Automation in Horticulture

Design and Evaluation of a Cost-Effective Parallel Robot Arm

by

Mathijs Dijkstra

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on November 29, 2023 at 14:30 AM.

Student number: 4561333
Project duration: April 1, 2023 – November 1, 2023
Thesis committee: Dr. ir. G. Smit, TU Delft, supervisor
Ir. J. Koudijzer, Festo B.V., daily supervisor

TU Delft

Preface

I extend my gratitude to Festo for their generous support in my graduation project, providing an enriching environment and essential components. Special thanks to Jan Koudijzer for his invaluable discussions and support, which enhanced my research experience. I also appreciate Gerwin Smit, TU Delft, for his insightful supervision and feedback. Their contributions have been instrumental in the success of this thesis.
I am grateful to Els and my mother for their loving support.

Mathijs Dijkstra
Delft, November 2023

Contents

1

Summary

This thesis presents the conceptual development of an innovative parallel manipulator that leverages the beneficial attributes of parallel kinematic structures. These attributes include increased stiffness, uniform load distribution, and reduced moving mass, which are strategically incorporated into the design to enhance resource efficiency. This approach aims to yield more robust, cost-effective, and reliable alternatives to conventional serial manipulators. The concept is synthesized from various literature sources, leading to a detailed kinematic analysis through inverse kinematics. This analysis provides valuable insights into the mechanism's functionality and facilitates iterative design improvements. Additionally, a dynamic behavior simulation of the mechanism is conducted using Simulink's Simscape. Based on these insights, a prototype is constructed to empirically validate the design concept. The final design demonstrates a compact, fully accessible workspace with a kinematic structure conducive to robust manipulation. The load distribution is optimized across all motors, maximizing motor capacity utilization and enabling efficient handling of peak loads.

2

Introduction

2.1. Automation is a must

Automation in horticulture has gained momentum in the challenge of providing citizens of the world with nutritious, healthy, and enough food. With the introduction of IoT, computer vision, robotics, and advanced control algorithms, this sector has been able to innovate rapidly, leading to innovations in automation, collaboration, sensing, control, and world perception.
Two main reasons the focus has shifted toward automation are the intensification of these agricultural fields [28] and the shrinking labor force available to process them [16]. This results in crop loss and increasing costs and this trend is predicted to continue [18]. Predictions are that the total labor gap will be around 50 percent in 2029 [3]. To ensure the future of these industries, there is a need for a reliable workforce.

2.2. Current situation

Currently, commercial crop-picking solutions have already entered the market. Like, Grow, Appharvest, and Dogtooth [27]. However, the adoption of this technology is still challenging. Vazconez et al. [40] show that specialty crops, like tomatoes and strawberries, are far behind on automation, compared to for example field crops. Factors that are responsible are in the field of limited accuracy, flexibility, and harvesting speed [30]. Humans excel in managing dynamic environments due to their remarkable ability to adapt quickly to sudden changes [19]. Lastly, the initial investment costs per picked unit of such a robot will leave many growers doubting, as this decides whether it will be commercially attractive [1].

2.3. Problem

Advancements in the design of manipulation methods have exploded, as the applications needed in these dynamic environments are rather endless. Think of grippers that can grasp different size crops, leaf cutters, and scissors for peduncles. However, the design of the robot arm itself is rather subordinate. Limited literature is found for the specific design of this part, and application designs tend to pick the ready-to-use options. The need for robotic manipulators with less than 6 DOFs can be questioned, as the ones with equal or more than 6 DOFs can be used for multiple applications. Kong et al. [22] cite a main reason to argue this statement. Cost reduction plays a huge part for potential customers of this technology. What would be the use of an over-engineered robot arm, if only half of its capacity can be used in practice? Would it be necessary to use a robot n-times the weight of the crop in order to handle it properly? For sure, designing a new robotic manipulator is a risky and valuable process. Therefore, the question becomes more important as these robot arms can be considered over-designed for the task at hand and therefore are too costly for the market of horticulture and perhaps other fields. Can the burden of costly robot arms be partially relieved by an applied robot arm design, taking into account the critical workspace, used materials, and required motor capacity?

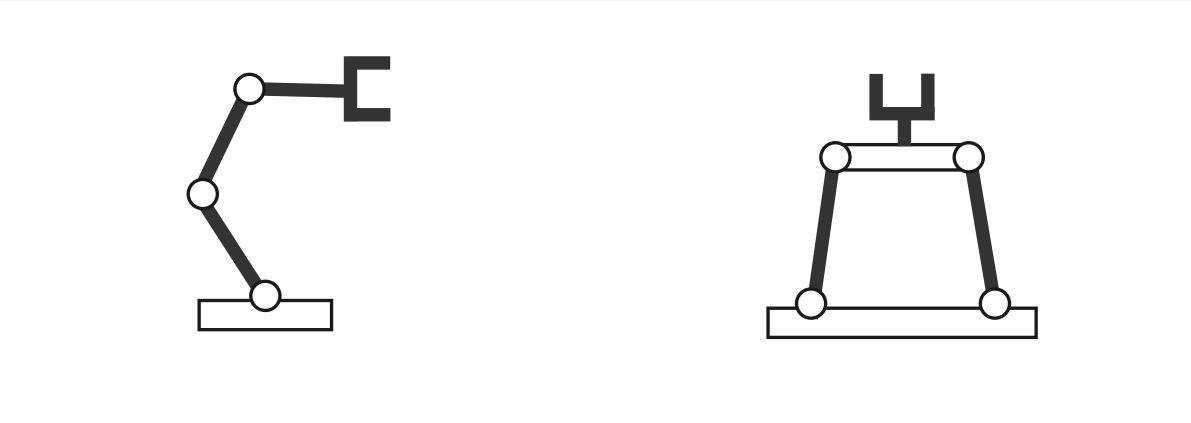


Figure 2.1: Schematic drawing of a serial vs. a parallel mechanism and their general lay-out

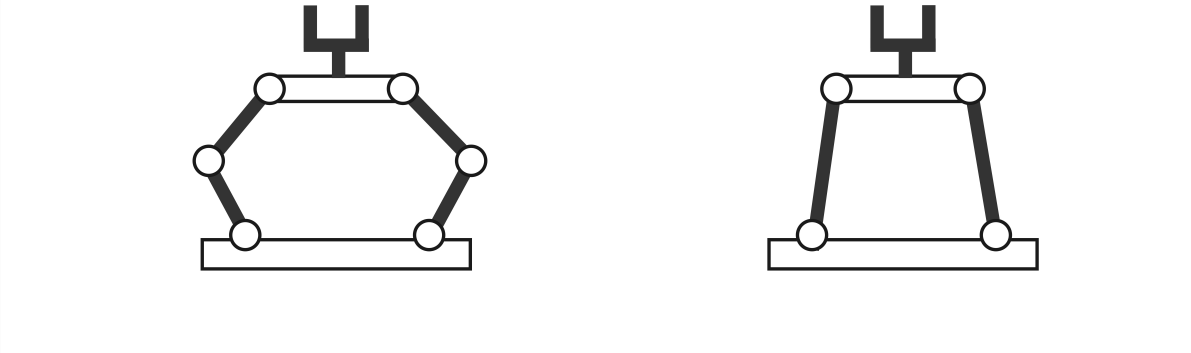


Figure 2.2: Schematic drawing of a serial vs. a parallel mechanism and their general properties

2.4. Introducing parallel manipulators

A promising possibility is the use of parallel manipulators (PM). PMs are a class of multi-closed-loop mechanisms. One may find this solution in applications like pick-and-place, or manufacturing. The configuration differs from conventional serial manipulators in that all actuators are situated at the base. Links are then used to connect these actuators to a common platform, where all ends of the links come together. This allows for superior mechanical properties, like high-speed movement, advantageous inertia effects, and high accuracy. By using closed kinematic chains, the load is shared among multiple actuators, which ensures better load transmission and stiffness, according to Uchiyama et al. [39]. Thus ensuring a better weight-to-payload ratio. This is a major advantage over serial manipulators, where every motor has to compensate for the weight of the following motors, resulting in a higher need for motor capacity as schematically shown in figure 2.1. The major weakness of this type of manipulator is the small workspace compared to serial manipulators. This is due to mechanical interference. Mechanical interference can be described as physical constraints, which are caused by the limited range of motion of the actuators and dependencies between the linkages [29]. This also causes an increased presence of singularities. When the system is exposed to a singularity, the overall performance and motion are poorly executed, compromising stiffness, and acceleration, and therefore allowing for unpredictable behavior. In such an event, actuation forces could go to infinity and as a result, damage the system [24]. Parallel manipulators, as we know them, can be classified into two main categories: pure and hybrid parallel manipulators. The difference lies in the way the legs are built, like in figure 2.2, and this influences the performance of the robot [31]. For example, the Delta [32] has a hybrid structure, and the well-known Steward platform [9] has a pure structure. Although these manipulators have placed themselves in the heart of the industry, the major issue is still not resolved. The fact that uneven workspace forces industrial decision-makers to pick the serial alternative. Nevertheless, major

advancements have been made in research to accommodate the problem of limited workspace.

2.5. Goal

The central issue that steers this thesis becomes clear: the challenge of integrating a low-cost concept into handling operations. Recognizing the imperative for a solution that aligns with economic feasibility and operational capability, this thesis proposes an innovative approach, being a robotic manipulator that is not only cost-effective but also applied to reaching the required performance set by the harvesting application.

The drive for efficiency and precision has led to the exploration of parallel kinematic systems, which boast an array of benefits over their serial counterparts including increased stiffness, end-effector accelerations, lower system inertia, and accuracy. Such systems have found their niche in industrial environments where these characteristics are essential for tasks like pick-and-place operations, manufacturing, and packaging. However, this thesis seeks to pivot these beneficial properties into a material-efficient manipulator design. By adopting these parallel structures, we hypothesize that a more resource-conscious design could be achieved—one that necessitates fewer materials and less motor capacity, yet delivers comparable performance to serial manipulators. The pressing question this thesis endeavors to answer is whether a parallel manipulator can satisfy the intricate dexterous and workspace requirements necessary for precision harvesting. This is a question of balancing the fine line between the practical constraints of a workspace within a greenhouse and the robotic system's ability to maneuver with the dexterity required for harvesting crops.

2.6. Outline

In order to find the answers to these questions, this thesis is constructed methodically. Starting with a thorough research of existing parallel solutions in chapter 3, which tackles the challenges of parallel manipulators. In chapter 4 the findings are combined into a set of functional requirements and a final design is chosen. Chapter 5 goes into the technical validity of the chosen design and provides a first look at the expected behavior. Chapter 6 shows the final prototype together with the experimental validation. Finally, discussing the results is done in chapter 7 and conclusions are named in chapter 8.

3

State of the art

3.1. Introduction

After the problem is thoroughly described and the outline of this report is clear, the focus is now on researching what is already out there. The state of the art, which is an extensive literature research on workspace affecting aspects in parallel manipulators, is therefore given. This section seeks to create an overview of major distinctive designs, and their characteristic aspects. Therefore, finding an answer to the question of how the performance of these manipulators is measured. Also, what methods are used to predict the output behavior? This chapter serves as the cornerstone for the conceptualization of a novel breed of parallel manipulators, synthesizing past insights to forge a path for future innovation in robotic design.

3.2. State of the art

3.2.1. General perspective on parallel manipulators

There are several types of parallel manipulators, but let us first address the usages of these mechanisms. Parallel manipulators are typically suited for two types of applications. Namely, applications that require high stiffness or high payloads and applications that require high speeds and accelerations [14]. Eventually, a design is the product of the trade-off between these two properties. Therefore, it is essential to differentiate between the tasks and applications of these systems. As mentioned in the introduction concerning the limitations of these robots, it is vital to explore these challenges and determine the methods used to address them. Parallel manipulators often have a limited workspace because of physical boundaries, the motion range of actuators, mechanical interference, and system singularities [29]. Furthermore, complex architectures are common and present significant challenges in these fields. The literature identifies various measures that address these challenges. First of all, kinematic redundancy. Which is a state in which the manipulator has more actuators than the ones that are strictly necessary for a complete motion. Whenever a system contains redundancy, its solution for the inverse-kinematic problem has multiple, or sometimes infinite, solutions [25]. This property can be used to decrease the effect of singularities in the system, as a solution tends to be available at all times. The team of Clément Gosselin presents several designs that use this fact, and with impressive results. The 4DOF singularity-free parallel manipulator [13], can access any point within its attainable workspace while allowing unrestricted rotation of the end-effector. Various other designs, by Lacombe et al. [26], highlight this feature, demonstrating its impact on the end-effector's dexterity. This means it not only provides improved reachability but also superior orientation capabilities. While kinematic redundancy offers flexibility and adaptability, it also introduces complexity in terms of control and planning. Determining the best way to use the extra DOFs, especially in real-time or dynamic environments, can be computationally intensive.

3.2.2. Range of motion

Regarding the motion range restrictions of actuators, it implies that actuators cannot operate throughout their complete cycle as movement is halted elsewhere because of mechanical obstructions. One of the

main factors is the orientation of the actuators at the base. The orientation of the actuators in parallel manipulators highly affects their reach, dynamics, and overall performance. Especially, coupling effects on the output motion, more on that later. Nonetheless, various principles for future work have been suggested where the manipulator's base aligns its axis with that of the actuators. As demonstrated in Gosselin et al. [13], the base's column aligns its axis with the rotational actuators, permitting rotation around its base and affording it complete freedom. A variant of this manipulator employs exclusively linear actuators, which enable the mechanism to travel extensively along one axis without restriction. Yang et al. [17] detail the primary concept behind the linear delta, where the relative motion of the actuators leads to distinct end-effector movements. The previously mentioned design by Gosselin that avoids singularities [13] employs a similar approach, providing benefits in expanding the workspace. Appendix B.1.4 gives some more details on this topic.

3.2.3. Mechanical interference

Mechanical interference is also broadly discussed within literature and an often-called causative agent is the coupled character of parallel kinematics. At its core, coupled kinematics refers to the interconnected motion of various components within a system. When one component moves, it influences the motion or position of another component. Moreover, the desired performance of a coupled system hinges on a coordinated integration of multiple input actuators. As a result, coupled kinematics can heighten the susceptibility to singularities, both in inverse and direct forms (section 5.3.2). Decoupling the motion also results in more straightforward control. A collection of designs, collected by the name of multipterons, focuses on the application of decoupling the motion by generating a range of variations. Gosselin et al. showed a range of 3DOF (tripteron) and 4DOF (quadrupteron) [12], where every Cartesian direction is completely decoupled. Each axis functions as an independent actuator for one dimension, but this significantly limits the workspace. Another design found in a video on YouTube [33], enables itself full decoupled motion by integrating specific parameters found by using Screw Theory. Furthermore, Yang et al. [17] describe how simplified architectures can contribute to uncoupled motion and present a modest design capable of performing decoupled motion.

3.2.4. Methods for kinematic analysis

Most of the designs mentioned in the previous section are validated using computational models, as prototyping tends to be time-invested and costly. Various methods are used in the creation of these models in order to find a relation between the actuation input and the motion output, which is the basis of kinematics. One must know a bit about kinematics to understand the following. Kinematics is the study of motion without considering the forces that cause it. In robotics, kinematics allows understanding and prediction of the motion of a robot given certain inputs. For manipulators, particularly parallel manipulators, two main methods arise: forward kinematics and inverse kinematics. Forward kinematics entails determining the end-effector's position and orientation based on known joint parameters, whereas inverse kinematics seeks the required joint parameters for a specified end-effector position and orientation. For serial manipulators, both these problems present their distinct challenges. However, in the context of parallel manipulators, a stark divergence is observed. Specifically, the forward kinematic solution for parallel manipulators is not only non-trivial but often non-unique, leading to multiple possible configurations for the same set of joint parameters. This inherent ambiguity, coupled with the complexity of solving highly coupled nonlinear equations, makes the inverse kinematic problem, which can be decomposed and tackled chain-by-chain, a more tractable and preferred approach in practical applications. Geometric methods follow the architecture of the design in order to find a kinematic relation. For example the Screw Theory or loop closure equations [17]. Screw theory [11] primarily emphasizes synthesis, aiming to determine the parameters that ensure optimal machine operation. On the other hand, loop closure equations [10] are employed to establish kinematic relationships post-synthesis. While screw theory often has the upper hand due to its clarity in computations, establishing loop equations can be more tedious. Although, it allows for a more straightforward understanding of the problem.

