve providing desired foot and/or body trajectories which are used to compute the necessary joint states to be commanded by the controller, which is a problem solved using inverse kinematics. It is common practice for quadruped designs to consist of four independent and identical legs which are connected to a central floating base body, meaning that the computation of the inverse kinematics can be performed for each leg to independently solve for the positions of each foot relative to the body. Moreover, the kinematic structure for such quadrupeds is typically quite simple, allowing for straightforward analytical computation of the inverse kinematics based on knowledge of the geometry. The kinematic leg couplings present in the proposed quadruped complicate the process of defining the inverse kinematics in an analytical way, since the legs are not kinematically independent from one another and the many passive joints in the system depend on the closed-loop and ground contact constraint equations to be fully defined. Additionally, the non-linear nature of the forward kinematics equations means that a closed-form solution may not always exist or may be difficult to find (Siciliano et al. 2009). For these reasons, it is easier to adopt a numerical approach for solving the inverse kinematics problem, where the target equations are reformulated as an optimization problem to find suitable joint values. Specifically, non-linear least squares fitting is an optimal optimization method due to the non-linearity of the functions and is set up as follows:

$$\mathbf{q}^* = \arg\min_{\mathbf{q}} \sum_{i=1}^{m} \|\mathbf{e}_i(\mathbf{q})\|^2 \tag{2.3.24}$$

where $\mathbf{q}$ denotes the joint state vector, $\mathbf{e}_i(\mathbf{q})$ denotes a target error function which is to be minimized and m is the number of individual error functions to be minimized simultaneously. The optimization goal is therefore to determine the values for $\mathbf{q}$ which result in a minimization of the sum of squared error functions.
Before setting up the optimization problem, it is important to know the size of $\mathbf{q}$, as the number of independent equations to solve for should be equal to the number of joint variables to have a fully-determined system of equations. Including the six virtual floating base joints, there are 65 total joints, however the kinematic representation ignores the joints at the closed chain cutting points for the planar loops, leaving a total of 57 joints, so $\mathbf{q} \in \mathbb{R}^{57\times 1}$. A detailed list of all of the joint variables included in $\mathbf{q}$ is provided in Appendix C. In order to create a fully-determined system of equations to solve for, there must then be at least 57 independent equations which represent the entire kinematics of the system, all of which have already been defined in Chapter 2.3.1.
The first set of equations represent the positions of the four feet w.r.t the world frame. This is a primary goal in the inverse kinematics problem since locomotion tasks require computation of the joint values from desired foot position values. The equations defining the foot position values can be extracted from the homogeneous transformation matrices, represented as follows:

$$\mathbf{T}^w_{j_n+3}(\mathbf{q}) = \begin{bmatrix} \mathbf{R}^w_{j_n+3}(\mathbf{q}) & \mathbf{p}^w_{j_n+3}(\mathbf{q}) \\ \mathbf{0}^T & 1 \end{bmatrix} \forall n \in \{1,2,3,4\} \tag{2.3.25}$$

where $\mathbf{R}^w_{j_n+3}$ denotes the rotation of frame j_n+3 w.r.t. frame w and $\mathbf{p}^w_{j_n+3}(\mathbf{q})$ denotes the position of frame j_n+3 w.r.t. frame w. The desired foot positions w.r.t. the world frame can then be represented by $\mathbf{x}_n \in \mathbb{R}^{3\times 1}$. For some given $\mathbf{x}_n$, it is desired that the error between $\mathbf{x}_n$ and $\mathbf{p}^w_{j_n+3}(\mathbf{q})$ is zero, or as close to zero as possible, so the foot position equations can naturally be reformulated into 12 error functions represented as follows:

$$\mathbf{e}_{f1}(\mathbf{q}) = \mathbf{p}^w_{j_1+3}(\mathbf{q}) - \mathbf{x}_1 \tag{2.3.26}$$

$$\mathbf{e}_{f2}(\mathbf{q}) = \mathbf{p}^w_{j_2+3}(\mathbf{q}) - \mathbf{x}_2 \tag{2.3.27}$$

$$\mathbf{e}_{f3}(\mathbf{q}) = \mathbf{p}^w_{j_3+3}(\mathbf{q}) - \mathbf{x}_3 \tag{2.3.28}$$

$$\mathbf{e}_{f4}(\mathbf{q}) = \mathbf{p}^w_{j_4+3}(\mathbf{q}) - \mathbf{x}_4 \tag{2.3.29}$$

The next set of equations represent the position of the base frame b w.r.t. the world frame w as defined by Eq. 2.3.3 and are formulated in the same way as the foot position equations. If $\mathbf{x}_b$ is used to represent the desired base frame position and $\mathbf{p}^w_b$ to represent the translational part of the transformation matrix from frame w to b, then 3 more error functions can be defined as:

$$\mathbf{e}_b(\mathbf{q}) = \mathbf{p}_b^w(\mathbf{q}) - \mathbf{x}_b \tag{2.3.30}$$

Another two error functions can be obtained from a reformulation of the pivot joint equalities which were introduced to simplify the representation of the opposite leg couplings into planar mechanisms. These are defined as follows:

$$\mathbf{e}_{p1}(\mathbf{q}) = \vartheta_{f+1'} - \vartheta_{p_3} \tag{2.3.31}$$
$$\mathbf{e}_{p2}(\mathbf{q}) = \vartheta_{f'+1'} - \vartheta_{p_4} \tag{2.3.32}$$

The remaining equations must represent the constraints which were introduced in Chapter 2.3.1 for each of the formerly closed kinematic chains. The first set comes from the closed loop in the pantograph legs, defined by Eq. 2.3.6. It was previously determined that only the positional constraints corresponding to $\mathbf{p}_x$ and $\mathbf{p}_y$ are independent and necessary, so 8 error functions for the four pantographs can be represented as follows:

$$\mathbf{e}_{l1}(\mathbf{q}) = \begin{bmatrix} e_x(\mathbf{q}) \\ e_y(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{j_1}^{j_1+2}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{j_1+2,x}^{j_1}(\mathbf{q}) \\ \mathbf{p}_{j_1+2,y}^{j_1}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_1+3',x}^{j_1}(\mathbf{q}) \\ \mathbf{p}_{j_1+3',y}^{j_1}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.33}$$

$$\mathbf{e}_{l2}(\mathbf{q}) = \begin{bmatrix} e_x(\mathbf{q}) \\ e_y(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{j_2}^{j_2+2}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{j_2+2,x}^{j_2}(\mathbf{q}) \\ \mathbf{p}_{j_2+2,y}^{j_2}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_2+3',x}^{j_2}(\mathbf{q}) \\ \mathbf{p}_{j_2+3',y}^{j_2}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.34}$$

$$\mathbf{e}_{l3}(\mathbf{q}) = \begin{bmatrix} e_x(\mathbf{q}) \\ e_y(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{j_3}^{j_3+2}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{j_3+2,x}^{j_3}(\mathbf{q}) \\ \mathbf{p}_{j_3+2,y}^{j_3}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_3+3',x}^{j_3}(\mathbf{q}) \\ \mathbf{p}_{j_3+3',y}^{j_3}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.35}$$

$$\mathbf{e}_{l4}(\mathbf{q}) = \begin{bmatrix} e_x(\mathbf{q}) \\ e_y(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{j_4}^{j_4+2}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{j_4+2,x}^{j_4}(\mathbf{q}) \\ \mathbf{p}_{j_4+2,y}^{j_4}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_4+3',x}^{j_4}(\mathbf{q}) \\ \mathbf{p}_{j_4+3',y}^{j_4}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.36}$$

where no modifications to the originally provided constraint equations are necessary since they already represent the position errors between coordinate frames. The second set of constraint equations comes from the closed loop in the adjacent leg couplings, defined by Eq. 2.3.11. In this case, it was determined that all six constraints are independent and necessary to define the loops, therefore 24 error functions can be represented as follows:

$$\mathbf{e}_{a1,p}(\mathbf{q}) = \mathbf{R}_b^{k_1+6}(\mathbf{q}) \left(\mathbf{p}_{k_1+6}^b(\mathbf{q}) - \mathbf{p}_{k_2'+1}^b(\mathbf{q}) \right) \tag{2.3.37}$$

$$\mathbf{e}_{a1,o}(\mathbf{q}) = \mathbf{z}_{k_1+6}^b(\mathbf{q}) - \mathbf{z}_{k_2'+1}^b(\mathbf{q}) \tag{2.3.38}$$

$$\mathbf{e}_{a2,p}(\mathbf{q}) = \mathbf{R}_b^{k_2+6}(\mathbf{q}) \left(\mathbf{p}_{k_2+6}^b(\mathbf{q}) - \mathbf{p}_{k_3'+1}^b(\mathbf{q}) \right) \tag{2.3.39}$$

$$\mathbf{e}_{a2,o}(\mathbf{q}) = \mathbf{z}_{k_2+6}^b(\mathbf{q}) - \mathbf{z}_{k_3'+1}^b(\mathbf{q}) \tag{2.3.40}$$

$$\mathbf{e}_{a3,p}(\mathbf{q}) = \mathbf{R}_b^{k_3+6}(\mathbf{q}) \left(\mathbf{p}_{k_3+6}^b(\mathbf{q}) - \mathbf{p}_{k_4'+1}^b(\mathbf{q}) \right) \tag{2.3.41}$$

$$\mathbf{e}_{a3,o}(\mathbf{q}) = \mathbf{z}_{k_3+6}^b(\mathbf{q}) - \mathbf{z}_{k_4'+1}^b(\mathbf{q}) \tag{2.3.42}$$

$$\mathbf{e}_{a4,p}(\mathbf{q}) = \mathbf{R}_b^{k_4+6}(\mathbf{q}) \left(\mathbf{p}_{k_4+6}^b(\mathbf{q}) - \mathbf{p}_{k_1'+1}^b(\mathbf{q}) \right) \tag{2.3.43}$$

$$\mathbf{e}_{a4,o}(\mathbf{q}) = \mathbf{z}_{k_4+6}^b(\mathbf{q}) - \mathbf{z}_{k_1'+1}^b(\mathbf{q}) \tag{2.3.44}$$

The last set of constraint equations comes from the two opposite leg couplings, each of which has two closed loops. The constraint equations for the opposite coupling between legs 1 and 3 are given by Eq. 2.3.16 and 2.3.17, while the constraint equations for the opposite coupling between legs 2 and 4 are given by Eq. 2.3.22 and 2.3.23. Just as with the pantograph leg loops, it was determined that each of the opposite leg coupling loops can be viewed as planar, so only the positional constraints corresponding to $\mathbf{p}_x$ and $\mathbf{p}_z$ are independent and necessary. The constraint equations for the four opposite coupling loops can therefore be reformulated into 8 error functions as follows:

$$\mathbf{e}_{o1} = \begin{bmatrix} e_{x,o1}(\mathbf{q}) \\ e_{z,o1}(\mathbf{q}) \end{bmatrix} = \mathbf{R}_b^{f+2}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{f+2,x}^b(\mathbf{q}) \\ \mathbf{p}_{f+2,z}^b(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_1+2'',x}^b(\mathbf{q}) \\ \mathbf{p}_{j_1+2'',z}^b(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.45}$$

$$\mathbf{e}_{o2} = \begin{bmatrix} e_{x,o2}(\mathbf{q}) \\ e_{z,o2}(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{p_3}^{f+3'}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{f+3',x}^{p_3}(\mathbf{q}) \\ \mathbf{p}_{f+3',z}^{p_3}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_3+2'',x}^{p_3}(\mathbf{q}) \\ \mathbf{p}_{j_3+2'',z}^{p_3}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.46}$$

$$\mathbf{e}_{o3} = \begin{bmatrix} e_{x,o3}(\mathbf{q}) \\ e_{z,o3}(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{p_2}^{f'+2}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{f'+2,x}^{p_2}(\mathbf{q}) \\ \mathbf{p}_{f'+2,z}^{p_2}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_2+2'',x}^{p_2}(\mathbf{q}) \\ \mathbf{p}_{j_2+2'',z}^{p_2}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.47}$$

$$\mathbf{e}_{o4} = \begin{bmatrix} e_{x,o4}(\mathbf{q}) \\ e_{z,o4}(\mathbf{q}) \end{bmatrix} = \mathbf{R}_{p_4}^{f'+3'}(\mathbf{q}) \left(\begin{bmatrix} \mathbf{p}_{f'+3',x}^{p_4}(\mathbf{q}) \\ \mathbf{p}_{f'+3',z}^{p_4}(\mathbf{q}) \end{bmatrix} - \begin{bmatrix} \mathbf{p}_{j_4+2'',x}^{p_4}(\mathbf{q}) \\ \mathbf{p}_{j_4+2'',z}^{p_4}(\mathbf{q}) \end{bmatrix} \right) \tag{2.3.48}$$

With each of the individual target error functions defined, they can then be compiled into a single system of equations which is represented as follows:

$$\mathbf{e}(\mathbf{q}) = \begin{bmatrix} \mathbf{e}_{f1}(\mathbf{q}) \\ \mathbf{e}_{f2}(\mathbf{q}) \\ \mathbf{e}_{f3}(\mathbf{q}) \\ \mathbf{e}_{f4}(\mathbf{q}) \\ \mathbf{e}_{b}(\mathbf{q}) \\ \mathbf{e}_{p1}(\mathbf{q}) \\ \mathbf{e}_{p2}(\mathbf{q}) \\ \mathbf{e}_{l1}(\mathbf{q}) \\ \mathbf{e}_{l2}(\mathbf{q}) \\ \mathbf{e}_{l3}(\mathbf{q}) \\ \mathbf{e}_{l4}(\mathbf{q}) \\ \mathbf{e}_{a1,p}(\mathbf{q}) \\ \mathbf{e}_{a1,o}(\mathbf{q}) \\ \mathbf{e}_{a2,p}(\mathbf{q}) \\ \mathbf{e}_{a2,o}(\mathbf{q}) \\ \mathbf{e}_{a3,p}(\mathbf{q}) \\ \mathbf{e}_{a3,o}(\mathbf{q}) \\ \mathbf{e}_{a4,p}(\mathbf{q}) \\ \mathbf{e}_{a4,o}(\mathbf{q}) \\ \mathbf{e}_{o1}(\mathbf{q}) \\ \mathbf{e}_{o2}(\mathbf{q}) \\ \mathbf{e}_{o3}(\mathbf{q}) \\ \mathbf{e}_{o4}(\mathbf{q}) \end{bmatrix} \tag{2.3.49}$$

This system of error functions can then be passed to the non-linear least squares optimization problem from Eq. 2.3.24 to compute numerical solutions for $\mathbf{q}$. It is important to note that numerical optimization methods such as this one do not always result in unique solutions due to the possible existence of multiple local minima. To further ensure that solutions obtained for $\mathbf{q}$ are valid and physically feasible, the initial guess for $\mathbf{q}$ should be chosen wisely and joint limits can be introduced to the optimization problem to represent physical constraints of the system.