3.3. Conclusion

In conclusion, the study of parallel manipulators has demonstrated that while they offer significant advantages in terms of stiffness, payload capacity, speed, and acceleration, these benefits come with

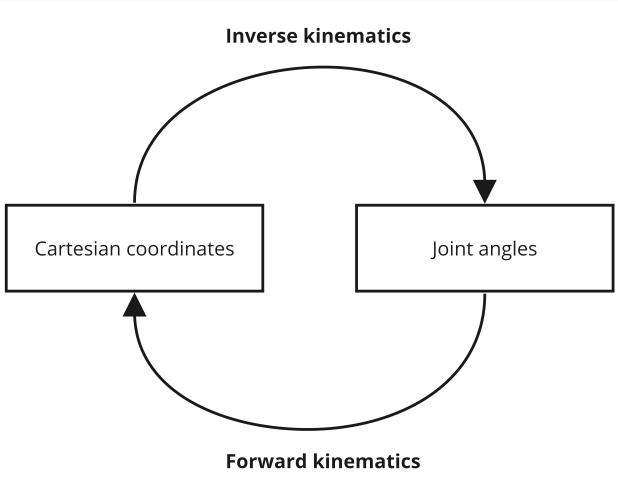


Figure 3.1: How Cartesian coordinates are mapped and the other way around.

inherent challenges such as limited workspace, mechanical interference, and complexities in control due to kinematic redundancy. Approaches like aligning actuator bases, employing linear delta mechanisms, and decoupling motion via specialized designs like multipterons have shown the potential to alleviate some of these issues. Computational models leveraging forward and inverse kinematics have been key in design validation, with methods like Screw Theory and loop closure equations playing crucial roles in understanding and optimizing kinematic relationships. These findings underscore the importance of tailored designs and computational kinematics to fully harness the capabilities of parallel manipulators in various applications.

4

Design generation

4.1. Introduction

At the outset of this chapter, we synthesize our understanding of parallel manipulators and aim to align their capabilities with practical needs. The process begins by delving into the specific functional requirements that govern the deployment of robotic arms in real-world settings. Through discussions with experts (FlexCraft and Greentech appendix G) particularly in agricultural settings, there is a significant emphasis on ensuring maximum reachability. Moreover, the motion trajectory of these robotic systems must be seamless, avoiding singularities that could compromise efficiency or safety in critical zones. Finally, given the tight space limitations, that are often found in settings that handle sensitive goods, it is imperative to design for optimal spatial utilization to minimize the chances of damaging the goods. This chapter presents a final design proposal that marries the technical advantages of parallel manipulators with the concrete operational requirements gathered, setting the stage for a concept that promises both innovation and applicability.

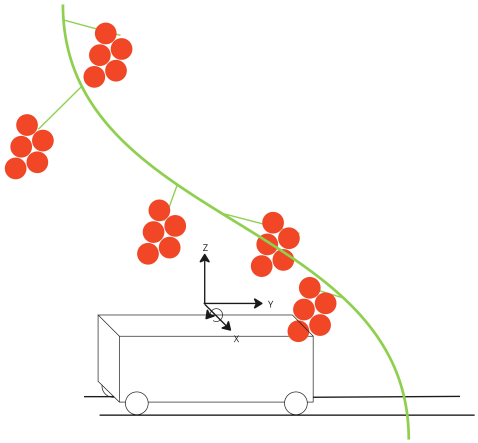
4.2. Functional requirements

4.2.1. Full reachability

As described in the introduction, the main disadvantage of parallel robots is their total reachable workspace. In order to manipulate the total space, the system must follow the principles of 3T1R, which is 3-translational (X, Y, Z) movement and 1-rotational movement (around X). Figure 4.1a shows a schematic image of how this would look in the case of the vine tomato. In order to reach the whole Z-range, and thus all the unevenly spaced crops, the manipulator should be able to travel over a large distance. It is assumed that the cart moves at a constant speed across the Y-axis. When controlled perfectly, this could be sufficient to harvest the crop at the right point. However, in practice, this is not the case as it takes some time to hold and cut the stem. Therefore, the ability to move individually needs to be provided in the Y-axis. Next to this, the manipulator should be able to move in the direction of the crops. The crops are at different distances across the X-axis, therefore the manipulator must compensate for this matter. As figure 4.1a shows, all the crops are at random angles around the main stem. Our manipulator needs to accommodate this and thus needs some degree of rotation around the X-axis.

4.2.2. Smooth trajectory

A smooth trajectory is classified as an uninterrupted path within the workspace. Every pose of a robot comes with a certain workspace in which it can operate. This workspace can be prone to singularities. Farhadi [10] presents singularities as being borderlines which diminishes the performance of the manipulator at that line. Therefore, these can not be crossed during operation. The same singularities cause the PM to be bound to a certain workspace. Wen et al. [41] discuss the dependencies between the reachable workspace and the height of the platform with parallel manipulators. It shows that the shape of the outer boundaries is unique for every PM. Another example from Rudmin [33], where the PM's platform is unable to move at maximum height. This means, that for higher heights of the platform, PMs show lesser performance in the sense of manipulability. It can now be concluded that there is a



(a) Schematic drawing of the theoretical design space

(b) Real vine tomatoes in horticulture [34]

Figure 4.1: Representation of the stem and crops of vine tomatoes in horticulture

trade-off between the translation, orientation, and height of the end-effector and the parallel actuation.

4.2.3. Footprint

The footprint of the robot is the volume of space it needs, in order to operate properly. The robot has to operate in tight spaces with the hazard of damaging stems, crops, and potentially other robots or humans. It is therefore important to provide a compact design, which can easily change its occupational space in order to move past vulnerable objects.

4.3. Selection

4.3.1. Design generation

Table 4.1 shows a range of different types of parallel manipulators, which are further examined in Appendix B.1. All designs and their key points are subjected to the three functional requirements: full reachability, smooth trajectory, and footprint.

With these requirements, the design process was conducted using a more iterative approach. Consisting of a deliberate process of repeated refinement and improvement, the final design originated. This part will describe the method that was used to come to that final design. It is clear now, that conventional parallel solutions will not provide the workspace that is needed for this specific solution. Therefore, a selection of similar principles was made. Using the criteria above, the principle described in Gosselin et al. [13] and Isaksson et al. [21] (B.1.2) was selected as the most promising. Taking the linear actuation principle as an example for generating 3T1R motion, which is a motion device capable of translating in 3 axes and rotating in 1, a minimum of at least 4 linear actuators must be placed. As the crops are vertically, the linear actuators are also placed vertically. Appendix B.2 shows the different iterations. The challenge for now is how to produce 3T1R motion, from an input of 4 linear actuators. The motion can be generally described as follows. Translation over the Z-axis is rather straightforward, as all linear actuators have to move in the same direction. It was also quickly discovered that motion in the X-axis means that the up and bottom sliders must have opposite directions. Translation in the Y direction is done by rotating around the Z-axis, while compensating for height differences with motion in X. At last, rotation around the X-axis can be generated on one side, moving both carts up, and on the other side moving them down. These motion sequences guarantee 3T1R motion.

4.3.2. Final design

The goal of this manipulator is to reach 3-dimensional translation (X,Y,Z) and 1-dimensional rotation (around X). The root of the design consists of 2 4-bar mechanisms which are actuated by sliders at the base. The simulated gripper constraints the distance in the Y-axis of the 4-bar mechanisms and allows

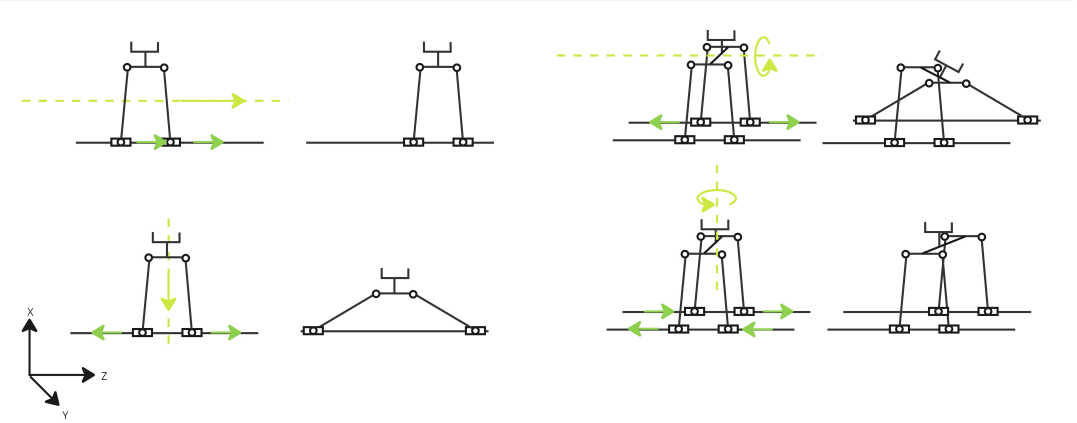


Figure 4.2: Intended 3T1R motion

Appendix	Source	Principle	Type	Score		
				Full reachability	Smooth trajectory	Footprint
B.1.1	[32]	Delta	Hybrid	+-	+	-
B.1.2	[32]	Linear Delta	Pure	+-	–	+
B.1.2	[21]	Linear Delta [1]	Pure	++	++	+
B.1.2	[21]	Linear Delta [2]	Pure	++	++	+
B.1.4	[21]	Axis Symmetry	Hybrid	-	++	+
B.1.2	[13]	Linear Delta	Pure	-	++	+
B.1.3	[33]	Multipteron	Hybrid	+	+-	+-
B.1.3	[12]	Multipteron	Hybrid	-	++	-

Table 4.1: Selection criteria different design principles

rotation in the X-axis and Z-axis. The simulated gripper consists of a prismatic extender and a gripper. In order to constrain the end-effector in rotating around the Y-axis, an extra leg is added to the upper, creating a parallelogram. The relative movement of the sliders is responsible for the movement of the end-effector. More details are found in appendix B.2.1. Figure 4.2 shows the possible movements of the end-effector with the sequence of the corresponding slider. Dimensional analysis of the legs, the prismatic extender, and the platform are of great importance as these parameters will determine the performance of the overall robot. In order to achieve lightweight designs, only kinematic chains which are axially loaded are considered. In order to avoid collisions, all chains are co-linear. All the joints used are revolute joints, which provide simplicity and a solid motion. The use of parallelograms locks the orientation, as the axes at the end of the chains remain parallel. The following chapter will dive deeper into the performance of the design. Appendix B.2.2 describes a prediction of how the reaction forces will look like. These will be further researched in section 5.5.

4.4. Conclusion

The final design adeptly combines the functional requirements with the advantages of parallel manipulators, presenting a compact solution for precision crop handling. With its innovative 4DOF motion capability and vertical linear actuators, the manipulator ensures tailored reach and smooth, uninterrupted movements across the workspace. The strategic use of a dual 4-bar mechanism, parallelograms, and revolute joints results in a minimized footprint while maintaining operational precision. This design meets the specific demands of agricultural environments, illustrating the practical application of parallel manipulator principles in real-world scenarios.

5

Analysis

5.1. Introduction

Now that a concept design has been created, which incorporates parallel principles in a novel robotic manipulator design. It is important to systematically assess the performance of the robotic manipulator to ensure its design functions as intended. By analyzing the kinematics and finding a relation between the actuation and the actual transformation of the end-effector. Using this relation, a range of performance measures are simulated in order to predict the workspace, maneuverability, and validity of the design, with consideration of the necessary assumptions.

5.2. Assumptions

In the study and simulation of robotic systems, various assumptions, and constraints are often introduced to simplify complex real-world scenarios, thus making it possible to develop theoretical models and analytical solutions. This section presents an overview of some common constraints adopted in the realm of robotics simulations:

- **No External Radial Forces:** This constraint posits that the system operates free from any radial forces. In other words, only gravitational forces are considered.
- **Perfect Joints:** Any junction or articulation in the robotic system is assumed to be ideal. This means the joints possess no friction, ensuring unhindered motion between interconnected components.
- **Motors:** Motors integrated into the system function optimally. There is no slipping, ensuring a reliable transfer of power.
- **Beams:** Beams, which serve as primary structural elements, are assumed to be perfectly rigid. This ensures that they remain undeformed regardless of the loads or forces applied.
- **Inertia and Weight:** Simplifying the dynamics, the weight and inertia of the robot's components are considered negligible. This often eases the computation and analysis of the system's motion.
- **Initial Conditions:** Each component of the system begins its operation from a predetermined and consistent initial state. There is an absence of initial deformities or displacements.
- **Environment:** The simulation environment is controlled and consistent. Whether it's a vacuum or standard atmospheric conditions, factors like temperature, humidity, and air resistance are disregarded.
- **Electrical and Electronic Delays:** To ensure real-time response and execution, all electronic components, from sensors to actuators, operate without any latency. The flow of information is instantaneous.

- **Perfect Power Supply:** All electronic components, be it motors or sensors, receive an unvarying and continuous power supply, ensuring consistent performance.
- **No Wear and Tear:** A crucial simplification, components are presumed to always operate at their peak performance, with no degradation over time.
- **Perfectly Aligned:** Manufacturing and assembly imperfections are overlooked. Every component fits and aligns flawlessly
- **Kinematic Linearity:** The relationships governing the system's motion are linear, which often simplifies the mathematical modeling and analysis.
- **Sensors:** Should the system utilize sensors, they are assumed to be impeccable—characterized by zero error, infinite resolution, and complete absence of noise.

5.3. Kinematics

5.3.1. Constraining the system

First, it is important to make sure the system is properly constrained. This means the system does not perform unintended motion. The Degrees of Freedom (DOF) is the indicator of the motion freedom a system has. For example, a 1DOF system is able to move in one specific axis. For a completely constrained system, the DOF is 0, which indicates that the system is locked and unable to move. Every DOF a system has can be actuated in order to perform certain motions, which is important for this case. The DOF of a system is determined by the freedom a rigid link has, which is 6 for a spatial case. This freedom is constrained by joints, for example, a revolute joint, which limits the link to only rotation, or 1DOF. Determining the DOFs for a special system, like this one, is a tricky one. The Kutzbach-Grübler criterion [37] is often used. However, this formula treats every constraint as an active one. In parallel mechanisms, like this one, joints work together to constrain a motion, which makes some of these dependent constraints. Because their motion depends on the constraint of another joint. Dai et al. [7] propose a variation on the Kutzbach-Grübler equation, which accounts for these dependent constraints.

$$m = d \cdot (n - g - 1) + \sum_{i=1}^{g} f_i + c \tag{5.1}$$

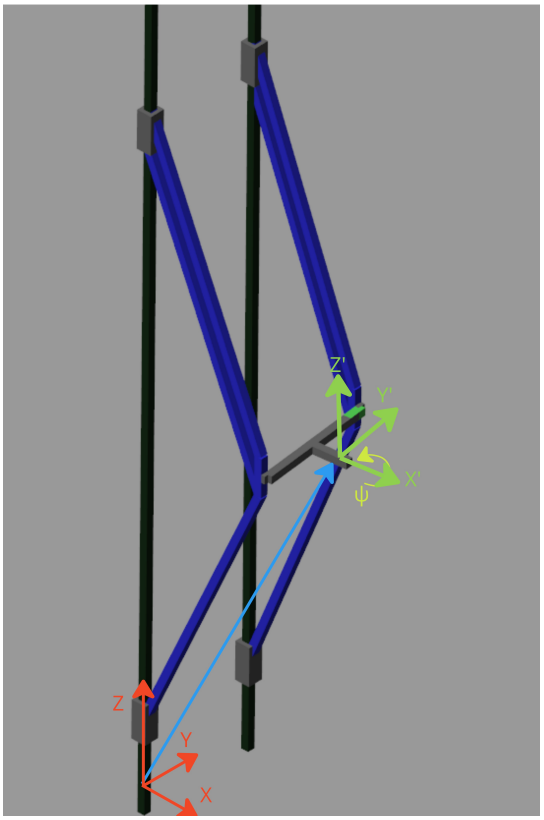
Where m is the system's DOF, n is the number of rigid links (bodies), including the rigid world, in the system, d is the spatial freedom being 6 for a spatial case, g is the number of constraints, f_i is the number of DOFs per constraint and c counts for all dependent constraints. The simulated model uses 16 revolute joints, 1 passive prismatic joint, and 4 actuated prismatic joints. Also, 16 bodies, plus the rigid world (total 17), are present. When accounted for all the passive joints, equation 5.1 gives a DOF of 4, as shown in appendix B.3.

$$m = 6 \cdot (17 - 21 - 1) + 21 + 13 = 4 \tag{5.2}$$

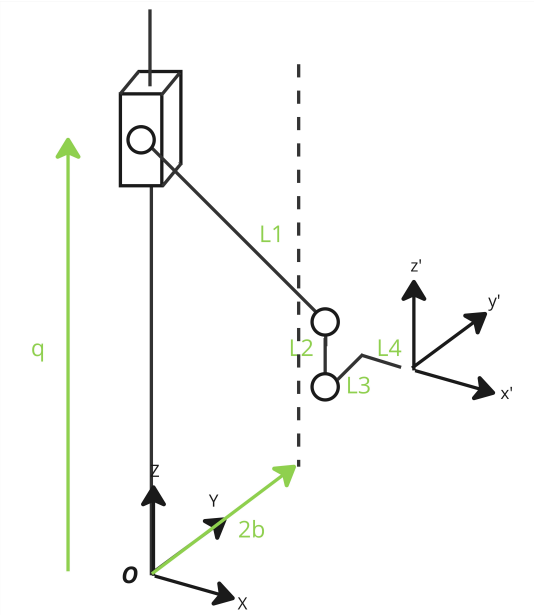
This number is equaled by the number of actuated joints, which means every DOF can be controlled using the actuators. This results in a system's DOF = 0, when all the actuators are locked.

5.3.2. Inverse Kinematics

Now that is known that the system is properly constrained, a set of equations must be found in order to establish the relation between the Cartesian position of the end-effector and the position of the actuated prismatic joints, which is inverse kinematics. Therefore, the assumption is that the Cartesian position of the end-effector is known. Although, the method of forward kinematics has been tried, but has not led to a solution. Vector loop equations are used to find all the unknowns in the system. This method uses vectors to represent the open chain to the end-effector and adds a straight vector from the base to the end-effector to close the vector loop. This process is shown in figure C.1. In this case, it produces 8 equations to solve the total of 8 unknowns ($\alpha_1, \alpha_2, S_{extend}, \phi, q_1, q_2, q_3, q_4$) in the system, which is further explained in appendix C.2 and the code is found in appendix F.2. The solution of the variables $\alpha_1, \alpha_2, S_{extend}$, and ϕ is not unique, which means there exists a second solution. The second solution



(a) Mapping world frame to the end-effector



(b) Description of the variables for the upper left chain

Figure 5.1: Supportive figures for understanding the system

L_1 [m]	L_2 [m]	L_3 [m]	L_4 [m]	b [m]
0.57	0.05	0.16	0.10	0.16

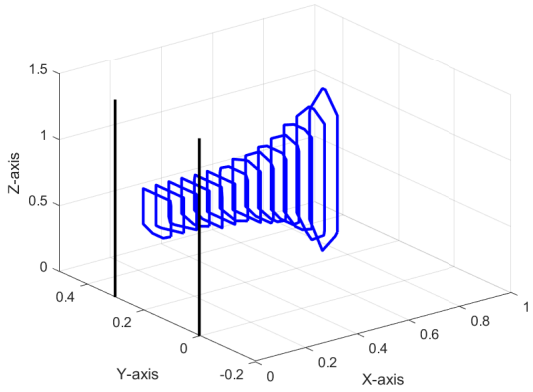
Table 5.1: Parameters found in figure 5.1b

exists only for the situation in which the direction of movement in X is flipped. In this practice, this is not possible and thus these solutions are taken out. Notable is the origin of the movement in the Y-axis, which is a coupled motion consisting of a rotation around the Z-axis and a translation in the X-direction. What is left is the kinematic transformation from the Cartesian output to the position of the sliders in the form of $q_i = f(x, y, z, \psi)$. Where q_i is the position of the prismatic actuator. Furthermore, the input values $x, y, z,$ and ψ, are to be given by the user. Appendix C.2 shows the details of acquiring the equations. The parameters for further analysis are shown in table 5.1.

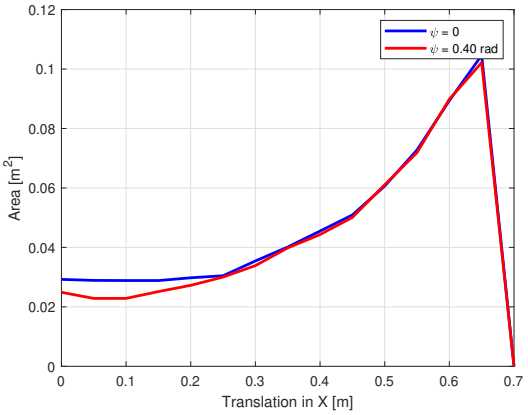
5.4. Workspace

5.4.1. Reachable workspace

Analyzing the workspace offers insight into the performance of the system. More precisely, the reachable workspace is an indicator of where the end-effector can reach. Therefore it is essential to have information about the volume that can be reached in a singular-free way and in any end-effector position. Using the inverse kinematic relation from section 5.3.2, an iteration is done through the total workspace of in total n^3 points, with $n = 40$ (Matlab code in appendix F.4). Here, the power three stands for the three Cartesian dimensions X, Y, Z. For every point in space, the physical conditions of the sliders are examined and are validated or discarded according to this. The results are then shown in figure 5.2a as the outer profile against the translation in X of the reachable workspace. As discussed, extra rotation influences the reachability of the end-effector and is therefore also plotted in figure 5.2b. For reference, the black lines indicate the linear actuators.



(a) Graph of the workspace surface as a function of the translation in X-direction



(b) Surface graph

Figure 5.2: Area as a function of the translation in X

5.4.2. Singularities

One main reason for not reaching all the points in the workspace is the singularities. As is explained in section 3, singularities cross the workspace as lines and degenerate the performance of the system. It is important to map these singularities, in order to avoid them. In order to further analyze the system and its singularities, the Jacobian matrix is calculated. Equation 5.3 shows this relation, where $\vec{\dot{x}}$ output velocity and $\vec{\dot{q}}$ is the input velocity [10]. A is the direct-kinematic Jacobian and B is the inverse-kinematic Jacobian. Hereafter called by respectively $\mathbf{J}$ and $\mathbf{J^{-1}}$. The Jacobian is presented using the equations for the input positions, differentiated over the Cartesian output of the end-effector, like in equation A.1.

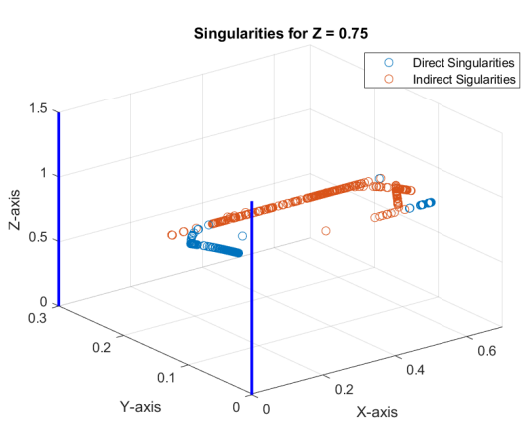
$$\mathbf{A} \cdot \dot{x} + \mathbf{B} \cdot \dot{q} = 0 \tag{5.3}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial q_1}{\partial x} & \frac{\partial q_1}{\partial y} & \frac{\partial q_1}{\partial z} & \frac{\partial q_1}{\partial \psi} \\ \frac{\partial q_2}{\partial x} & \frac{\partial q_2}{\partial y} & \frac{\partial q_2}{\partial z} & \frac{\partial q_2}{\partial \psi} \\ \frac{\partial q_3}{\partial x} & \frac{\partial q_3}{\partial y} & \frac{\partial q_3}{\partial z} & \frac{\partial q_3}{\partial \psi} \\ \frac{\partial q_4}{\partial x} & \frac{\partial q_4}{\partial y} & \frac{\partial q_4}{\partial z} & \frac{\partial q_4}{\partial \psi} \end{bmatrix} \tag{5.4}$$

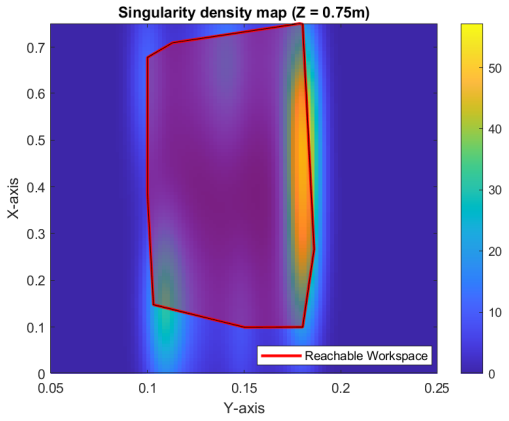
The Jacobian matrix relates the velocities in the joint space to the velocities in the task or Cartesian space for robotic manipulators, using the derivatives of q_1, q_2, q_3, q_4. To analyze for type I singularities, or direct singularities, one can take the determinant of the Jacobian and equal it to zero: $det(\mathbf{J}) = 0$. For type II, or indirect singularities, one takes the inverse of the Jacobian and proceeds the same way: $det(\mathbf{J^{-1}}) = 0$. These sets of equations can be solved for the input variables: X, Y, Z, ψ. In Matlab, three different cases are run which require a non-linear solving technique. The Levenberg-Marquardt algorithm is then used to iterate to a solution, as it deals with a set of non-square systems. Two different cases are solved:

- $det(\mathbf{J}) = 0$. Type I or direct singularities occur when a robot's end-effector is in a configuration where certain directions of movement are unachievable regardless of joint velocities. This causes small forces or torques applied to the robot's end-effector can translate into very large, sometimes infinite, forces or torques in the actuators.
- $det(\mathbf{J^{-1}}) = 0$. Type II singularities often cause the robot to have redundant degrees of freedom, which makes it uncontrollable in these regions.

The optimization function takes a raster of n^2, with $n = 20$, initial guesses as input, and shows a heatmap of the end-effector position at $Z = 0.75$, in figure 5.3b. This can be interpreted as the configurations in figure 5.4. Details of the code used are in appendix F.3.

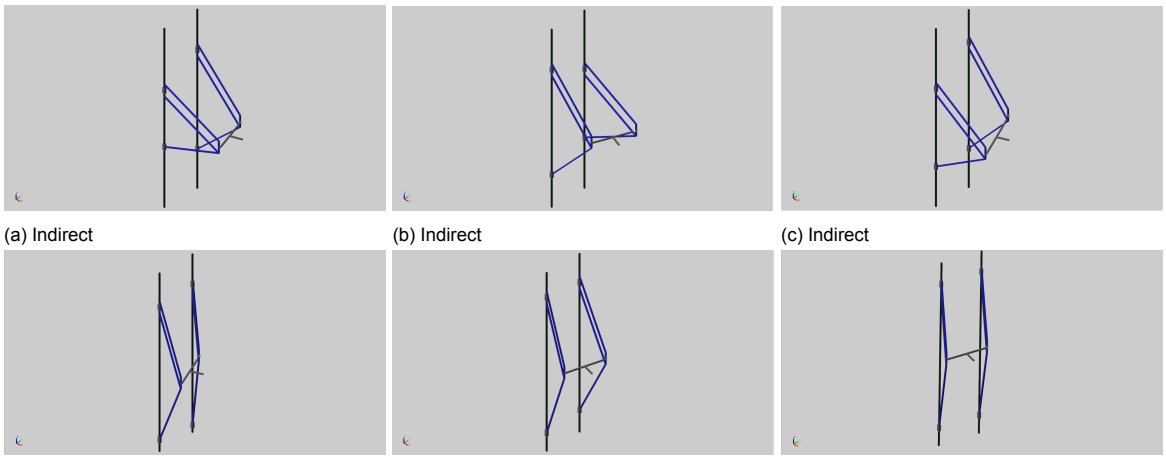


(a) Indirect and direct singularities for Z = 0.75m



(b) Color map showing the intensity of the singularities within the reachable workspace for Z = 0.75m

Figure 5.3: Singurities at the Z-plane (Z = 0.75m)



(a) Indirect (b) Indirect (c) Indirect

(d) Direct (e) Direct (f) Direct

Figure 5.4: Direct and inverse singular positions

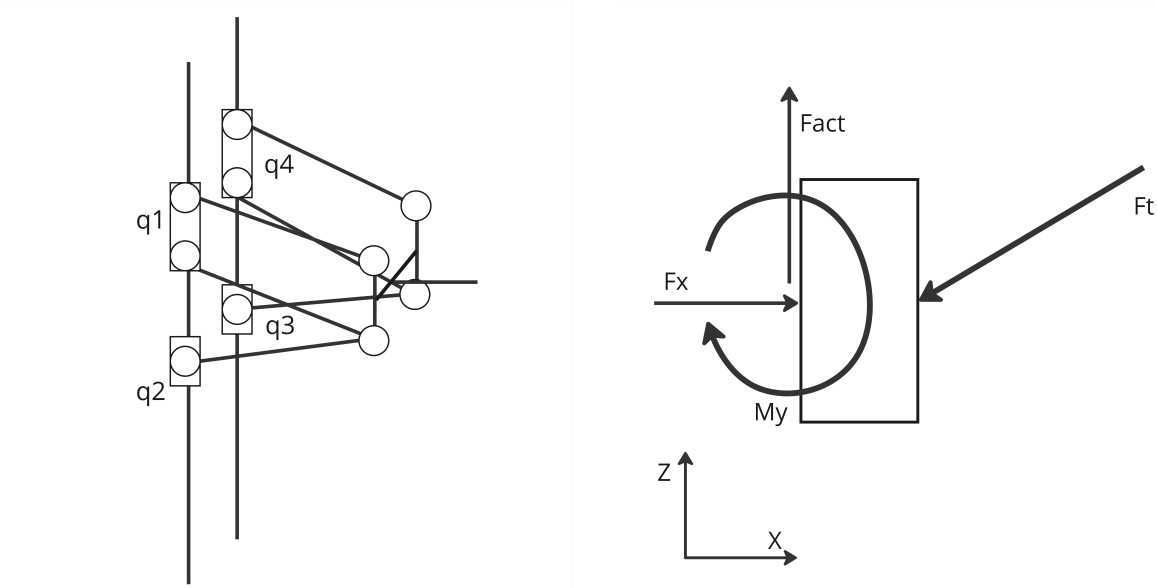
5.5. Simulation

Now that is know how to move the sliders, in order to move the end-effector to our points of interest, one must validate the assumptions made. In Simulink, a physical model is created which takes the slider position as an input. In order for this to follow a preconceived path, a trajectory planner (Appendix C.1) is used and fed to the transformation from section 5.3.2. The trajectory path integrates various waypoints, time intervals, and sample counts to produce a smooth route based on a trapezoidal velocity profile of the end-effector, implying a constant acceleration at the end-effector. Furthermore, the components' inertial properties are shown in table 5.2. Either a point mass or a density combined with geometric properties is simulated. The simulation solves for reaction forces, explained in figure 5.5, on the sliders, as well as velocity and acceleration. The position of the end-effector is tracked and shown through an animation found at http://bit.ly/videosthesis. Each of the motions described in figure 4.2 is reproduced in the model, namely translation in X, Y, and Z and rotation around X called ψ. The Z motion has been separated into Z and $Z*$, where Z implies $X = Xmin$ and Z^* implies $X = Xmax$. These motions show very different behavior and cannot be captured in one plot. The simulation output is shown in table 5.3. Eventually, the maximum motor torque (T_{act}) is acquired. Finally, the normalized Root-Mean-Squared error between the input position and the output position of the sliders is found in the last column to test the accuracy of the simulation. Appendix C explains more about the original figures. The images from figure 5.7 to figure 5.11 illustrate the anticipated trajectory of the end-effector alongside the simulated pathway. For context, the linear guides are depicted with blue lines. Adjacent to this trajectory, there

	Guide	Slider	Leg	Gripper	End-effector
Mass [kg]	-	-	-	-	1.0
ρ $[kg/m^3]$	2700	2500	2200	2200	-

Table 5.2: Simulation values

is a graph displaying the actuator torque (T_{act}), which represents the torque for each motor.



(a) Schematic overview of the design

(b) Force plot of $q1$ and $q3$, representing the reaction forces

Figure 5.5: Overview of the situation

5.6. Conclusion

In summary, the main findings of the text clarify the robotic system's theoretical underpinnings and simulated operational capabilities. It confirms that the manipulator's design adheres to the kinematic constraints, as outlined by the modified Grübler–Kutzbach criterion, ensuring the system is properly constrained with a determined degree of freedom. The inverse kinematics are successfully formulated, excluding non-viable solutions and allowing for precise control over the manipulator's end-effector through actuated prismatic joints. The workspace analysis provides valuable insights into the regions the end-effector can effectively reach, including the identification of singular points where movement constraints are observed. Lastly, the simulation validates the robotic system's performance against the established assumptions, with trajectory planning demonstrating a controlled path of motion and the actuators' reaction forces and torques indicating the system's dynamic responsiveness. The normalized Root-Mean-Squared error metric indicates the accuracy of the simulation when compared to the expected outcomes.

Action	Duration [s]		Max. M_y [Nm]	Max. F_x [N]	T_{act} [Nm]		norm RMSE model [m]
X-translation	2s	q_1	-12.34	-14.68	0.1160	X	0.1860
		q_2	7.792	14.68	0.1109	Y	-
		q_3	7.481	14.15	0.1094	Z	-
		q_4	-11.83	-14.15	0.1145	ψ	-
Y-translation	4s	q_1	-4.587	-3.689	0.08182	X	0.001400
		q_2	1.442	3.689	0.0623	Y	0.2060
		q_3	0.9961	3.193	0.09560	Z	-
		q_4	-3.512	-3.193	0.07988	ψ	-
Z-translation	5s	q_1	-1.799	-1.086	0.05623	X	-
		q_2	0.3631	1.086	0.05006	Y	0.00500
		q_3	0.3482	1.042	0.04852	Z	0.2130
		q_4	-1.736	-1.042	0.05469	ψ	-
Z-translation *	5s	q_1	-19.99	-10.82	0.06041	X	-
		q_2	15.63	10.82	0.05358	Y	0.1037
		q_3	14.99	16.03	0.05190	Z	0.1991
		q_4	-19.16	-16.03	0.05871	ψ	-
ψ-rotation	3s	q_1	-3.921	-2.239	0.06068	X	-
		q_2	0.6733	2.254	0.05551	Y	0.0144
		q_3	0.5199	2.306	0.04599	Z	0.00900
		q_4	-3.097	-2.287	0.05221	ψ	0.2365

Table 5.3: Force and torque output of the different motions. The last column interprets how accurately the simulation was able to follow the input (Appendix C).