2.3.3. Differential Kinematics

While the forward and inverse kinematics problems provide functional relationships between the joint states and the end effector (feet and base) poses, motion control schemes typically rely instead on velocity or acceleration mappings, which can be defined with differential kinematics. The Jacobian matrix, represented by $\mathbf{J}(\mathbf{q})$, is a joint state-dependent matrix which provides a direct mapping from joint velocities to end effector velocities at a given configuration. This mapping is denoted as follows:

$$\mathbf{v}_e = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{2.3.50}$$

where $\dot{\mathbf{q}}$ represents the joint velocity vector and $\mathbf{v}_e$ represents the end effector linear and rotational velocity in the form $\mathbf{v}_e = \begin{bmatrix} \dot{\mathbf{p}}_e & \boldsymbol{\omega}_e \end{bmatrix}^T$. Since this velocity mapping takes

the form of a linear equation, it can be seen that when the Jacobian matrix is square and non-singular, the joint velocities can also be uniquely computed from given end effector velocities by simple inversion of $\mathbf{J}(\mathbf{q})$ as follows:

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q})\mathbf{v}_e \tag{2.3.51}$$

For fully-actuated serial manipulators, the formulation for differential kinematics can be quite straightforward, with a single Jacobian matrix of dimension $6 \times n$ mapping the joint velocities to a single end effector's velocities, where n denotes the number of joints in the kinematic chain. In the proposed quadruped, however, the formulation of the differential kinematics is complicated by the parallel structure and the presence of many passive joints. Since velocities cannot be commanded to the passive joints, the inverse differential kinematics problem should not compute velocities for the entire set of joints, but rather only for a subset consisting of the six active joints. A mapping between the active joint velocities and the passive joint velocities is then necessary and can be derived using the same system constraints which were defined in Chapter 2.3.1 and used in 2.3.2 to fully constrain the passive joints.
A general method for formulating the differential kinematics problem for parallel mechanisms was outlined by Lynch and Park (2017), where a so-called *constraint Jacobian* derived from the Jacobians defined by the constraint equations is introduced and used for computing both the active and passive joint velocities separately. Specifically, the method provides a way of formulating a square and invertible Jacobian matrix for computing the active joint velocities as well as a configuration-dependent mapping from active to passive joint velocities. It can first been seen that the closed-loop constraint equations, originally expressed in terms of each branch's end-effector frame position and orientation, can equally be expressed in terms of end-effector frame linear and rotational velocity using the Jacobian matrices for the branches. For example, a closed loop represented by the constraint equation $\mathbf{T}_a(\boldsymbol{\theta}_a) = \mathbf{T}_b(\boldsymbol{\theta}_b)$ would be expressed as follows:

$$\mathbf{J}_a(\boldsymbol{\theta}_a)\dot{\boldsymbol{\theta}}_{\boldsymbol{a}} = \mathbf{J}_b(\boldsymbol{\theta}_b)\dot{\boldsymbol{\theta}}_{\boldsymbol{b}} \tag{2.3.52}$$

where simple algebraic rearrangement yields:

$$\mathbf{J}_a(\boldsymbol{\theta}_a)\dot{\boldsymbol{\theta}}_{\boldsymbol{a}} - \mathbf{J}_b(\boldsymbol{\theta}_b)\dot{\boldsymbol{\theta}}_{\boldsymbol{b}} = 0 \tag{2.3.53}$$

which can also be represented as:

$$\begin{bmatrix}\mathbf{J}_a(\boldsymbol{\theta}_a) & -\mathbf{J}_b(\boldsymbol{\theta}_b)\end{bmatrix}\begin{bmatrix}\dot{\boldsymbol{\theta}}_{\boldsymbol{a}} \\ \dot{\boldsymbol{\theta}}_{\boldsymbol{b}}\end{bmatrix} = 0 \tag{2.3.54}$$

In the case that multiple constraint equations exist, they can simply be stacked vertically, forming a system of constraint equations expressed in terms of configuration-dependent Jacobians and joint velocities. Looking again at the previous example, if an additional constraint equation represented by $\mathbf{T}_c(\boldsymbol{\theta}_c) = \mathbf{T}_d(\boldsymbol{\theta}_d)$ is introduced, the new formulation is given by:

$$\begin{bmatrix} \mathbf{J}_a(\boldsymbol{\theta}_a) & -\mathbf{J}_b(\boldsymbol{\theta}_b) & 0 & 0 \\ 0 & 0 & \mathbf{J}_c(\boldsymbol{\theta}_c) & -\mathbf{J}_d(\boldsymbol{\theta}_d) \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\theta}}_a \\ \dot{\boldsymbol{\theta}}_b \\ \dot{\boldsymbol{\theta}}_c \\ \dot{\boldsymbol{\theta}}_d \end{bmatrix} = 0 \tag{2.3.55}$$

For the proposed design, all of the constraint equations were collected in Chapter 2.3.2 and it was seen that most of the closed loops resulted in reduced sets of independent constraints due to their geometrical features. In order to compile the constraint equations into a form such as that of Eq. 2.3.55, it is therefore necessary to isolate the independent equations from their Jacobian matrix representations since the Jacobian matrix, like the transformation matrix, represents six degrees of freedom (three positional and three orientational) in its complete form.
The first of the closed-loop constraint equation sets comes from the pantograph structure in the four legs, which was shown previously to be planar, with only the positional constraints corresponding to $\mathbf{p}_x$ and $\mathbf{p}_y$ being independent. If the Jacobian matrix is expressed by:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_P \\ \mathbf{J}_O \end{bmatrix} \tag{2.3.56}$$

where $\mathbf{J}_P = \begin{bmatrix} \mathbf{J}_{P_x} & \mathbf{J}_{P_y} & \mathbf{J}_{P_z} \end{bmatrix}^T$ represents the individual joints' contributions to the linear velocities at the end effector and $\mathbf{J}_O = \begin{bmatrix} \mathbf{J}_{O_x} & \mathbf{J}_{O_y} & \mathbf{J}_{O_z} \end{bmatrix}^T$ represents their contributions to the angular velocities at the end effector, then the constraint equations for the pantograph closed loops expressed in terms of the Jacobians can be formulated as follows:

$$\begin{bmatrix} \tilde{\mathbf{p}}_0 \left(\mathbf{J}^{j_1}_{j_1+2}(\mathbf{q}) - \mathbf{J}^{j1}_{j_1+3'}(\mathbf{q}) \right) \\ \tilde{\mathbf{p}}_0 \left(\mathbf{J}^{j_2}_{j_2+2}(\mathbf{q}) - \mathbf{J}^{j2}_{j_2+3'}(\mathbf{q}) \right) \\ \tilde{\mathbf{p}}_0 \left(\mathbf{J}^{j_3}_{j_3+2}(\mathbf{q}) - \mathbf{J}^{j3}_{j_3+3'}(\mathbf{q}) \right) \\ \tilde{\mathbf{p}}_0 \left(\mathbf{J}^{j_4}_{j_4+2}(\mathbf{q}) - \mathbf{J}^{j4}_{j_4+3'}(\mathbf{q}) \right) \end{bmatrix} \dot{\mathbf{q}} = 0 \tag{2.3.57}$$

where $\mathbf{J}^i_j$ represents the Jacobian matrix for the kinematic branch ending with frame j, expressed in frame i, and

$$\tilde{\mathbf{p}}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.3.58}$$

allows the selection of only the first and second rows of each set of Jacobian constraint equations, which correspond to the linear velocities of the each end frame in x_{j_n} and y_{j_n}.
The next set of closed-loop constraint equations comes from the four adjacent leg couplings, which were shown before to each introduce six independent equations. This means that no isolation of rows from the Jacobians is necessary and the constraint equations can therefore be expressed in terms of the Jacobians as follows:

$$\begin{bmatrix} \mathbf{J}^b_{k_1+6}(\mathbf{q}) - \mathbf{J}^b_{k'_2+1}(\mathbf{q}) \\ \mathbf{J}^b_{k_2+6}(\mathbf{q}) - \mathbf{J}^b_{k'_3+1}(\mathbf{q}) \\ \mathbf{J}^b_{k_3+6}(\mathbf{q}) - \mathbf{J}^b_{k'_4+1}(\mathbf{q}) \\ \mathbf{J}^b_{k_4+6}(\mathbf{q}) - \mathbf{J}^b_{k'_1+1}(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = 0 \tag{2.3.59}$$

The last set of closed-loop constraint equations comes from the two opposite leg couplings, each of which was shown previously to consist of two separate closed loops which are represented as planar for the purpose of deriving the inverse kinematics. This representation depends on the introduction of two joint equalities for the aligned pivot joints, however the differential kinematics representation can be formulated without the need for these equalities. The opposite coupling between legs 1 and 3 begins with frame b and its first closed loop consists of only joints with parallel axes, so the representation of its closing conditions with two independent constraint equations (corresponding to linear motion in x_b and z_b) remains unchanged from the inverse kinematics representation. The second closed loop in the coupling between legs 1 and 3 has the same planar structure as the first loop, but includes a single pivot joint in each branch, both of which are aligned such that they create motion in y_b at the respective branch's end frame. All three of the positional constraint equations are therefore independent, but the orientational constraints can be seen as dependent since the orientation of each branch's end frame is fully defined by its position relative to frame b. The same logic applies to the two closed loops in the opposite coupling between legs 2 and 4, with the reference frame for both loops being frame p_2 instead of frame b. Each of the opposite leg couplings therefore yields five constraint equations, all of which can be expressed in terms of their Jacobian matrices as follows:

$$\begin{bmatrix} \tilde{\mathbf{p}}_1 \left(\mathbf{J}^b_{f+2}(\mathbf{q}) - \mathbf{J}^b_{j_1+2''}(\mathbf{q}) \right) \\ \tilde{\mathbf{p}}_2 \left(\mathbf{J}^b_{f+3'}(\mathbf{q}) - \mathbf{J}^b_{j_3+2''}(\mathbf{q}) \right) \\ \tilde{\mathbf{p}}_1 \left(\mathbf{J}^{p_2}_{f'+2}(\mathbf{q}) - \mathbf{J}^{p_2}_{j_2+2''}(\mathbf{q}) \right) \\ \tilde{\mathbf{p}}_2 \left(\mathbf{J}^{p_2}_{f'+3'}(\mathbf{q}) - \mathbf{J}^{p_2}_{j_4+2''}(\mathbf{q}) \right) \end{bmatrix} \dot{\mathbf{q}} = 0 \tag{2.3.60}$$

where the pre-multiplied matrices $\tilde{\mathbf{p}}_1$ and $\tilde{\mathbf{p}}_2$ facilitate selection of the corresponding independent rows of the Jacobian matrices and are defined as:

$$\tilde{\mathbf{p}}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.3.61}$$

$$\tilde{\mathbf{p}}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.3.62}$$

Aside from the closed-loop constraint equations, the contact points between the feet and the ground must also be represented by constraint equations in the differential kinematics problem. In the inverse kinematics representation, the positions of all four feet are provided to the numerical solver in the form of error "constraint" equations,

however static walking tasks involve ground contact of only three feet at any given time, thus the position of the swing foot, $\mathbf{p}^w_{j_n+3}$, is not truly a constraint but rather a goal which should be provided to the system as input. The same is true about the position of the base frame, $\mathbf{p}^w_b$, which was also provided to the inverse kinematics solver as an error "constraint". These six variables are therefore left out of the matrix of constraint equations in the differential kinematics representation and instead set aside for use later as system inputs. The formulation of the ground contact constraints then depends on which foot is chosen as swing foot, with the other three maintaining constant contact with the ground and thus having zero velocity. The nine positional constraints of the contact feet can therefore be expressed in terms of the Jacobians of their respective branches as follows:

$$\begin{bmatrix} \tilde{\mathbf{p}}_2 \mathbf{J}^w_{j_r+3}(\mathbf{q}) \\ \tilde{\mathbf{p}}_2 \mathbf{J}^w_{j_s+3}(\mathbf{q}) \\ \tilde{\mathbf{p}}_2 \mathbf{J}^w_{j_t+3}(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} = 0 \tag{2.3.63}$$

where $\{r, s, t\}$ is the set of feet which are in contact with the ground.
All of the defined constraint equations can then be stacked into a single matrix $\mathbf{H}(\mathbf{q})$, which is named the *constraint Jacobian*, and the resulting linear equation is given by:

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{q}} = 0 \tag{2.3.64}$$

where $\mathbf{H}(\mathbf{q}) \in \mathbb{R}^{51\times 57}$ and $\dot{\mathbf{q}} \in \mathbb{R}^{57\times 1}$. It is then possible to rearrange the joint variables such that the active and passive joints are separated by simply changing the order of the columns in the constraint Jacobian together with their corresponding rows in $\dot{\mathbf{q}}$. This rearranged representation is given by:

$$\begin{bmatrix} \mathbf{H}_a(\mathbf{q}) & \mathbf{H}_p(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_p \end{bmatrix} = 0 \tag{2.3.65}$$

which can equivalently represented as:

$$\mathbf{H}_a(\mathbf{q})\dot{\mathbf{q}}_a + \mathbf{H}_p(\mathbf{q})\dot{\mathbf{q}}_p = 0 \tag{2.3.66}$$

where $\mathbf{H}_a(\mathbf{q}) \in \mathbb{R}^{51\times 6}$, $\mathbf{H}_p(\mathbf{q}) \in \mathbb{R}^{51\times 51}$, $\dot{\mathbf{q}}_a \in \mathbb{R}^{6\times 1}$ and $\dot{\mathbf{q}}_p \in \mathbb{R}^{51\times 1}$. It can be seen that the matrix $\mathbf{H}_p(\mathbf{q})$ is square, so if it also has full rank and is non-singular, it can be inverted and Eq. 2.3.66 can be reformulated as:

$$\dot{\mathbf{q}}_\mathbf{p} = -\mathbf{H}_p^{-1}(\mathbf{q})\mathbf{H}_a(\mathbf{q})\dot{\mathbf{q}}_a \tag{2.3.67}$$

This equation provides a unique solution for the passive joint velocities $\dot{\mathbf{q}}_p$ if the actuated joint velocities $\dot{\mathbf{q}}_a$ are known. It is therefore still necessary to compute the actuated joint velocities for some given end effector velocities, which can be denoted by the vector $\mathcal{V}_s$. Recalling that the swing foot and base frame positions were set aside from the constraint Jacobian formulation to be used as system inputs, the linear velocities of these two frames can be directly stacked to form $\mathcal{V}_s \in \mathbb{R}^{6\times 1}$. It is then desired to derive the Jacobian matrix corresponding to the active joint velocities, $\mathbf{J}_a(\mathbf{q}) \in \mathbb{R}^{6\times 6}$, which satisfies the forward differential kinematics equation $\mathcal{V}_s = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_a$. Looking back to

Eq. 2.3.67, each row i of the mapping $-\mathbf{H}_p^{-1}(\mathbf{q})\mathbf{H}_a(\mathbf{q})$ can be denoted by $\mathbf{g}_i^T(\mathbf{q}) \in \mathbb{R}^{1\times 6}$, and the equation can be reformulated to represent a mapping between the active joint velocities and all joint velocities by inserting row vectors $\mathbf{e}_i^T(\mathbf{q})$ for $i = 1, ..., 6$ into the rows corresponding to the actuated joints, where $\mathbf{e}(\mathbf{q})$ is the identity matrix I_6. This new mapping can be denoted by $\mathbf{G}(\mathbf{q}) \in \mathbb{R}^{57\times 6}$ and yields the following equation, with the full representation of the mapping provided in Appendix D.

$$\dot{\mathbf{q}} = \mathbf{G}(\mathbf{q})\dot{\mathbf{q}}_a \tag{2.3.68}$$

Considering that the vector $\mathcal{V}_s$ denotes the stacked linear velocities of the swing foot frame and the base frame, the Jacobian matrix in the full forward differential kinematics equation $\mathcal{V}_s = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ must be represented by the stacked positional contribution rows of the Jacobian matrices corresponding to the kinematic chains of the swing foot and base frame, with the full equation then given as:

$$\mathcal{V}_s = \begin{bmatrix} \tilde{\mathbf{p}}_2\mathbf{J}^w_{j_n+3}(\mathbf{q}) \\ \tilde{\mathbf{p}}_2\mathbf{J}^w_b(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} \tag{2.3.69}$$

It is then possible to substitute Eq. 2.3.68 into Eq. 2.3.69 as follows:

$$\mathcal{V}_s = \begin{bmatrix} \tilde{\mathbf{p}}_2\mathbf{J}^w_{j_n+3}(\mathbf{q}) \\ \tilde{\mathbf{p}}_2\mathbf{J}^w_b(\mathbf{q}) \end{bmatrix} \mathbf{G}(\mathbf{q})\dot{\mathbf{q}}_a \tag{2.3.70}$$

where a representation for $\mathbf{J}_a(\mathbf{q})$ can be given by:

$$\mathbf{J}_a(\mathbf{q}) = \begin{bmatrix} \tilde{\mathbf{p}}_2\mathbf{J}^w_{j_n+3}(\mathbf{q}) \\ \tilde{\mathbf{p}}_2\mathbf{J}^w_b(\mathbf{q}) \end{bmatrix} \mathbf{G}(\mathbf{q}) \tag{2.3.71}$$

Since the actuated joint Jacobian $\mathbf{J}_a(\mathbf{q})$ is square, if it has full rank and is non-singular, it can be inverted to obtain the inverse differential kinematics equation as follows:

$$\dot{\mathbf{q}}_a = \mathbf{J}_a(\mathbf{q})^{-1}\mathcal{V}_s \tag{2.3.72}$$

This equation can be used to obtain unique solutions for the six actuator velocities given some desired reference velocities of the swing foot and base frame, with the passive joint velocities derived by simply plugging the actuator velocities into Eq. 2.3.67. It is important to note that in some cases the active joint Jacobian $\mathbf{J}_a(\mathbf{q})$ could become singular, leading to singularities and making inversion and computation of $\dot{\mathbf{q}}_a$ impossible, but the implementation of physical joint limits can easily prevent the system from going near singular poses.

2.3.4. Inverse Kinematics Control

With the inverse differential kinematics equation derived in the previous chapter, it is possible to compute actuator velocities from given swing foot and base frame linear velocities at an instance in discrete time. Since the goal is to show that a static walking gait with arbitrary foot placement is feasible, a simple inverse kinematic controller can be introduced which takes a desired trajectory for the swing foot and

base frames as input and computes the joint positions and velocities at each time step using the inverse differential kinematics evaluation from the previous time step. Because the computations are performed in discrete time, it is possible for drift to occur, whereby some error between the computed positions of the end effector frames and their desired positions exists. The size of this error depends largely on the size of the discrete time steps and the given reference velocities, namely larger time steps and reference velocities generally lead to larger errors (Siciliano et al. 2009). The error equation can be represented by:

$$\mathbf{e} = \mathbf{x}_d - \mathbf{k}(\mathbf{q}) \tag{2.3.73}$$

where $\mathbf{x}_d \in \mathbb{R}^{6\times 1}$ denotes the desired position of the swing foot and the base frame, and $\mathbf{k}(\mathbf{q})$ represents the forward kinematics computation for the swing foot and base frame given some joint configuration $\mathbf{q}$. This error should be taken into account by the inverse kinematics algorithm in such a way that the system is asymptotically stable, i.e. the error tends towards zero over the reference trajectory, so the following representation is adopted:

$$\dot{\mathbf{q}}_a = \mathbf{J}_a^{-1}(\mathbf{q})(\dot{\mathbf{x}}_d + \mathbf{K}\mathbf{e}) \tag{2.3.74}$$

where $\mathbf{K} \in \mathbb{R}^{6\times 6}$ denotes a positive-definite gain matrix whose eigenvalues determine how fast the error $\mathbf{e}$ converges to zero, namely larger values lead to faster convergence, with an upper limit determined by the discrete step size. A schematic view of the inverse kinematic algorithm is shown in Fig. 2.27, where it can be seen that the differential kinematics computations discussed in the previous section for deriving both the constraint Jacobian mapping between $\dot{\mathbf{q}}_a$ and $\dot{\mathbf{q}}$ and the actuated joint Jacobian $\mathbf{J}_a^{-1}(\mathbf{q})$ are necessary for calculating the relevant parameters. Since the differential kinematics solver produces joint velocities $\dot{\mathbf{q}}$ as output, an integration step is included to derive the joint positions $\mathbf{q}$ which are used in evaluation of the next time step. This integration is performed in discrete time as follows:

$$\mathbf{q}(t_{k+1}) = \mathbf{q}(t_k) + \dot{\mathbf{q}}(t_k)\Delta t \tag{2.3.75}$$

where Δt denotes the length of each discrete time step and $\dot{\mathbf{q}}(t_k)$ is computed via inverse differential kinematics.

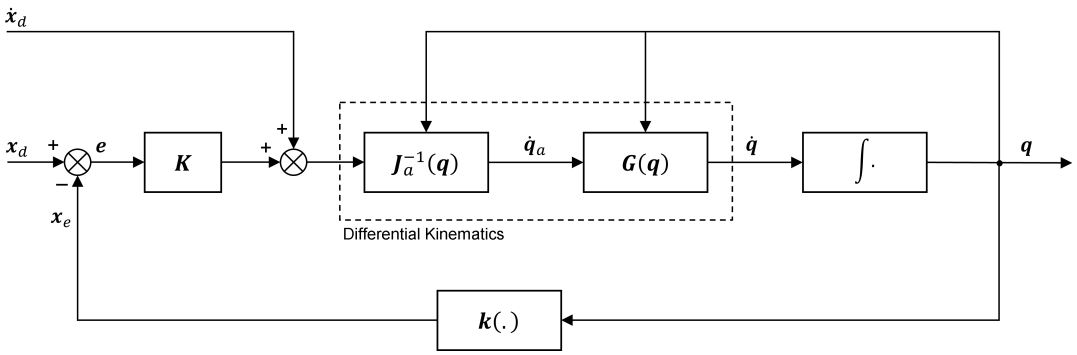


Figure 2.27.: Schematic view of the simple inverse kinematics control algorithm

It can be seen that the first iteration of the control loop already requires the joint state $\mathbf{q}$ to be defined in order to compute $\mathbf{J}_a(\mathbf{q})$, $\mathbf{G}(\mathbf{q})$ and $\mathbf{k}(\mathbf{q})$. These initial values can be conveniently calculated using the numerical inverse kinematics solver from Chapter 2.3.2. It then only remains to select appropriate values for the gain matrix K which lead to convergence of the position error and thus stable feedback control. In the case of this discrete time model, the values can be simply selected experimentally based on the chosen time step.

3

Testing and Results

This chapter outlines a software implementation of the proposed quadruped which was utilized to perform several tests in a simulated environment and collect data which is presented and discussed. Specifically, a CAD model of the quadruped was created in Creo Parametric and exported to URDF using Automatic Model Generation, then an inverse kinematic control algorithm was implemented in Matlab using the LucaDynamics library and a few static walking tests were performed to show that the desired system goals have been achieved.

3.1. Model Generation

The first step in implementation of the proposed design was to set up a CAD model which could realistically represent the kinematic structure and provide insight into its physical limitations. This was done using PTC Creo Parametric 7, which serves as the standard CAD modeling software at DLR's Robotics and Mechatronics Center (RMC). The software offers the ability to create so-called motion skeletons, which function as basic representations of kinematic structures using 2D sketches connected by motion joints. This feature allowed for quick modeling of the proposed concepts and testing of their motions before proceeding to a more detailed solid modeling approach. Once the final proposed model was implemented in the form of a motion skeleton, simple solid parts were designed to represent each of the links and assembled onto the motion skeleton model. For the purpose of testing the overall kinematic structure, the precise values of the link lengths were not critical, so they were initially selected arbitrarily. Coordinate frames were then attached to each of the model's parts according to the modified DH convention discussed and shown in Chapter 2.3.1. A rendered view of the solid 3D Creo model can be seen in Fig. 3.1, with additional views provided in Appendix E. Additionally, a full table of the DH parameter values used in the model is provided in Appendix F.
The next step was to export the Creo model to URDF using Automatic Model Generation, a tool developed at the RMC for rapid conversion of Creo configuration files into URDF or SDF files which can be used for simulation and control purposes. A few manual modifications to the resulting URDF file were then necessary to represent the model as desired. These changes included:

1. Addition of a world reference link
2. Addition of a ground link represented by a rectangular box visual geometry object
3. Change of joint type between ground link and Basis link from "fixed" to "floating"
4. Removal of joint connections for all closed-chain cutting joints
5. Addition and labeling of TCP links/joints for all branches

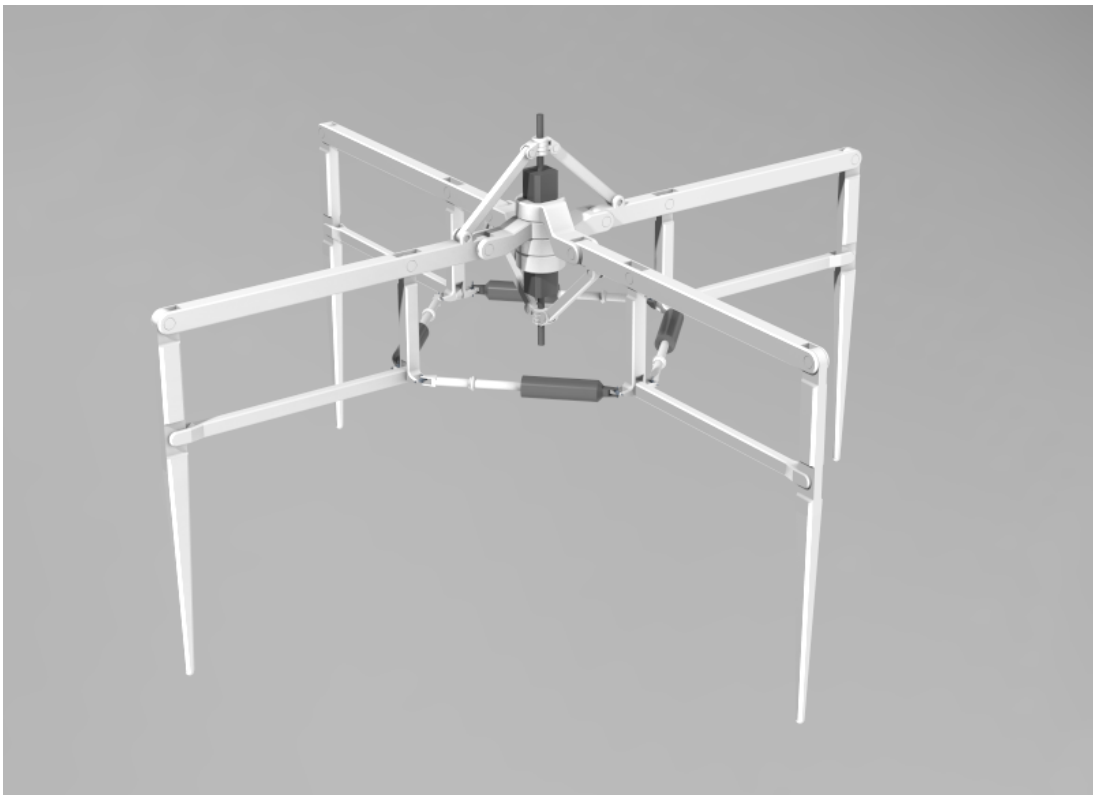


Figure 3.1.: Rendered image of the quadruped CAD model

3.2. MATLAB Implementation

With the URDF file modified to match the desired robot representation, it was then possible to implement the inverse kinematic control algorithm in MATLAB with the help of LucaDynamics, a compact MATLAB library developed at the RMC which performs computations for quantities used in rigid body robotics such as homogeneous transformations, Jacobian matrices and mass/Coriolis matrices. The library also features a 3D visualization tool similar to RViz from ROS which can be used to simulate the URDF file in different published joint configurations. As mentioned previously, the inverse kinematic control algorithm requires an initial joint state for its first iteration which can be computed using the numerical inverse kinematics solver, so this was naturally the chosen starting point for the implementation in MATLAB.