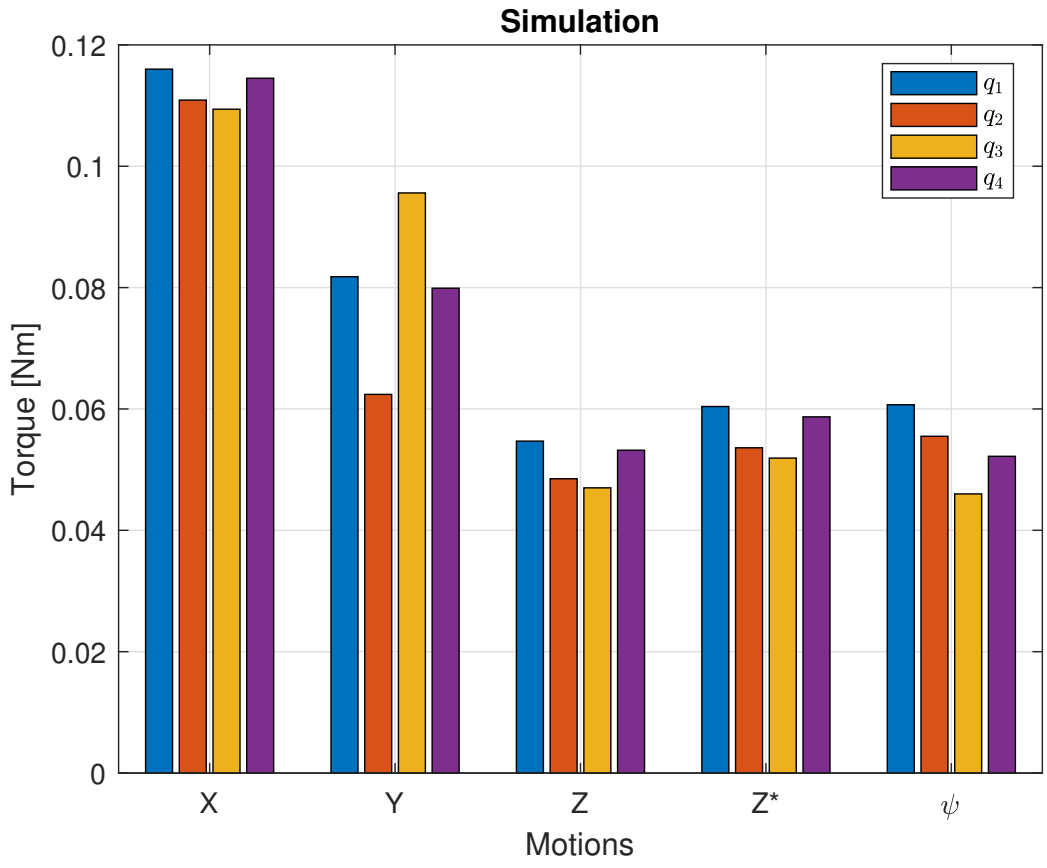


Figure 5.6: Maximum M_y and maximum F_z per motion

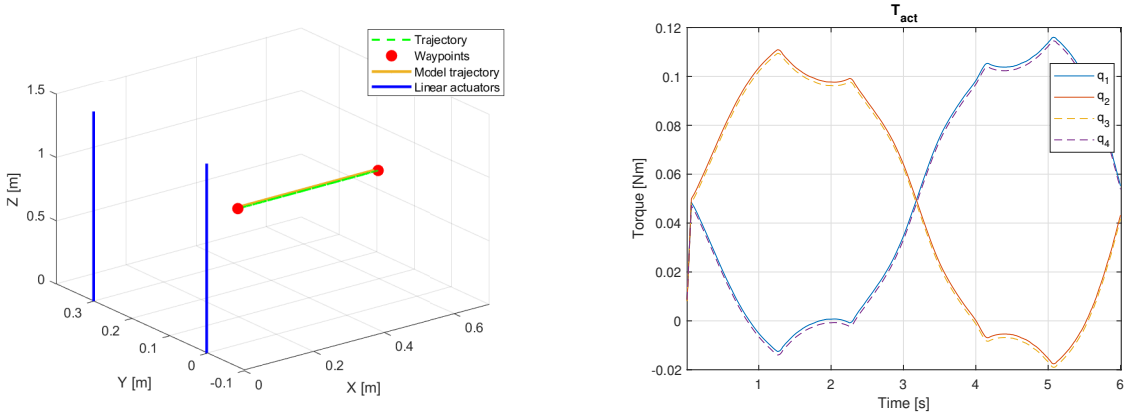


Figure 5.7: Path of the end-effector with the motor torque T_{act}

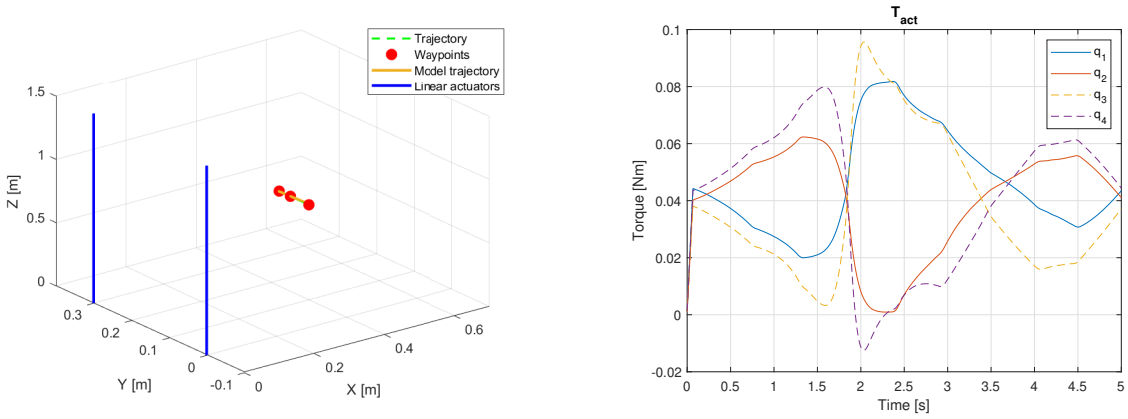


Figure 5.8: Path of the end-effector, while translating in Y, with the velocity and acceleration profile of the sliders

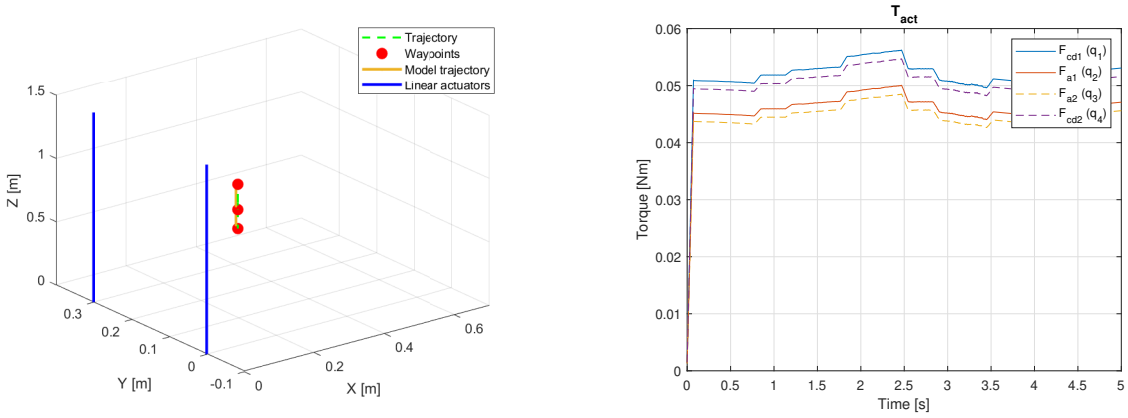


Figure 5.9: Path of the end-effector, while translating in Z, with the velocity and acceleration profile of the sliders

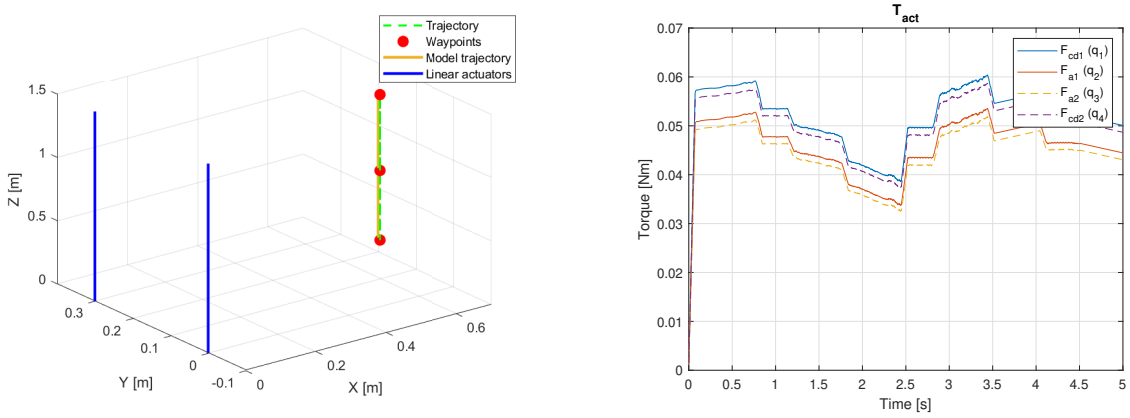


Figure 5.10: Path of the end-effector, while translating in Z, with the velocity and acceleration profile of the sliders

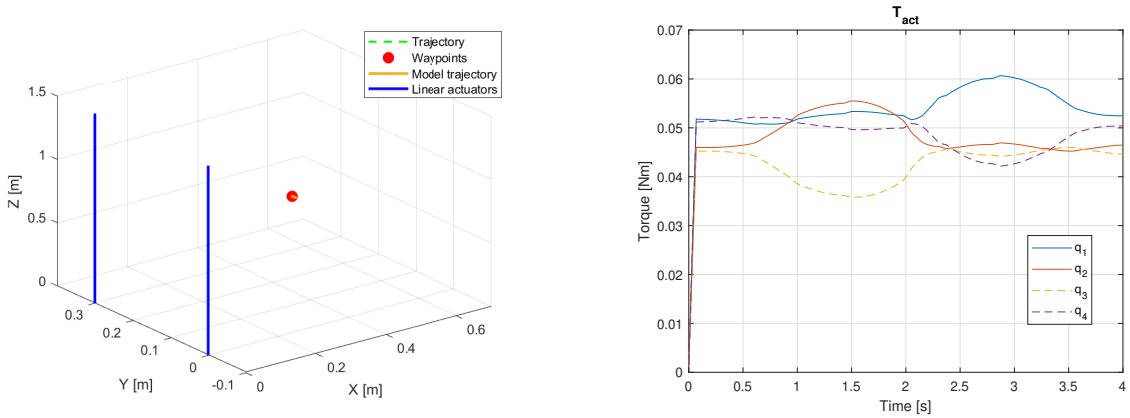


Figure 5.11: Path of the end-effector, while translating in ψ, with the velocity and acceleration profile of the sliders

6

Testing and validation

6.1. Introduction

The model, while idealized, doesn't encompass every constraint. To bridge the gap between theoretical and practical application, the Simulink model was transformed into a tangible representation of the design. An iterative approach involving design in SolidWorks followed by rapid prototyping was adopted to build the eventual prototype. More details on the working principles are found in Appendix D.1. Further exploration of the theoretical assumptions is conducted through empirical tests with the prototype, as detailed in the provided appendix. These experiments are crucial for verifying the practical workspace dimensions, the dynamics of reaction forces, and the actual motor torque necessary for operation. Which eventually will give insight into the practical implementation of the parallel principles. These experiments will refer to several motions made by the manipulator, which can be viewed in Google Drive (http://bit.ly/videosthesis).

6.2. Experiments

To validate the performance of the kinematic design, three experiments were conducted. The first experiment would determine the usable workspace of the manipulator, ensuring the end-effector can reach all desired positions, also including singularities. The second experiment focuses on the basic motions the manipulator can do in order to compare with the predicted torque values from section 5.5. The final experiment tests the payload capacity and stability of the mechanism. Here, the behavior of the mechanism is tracked while constantly increasing the load at the tip. At our disposal are several sensors in the motors, which can capture position, velocity, and motor torque. These motions were programmed using Codesys. For more details about the use of Codesys, please read appendix D.2.3.

6.3. Results

6.3.1. Kinematic test

In order to safely find the outer limits of the kinematic design, a program is used to step-wise increase the Cartesian coordinates and measure the real positional output. The range of measured values is to be found in table 6.1.

	ground	min	max	range
X [mm]	300	100	650	550
Y [mm]	150	100	180	80
Z_{Xmin} [mm]	750	600	950	350
Z_{Xmax} [mm]	750	200	1350	1150
ψ [rad]	0	-0.45	0.45	0.9

Table 6.1: Minima and maxima of the positional range of the kinematic design

6.3.2. Singularity test

Section 5.4.2 details the location of singularities at the workspace's outer edges. These singularities are categorized into two distinct areas of the workspace, as depicted in figure B.5. The motor torque response near region 2 is illustrated in figures 6.5, revealing an additional peak in motor torque. This occurs in the configuration represented in figure 6.3, which is near a singularity point. Indirect singularities are identified by an increase in DOF and are found at the edge of region 1. This is demonstrated in two different configurations, shown in figure 6.2 and figure 6.4. It is important to note that points beyond these designated regions are not considered, as they fall outside the workspace's reachable scope.

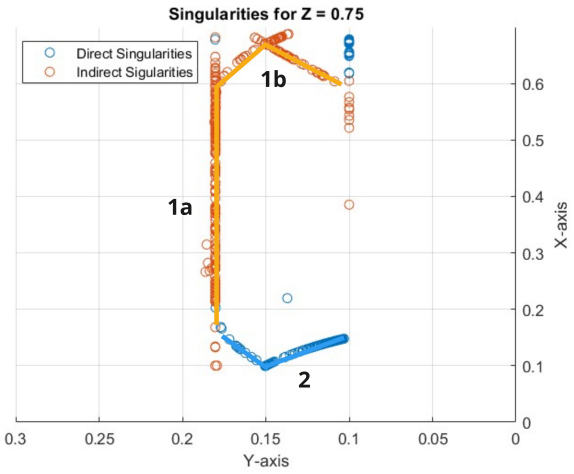


Figure 6.1: Interpretation of the singular regions shown in figure 5.3a

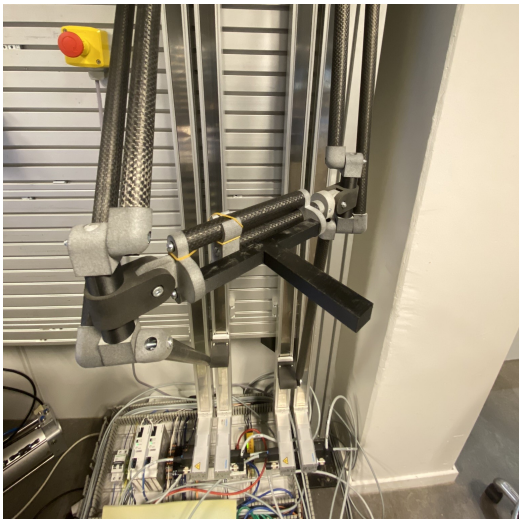


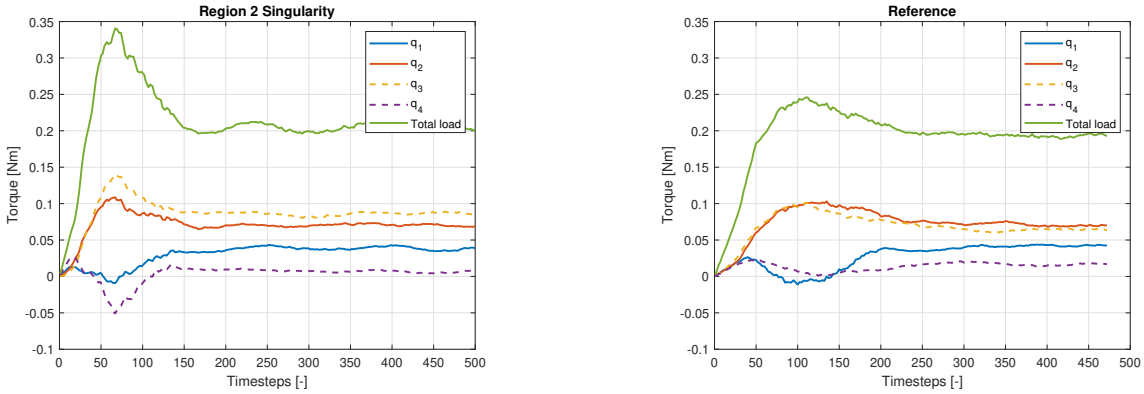
Figure 6.2: Singular point for region 1a

Figure 6.3: Singular point for region 2

Figure 6.4: Singular point for region 1b

6.3.3. Basic motion tests

The motions from table 5.3 are programmed into the PLC. Each motion is repeated three times. Sensor data on the torque in the motor is captured with every motion, which allows insight into the behavior of the kinematic design. Figure 6.6 shows the maximum motor torque in every motion, which indicates the distribution of the torque across all motors. It is shown that the range of these maxima lies within 10% of the mean value, which indicates a fairly distributed motor load. Furthermore, the torque distribution among the motors varies significantly based on the motion being executed. No single motor consistently



(a) Motion at close-to-singular point: X = [0.17 - 0.25]

(b) Reference motion: X = [0.32 - 0.40]

Figure 6.5: Comparable movements showing the difference when close-to-singularity

produces the highest torque across all motions. This variability indicates that each motor might be specialized or more involved in specific motions. For example, motor q1 might be more influential in Z motion, while motor q3 seems to be essential for X motion.
The repeatability of the runs is shown in figure 6.7, where the normalized RMSE is calculated of the **total torque** at every time instance. The mean of the standard deviation tells about the consistency with which the tests can be conducted. If there are large dissimilarities, then there might be inaccuracies in the system. The motions that require all sliders to do the same procedure (Z and Z*), figures 6.7c and 6.7d, seem to be rather precise. There is also a recognizable similarity in figure 6.7a, which requires the sliders to perform the same, but opposite, motion (X). The motion that requires all sliders to move individually (ψ and Y), like in figure 6.7b and 6.7e, contains more disturbances.