3.2.1. Forward/Inverse Kinematics Solvers

A function was first created for symbolically computing the forward kinematic transformations for each of the open-chain branches in the kinematic tree using the modified DH convention with the parameters from Chapter 2.3.1 and Appendix F. A separate inverse kinematics function was then created which takes the resulting symbolic forward kinematic transformations along with the desired feet and base positions and an initial guess for the joint states as input arguments. This function then computes and compiles the relevant error functions and utilizes MATLAB's nonlinear least-squares solver `lsqnonlin` to calculate a solution for the joint states.
It was desired that the joint configuration corresponding to the default URDF robot pose shown in Fig. 3.1 be considered the home configuration, where all joint state values are equal to zero, so as to provide an easy-to-define starting pose. Since many of the joint values according to the modified DH convention are not equal to zero in this configuration, it was necessary to compensate these variables in the symbolic forward kinematics function with the true joint values obtained from the URDF file. These compensated values are provided in Appendix G. With the joints compensated, it was then possible to test the inverse kinematic solver's ability to admit feasible solutions using a simple algorithm given as follows:

Algorithm 3.1.: Simple inverse kinematics path following

Require: H: symbolic forward kinematic transformations, q: initial guess for q, $p_{initial}$: initial feet and base position, p_{end}: final feet and base position, h: foot lifting height, T: total time, dt: timestep

function GENERATETRAJECTORY($p_{initial}, p_{end}, h, T, dt$)
  return t, p ▷ t: time vector, p: feet/base position trajectory
for length of t **do**
  function INVERSEKINEMATICS(p, q, H)
    collect error functions f_e
    q ← LSQNONLIN(f_e, q) ▷ q: nonlinear least squares solution
  compute swing foot position H_f ▷ using LucaDynamics
  $e \leftarrow p - H_f$ ▷ error between desired & actual foot position
  save q and e for plotting

Because the solver is only used for computing the initial joint configuration for inverse kinematic control, it was tested on only a basic single-leg lifting task. The robot was commanded to start in its home configuration, lift its first foot and move it vertically to a given height, then place it back in the original position, with the robot base position staying stationary. To measure the accuracy of the resulting joint trajectories, the Euclidean distance error between the desired and actual swing foot position was computed for each time step. Likewise, the same measurement was computed for the base frame position. Fig. 3.2 shows the evolution of the swing foot and base frame position errors for the given trajectory, where it can be seen that the swing foot error remains quite small throughout the computed trajectory, reaching a maximum value of

only 0.032 cm but remaining below 0.01 cm for the majority of the computed trajectory. The base frame error, on the other hand, reached a maximum of only 0.006 cm and remained at virtually zero for most of the trajectory. Additionally, it can be seen in Fig. 3.3 that the resulting actuated joint trajectories follow smooth and continuous paths, further proving that the proposed system is at least capable of continuous motion along the arbitrarily provided trajectory. It must be noted however that the numerical inverse kinematics solver is not guaranteed to result in a feasible solution and might depend heavily on the provided initial joint conditions to reach a local minimum. The inverse kinematics solution also does not distinguish between active and passive joints, instead computing position values for all joints simultaneously, which in practical terms is not useful unless all the joints are individually actuated. This approach therefore does not help in proving that control of the system is possible with only the six actuated joints, but gives some useful insight into the feasibility of its motion with all of the introduced constraints.

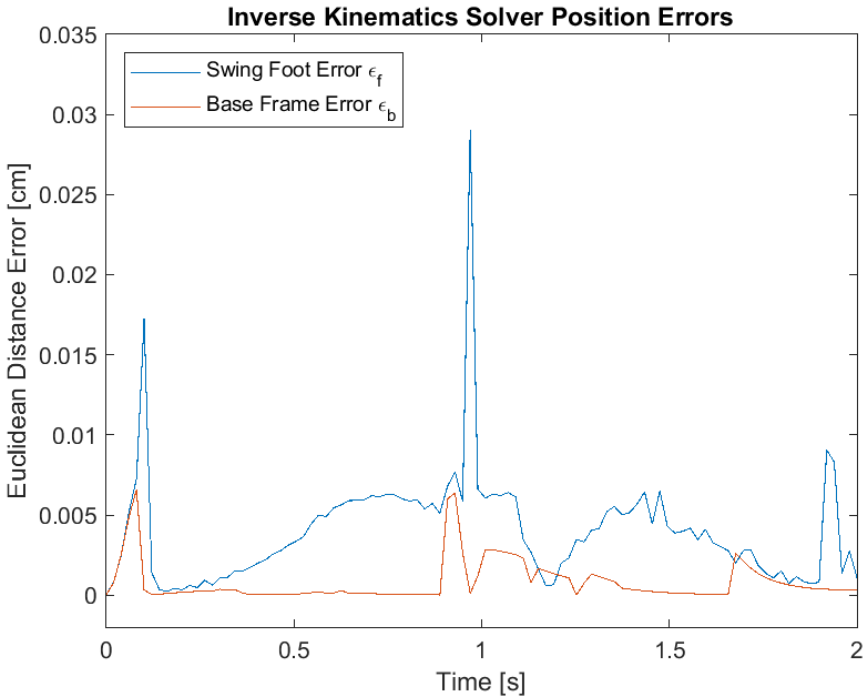


Figure 3.2.: Evolution of swing foot and base frame position errors for a simple joint trajectory computed with the numerical inverse kinematic solver

3.2.2. Trajectory Planner

An important part of quadruped locomotion is the ability to define desired trajectories which should be followed by the feet and/or body, namely reference positions and velocities parameterized by time. The level of complexity in chosen planning methods for such trajectories varies greatly, with each depending on the specific locomotion task for which it is used. For example, Wang et al. (2017) proposed a planning method which minimizes fluctuations of the center of mass in the z direction and Chen, Li, and

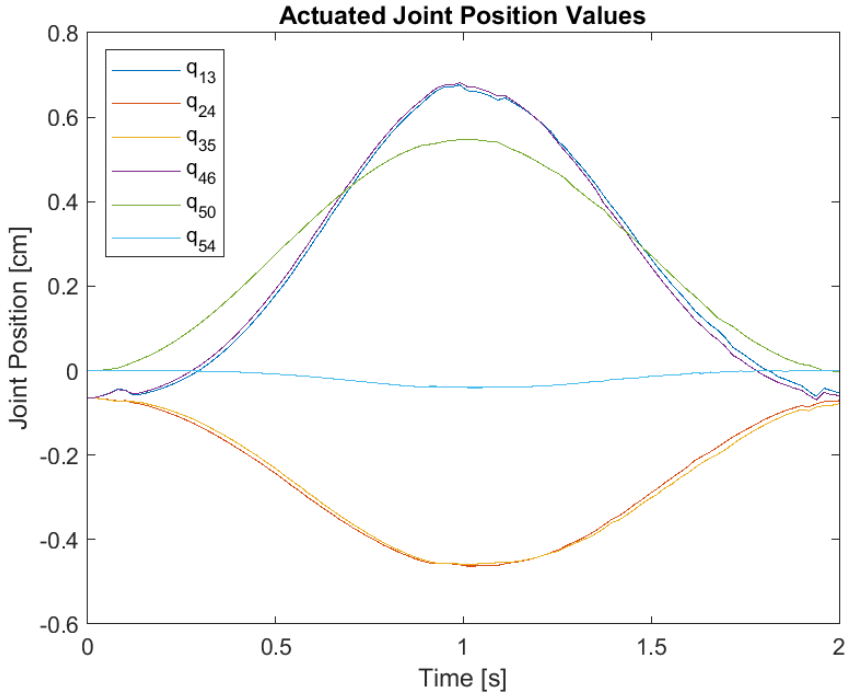


Figure 3.3.: Evolution of the actuated joint positions for a simple joint trajectory computed with the numerical inverse kinematics solver

Huang (2014) proposed swing leg and center of mass trajectories for stable crawling based on Double-Support Triangle (DST) theory. Since the focus of this thesis is on the kinematic design, namely the capability of the proposed model to achieve a basic static crawl, a complicated method for stable center of mass trajectory planning is not necessary or of interest. It is however still necessary to define a method for defining simple point-to-point trajectories for the feet and base which can be provided to the inverse kinematic control algorithm to compute the required actuator velocities and produce walking motions.
A simple trajectory planner based on the one used by Ma, Tomiyama, and Wada (2005) was therefore employed, with individual trajectories being defined by a cosine function with maximum height h which connects a start position $\mathbf{p}_{start}$ to an end position $\mathbf{p}_{end}$ on the ground plane. The reference positions and velocities are computed as follows:

$$\begin{aligned}
\mathbf{p}_{ref} &= \begin{bmatrix} \frac{p_{start,x}-p_{end,x}}{2}\cos\left(\frac{\pi}{T}\mathbf{t}\right) \\ \frac{p_{start,y}-p_{end,y}}{2}\cos\left(\frac{\pi}{T}\mathbf{t}\right) \\ \frac{h}{2}\left(1-\cos\left(\frac{2\pi}{T}\mathbf{t}\right)\right) \end{bmatrix} + \frac{1}{2}\left(\mathbf{p}_{start}+\mathbf{p}_{end}\right) \\
\dot{\mathbf{p}}_{ref} &= \begin{bmatrix} -\frac{1}{2}\left(p_{start,x}-p_{end,x}\right)\frac{\pi}{T}\sin\left(\frac{\pi}{T}\mathbf{t}\right) \\ -\frac{1}{2}\left(p_{start,y}-p_{end,y}\right)\frac{\pi}{T}\sin\left(\frac{\pi}{T}\mathbf{t}\right) \\ h\frac{\pi}{T}\sin\left(\frac{2\pi}{T}\mathbf{t}\right) \end{bmatrix}
\end{aligned} \tag{3.2.1}$$

where T denotes the total time of the desired motion and t is a vector of time stamps. For the purpose of demonstrating simple static walking, it was also sufficient to use this same method for planning the trajectories of the robot's base frame, with the value

of h simply being set to zero so that the body only moves along the $x-y$ plane. A visual representation of the type of trajectory used here can be seen in Fig. 3.4.

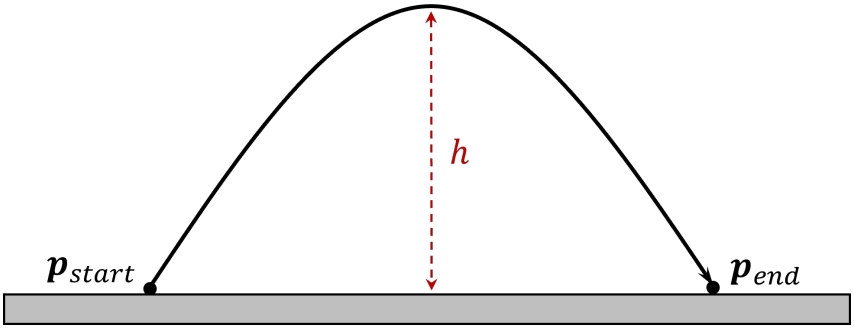


Figure 3.4.: A simple cosine function foot trajectory

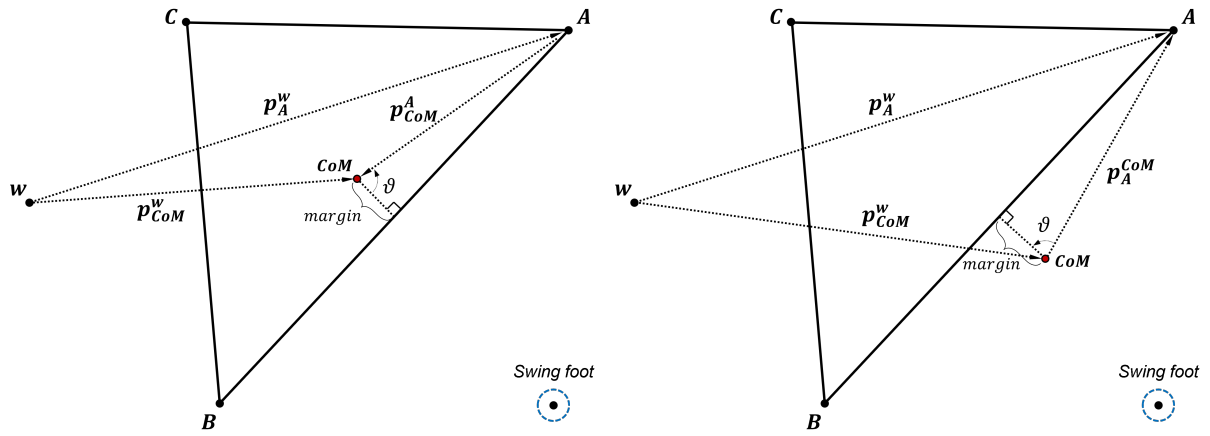
The static walking locomotion gait involves moving a single foot at a time, with the remaining three feet remaining planted on the ground. The contact points of these three feet form the support polygon, which determines a region of static stability for the center of mass. As was discussed previously, the proposed design features common drive actuation whereby the selection of the swing foot is determined by the location of the center of mass, therefore for certain desired leg lifting tasks it is necessary for the center of mass to be shifted within the respective support polygon. Having the center of mass situated inside a support polygon is alone not enough to ensure successful motion since a position very close to the support line could result in low stability. It is therefore a common approach to use a measure of the *stability margin*, which is based on the distance of the center of mass from the support line, when determining movements of the center of mass. A planning method was implemented in MATLAB which calculates the Euclidean distance from the center of mass to the support line, determines if the CoM is within the support polygon, then calculates a vector orthogonal to the support line pointing in the direction of increasing stability. The robot's base frame is then moved in this direction until the desired margin of stability is reached. This scheme is executed before each foot placement task, with the support polygon determined by the desired swing leg.
Referring to Fig. 3.5, the margin of stability is computed as follows:

$$margin = |\frac{mp_{CoM,x} - p_{CoM,y} + c}{\sqrt{m^2 + b^2}}| \tag{3.2.2}$$

where

$$m = \frac{p_{B,y} - p_{A,y}}{p_{B,x} - p_{A,x}}$$
$$c = p_{A,y} - mp_{A,x}$$

and where $\mathbf{p}_{CoM}$, $\mathbf{p}_A$ and $\mathbf{p}_B$ denote the positions of **CoM**, **A** and **B** w.r.t. the world.
To determine whether the CoM is within the support polygon, a strategy was employed which first computes the areas of the four triangles formed by connecting points **A**, **B**, **C** and **CoM**. The center of mass is then located within the support polygon formed



(a) CoM within the support polygon

(b) CoM outside the support polygon

Figure 3.5.: Stability margin and CoM shift direction calculation

by points **A**, **B** and **C** only if $Area_{A,B,CoM} + Area_{A,C,CoM} + Area_{B,C,CoM} = Area_{A,B,C}$. Finally, the direction vector is computed based on whether or not the CoM is within the support polygon. Referring again to Fig. 3.5, the angle ϑ is first computed as follows:

$$\vartheta = \cos^{-1}\left(\frac{margin}{\|\mathbf{p}_{CoM} - \mathbf{p}_A\|}\right) \tag{3.2.3}$$

If the CoM lies within the support polygon, the direction vector is computed by:

$$direction = \begin{bmatrix} \cos(2\pi - \vartheta) & -\sin(2\pi - \vartheta) \\ \sin(2\pi - \vartheta) & \cos(2\pi - \vartheta) \end{bmatrix} * \mathbf{p}^A_{CoM} \tag{3.2.4}$$

If the CoM lies outside the support polygon, the direction vector is computed by:

$$direction = \begin{bmatrix} \cos(\vartheta) & -\sin(\vartheta) \\ \sin(\vartheta) & \cos(\vartheta) \end{bmatrix} * \mathbf{p}^{CoM}_A \tag{3.2.5}$$

This scheme provides a way for determining the necessary direction to shift the base frame before a leg can be lifted and was integrated into the inverse kinematic control algorithm to produce several test motions, as discussed further in the next section.