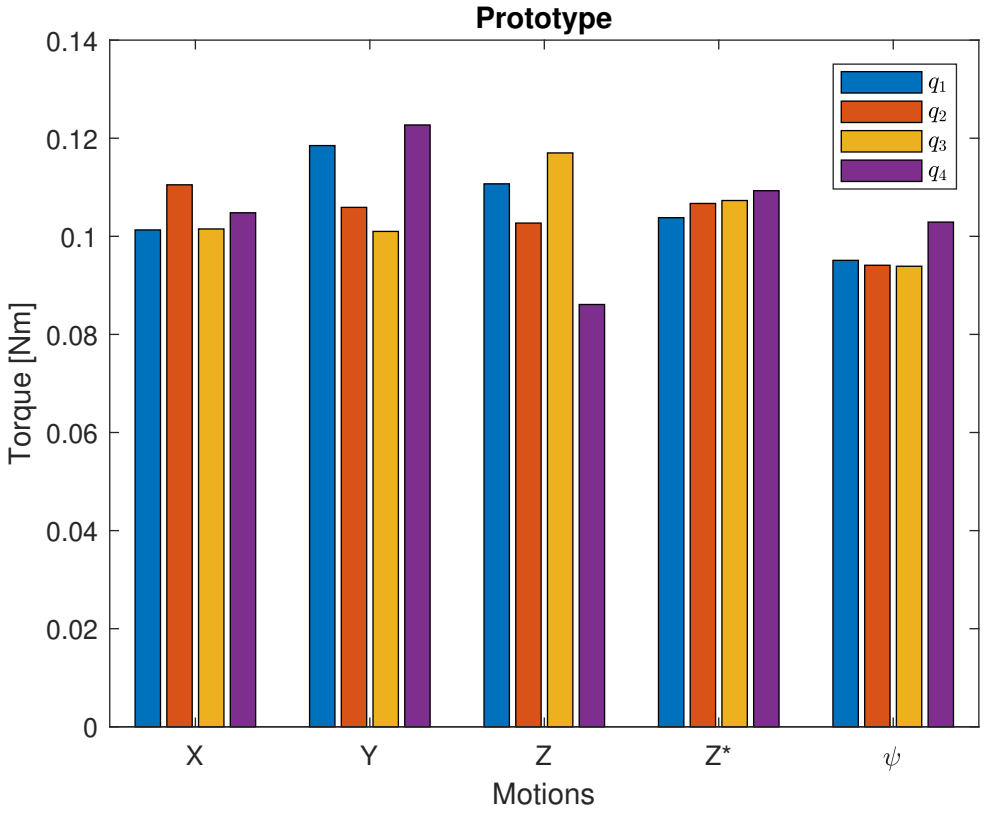
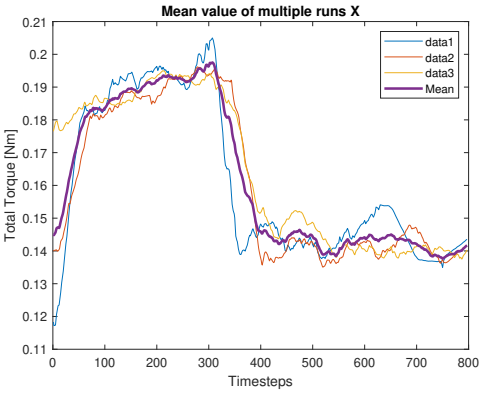
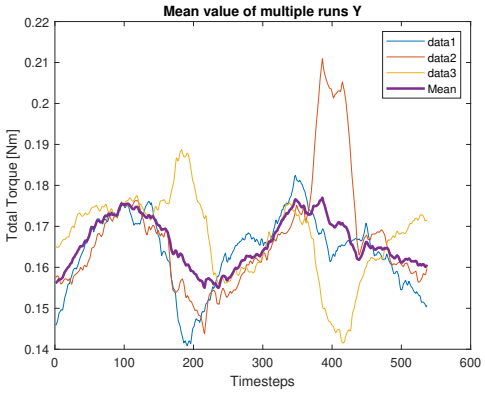


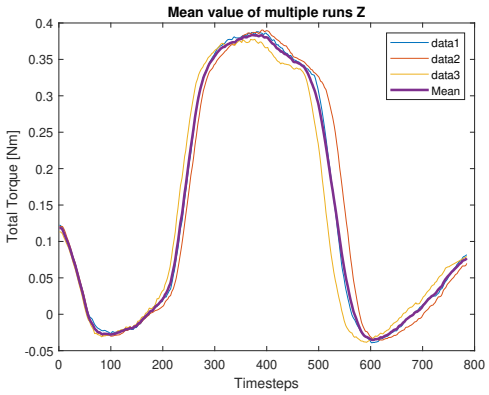
Figure 6.6: Maximum torque per motion for $v_{slider} = 1.2m/s$ and $a_{slider} = 6m/s^2$



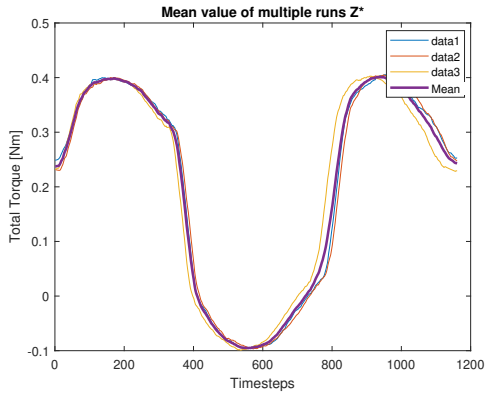
(a) Normalized RMSE = 0.2701



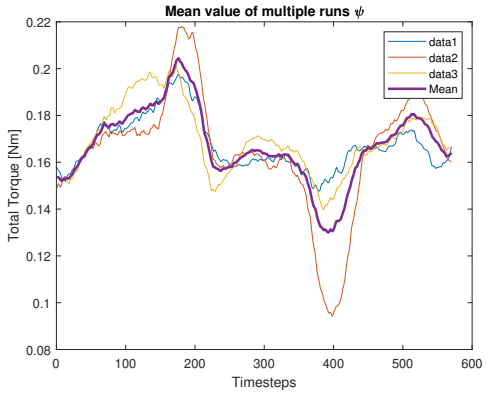
(b) Normalized RMSE = 0.9079



(c) Normalized RMSE = 0.1250



(d) Normalized RMSE = 0.1033



(e) Normalized RMSE = 0.4019

Figure 6.7: Multiple runs for every motion showing the repeatability of the test

6.3.4. Payload capacity and stability test

This examination focuses on evaluating the structural boundaries of the kinematic configuration. Here the goal is to find a relation between the weight at the tip and the torque distribution. The PLC has been programmed with three movement sequences, being the translation in X, Y, and Z. Stepwise, the amount of weight at the tip is increased. Each run consists of three repetitions of the motion. Figure 6.8 shows how the mass at the tip can be increased, while velocity and acceleration remain constant. The torque at the motors tends to have a linear relation. Figure 6.9 shows an extrapolation of that graph indicating the theoretical range of mass. The theoretical limit of the motors combined lies at 2.0 Nm.

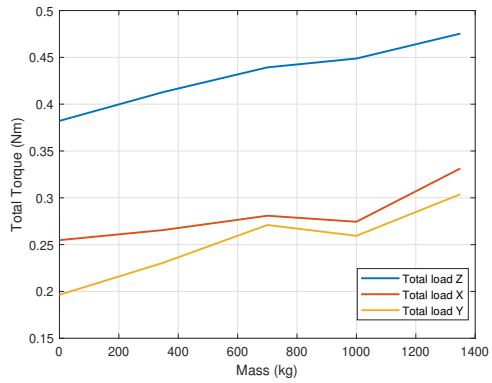


Figure 6.8: Measured torque against the extra applied mass at the tip, while velocity and acceleration are kept constant

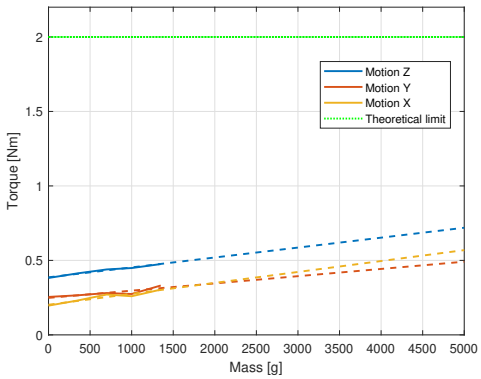


Figure 6.9: Extrapolated data from figure 6.8 showing the theoretical mass range

6.4. Conclusion

In conclusion, the journey from a Simulink model to a physical prototype has validated the initial theoretical assumptions while exposing practical limitations. The workspace dimensions of the prototype were confirmed to align with the expected ranges, affirming the design's spatial efficacy. Motor torque distribution was found to be balanced across varied movements, albeit with a notable variation that underscores the complexity of dynamic force interactions within the system. Repeatability tests revealed high precision in synchronized slider movements but indicated a need for improvement in independent slider operations. The payload tests demonstrated a clear linear torque-to-mass relationship, pointing to predictable and scalable performance under increasing loads. Ultimately, this empirical evaluation has not only corroborated the theoretical model but also highlighted areas for refinement, ensuring that future iterations of the design will enhance precision and reliability. The findings pave the way for further research and development, with the goal of achieving an even more robust and accurate parallel kinematic mechanism.

7

Discussion

7.1. Introduction

At the start of this thesis, the problem of the technological adoption of picking robots in greenhouses was addressed. One of the reasons that is mentioned is the high investment costs of robotic solutions, while only a part of the robot's capacity. Therefore, the main goal was to design a robotic manipulator, which is cost-effective and capable of the task of harvesting. This translation resulted in a parallel manipulator, which assumably offers greater stiffness, reduced moving mass, and uses less material. However, to address the frequent issue of limited workspace, a decision was made to conceptualize a kinematic design with 4DOFs, powered solely by linear actuators. Called the linear delta. The design of the linear delta has been elaborated in the case of the tomato harvester. The sector of horticulture has proven to benefit from an applied robot arm that is better suited for payload, reachability, uninterrupted trajectories, and volume occupation.

7.2. Key findings

7.2.1. Payload and load distribution

The payload-to-weight ratio is higher compared to ordinary serial robot arms, which is examined in section 6.3.4. It was shown that the capacity of the manipulator exceeds the load of the task many times, a vine tomato in this case. The parallel kinematic structure allows for efficient distribution of the weight of the manipulator, as it is only moving the weight of the carbon instead of heavier parts, like other motors. That is also why parallel principles contribute to cost efficiency because the motors are placed at the base. Next to a high payload, it is important to address how this load is distributed along the motors, in order to fully utilize its capacity. The basic motion test from section 6.3.3 shows that different motions, have different peak producers of motor torque, however always within a certain range compared to each other. The benefits of a fairly distributed load are described in Damerla et al. [8], when the load can be shared it tends to be more predictable and is less vulnerable to errors and sudden peak load. Additionally, fewer peak loads and unpredictable behavior lead to better use of motor capacity in the end. The increase of the load means an increase of reaction forces at the guides. Not only does this influence the motor torque, but it also requires the guides to be robust in order to withstand these forces. The reaction forces were simulated and shown in table 5.3 and these results influenced the choice of components. These values can also influence the dynamic limits of the system, so care is needed when selecting components or determining speeds and accelerations.

7.2.2. Workspace

Analysis of the reachable workspace has led to the visualization of the workspace and its vulnerabilities. Each manipulator's workspace has its own unique shape, as stated by Yang et al. [42], which is characterized by the kinematic design. For the specific design examined in this context, the workspace demonstrates enhanced reachability when the end-effector is positioned further from its base. Additionally, the capability of the end-effector to rotate up to 0.4 radians at every point within the workspace further exemplifies the design's operational flexibility. In summary, the findings regarding the shape

and capabilities of the manipulator's workspace are in line with existing literature, reinforcing the critical role of kinematic design in determining a robotic arm's operational range and capabilities. As described in chapter 2, the workspace of a parallel manipulator is crossed with singularities, making it an important aspect to consider. The singularities are shown in section 5.4.2, where they are all found at the borders of the workspace. The location of these points seems logical when the kinematic model counts for X, Y, and Z translation. Y therefore has a range, in which the coupled motion can provide sufficient output. Outside of this range, the tip of the end-effector starts to rotate which means it gains a degree of freedom, which becomes worse when the end-effector moves more out of range. The difference in singular points between left and right has to do with the non-symmetric character of the end-effector. One direction of Y has better reach. The singular points of region 1b, in figure B.5, happen when the manipulator is fully out and the legs approach a parallel configuration. At this point, an extra DOF is introduced which allows rotation about the Y-axis, which leads to the unavoidable collapse of the mechanism. Secondly, direct singularities are also found at the minima for X as the legs are almost co-linear, which has been shown to cause high motor torques and degenerated positional output of the end-effector. Therefore, one must know that the system can be unpredictable in these regions. Within the regions, the workspace is very reliable. The uncoupled character of this manipulator offers an uninterrupted workspace within the singular-free part. This makes it extremely useful for applications where the operational space is known.

7.3. Interpretation

This thesis has shown that a different approach to manipulator design can make the payload-to-weight ratio seem more fair, as the weight of the manipulator is more comparable to the weight of the crop. Furthermore, the interpretation of the robot in horticulture sends us back to figure 4.1a, where the cart was initialized. Now the manipulator is placed on the side of the cart and moves at a constant speed. This speed is determined by the speed of harvesting. Assumed is that previous steps, like detection and ordering the task, are all completely correct. Then the system moves towards the position and starts its harvesting task. Since Y has a range of 8 cm and no singularities are within the workspace, it can hold the position while the cart is driving. Assume the harvesting task takes 8 seconds (holding and cutting the peduncle), which results in a constant cart speed of $0.01m/s$. This may not be effective enough for a healthy yield. However, one could choose to implement multiple manipulators which can increase the yield. Since the reachable workspace is only in front of the manipulator, no communication is necessary as chances of collision are nihil. The workspace must match the space where crops are present. Since the workspace increases for larger X values, a division is made between just transportation of the end-effector and effective harvesting as shown in figure 7.1. Ultimately, the geometric parameters and placement can be changed slightly to fit the constraints.

7.4. Limitations

There are some strong limitations on the use and testing of this system. Throughout this research, no other external forces except for gravitational forces have been considered, making the mechanism prone to damage caused by small collisions. Although it shows excellent resistance against gravitational forces, it is susceptible to radial forces and moments. The advantageous stiffness comes at a price, which is the many rotational joints the system possesses. Every joint, as with every attachment point, comes with a certain backlash which obviously adds up to the end-effector. This affects the accuracy of the system as well as the controllability. Seen in the experiments the repeatability then becomes harder, as redundancies in the system appear, which allows for discrepancies in the output data. This phenomenon is shown in figure 6.7. When there is no rotation of joints, measured data has a small RMSE value. However, when there are increased rotational activities in the joints the data tends to be more unclear. Therefore it is important to put effort into finishing the joints. Further inaccuracies can be tackled with closed control, by measuring positional data at the tip. A transformation between the Cartesian coordinates of the end-effector and the positions of the sliders was made using inverse kinematics. The plot in figure 5.2a shows the outline of every valid point in 3D space. The shape is this outline depends on the configuration of the parallel manipulator and explains the edges, which indicates that one of the sliders has reached its practical limits. Optimal geometric parameters need to be calculated in order to reach the maximum reachable workspace. Lastly, the prototype does not have an active controller to control the path of the end-effector. Appendix E shows how the prototype

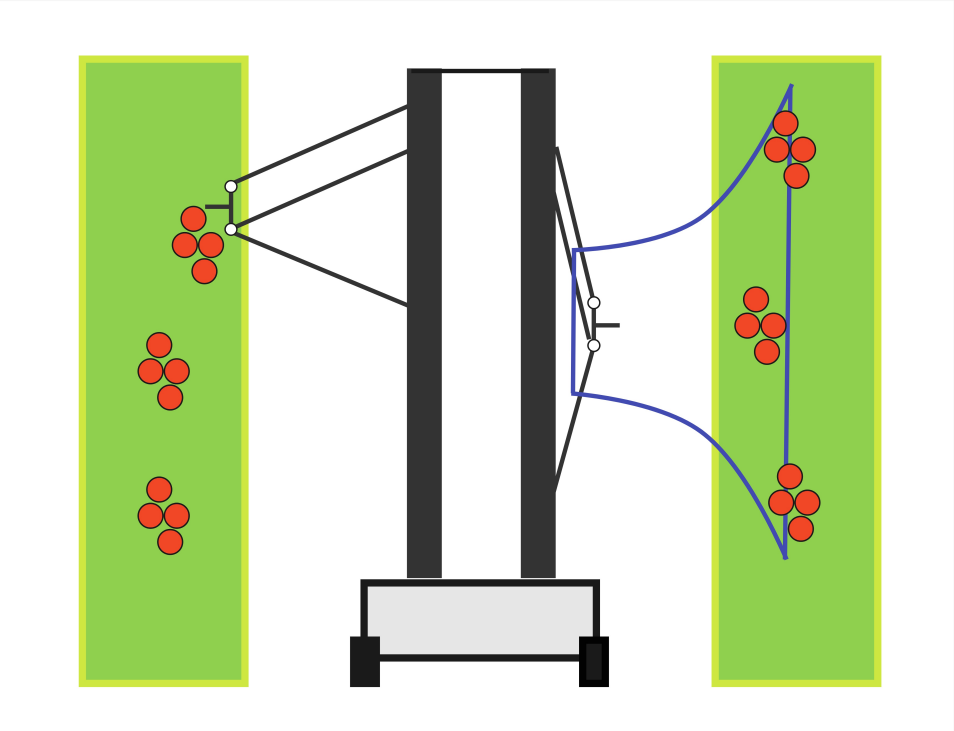


Figure 7.1: Implementation of the design in a greenhouse, showing the range of the workspace (blue) that can be used to harvest effectively, viewed from the back of the cart

is only focused on the endpoint, so no optimized path is acquired. This can be seen in several of the results. More research into active control of the path trajectory will lead to even better load distribution and smoother paths.

7.5. Recommendations and future work

The limitations describe some issues that still need to be resolved in order to let the manipulator operate in the industry. External forces are not accounted for, which means that the design needs to be improved in order to facilitate this. One option could be to increase the angle (Z-axis), from the slider to the end-effector, in order to apply axial load to the legs when an external force in Y is applied. Now, this angle is 90 degrees, which causes the bracket to deliver all reaction force, which leaves this part very vulnerable. On the other hand, this will sacrifice some of the workspace. Therefore, more research has to go into finding the optimal angle and workspace.
Furthermore, the design is an obvious proof of concept. Therefore, geometric parameters are made with educational guesses but are hardly optimized. More research could be done on the optimal geometric parameters and their influence on the workspace.
The presented design is critically constrained, which implies it has an equivalent number of degrees of freedom, as it has actuators to actuate these. Most manipulators found in the industry possess some form of redundancy, which means it has more actuators than there are degrees of freedom. This offers extensive possibilities in dexterity and workspace, however comes with more complex control. This design could benefit from more research to a redundant degree of freedom and what the possibilities could be. For example, Isaksson et al. [21] proposes a design that uses a degree of kinematic redundancy to operate a gripper. Other researchers, like Kotlarski et al. [23], use it for maximizing the workspace.
More attention needs to go to the implementation of the joints, for improving the accuracy by minimizing the backlash. In order for the system to form a stiff structure, capable of withstanding forces from all directions, it needs proper joint assembly. Since many joints are used, the chances of backlash increase. A way to reduce the number of assembled parts, one could consider compliant joints. Research would focus on compliant joints and the required range of motion. However, one must take care

not to sacrifice the stiffness of the mechanism.
The system now depends on rigid guides to withstand the reaction moment and forces, these reaction forces could form a challenge in the long run. Therefore, design choices need to be implemented which limit the torque on the sliders. This can be done by designing smaller slider brackets, with a smaller distance from the cart to the joint.
The position of the end-effector is not actively controlled in this mechanism, only the position of the end-effector. For better and smoother motion, a closed-loop controller should be implemented to capture the position of the end-effector and then control the motion of the sliders. This will further increase the robustness of the design, as disturbances are cancelled out.

8

Conclusion

This thesis researched the properties of parallel manipulators implemented in a novel manipulator design. In order for this design to come to life, literature research has led to the creation of a 4DOF linear delta manipulator. Assumed was that by using the parallel principles, one could reach better load distribution and more robust structures with fewer materials, like motors and other parts. The assumption of load distribution was successfully validated firsthand with the simulation and the prototype. For different movements, this showed an equal distribution of the load, which tempers the maximum torque peak of individual motors resulting in more predictable behavior and efficient use of a motor's capacity. The simulation also showed that the response of the system to the dynamic requirements is in fact a realistic outcome. This opened up the way for the real-life prototype to be built and tested. Experiments with this prototype have shown that the kinematic structure is in fact contributing to an exceptionally stiff structure that can hold its weight many times. Therefore validating the assumption that this minimalistic design can be used for lightweight manipulator design. The simulation and the prototype underlined the workspace of the kinematic design, which is influenced strongly by its kinematic design. However, the outer shape of this volume remains the same. Translational and rotational capabilities are guaranteed within this workspace. Singularities at the border have been localized and their behavior is mapped. The parallel placement of the guides combined with the kinematic design opens up an 'unlimited axis', which can be further researched. In the end, a comprehensive understanding of the inherent characteristics of this kinematic configuration has been established, showing that through strategic design and meticulous engineering, the challenges of maintaining precision and strength in lightweight robotic manipulators can be effectively addressed.

This research makes a substantial contribution to the field by illustrating how the integration of minimalistic design principles with parallel kinematic architectures can result in manipulators that are robust, efficient, and agile. The findings regarding the shape and capabilities of the manipulator's workspace align with existing literature, further reinforcing the vital role of kinematic design in defining a robotic arm's operational range and capabilities. These insights highlight the significance of customizing designs to suit specific applications while also paying close attention to workspace limitations and vulnerabilities. This fusion of minimalist design with functional kinematics presents a promising avenue for the development of advanced robotic systems.

Bibliography

[1] A. K. Al-Amin et al. “Economics of field size and shape for autonomous crop machines”. In: *Precision Agriculture* 24.5 (2023), pp. 1738–1765. DOI: 10.1007/s11119-023-10016-w.

[2] Eray A. Baran et al. “Unified Kinematics of Prismatically Actuated Parallel Delta Robots”. In: *Robotica* (2019). ISSN: 14698668. DOI: 10.1017/S0263574719000092.

[3] Alex Barnard. *Growing horticulture: Starting the conversation on Labour Shortage Solutions*. Jan. 2022. URL: https://www.fruitandveggie.com/growing-horticulture-starting-the-conversation-on-labour-shortage-solutions/.

[4] N. Baron, A. Philippides, and N. Rojas. “A novel kinematically redundant planar parallel robot manipulator with full rotatability”. In: *Journal of Mechanisms and Robotics* 11 (1 2019). DOI: 10.1115/1.4041698.

[5] M Bouri and R Clavel. *The Linear Delta: Developments and Applications*. 2010.

[6] Olivier Company, Vincent Pierrot FrancoisVincent Nabat, and Maria de la O Rodriguez. *Schoenflies Motion Generator: A New Non Redundant Parallel Manipulator with Unlimited Rotation Capability*. IEEE, 2005. ISBN: 078038914X.

[7] Jian S. Dai, Zhen Huang, and Harvey Lipkin. “Mobility of overconstrained parallel mechanisms”. In: *Journal of Mechanical Design* 128.1 (2004), pp. 220–229. DOI: 10.1115/1.1901708.

[8] Revanth Damerla and Shorya Awtar. “Constraint-based analysis of parallel kinematic articulated wrist mechanisms”. In: *Journal of Mechanisms and Robotics* 13.3 (2021). DOI: 10.1115/1.4049947.

[9] Bhaskar Dasgupta and T.S. Mruthyunjaya. “The Stewart platform manipulator: a review”. In: *Mechanism and Machine Theory* 35.1 (2000), pp. 15–40. DOI: https://doi.org/10.1016/S0094-114X(99)00006-3.

[10] D. Farhadi. *Week 4. Kinematic Analysis and Synthesis in rigid body mechanism (Part 2)*. [Powerpoint slides]. 2021.

[11] M. Gabardi, M. Solazzi, and A. Frisoli. “An optimization procedure based on kinematics analysis for the design parameters of a 4-UPU parallel manipulator”. In: *Mechanism and Machine Theory* 133 (2019), pp. 211–228.

[12] C. Gosselin. “Compact dynamic models for the tripteron and quadrupteron parallel manipulators”. In: *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering* 223 (1 2009), pp. 1–11. ISSN: 09596518. DOI: 10.1243/09596518JSCE605.

[13] C. Gosselin, T. Laliberté, and A. Veillette. “Singularity-Free Kinematically Redundant Planar Parallel Mechanisms With Unlimited Rotational Capability”. In: *IEEE Transactions on Robotics* 31 (2 2015), pp. 457–467. DOI: 10.1109/TRO.2015.2409433.

[14] Clement Gosselin and Louis Thomas Schreiber. “Kinematically Redundant Spatial Parallel Mechanisms for Singularity Avoidance and Large Orientational Workspace”. In: *IEEE Transactions on Robotics* 32 (2 Apr. 2016), pp. 286–300. ISSN: 15523098. DOI: 10.1109/TRO.2016.2516025.

[15] Clément Gosselin and Louis-Thomas Schreiber. “Redundancy in Parallel Mechanisms: A Review”. In: *Applied Mechanics Reviews* 70.1 (2018).

[16] Tony E. Grift et al. “A review of automation and robotics for the bio-industry”. In: 2008. URL: https://api.semanticscholar.org/CorpusID:13511513.

[17] Guilin Yang Mechatronics Group et al. *Kinematic design of a six-DOF parallel-kinematics machine with decoupled-motion architecture*. Oct. 2004. URL: https://dl.acm.org/doi/abs/10.1109/TRO.2004.829485.

[18] *Growers fear for future as rising costs and workforce shortages impact horticulture sector*. Nov. 2022. URL: https://www.nfuonline.com/media-centre/releases/growers-fear-for-future-as-rising-costs-and-workforce-shortages-%5C%5C%20impact-horticulture-sector/.

[19] Hermann J. Heege. “Heterogeneity in Fields: Basics of Analyses”. In: 2013. URL: https://api.semanticscholar.org/CorpusID:127154423.

[20] M Isaksson, C Gosselin, and K Marlow. “Singularity analysis of a class of kinematically redundant parallel Schonflies motion generators”. In: *MECHANISM AND MACHINE THEORY* 112 (2017), pp. 172–191. ISSN: 0094-114X. DOI: 10.1016/j.mechmachtheory.2017.01.012.

[21] Mats Isaksson, Clément Gosselin, and Kristan Marlow. “An introduction to utilising the redundancy of a kinematically redundant parallel manipulator to operate a gripper”. In: *Mechanism and Machine Theory* 101 (July 2016), pp. 50–59. ISSN: 0094114X. DOI: 10.1016/j.mechmachtheory.2016.03.006.

[22] Xianwen Kong and Clément Gosselin. “Type synthesis of parallel mechanisms”. In: *Springer Tracts in Advanced Robotics* 33 (2007). DOI: 10.1007/978-3-540-71990-8.

[23] J Kotlarski, B Heimann, and T Ortmaier. “Influence of kinematic redundancy on the singularity-free workspace of parallel kinematic machines”. In: *Frontiers of Mechanical Engineering* 7 (2 2012), pp. 120–134. DOI: 10.1007/s11465-012-0321-8.

[24] Jens Kotlarski, Bodo Heimann, and Tobias Ortmaier. “Experimental validation of the influence of kinematic redundancy on the pose accuracy of parallel kinematic machines”. In: 2011, pp. 1923–1929. ISBN: 9781612843865. DOI: 10.1109/ICRA.2011.5980056.

[25] Jens Kotlarski et al. *Optimization strategies for additional actuators of kinematically redundant parallel kinematic machines*. IEEE, 2010. ISBN: 9781424450404.

[26] Jonathan Lacombe and Clément Gosselin. “Singularity analysis of a kinematically redundant (6+2)-DOF parallel mechanism for zero-torsion configurations”. In: *Mechanism and Machine Theory* 170 (2022), p. 104682.

[27] Research ltd and Markets. *Smart harvest market by component, by technology, by Crop Type: Global Opportunity Analysis and Industry Forecast, 2020-2030*. URL: https://shorturl.at/glMV7.

[28] Kenji Nakamura. “Development of Efficient Production System”. In: *Plant Factory Using Artificial Light* (2019). URL: https://api.semanticscholar.org/CorpusID:117297921.

[29] Y. D. Patel and P. M. George. “Parallel Manipulators Applications A Survey”. In: *Modern Mechanical Engineering* 2.1 (2012), pp. 57–64.

[30] Erik Pekkeriet and Eldert J. van Henten. “Current developments of high-tech robotic and mechatronic systems in horticulture and challenges for the future”. In: 2011. URL: https://api.semanticscholar.org/CorpusID:55083192.

[31] Mohammad Reza Chalak Qazani et al. “Performance Evaluation and Calibration of Gantry-Tau Parallel Mechanism”. In: *Iranian Journal of Science and Technology - Transactions of Mechanical Engineering* 44 (4 Dec. 2020), pp. 1013–1027. ISSN: 23641835. DOI: 10.1007/s40997-019-00322-y.

[32] R.Clavel. 1990-12-11.

[33] Rudmin. *Parallel Axis Tripteron Concept*. Zaber Technologies, Oct. 2022. URL: https://www.youtube.com/watch?v=6EtXycVGJg4.

[34] Maureen Sondag. *The Netherlands, a significant partner in Canada’s Greenhouse Industry*. May 2020. URL: https://www.agroberichtenbuitenland.nl/actueel/nieuws/2020/02/03/the-netherlands-a-significant-partner-in-canada%E2%80%99s-greenhouse-industry.

[35] *Stepper motors EMMS-ST*. English. Festo. 14 pp. 06-2023.

[36] Michael Stock and Karol Miller. “Optimal kinematic design of spatial parallel manipulators: Application to linear delta robot”. In: *Journal of Mechanical Design, Transactions of the ASME* 125 (2 June 2003), pp. 292–301. ISSN: 10500472. DOI: 10.1115/1.1563632.

[37] Tom Tenner. *The kutzbach criterion*. Mar. 2020. URL: https://medium.com/@tomtenner/the-kutzbach-criterion-e1592ef88bd1.

[38] *Toothed belt axes ELGC-TB-KF*. English. Festo. 26 pp. 08-2022.

[39] Masaru Uchiyama. “Structures and characteristics of parallel manipulators”. In: *Advanced Robotics* 8.6 (1993), pp. 545–557. DOI: 10.1163/156855394x00248.

[40] Juan P. Vasconez, George A. Kantor, and Fernando A. Auat Cheein. “Human–robot interaction in agriculture: A survey and current challenges”. In: *Biosystems Engineering* 179 (2019), pp. 35–48. DOI: https://doi.org/10.1016/j.biosystemseng.2018.12.005.

[41] K F Wen et al. “A Backdrivable Kinematically Redundant (6+3)-Degree-of-Freedom Hybrid Parallel Robot for Intuitive Sensorless Physical Human-Robot Interaction”. In: *IEEE TRANSACTIONS ON ROBOTICS* 37 (4 2021), pp. 1222–1238. ISSN: 1552-3098. DOI: 10.1109/TRO.2020.3043723.