3.2.3. Inverse Kinematics Controller

In Chapter 2.3.4, it was seen that a simple inverse kinematic control algorithm with position error feedback can be used to compute joint velocities for the six actuators, given a reference trajectory for the swing foot and base frame. With the trajectory planner presented in the previous section, it was then possible to implement the control algorithm in MATLAB with the help of the LucaDynamics library, which was used for computing all of the homogeneous transformation matrices and Jacobian matrices, as well as the center of mass position. The resulting implementation allows for any arbitrary foot stepping order to be provided along with the desired step length and direction of each step w.r.t. the world frame. The algorithm then loops through

each desired step, generates a corresponding motion trajectory, and computes the joint velocities using the differential kinematics approach from Chapter 2.3.3. Various metrics are also collected throughout the process to quantify the results and to run simulations in LucaDynamics' lightweight visualizer. A pseudo-code representation of the implemented approach is provided in Appendix H

3.3. Static Walking Tests

With the proposed trajectory planner and inverse kinematic control algorithm fully implemented in MATLAB, it was possible to perform a series of tests to demonstrate the behavior of the system in simulation and to quantify the results with some collected metrics. Specifically, three different static walking tasks were simulated with the center of mass and contact feet position data tracked throughout each task. Additionally, the Euclidean distance error between the actual and reference swing foot and base frame positions was collected to demonstrate the accuracy of the admitted results. Finally, the condition number for inversion was computed at each time-step for the actuated joint Jacobian matrix to show that no singular or near-singular poses occurred during the commanded locomotion tasks.

3.3.1. Forward Static Walking Test

The first of the tests involved commanding the robot to walk in a straight line in the world frame x direction using a 3-4-2-1 step pattern. Each step consisted of two phases: shifting of the base frame to achieve the desired stability margin, and movement of the selected swing leg along the commanded trajectory. For the test, the following parameters were chosen arbitrarily:

Test Parameter	Value
step length	0.1 m
total step/shift time	2 s
time-step	0.01 s
desired stability margin	0.04 m

Table 3.1.: Forward Static Walking Test Parameters

The position error gain matrix $\mathbf{K}$ was chosen experimentally to achieve a balance between being too small and leading to larger errors, and being too large for the given discrete time-steps, thus leading to instability of the error system. A good balance was found with $\mathbf{K} = 20 * \mathbf{I}_6$, where $\mathbf{I}_6$ denotes the 6×6 identity matrix. The resulting actuated joint trajectories were found to follow continuous paths as shown in Fig. 3.6 and displayed no unreasonably sudden changes in value. The evolution of the Euclidean distance errors between the actual and reference task space positions is displayed in Fig. 3.7a, where it can be seen that both errors remained extremely small for all phases in the static walk. Specifically, the swing foot error stayed below a maximum value of 0.014 cm while the base frame error never exceeded 0.004 cm,

showing that the accuracy of the actual task-space trajectory was quite high. The condition number of the active joint Jacobian matrix was also computed for each time-step and is shown in Fig. 3.7b, where it can be seen that the matrix remained reliably well-conditioned throughout the commanded trajectories, fluctuating within a range of 11.5 to 15. It can therefore be presumed that the system did not move near any singular poses, since the condition number would become extremely large in the vicinity of one.

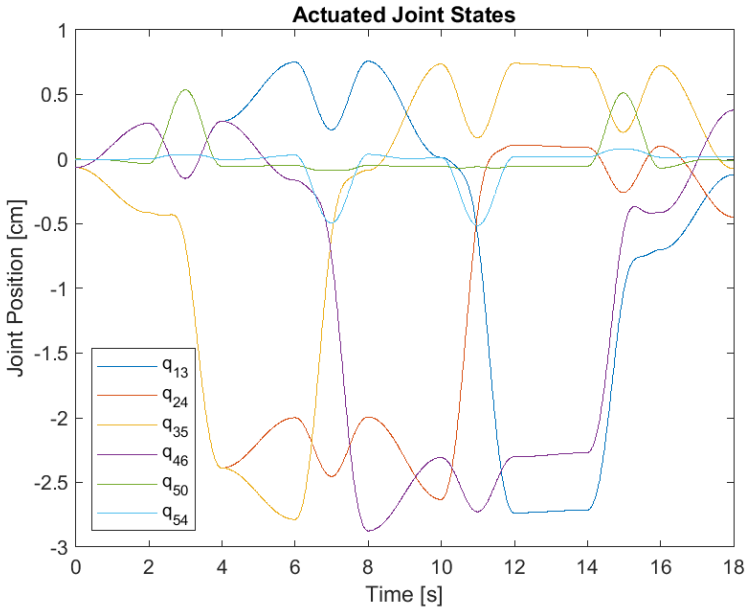
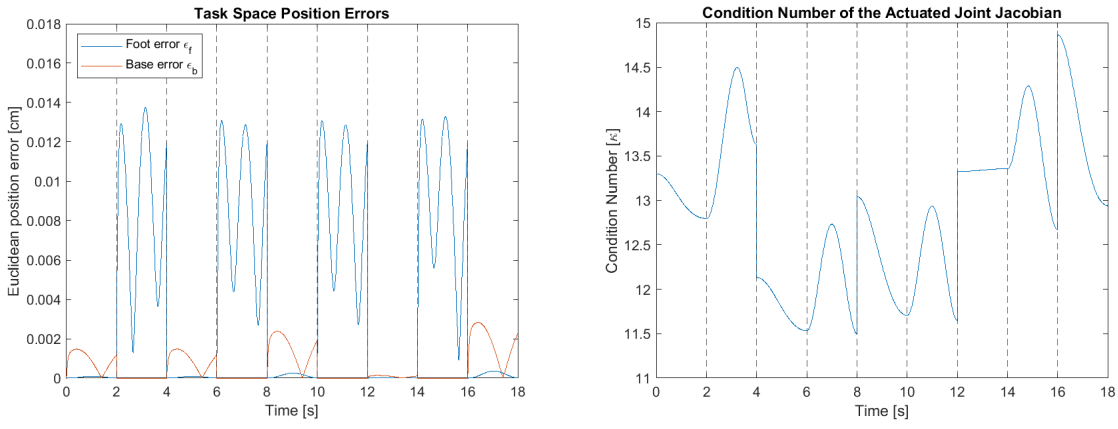


Figure 3.6.: Evolution of active joint positions for forward static walking test



(a) (b)

Figure 3.7.: Evolution of task space errors and condition number of the active joint Jacobian for forward static walking test

In order to confirm that static walking stability was maintained, the center of mass position was tracked and plotted for each of the gait phases as seen in Fig. 3.8. For each of the pre-step phases, it can be seen that the center of mass is shifted within the

support polygon in a direction orthogonal to the support line, and during all of the step phases the center of mass position successfully stays within the support zone.

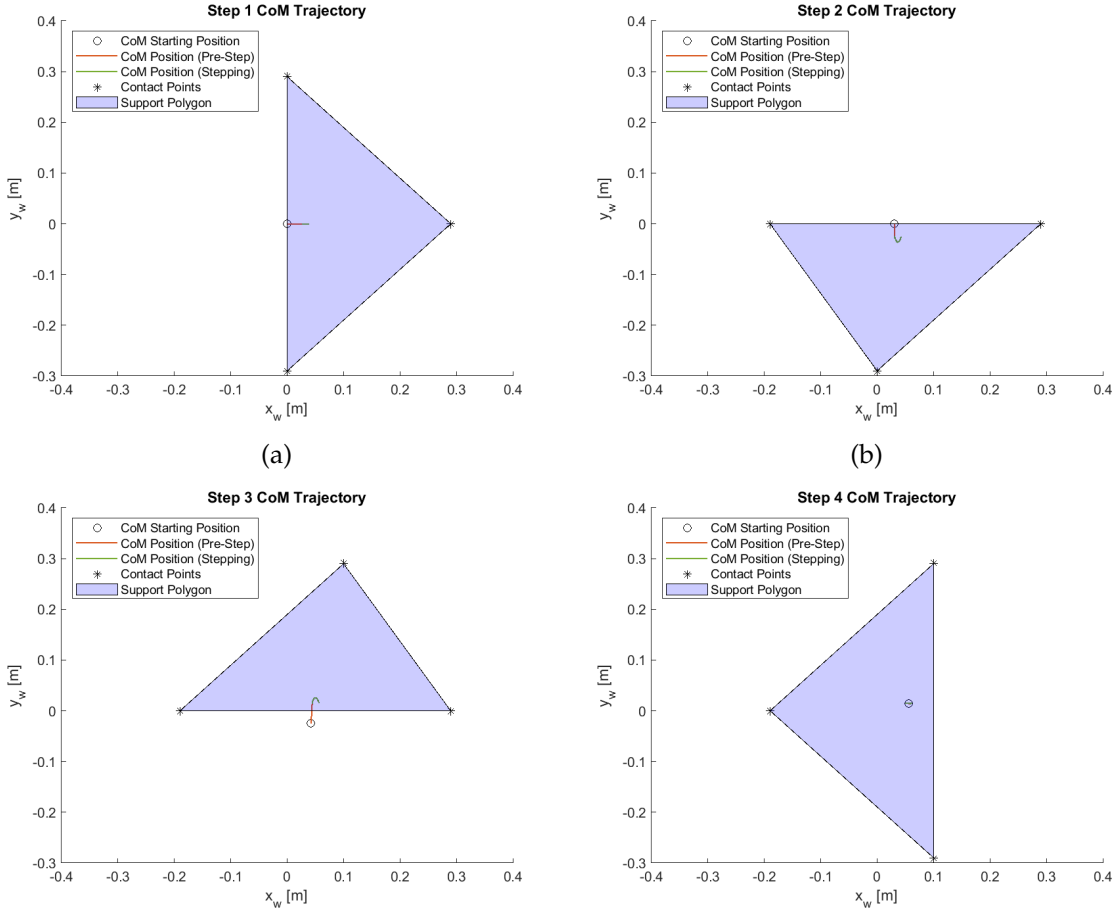


Figure 3.8.: Center of mass trajectory for each step in the forward walking test

3.3.2. Diagonal Static Walking Test

The second test also commanded the robot to follow a straight-line motion with a 3-4-2-1 step pattern, but in a direction rotated $\pi/4$ radians counter-clockwise from the first test, which can be referred to as the diagonal direction. The parameters chosen for the test were the same as those given by Table 3.1 for the first test and the same position error gain matrix was used. Once again, the resulting joint trajectories maintained smooth and continuous paths, with all joints staying within reasonable limits, as seen in Fig. 3.9. The evolution of the resulting Euclidean distance errors can be seen in Fig. 3.10a, with the values again staying quite small. Specifically, the error for the swing foot position stayed below 0.013 cm and the error for the base frame did not go above 0.003 cm. Fig. 3.10b shows the evolution of the actuated joint Jacobian's condition number, which again exhibited quite well-conditioned behavior for the entire walking task, staying within a range of 10.5 to 14.5.

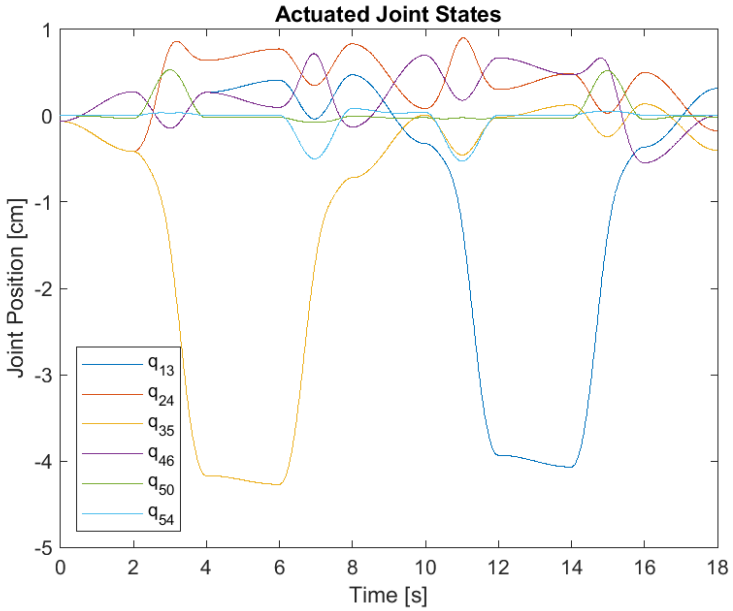
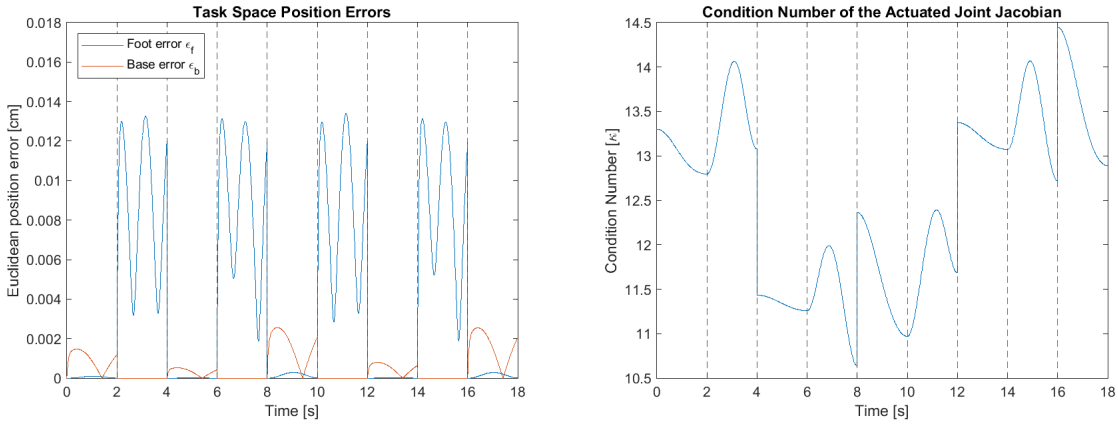


Figure 3.9.: Evolution of active joint positions for diagonal static walking test



(a) (b)

Figure 3.10.: Evolution of task space errors and condition number of the active joint Jacobian for diagonal static walking test

Again, the center of mass position was tracked for each phase in the walking test and is visualized in Fig. 3.11. Just as with the first test, the center of mass successfully shifted into the support polygon before each step and remained in the support zone for the duration of all the swing stages.