[42] Guilin Yang et al. “Kinematics Analysis and Design Optimization of Novel 3T1R Parallel Manipulator”. In: *Nongye Jixie Xuebao/Transactions of the Chinese Society for Agricultural Machinery* 48.12 (2017), 386–394 and 420. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85044267247&doi=10.6041%2fj.issn.1000-1298.2017.12.048&partnerID=40&md5=f8383aab97885c20ab26af59ccf37303.

A

Extra information

A.1. Parallel manipulators

Different types of parallel manipulators have been researched in the past. In 1990, Clavel et al. [32] laid the foundation of the parallel robots we know best: the Delta mechanism. It has three kinematic serial chains, that can translate the platform in the Cartesian space by employing an actuated revolute joint in series with a universal joint, a second revolute joint, and a passive universal joint [36]. In the 2000s, ABB launched the FlexPicker which incorporated the positive aspects into an industrial machine. Variations on this concept use different types of actuators, namely prismatic actuators. Like Baran et al. [2] who describe the principle and different factors of using prismatic actuators as the actuation method. Gosselin et al. [12] present the design of the multipteron. Which consists of multiple serial orthogonal legs connected at a platform, driven by independent linear actuators. Here the revolute joints connected to the manipulated platform are orthogonal to each other and parallel to the one at the base. This configuration ensures controllability in all directions. The system is therefore fully decoupled. As a disadvantage, the revolute joint connecting the links in every leg compromises the pure parallel properties of the leg. Compared to pure parallel manipulators, the links of serial manipulators are subjected to bending. Therefore, inertia and mass properties have a larger impact on the dynamics of the system at higher speeds.

A.2. Singularities

Every system, parallel or not, is exposed to singularities. Three types of singularity can occur in a robotic system. Type I, or inverse-kinematic singularities. This is the situation where the end-effector experiences zero velocity and loses a DOF. These regions are typically found at the outer border of the workspace. Type II, or direct-kinematic singularities. This is the situation where the output gains a degree of freedom, while the input is locked. Type III, or combined singularities, happens when the input link can move in a certain range, while the output link is at zero velocity [23][10]. Singularities can be interpreted as borders, instead of regions, which an end-effector can not pass properly. Mathematically, these occur when the matrices, formulating the relationship between the direct Jacobian and inverse Jacobian matrices, respectively J_q and J_x in eq. A.1, are not of full rank [4] [23]. In the case of type III singularity, both are not of full rank. This relationship can be uniquely described in a kinematic model A.1.

$$J_x \dot{x} + J_q \dot{q} = 0 \tag{A.1}$$

As a result of implementing kinematic redundancy, the elements of the Jacobian matrices can be directly affected. As a result, the singularity loci and, therefore, the performance of the mechanism can be modified while operating the system [**KotlarskiOPT2010**]. Another, more intuitive, way of determining these singularities is finding the instantaneous center of rotation (ICR). The relative movement of the platform to the base can be outlined by the ICR [**BaronNOV2019**]. Plucker lines, which are colinear lines with links connecting the platform to the other, can cross each other in space. As long as these points are separated, the system is rigid. Whenever these lines coincide or run colinear the end-effector experiences type II singularities [41].

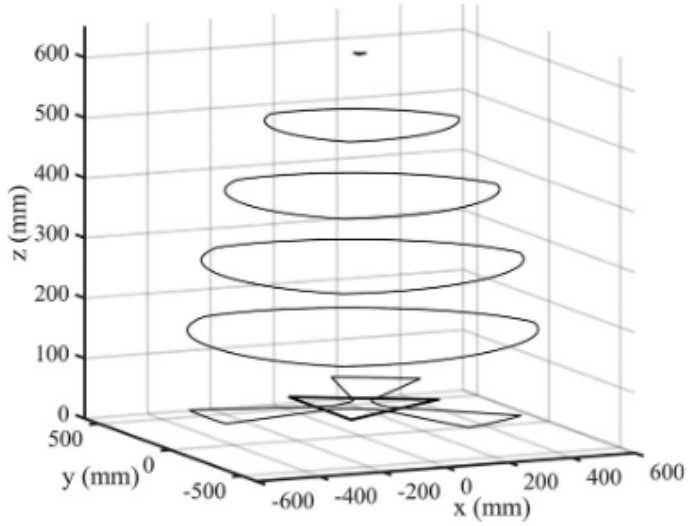


Figure A.1: Example of the change in the translational workspace as a function of the height of the platform [41]

A.3. Workspace

A.3.1. Orientational workspace

Spatial workspaces are height-dependent, meaning that different operational workspaces exist for different heights of the platform. The volume of the robot is generally not so important, more is the continuity of the workspace [**Wen2019**]. This example is found in Wen et al. [41], where an outline of the translational workspace is given as a function of the height. The result is a non-linear change in area and shape, as shown in figure A.1. Translational workspace can always be increased by scaling up the geometric parameters. This is not the case with an orientational workspace. Gosselin et al. [13] shows that partial manipulability is still provided for all values of the height of the platform. By using pure parallel legs and kinematic redundancy, one can conclude that this type of system can achieve reachability for all values of height.

A.3.2. Kinematic Redundancy

In order to increase the workspace, Gosselin et al. [15] started with the incorporation of kinematic redundancy in the system. Kinematic redundancy is the over-actuation of a system, in a way that is caused by having more actuators present than there are degrees of freedom at the end-effector. it uses this additional degree of freedom to reconfigure certain parts of the mechanism. Significant results are presented in the fields of workspace enlargement and singularity avoidance. In fact, Gosselin et al. [13] show that with 1 additional degree of redundancy, all singularities can be avoided by proper reconfiguration. The workspace enlargement is also thoroughly researched by Kotlarski et al. [23] where it was shown that the workspace is increasing exponentially with the number of additional degrees of freedom. However, that comes at the price of increasing complexity for trajectory planning. Also, this paper presents the trade-off between the additional complexity and the benefits.

A.3.3. Axial-symmetric

Axis-symmetrical designs own the property that the workspace is similar in all radial directions. The system is built from several rotating actuators, sharing the same axis, actuating an arm that is connected to a link with a revolute joint. The system presented in Company et al. [6] allows for spatial translation and unlimited rotation at the end-effector. The main drawback of this non-redundant mechanism is that at least one of the kinematic chains is subjected to torsion and bending [20]. This may increase the risk of low stiffness. Therefore it requires a stiff and heavy structure [21] and thus a higher mass-to-payload ratio. A different strategy is used in Gosselin et al. [14], however, it deals with the same problem.

A.3.4. Gantry

Other methods to increase the workspace are facilitated by the use of gantry systems. Instead of rotating actuation, gantry-type manipulators use parallel prismatic actuators. Reza et al. [31] describe the main advantage of parallel legs over serial as having higher stiffness, acceleration, and accuracy characteristics. Gantry-type designs focus on the translational workspace. Following the base principle of gantry systems, the manipulators are actuated by parallel prismatic actuators. The linear Delta [5] is a variation on the Delta mechanism discussed before. The linear Delta actuation consists of prismatic actuators and is oriented parallel in such a way that the manipulated platform is located outside of the actuated guideways. The actuators' translation is one-dimensional. Different architectures have been presented, like in Isaksson et al. [20] where lightweight design is the focus. The major advantage of this mechanism is the theoretical workspace. One can make the guideways as long as needed, without compromising the workspace. Also, the cross-section, perpendicular to the plane of the guideways, is constant over the length of the guideways [36]. Therefore, all properties related to the mechanism (stiffness, manipulability, etc.) are constant over the length. Another design that shares this principle is shown in Gosselin et al. [13]. A 2-dimensional representation of 2 translations and 1 rotation in the plane. It shows the combination of a prismatically actuated pure parallel manipulator with kinematic redundancy. Furthermore, kinematic redundancy helps to avoid certain singular positions. Therefore, this design can achieve smooth trajectories through the reachable workspace of the manipulator. The design presented in Isaksson et al. [21] can be utilized in several actuation modes. One is described with 3 parallel guideways and one with 5. Unlimited configurations and architectures could be designed. However, Isaksson et al. [21] focus on resource-based design. This method focuses on the feasibility of the design with practical considerations such as moving mass, number of required components, and practicality of the components.

A.4. Type synthesis

All the poses (position, orientation) a PM can perform, are captured in the motion pattern. Which classifies the DOFs as translational, spherical, rotational, or any other motion. Type synthesis of PMs is the study of finding all possible types of PMs generating specific motion patterns of the moving platform. One can use type synthesis for designing PMs with a specified number of DOFs using the mobility criterion 5.1 as a basis, which is a systematic approach. Type synthesis based on the motion pattern is also common, due to the great variety of applications. Research shows that most published works are on a case basis [22]. One prominent example of a motion pattern is the common Schönflies motion (3T1R), which enables the end-effector to move in Cartesian space and orient around a given axis. Careful consideration should be given to actuation decisions, as the selection of actuated joints should not be arbitrary. In the typical configuration, the PMs DOF should be zero when all actuators are locked.[22]. The selection of the number of legs in a parallel mechanism (PM) depends on various factors, including the arrangement of actuators, the DOF of each actuator, the desired workspace, and the stiffness requirements [22]. To achieve a larger workspace, it is possible to have fewer legs than DOFs. Conversely, to increase the stiffness of the mechanism, one can opt for more legs than DOFs. Although not explicitly stated, it is preferable to use identical legs in the design of a PM.

A.5. Parallel placed guides

The motion in the Z-axis can be provided in several ways. One option is provided by one motor moving a parallel robot as a whole. This allows for full reachability along the Z-axis, but it also needs one extra powerful actuator. By introducing the gantry and prismatic actuated systems from the introduction, these principles could be combined. It has already been proven that using prismatic actuators as a form of actuation can develop powerful tools. Furthermore, it allows for theoretically unlimited movement in a certain direction without introducing additional actuators. There remains a challenge, namely that all prismatic actuators must be parallel to one another. Unlike traditional parallel robots, like the Delta, which form a 120^o angle with each other, constraining all translational motion of the robot. By placing actuators in parallel, the design of the manipulator changes completely to the ones that have been introduced already, in order to facilitate the motion in the X-direction. The examples from Isaksson et al. [21], Gosselin et al. [13], and Rudmin [33] (who presents a concept to the idea of Nicholas Seward) form an example of how parallel actuation can be used. One aspect pops from these designs, as they are individually unable to produce 4 degrees of freedom with only parallel actuators. It can be concluded

that the parallel actuation constrains some of the motion in the end-effector. By properly constraining the system, Rudmin [33] is able to facilitate all translational motion. However, no orientational motion can be realized. Next to that, it uses serial chains, which allows for more translational freedom but comes at the cost of diminishing parallel properties.

B

Design

B.1. Selection

B.1.1. Delta

The original rotational Delta [32] is known for its high-speed dynamics and excellent weight-to-payload ratio. It consists out of 3 rotational motors, at 120 degrees with each other, moving an arm. This arm is connected to a leg under an angle, resulting in a closed-loop kinematic chain at the platform below. The speed and torque, combined with short distances, of the motors make it a highly dynamic mechanism that is only limited by the speed and torque of the motors themselves. The workspace is limited to the position where the Delta is stationed. Within this workspace, Delta robots allow for relatively large motion in Z-axis (downward). However, this is not suitable for the application at hand, because it lacks movement radial to the Z-axis, required for full reachability. Delta robots have to deal with the trade-off, between workspace and height, discussed in Appendix A.3.1, as mechanical interference is present. The inner state shows the minimum volume the Delta possesses, which is more than the base volume.

B.1.2. Linear Delta

The linear Delta from Clavel et al. [32] and Isaksson et al. [21] differs from the original Delta as it uses linear actuators. Thereby, it improves its range of motion greatly, as on direction is theoretically endless. However, it has to give in to the high-speed performance. Now the speed and acceleration are limited to the speed and acceleration of the linear motors, which are often not as fast as rotational actuators. Also, the traveled actuator distance is far greater. Nevertheless, the footprint is smaller as the inner state can approach the base volume, making it less bulky and more suitable for motion in tight spaces.

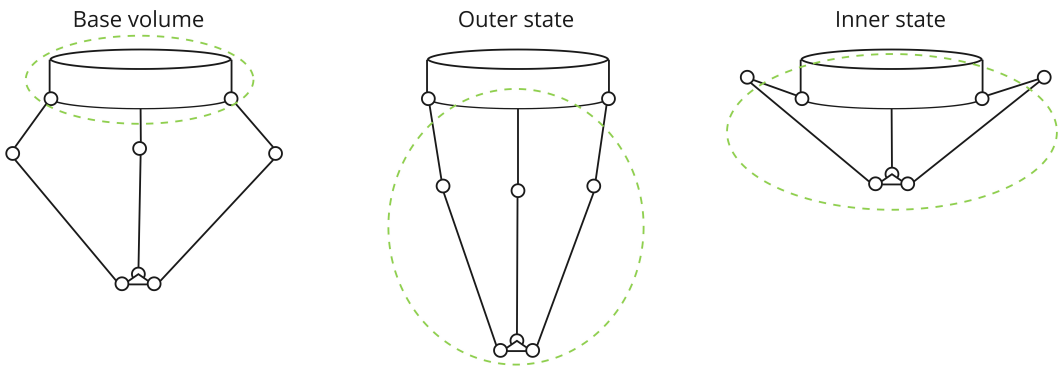


Figure B.1: Schematic representation of the Delta, showing base volume, outer state, and inner state

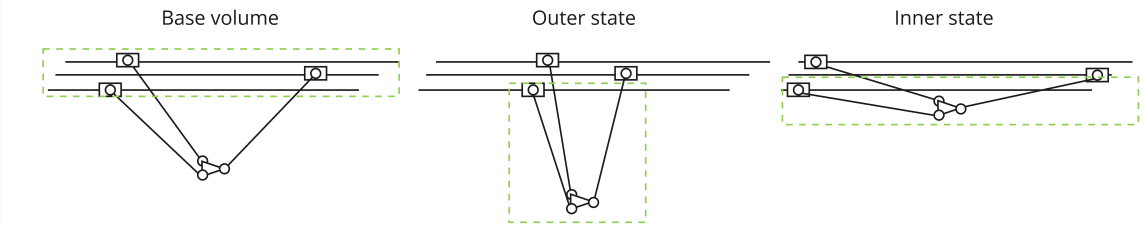


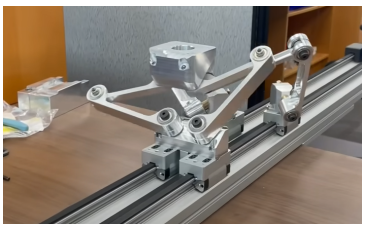
Figure B.2: Schematic representation of the linear Delta, showing base volume, outer state, and inner state

B.1.3. Multipteron

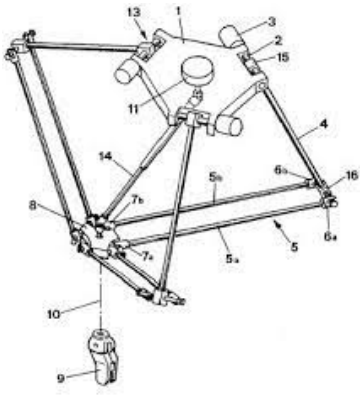
Gosselin et al. [12] propose the designs of respectively the tripteron and quadrupteron parallel manipulators, or combined multipterons. It uses only linear actuators to translate a platform. Because it scores worse on all criteria, it will not be further discussed in this section. The reader is forwarded to the paper of Gosselin.

B.1.4. Axis Symmetry

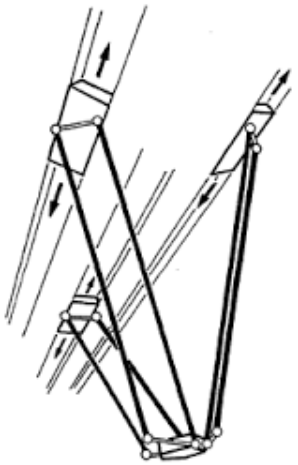
Isaksson et al. [20] propose another differentiation on the Delta, where all the rotational actuators share the same axis. By holding on to the rotational motors, it maintains its high-speed character. The workspace also increases, as it is now able to move freely around the axis. Because it counts 5 different legs, it counts for a volume larger than the base volume, comparable to the volume of the original Delta in figure B.1.