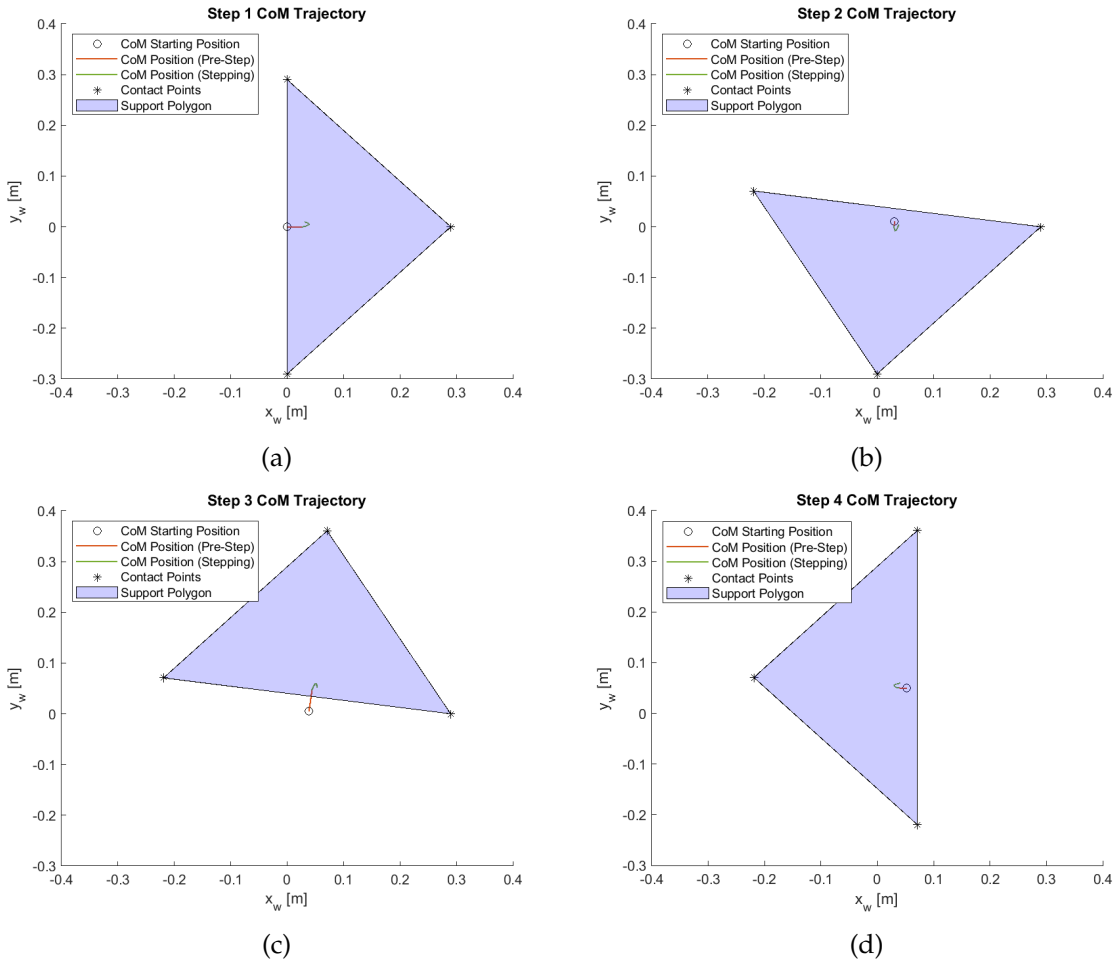


Figure 3.11.: Center of mass trajectory for each step in the diagonal walking test

3.3.3. Rotation Static Walking Test

The final test which was performed involved commanding the robot to rotate $\pi/8$ radians clockwise with a 1-2-3-4 stepping pattern. The individual foot trajectories were computed in the same way as before, but with their end positions determined by rotating the vector positions of the feet $-\pi/8$ about the world frame z axis. Again, the same parameters given in Table 3.1 as well as the same gain matrix $\mathbf{K}$ were used. The resulting joint trajectories again exhibited smooth, continuous behavior as seen in Fig. 3.12. The task space errors also remained within a very similar range of values as the previous test, with the foot error not exceeding 0.013 cm and the base frame error not exceeding 0.002 cm, as seen in Fig. 3.13a. The condition number also maintained a healthy range of well-conditioned values between 11.5 and 14, as shown in Fig. 3.13b.

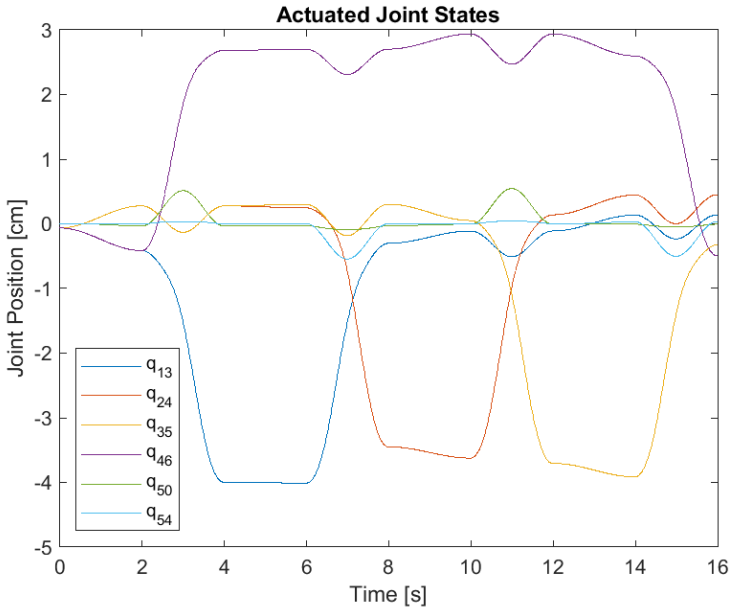
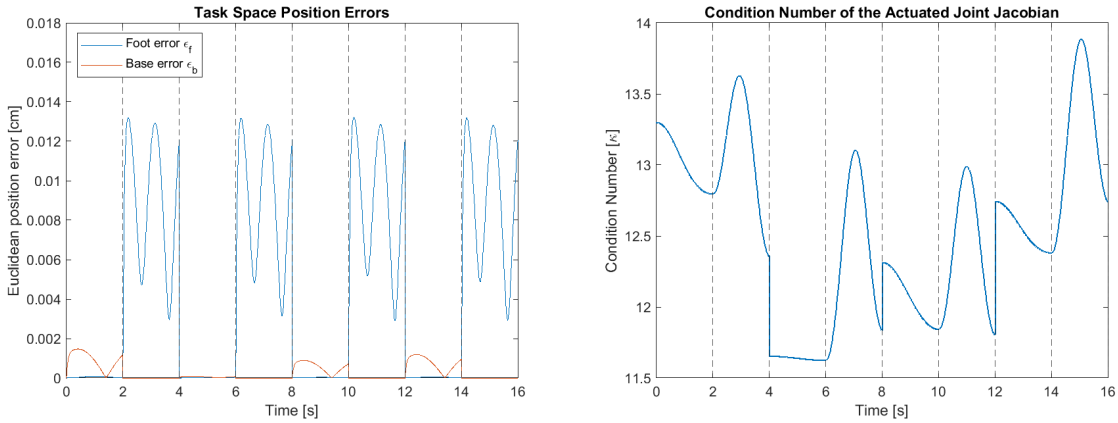


Figure 3.12.: Evolution of active joint positions for rotation static walking test



(a) (b)

Figure 3.13.: Evolution of task space errors and condition number of the active joint Jacobian for rotation static walking test

Finally, just as with the previous two tests, the center of mass position successfully shifted into the support polygon before each step and remained in the stability zone during all four swing stages, as seen in Fig. 3.14. The results for all three tests were very similar, with the actuated joint Jacobian being well-conditioned throughout all the trajectories and the task-space position errors remaining extremely low for the chosen gain matrix values. This shows that the computed joint trajectories result in task-space trajectories which very accurately follow the desired reference values. Furthermore, the well-conditioned nature of the actuated joint Jacobian matrix indicates that the system does not go near any singular poses, since the condition number would become extremely large in the vicinity of one.

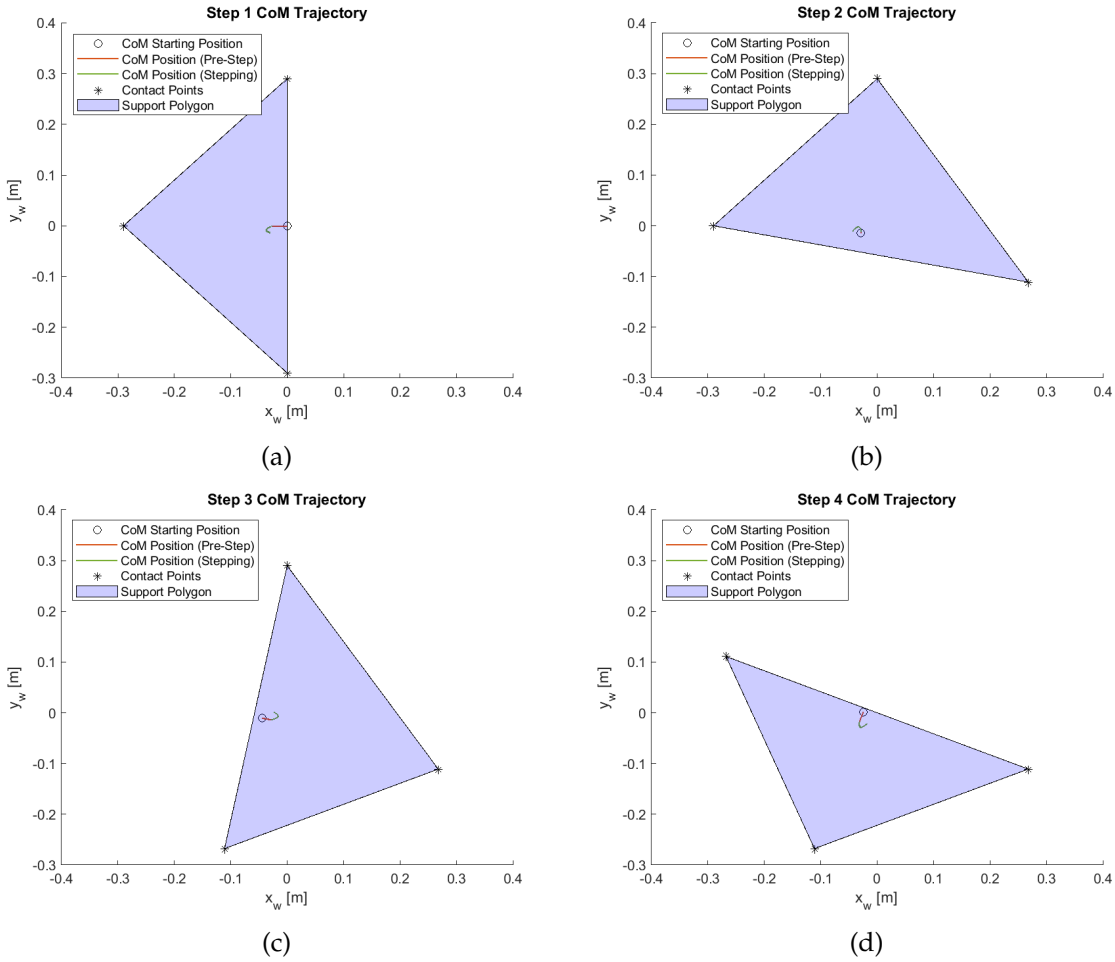


Figure 3.14.: Center of mass trajectory for each step in the rotation walking test

4

Conclusion and Outlook

The goal of this thesis was to design a novel quadruped kinematic structure which allows for lifting and placing of each of the four feet freely in three dimensions using only six actuated degrees of freedom. A solution was proposed which features pantograph legs connected by a central pivot axis and two types of kinematic couplings between the adjacent and opposite leg pairs. A kinematic analysis was performed on the system, where it was shown that its parallel rigid body structure can be represented as a serial rigid body tree, with the introduction of constraint equations to represent the closing conditions of each parallel loop. It was shown that the inverse kinematics problem can be solved numerically using nonlinear least squares optimization, with the cost function represented by a system of the constraint equations. These constraint equations were further represented in terms of differential kinematics in order to derive an inverse kinematics control algorithm which accepts base frame and swing foot linear velocities as input and produces corresponding velocities for the six actuators. The proposed model was implemented in Creo Parametric and exported to URDF, then MATLAB implementations of the inverse kinematics solver and control algorithm were used to verify the behavior of the system by testing a few different motions and gait patterns in simulation. The control algorithm was shown to produce smooth joint trajectories for all tested walking patterns, and the swing foot and base frame position errors remained extremely small, showing that the computed joint trajectories were quite accurate. The center of mass was successfully shifted within the support polygon before each walking step and remained in the support zone throughout all swing stages, showing that static stability can be maintained during crawling. Additionally, the active joint Jacobian matrix was shown to remain well-conditioned throughout the tested trajectories, meaning that no singular or near-singular poses were experienced.

Although it was proven that the desired motions are possible in simulation, the proposed model depends on the assumption of no-slip conditions, with the ground contacts being modeled as ball joints. In reality, foot slipping is a possibility and depends on both the forces experienced between the feet and the ground and their friction coefficients. Since the kinematic representation of the model depends heavily on ideal no-slip contacts with the ground, the presence of slipping could result in undesired and/or uncontrollable motions. It would therefore be of high interest for future work regarding the proposed model to involve analysis of the ground contact forces, partic-

ularly the tangential forces, to determine if slipping is likely to occur in reality.
In addition, this work focused solely on the kinematics of the proposed system and did not go as far as deriving the dynamic representation. It was assumed that simply keeping the center of mass within the support polygon would result in stability during static walking, but given large enough motion velocities, the dynamic properties such as Coriolis and inertial forces could have a negative effect on the stability. On the other hand, there is also the potential of exploiting these dynamic effects to improve the efficiency of the robot's motions, especially through the implementation of joint elasticities which store and release energy, lessening the power requirements from the motors. A next step in development of the proposed concept would therefore be expansion to a dynamic model in order to properly account for all forces acting on the system in the simulation. It would then be possible to better analyze the stability of the chosen walking gaits and to investigate the effects elasticities would have on the system.
Lastly, the geometrical parameters of the implemented solution were chosen arbitrarily for the purpose of basic demonstration of the kinematic structure, but may not be optimal with regards to torques/forces experienced at the passive joints and motors. Further analysis is required to determine if the chosen parameters result in a good balance between motor size/weight and torque requirements for the expected motions. Furthermore, a useful approach could be to determine optimal parameter values with the help of a numerical optimization problem which aims to minimize, for example, the required actuator torques.

Bibliography

Arikawa, K. and S. Hirose (Feb. 2007): "Mechanical design of walking machines". In: *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences* 365, pp. 171–83.

Bledt, G., M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim (2018): "MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2245–2252.

Chen, B., S. Li, and D. Huang (2014): "Quadruped robot crawl gait planning based on DST". In: *Proceedings of the 33rd Chinese Control Conference*, pp. 8578–8582.

Craig, J. J. (2005): *Introduction to Robotics: Mechanics and Control*. 3rd. USA: Pearson Education, Inc.

Denavit, J. and R. S. Hartenberg (June 1955): "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices". In: *Journal of Applied Mechanics* 22.2, pp. 215–221. eprint: https://asmedigitalcollection.asme.org/appliedmechanics/article-pdf/22/2/215/6748803/215_1.pdf. URL: https://doi.org/10.1115/1.4011045.

Hirose, S. and M. Sato (1989): "Coupled drive of the multi-DOF robot". In: *Proceedings, 1989 International Conference on Robotics and Automation*, 1610–1616 vol.3.

Hirose, S., K. Yoneda, K. Arai, and T. Ibe (1991): "Design of prismatic quadruped walking vehicle TITAN VI". In: *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, 723–728 vol.1.

Hirose, S., K. Yoneda, and H. Tsukagoshi (1997): "TITAN VII: quadruped walking and manipulating robot on a steep slope". In: *Proceedings of International Conference on Robotics and Automation*. Vol. 1, 494–500 vol.1.

Hirose, S. (1984): "A Study of Design and Control of a Quadruped Walking Vehicle". In: *The International Journal of Robotics Research* 3.2, pp. 113–133. eprint: https://doi.org/10.1177/027836498400300210. URL: https://doi.org/10.1177/027836498400300210.

Hirose, S., Y. Fukuda, K. Yoneda, A. Nagakubo, H. Tsukagoshi, K. Arikawa, G. Endo, T. Doi, and R. Hodoshima (2009): "Quadruped walking robots at Tokyo Institute of Technology". In: *IEEE Robotics Automation Magazine* 16.2, pp. 104–114.

Hutter, M., C. Gehring, M. Bloesch, M. Hoepflinger, C Remy, and R. Siegwart (Sept. 2012): "StarlETH: a Compliant Quadrupedal Robot for Fast, Efficient, and Versatile Locomotion". In: pp. 483–490.

Hutter, M., C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger (2016): "ANYmal - a highly mobile and dynamic quadrupedal robot". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 38–44.

Kaneko, M., M. Abe, and S. Tachi (1986): "Basic considerations of the degrees of freedom of multi-legged locomotion machines". In: *Adv. Robotics* 1, pp. 101–116.

Lynch, K. M. and F. C. Park (2017): *Modern Robotics: Mechanics, Planning, and Control*. 1st. USA: Cambridge University Press.

Ma, S., T. Tomiyama, and H. Wada (2005): "Omnidirectional static walking of a quadruped robot". In: *IEEE Transactions on Robotics* 21.2, pp. 152–161.

Raibert, M. H. (1986): *Legged Robots That Balance*. USA: Massachusetts Institute of Technology.

Seidel, D., M. Hermann, T. Gumpert, F. C. Loeffl, and A. Albu-Schäffer (2020): "Using Elastically Actuated Legged Robots in Rough Terrain: Experiments with DLR Quadruped bert". In: *2020 IEEE Aerospace Conference*, pp. 1–8.

Semini, C., V. Barasuol, J. Goldsmith, M. Frigerio, M. Focchi, Y. Gao, and D. G. Caldwell (2017): "Design of the Hydraulically Actuated, Torque-Controlled Quadruped Robot HyQ2Max". In: *IEEE/ASME Transactions on Mechatronics* 22.2, pp. 635–646.

Siciliano, B., L. Sciavicco, V. Luigi, and G. Oriolo (Jan. 2009): "Robotics: Modelling, planning and control". In: pp. 1–623.

Stewart, D. (1965): "A Platform with Six Degrees of Freedom". In: *Proceedings of the Institution of Mechanical Engineers* 180.1, pp. 371–386. eprint: https://doi.org/10.1243/PIME_PROC_1965_180_029_02. URL: https://doi.org/10.1243/PIME_PROC_1965_180_029_02.

Wang, Z., C. Sun, G. Deng, and A. Zhang (2017): "Locomotion planning for quadruped robot over rough terrain". In: *2017 Chinese Automation Congress (CAC)*, pp. 3170–3173.

Yoneda, K. and S. Hirose (1992): "Dynamic and static fusion gait of a quadruped walking vehicle on a winding path". In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 143–148 vol.1.

Yoneda, K. (2007): "Light Weight Quadruped with Nine Actuators". In: *Journal of Robotics and Mechatronics* 19.2, pp. 160–165.

Yoneda, K., Y. Ota, F. Ito, and S. Hirose (2001): "Quadruped Walking Robot with Reduced Degrees of Freedom". In: *Journal of Robotics and Mechatronics* 13.2, pp. 190–197.

Zhang, J., J. Shen, and D. W. Hong (2020): "Kinematic Analysis and Design Optimization for a Reduced-DoF Quadruped Robot with Minimal Torque Requirements". In: *2020 17th International Conference on Ubiquitous Robots (UR)*, pp. 198–204.

A

Common Drive Quadrupeds

The following three figures are expanded versions of the sub-figures in Fig. 2.5.

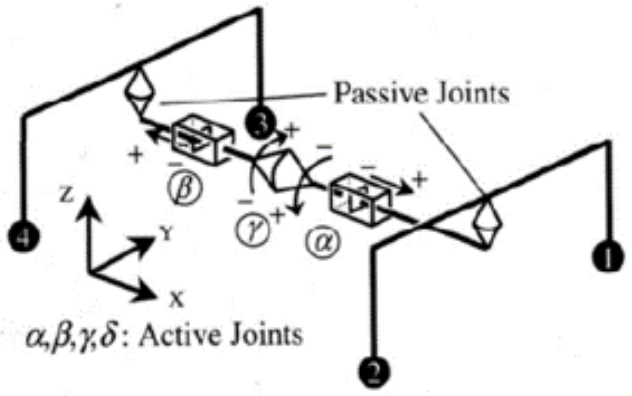


Figure A.1.: 3-active DoF quadruped proposed by Yoneda et al. (2001)

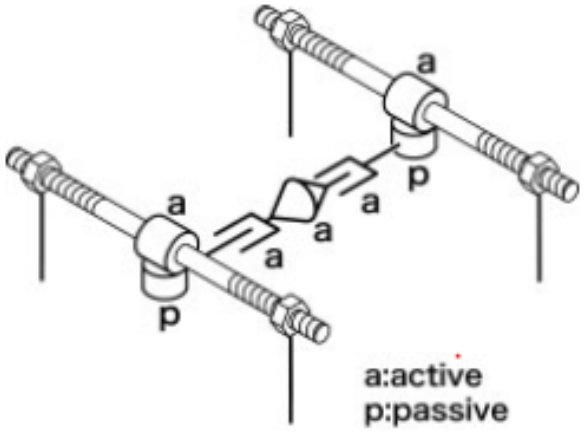


Figure A.2.: 5-active DoF quadruped proposed by Yoneda (2007)

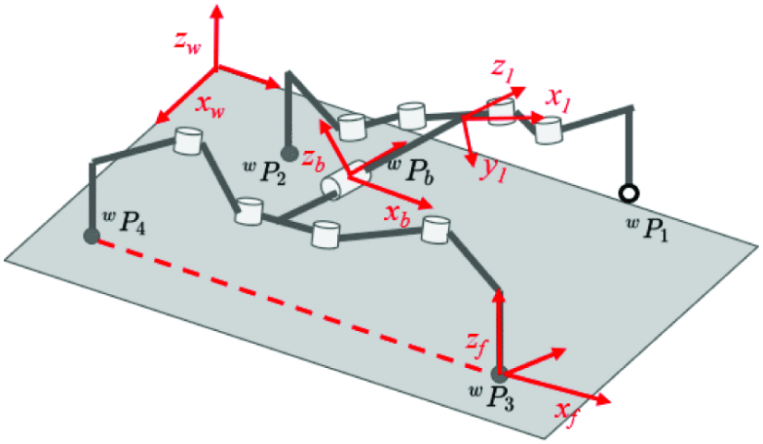


Figure A.3.: 9 DoF quadruped proposed by Zhang, Shen, and Hong (2020)

B

Joint DoF Tables

Joint	Type	DoF
j_1	Revolute	1
j_2	Revolute	1
j_3	Revolute	1
j_4	Revolute	1
j_5	Revolute	1
j_6	Ball	3
j_7	Prismatic	1
j_8	Universal	2
j_9	Ball	3
j_{10}	Prismatic	1
j_{11}	Universal	2
Total		17

Table B.1.: Joint types and DoF for the single-leg mechanism in Fig. 2.7

Joint	Type	DoF
j_1	Revolute	1
j_2	Revolute	1
j_3	Revolute	1
j_4	Revolute	1
j_5	Revolute	1
j_6	Revolute	1
j_7	Ball	3
j_8	Prismatic	1
j_9	Universal	2
j_{10}	Ball	3
j_{11}	Prismatic	1
j_{12}	Universal	2
Total		18

Table B.2.: Joint types and DoF for the single-leg mechanism in Fig. 2.8

	Joint	Type	DoF
$\times$4 legs	j_1	Revolute	1
	j_2	Revolute	1
	j_3	Revolute	1
	j_4	Revolute	1
	j_5	Revolute	1
	j_6	Revolute	1
	j_7	Ball	3
	j_8	Prismatic	1
	j_9	Universal	2
	j_{10}	Ball	3
	j_{11}	Prismatic	1
	j_{12}	Universal	2
	–	Floating Base	6
Total			78

Table B.3.: Joint types and DoF for the four-leg mechanism in Fig. 2.9

	Joint	Type	DoF
×4 legs	j_1	Revolute	1
	j_2	Revolute	1
	j_3	Revolute	1
	j_4	Revolute	1
	j_5	Revolute	1
	j_6	Revolute	1
	j_7	Ball	3
	j_8	Prismatic	1
	j_9	Universal	2
	–	Floating Base	6
Total			54

Table B.4.: Joint types and DoF for the four-leg mechanism with adjacent leg couplings in Fig. 2.10

	Joint	Type	DoF
×4 legs	j_2	Revolute	1
	j_3	Revolute	1
	j_4	Revolute	1
	j_5	Revolute	1
	j_6	Revolute	1
	j_7	Ball	3
	j_8	Prismatic	1
	j_9	Universal	2
×3 legs	j_1	Revolute	1
	–	Floating Base	6
Total			53

Table B.5.: Joint types and DoF for the four-leg kinematically reduced mechanism in Fig. 2.12

	Joint	Type	DoF
×4 legs	j_2	Revolute	1
	j_3	Revolute	1
	j_4	Revolute	1
	j_5	Revolute	1
	j_6	Revolute	1
	j_7	Ball	3
	j_8	Prismatic	1
	j_9	Universal	2
×3 legs	j_1	Revolute	1
×2 couplings	j_{10}	Prismatic	1
	j_{11}	Revolute	1
	j_{12}	Revolute	1
	j_{13}	Revolute	1
	j_{14}	Revolute	1
	j_{15}	Revolute	1
	–	Floating Base	6
Total			65

Table B.6.: Joint types and DoF for the final four-leg mechanism in Fig. 2.15

C

Joint State Vector

The joint state vector $\mathbf{q}$ is of dimension $\mathbb{R}^{57\times 1}$ and is made up of variables defined in the DH parameters tables, plus the six virtual floating base joint variables. The full state vector is represented as follows:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_{l1} \\ \mathbf{q}_{ac1} \\ \mathbf{q}_{l2} \\ \mathbf{q}_{ac2} \\ \mathbf{q}_{l3} \\ \mathbf{q}_{ac3} \\ \mathbf{q}_{l4} \\ \mathbf{q}_{ac4} \\ \mathbf{q}_{oc1} \\ \mathbf{q}_{oc2} \end{bmatrix} \tag{C.0.1}$$

where:

$$\mathbf{q}_b = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix} = \begin{bmatrix} x_b \\ y_b \\ z_b \\ \phi_b \\ \vartheta_b \\ \psi_b \end{bmatrix} \tag{C.0.2}$$

$$\mathbf{q}_{l1} = \begin{bmatrix} q_7 \\ q_8 \\ q_9 \\ q_{10} \end{bmatrix} = \begin{bmatrix} \vartheta_{j_1} \\ \vartheta_{j_1+1} \\ \vartheta_{j_1+1'} \\ \vartheta_{j_1+2'} \end{bmatrix} \tag{C.0.3}$$

$$\mathbf{q}_{ac1} = \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \\ q_{14} \\ q_{15} \\ q_{16} \end{bmatrix} = \begin{bmatrix} \vartheta_{k_1+1} \\ \vartheta_{k_1+2} \\ d_{k_1+3} \\ \vartheta_{k_1+4} \\ \vartheta_{k_1+5} \\ \vartheta_{k_1+6} \end{bmatrix} \tag{C.0.4}$$

$$\mathbf{q}_{l2} = \begin{bmatrix} q_{17} \\ q_{18} \\ q_{19} \\ q_{20} \\ q_{21} \end{bmatrix} = \begin{bmatrix} \vartheta_{p_2} \\ \vartheta_{j_2} \\ \vartheta_{j_2+1} \\ \vartheta_{j_2+1'} \\ \vartheta_{j_2+2'} \end{bmatrix} \tag{C.0.5}$$

$$\mathbf{q}_{ac2} = \begin{bmatrix} q_{22} \\ q_{23} \\ q_{24} \\ q_{25} \\ q_{26} \\ q_{27} \end{bmatrix} = \begin{bmatrix} \vartheta_{k_2+1} \\ \vartheta_{k_2+2} \\ d_{k_2+3} \\ \vartheta_{k_2+4} \\ \vartheta_{k_2+5} \\ \vartheta_{k_2+6} \end{bmatrix} \tag{C.0.6}$$

$$\mathbf{q}_{l3} = \begin{bmatrix} q_{28} \\ q_{29} \\ q_{30} \\ q_{31} \\ q_{32} \end{bmatrix} = \begin{bmatrix} \vartheta_{p_3} \\ \vartheta_{j_3} \\ \vartheta_{j_3+1} \\ \vartheta_{j_3+1'} \\ \vartheta_{j_3+2'} \end{bmatrix} \tag{C.0.7}$$

$$\mathbf{q}_{ac3} = \begin{bmatrix} q_{33} \\ q_{34} \\ q_{35} \\ q_{36} \\ q_{37} \\ q_{38} \end{bmatrix} = \begin{bmatrix} \vartheta_{k_3+1} \\ \vartheta_{k_3+2} \\ d_{k_3+3} \\ \vartheta_{k_3+4} \\ \vartheta_{k_3+5} \\ \vartheta_{k_3+6} \end{bmatrix} \tag{C.0.8}$$

$$\mathbf{q}_{l4} = \begin{bmatrix} q_{39} \\ q_{40} \\ q_{41} \\ q_{42} \\ q_{43} \end{bmatrix} = \begin{bmatrix} \vartheta_{p_4} \\ \vartheta_{j_4} \\ \vartheta_{j_4+1} \\ \vartheta_{j_4+1'} \\ \vartheta_{j_4+2'} \end{bmatrix} \tag{C.0.9}$$

$$\mathbf{q}_{ac4} = \begin{bmatrix} q_{44} \\ q_{45} \\ q_{46} \\ q_{47} \\ q_{48} \\ q_{49} \end{bmatrix} = \begin{bmatrix} \vartheta_{k_4+1} \\ \vartheta_{k_4+2} \\ d_{k_4+3} \\ \vartheta_{k_4+4} \\ \vartheta_{k_4+5} \\ \vartheta_{k_4+6} \end{bmatrix} \tag{C.0.10}$$

$$\mathbf{q}_{oc1} = \begin{bmatrix} q_{50} \\ q_{51} \\ q_{52} \\ q_{53} \end{bmatrix} = \begin{bmatrix} d_f \\ \vartheta_{f+1} \\ \vartheta_{f+1'} \\ \vartheta_{f+2'} \end{bmatrix} \tag{C.0.11}$$

$$\mathbf{q}_{oc2} = \begin{bmatrix} q_{54} \\ q_{55} \\ q_{56} \\ q_{57} \end{bmatrix} = \begin{bmatrix} d_{f'} \\ \vartheta_{f'+1} \\ \vartheta_{f'+1'} \\ \vartheta_{f'+2'} \end{bmatrix} \tag{C.0.12}$$

D

Differential Kinematics Representation

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} \mathbf{g}_1^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{12}^T(\mathbf{q}) \\ \mathbf{e}_1^T \\ \mathbf{g}_{13}^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{22}^T(\mathbf{q}) \\ \mathbf{e}_2^T \\ \mathbf{g}_{23}^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{32}^T(\mathbf{q}) \\ \mathbf{e}_3^T \\ \mathbf{g}_{33}^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{42}^T(\mathbf{q}) \\ \mathbf{e}_4^T \\ \mathbf{g}_{43}^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{45}^T(\mathbf{q}) \\ \mathbf{e}_5^T \\ \mathbf{g}_{46}^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{48}^T(\mathbf{q}) \\ \mathbf{e}_6^T \\ \mathbf{g}_{49}^T(\mathbf{q}) \\ \vdots \\ \mathbf{g}_{51}^T(\mathbf{q}) \end{bmatrix} \tag{D.0.1}$$

E

CAD Model

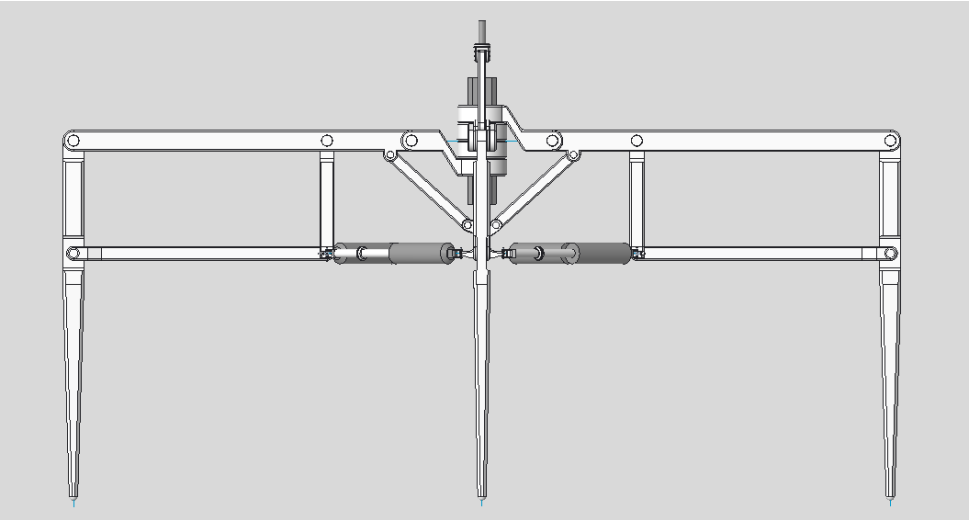


Figure E.1.: Front view

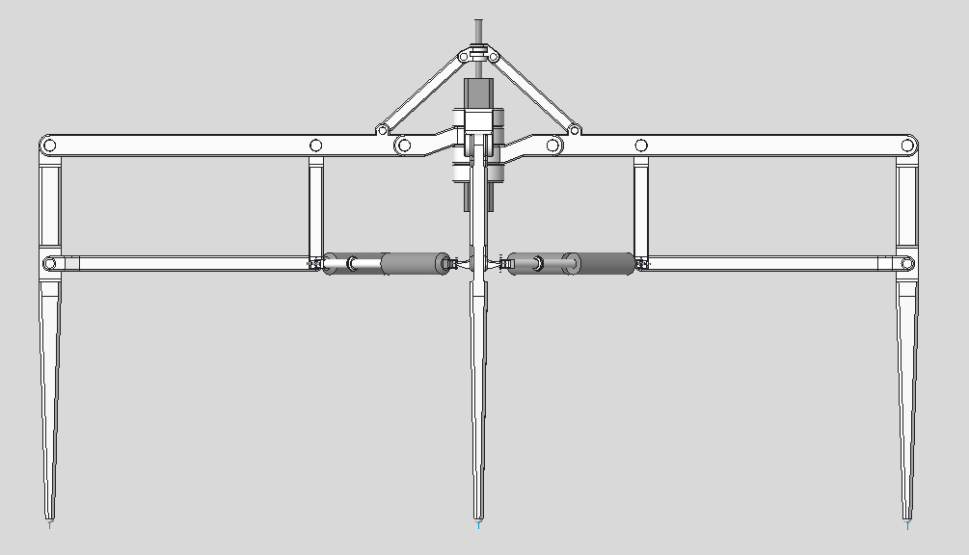


Figure E.2.: Side view

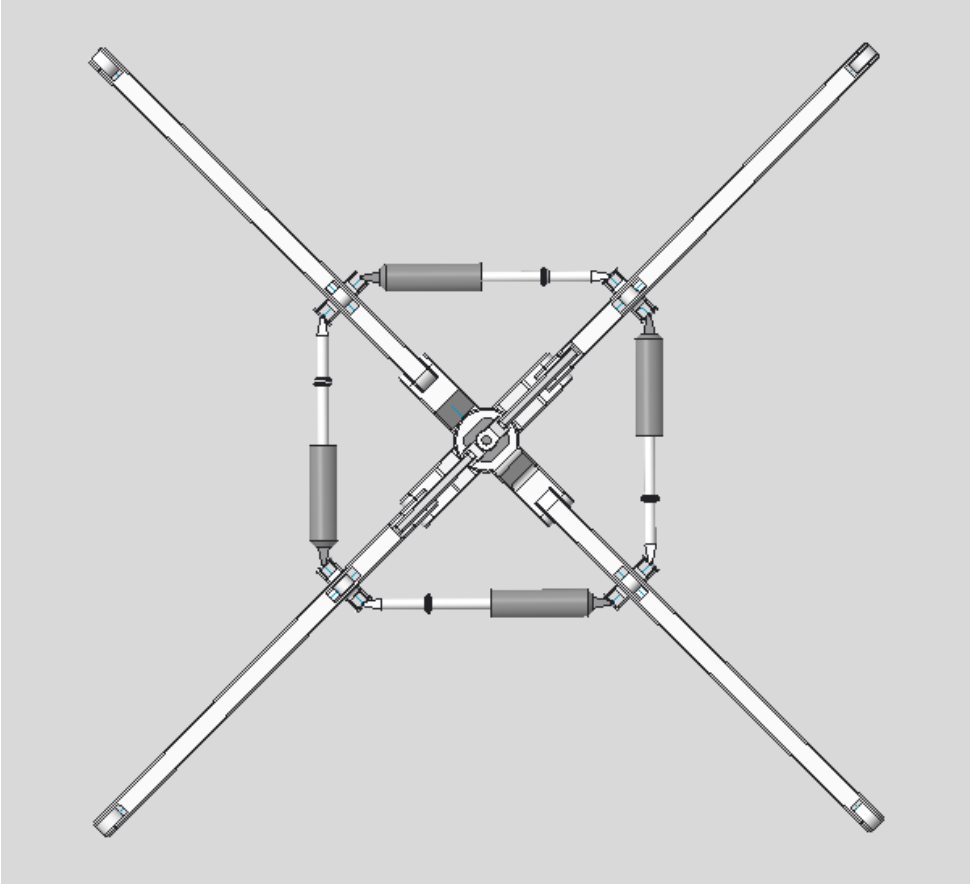


Figure E.3.: Top view

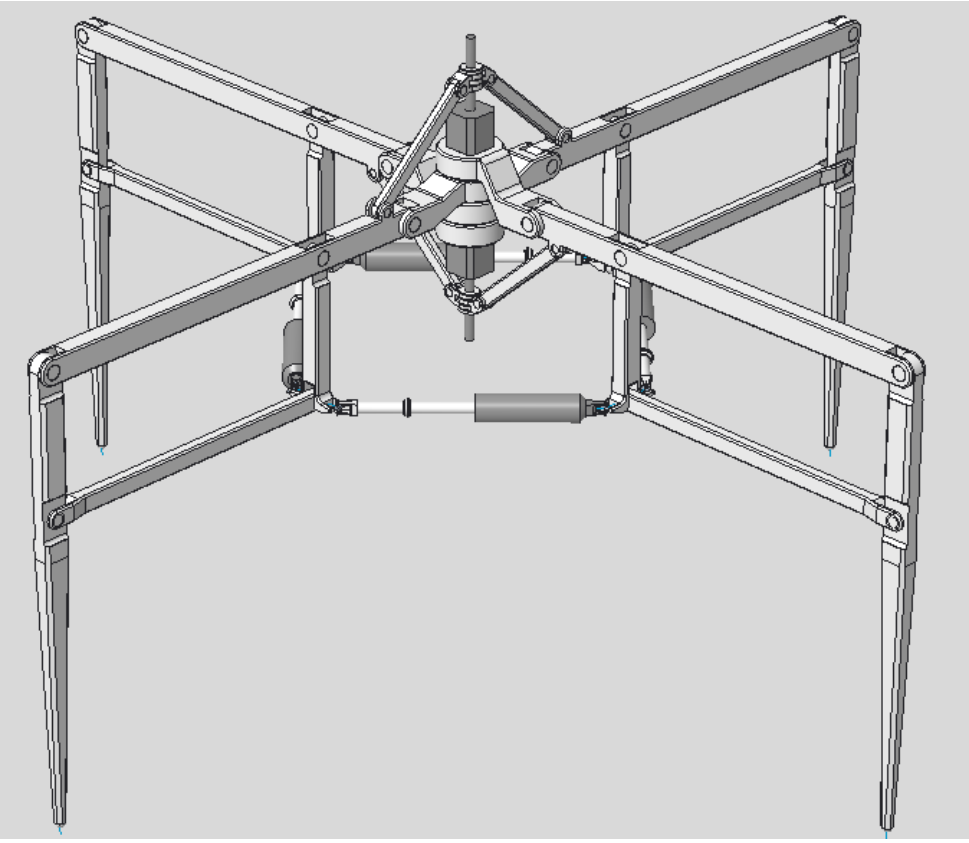


Figure E.4.: Isometric view

F

Model Parameter Values

Parameter	Value (m)
a_{j_n}	0.05
a_{j_n+1}	0.24
a_{j_n+2}	0.08
$a_{j_n+1'}$	0.06
$a_{j_n+2'}$	0.08
$a_{j_n+3'}$	0.18
a_{j_n+3}	0.18
a_{k_n}	0.08
a_{k_n+1}	0.015
d_{k_n+4}	0.035
$a_{k'_n}$	0.08
$a_{k'_n+1}$	0.015
$a_{j_n+1''}$	0.015
$a_{j_n+2''}$	0.01
a_{f+1}	0.01
a_{f+2}	0.07433
$a_{f+2'}$	0.01
$a_{f+3'}$	0.07433
$a_{f'+1}$	0.01
$a_{f'+2}$	0.07433
$a_{f'+2'}$	0.01
$a_{f'+3'}$	0.07433

Table F.1.: DH parameter values used for the experimental implementation

G

Compensated Joint Variables

Joint Variable	Compensated Form
q_3	$q_3 + 0.26$
q_8	$q_8 - \pi/2$
q_9	$q_9 - \pi/2$
q_{10}	$q_{10} + \pi/2$
q_{12}	$q_{12} + \pi/4$
q_{13}	$q_{13} + 0.1$
q_{15}	$q_{15} + 3\pi/4$
q_{16}	$q_{16} + \pi/2$
q_{17}	$q_{17} - \pi/2$
q_{19}	$q_{19} - \pi/2$
q_{20}	$q_{20} - \pi/2$
q_{21}	$q_{21} + \pi/2$
q_{23}	$q_{23} + \pi/4$
q_{24}	$q_{24} + 0.1$
q_{26}	$q_{26} + 3\pi/4$
q_{27}	$q_{27} + \pi/2$
q_{28}	$q_{28} - \pi$
q_{30}	$q_{30} - \pi/2$
q_{31}	$q_{31} - \pi/2$
q_{32}	$q_{32} + \pi/2$
q_{34}	$q_{34} + \pi/4$
q_{35}	$q_{35} + 0.1$
q_{37}	$q_{37} + 3\pi/4$
q_{38}	$q_{38} + \pi/2$
q_{39}	$q_{39} - 3\pi/2$
q_{41}	$q_{41} - \pi/2$
q_{42}	$q_{42} - \pi/2$
q_{43}	$q_{43} + \pi/2$
q_{45}	$q_{45} + \pi/4$
q_{46}	$q_{47} + 0.1$
q_{48}	$q_{48} + 3\pi/4$

q_{49}	$q_{49} + \pi/2$
q_{50}	$q_{50} + 0.06$
q_{51}	$q_{51} - 0.737815$
q_{52}	$q_{52} + \pi$
q_{53}	$q_{53} - 0.737815$
q_{54}	$q_{54} + 0.06$
q_{55}	$q_{55} + 0.737815$
q_{56}	$q_{56} + 3\pi/2$
q_{57}	$q_{57} + 0.737815$

Table G.1.: Joint variables & their compensated forms for the forward kinematics function

H

Inverse Kinematics Control Algorithm

Algorithm H.1.: Inverse Kinematics Control

Require: q: initial joint positions, dq: initial joint velocities, p reference trajectory positions, dp: reference trajectory velocities, t: time vector, dt: timestep, K: position error gain matrix, $foot$: desired swing foot

1: **for** all i in t **do**
2: Compute all **H** and **J** ▷ using LucaDynamics algorithm
3: e $\leftarrow p(i) - x_{f,b}$ ▷ $x_{f,b}$ is the actual foot & base frame position
4: $dx\_d \leftarrow dp(i) + K * e$
5: **function** DIFFKIN($H, J, dx\_d, foot$)
6: Transform Jacobians into respective frames
7: Compile constraint Jacobian J_temp
8: Separate J_temp by active/passive joints (H_a and H_p)
9: $J\_con \leftarrow -(H\_p)^{-1} H\_a$
10: Reintroduce active variable rows to J_con
11: $J\_a \leftarrow [\mathbf{J}_{foot}; \mathbf{J}_{base}] * J\_con$ ▷ compute active joint Jacobian
12: Transform dx_d into end-effector frames (v_f)
13: $dq\_a \leftarrow (J\_a)^{-1} * v\_f$ ▷ compute active joint velocities
14: $dq \leftarrow J\_con * dq\_a$ ▷ map to all joint velocities
15: **return** q_dot
16: $q \leftarrow q + dq * dt$ ▷ update joint state values