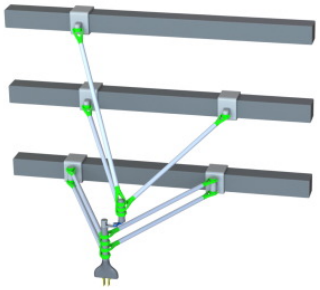
(a) Multipteron design from Rudmin [33]



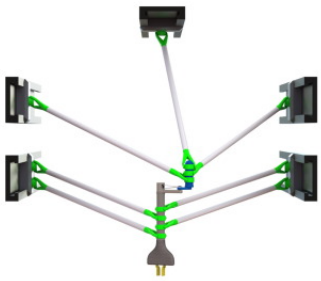
(b) Design of the Delta manipulator by R. Clavel [32]



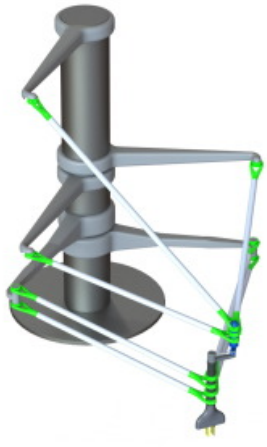
(c) Design of the linear Delta by R. Clavel [32]



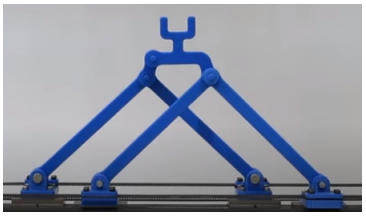
(d) Linear delta [1] with kinematic redundancy by Isaksson et al. [21]



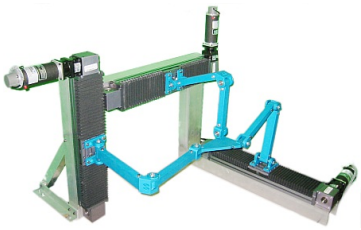
(e) Linear delta [2] with kinematic redundancy by Isaksson et al. [21]



(f) Axis-Symmetric Delta with kinematic redundancy by Isaksson et al. [21]



(g) 2D linear parallel manipulator with kinematic redundancy by Gosselin et al. [13]



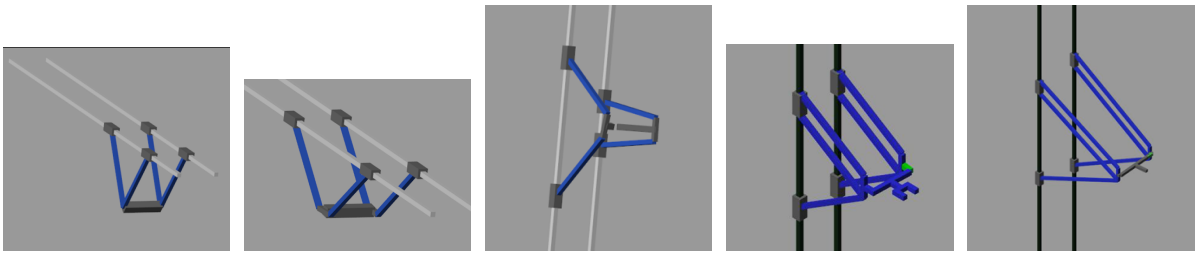
(h) Tripteron by Gosselin et al. [13]

B.2. Process

B.2.1. Iterations

With this idea being the foundation of the manipulator, a model is created in Matlab/Simulink. Starting with the simplest version is shown in figure B.4a. The model is constrained and performs a 2T1R motion. However, the rotation around the Y-axis should be constrained, as the end-effector must remain parallel to the crops. A valid way to implement this is by adding a link that secures the parallel character of the end-effector, as seen in figure B.4c. Replacing the upper leg with a parallelogram structure, the parallel structure is reliable for a large range of motion. By constraining rotation in the Y-axis, and relieving the rotational constraints in Z and X, the system should be able to perform as intended. However, rotation is still not possible over Z, because the platform itself is modelled rigidly. Consequently, the rotation is blocked, because the distance between both coupling links is not constant. The next

iteration, figure B.4d, accounts for this, by using an extra link that extends the platform by rotation, with the intention of maintaining a constant position by control of the end-effector's position. Now, the system allows for 3T1R motion patterns when the platform is exactly between both actuators. By adding an extra link to the other side of the coupler, we add a constraint in the rotational Y-direction. Now, the mechanism is fully constrained. Although the rotation around the Z-axis is provided correctly, it shows a weak spot in the design. Namely, a singularity in the extra link connected to the platform. This link allows for extension, in order to facilitate rotation. The last iteration, figure B.4e, provides a solution to this problem, and the revolute joint is replaced by a prismatic joint. Therefore, it cancels the singular character of the platform and it improves the predictability of the motion of the end-effector.



(a) First iteration (b) Second iteration (c) Third iteration (d) Fourth iteration (e) Fifth iteration

Figure B.4: Iterations shown in Simscape

B.2.2. Force plot

A back of the envelope drawing shows the expected distribution of the static forces in figure B.7. Looking at the force plot of the manipulator, three joints are simulated named A, B, and C. Only reaction forces F_a, F_b, and F_c can be generated to compensate F_L. It can be quickly seen that these reaction forces cause a resultant moment in the sliders, which needs to be withstood by the guide itself. The couple on the double slider, caused by F_a and F_b will probably cause higher torques on the upper slider. The other reaction force F_x will be distributed between both sliders, as $\sum Fx = 0$. The force resultant force in the negative Z direction will be directly withstood by the motor itself.

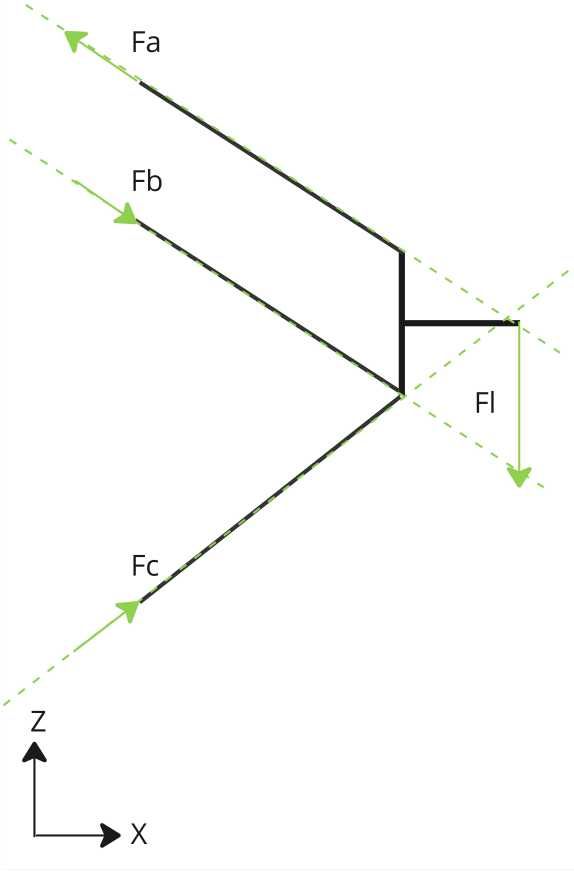


Figure B.5: Force plot of the manipulator

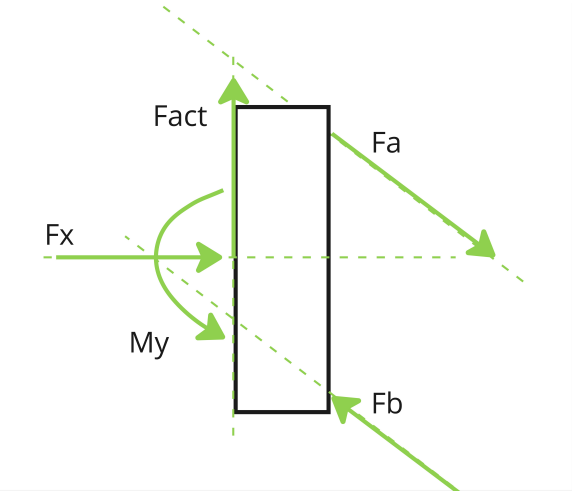


Figure B.6: Force plot of the upper slider

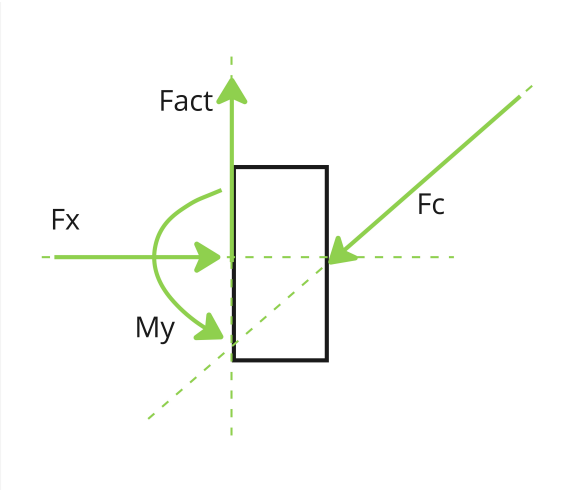


Figure B.7: Force plot of the lower slider

B.3. Constraining the system

The initial Kurzbach-Grübler criterion has been worked out for this case. However, this method resulted in a DOF of less than 0, as seen in equation B.2.

$$m = 6 \cdot (n-1) - \sum f_i \tag{B.1}$$

$$m = 6 \cdot (17-1) - 5 \cdot 21 = -9 \tag{B.2}$$

This means the system is overconstrained. It became clear that this system contains dependent constraints and that these need to be identified. Using the modified Kutzbach-Grübler criterion from Dai et

al. [7], two separate approaches have been tried. The first approach counts all the bodies and joints and substracts the dependent joints, which is every joint except for the actuated prismatic ones and 2 upper and 2 lower base revolute joints. These joints determine the motion for further joints along the manipulator, as seen in figure B.8. The second approach assumes that the parallelogram structures on both sides have 2 DOFs each. These are constrained by the end-effector and look like equations B.3. Where the prismatic actuator is the dependent constraint (c=1).

$$m = 6 \cdot (7 - 7 - 1) + 9 + 1 = 4 \tag{B.3}$$

Figure B.8: Approach 1

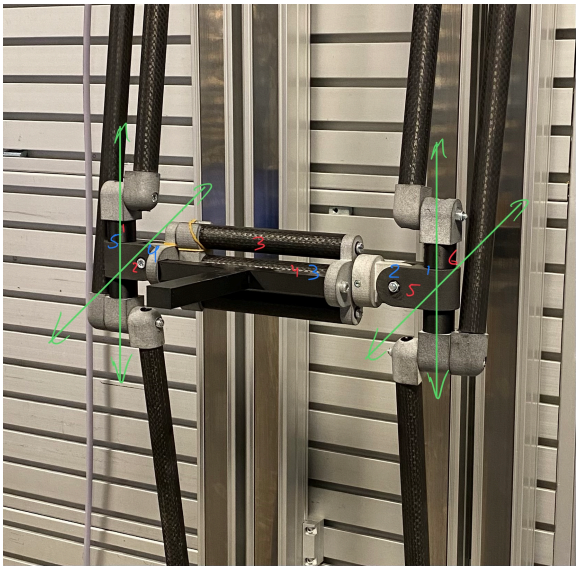


Figure B.9: Approach 2

Figure B.10: Two different situations for determining the DOFs of the mechanism

C

Simulation

C.1. Path trajectories

C.1.1. Sinusoïdal waves

To test the behavior of the system, a path trajectory is needed. The first attempt exists of controlling the sliders, without knowing the pose of the end-effector. By implementing a sinusoïdal wave, the behavior of the end-effector is exposed and certain constraints are shown.

C.1.2. Trapezoidal velocity

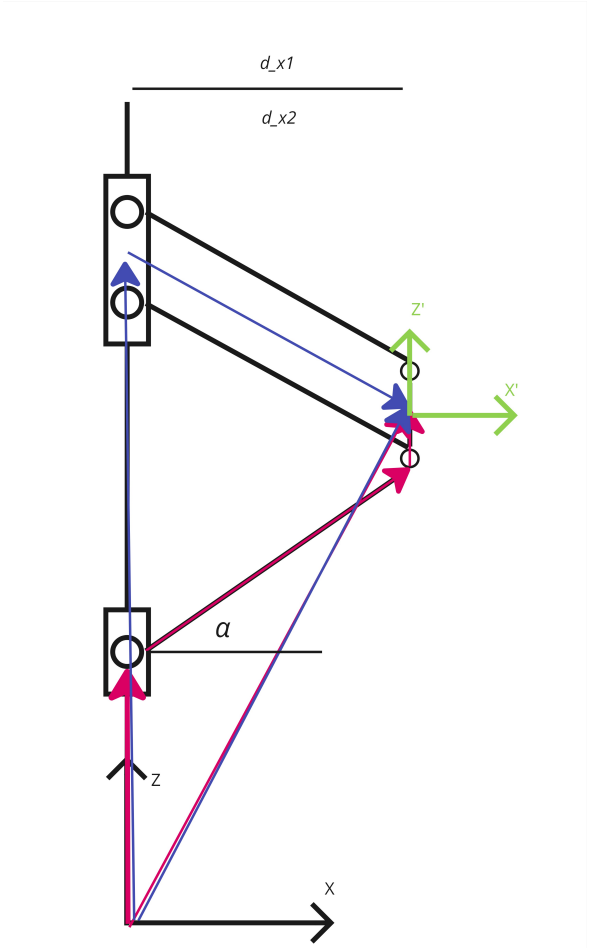
This built-in Matlab function takes the waypoints of the path and the corresponding time of arrival as input. Next, it computes the positions, velocity, and acceleration profiles of the end-effector. Eventually, this model was better compared to the other models, as it allowed for smoother speed and acceleration profiles for the sliders, while still following the expected path of the end-effector.

C.2. Inverse kinematics

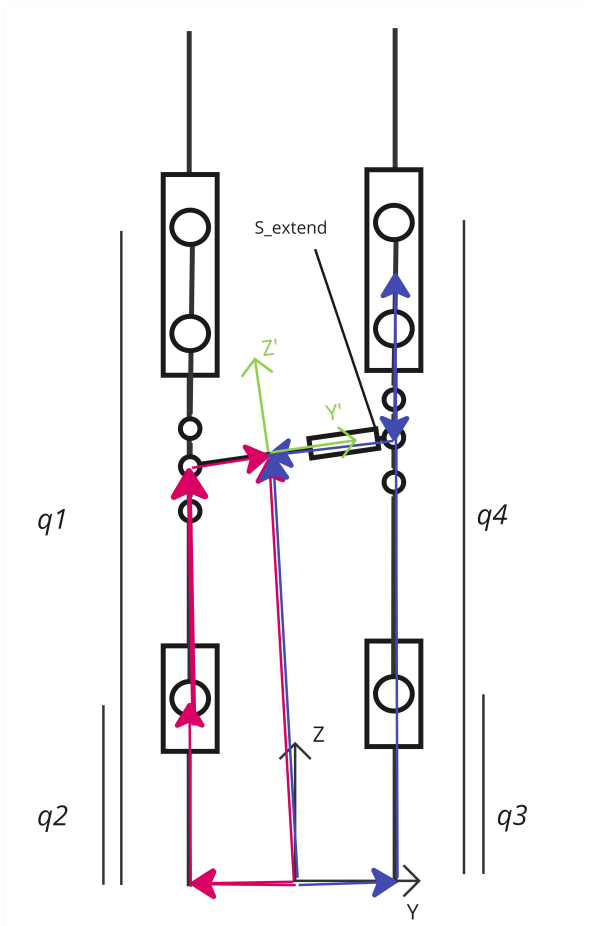
In order to find the relation between the spatial pose of the end-effector and the position of the sliders, vector loop equations are used. In three different views, multiple vector loops are found, as shown in figure C.1. The chain of vectors, representing the beams, is equal to the vector that closes the chain. For x unknowns, there need to be x equations. By solving this system of equations, the relation between input and output is found.

Unknowns that are solved for, are:

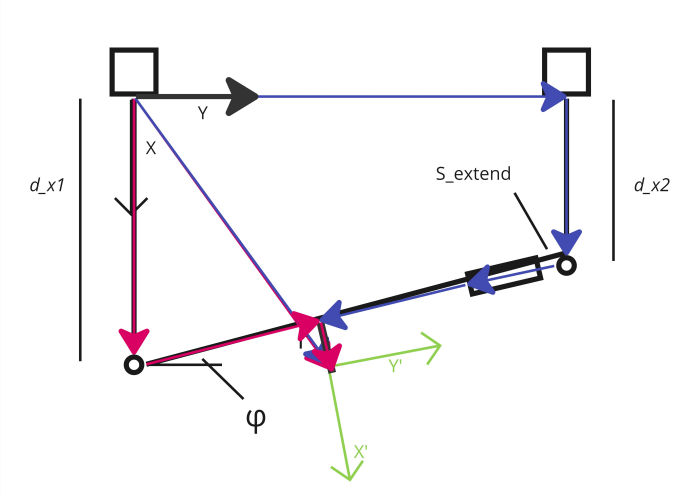
- S_{extend}, being the extension of the prismatic joint at the end-effector
- α, being the angle between the guide and the first leg
- ϕ, being the angle that the end-effector makes when the height of both the 4-bar mechanisms is not equal.



(a) Side view



(b) Front view



(c) Above view

Figure C.1: Closed vector chains for three different viewpoints

$$
\begin{aligned}
q_1 &= z + \frac{b\sin(\psi)}{2} \\
&\quad + L\sin\left(\left|\operatorname{acos}\left(\frac{x + \frac{L_{\mathrm{EE}}\sin(z)}{2} + L_{\mathrm{grip}}\sin(z)}{L}\right)\right|\right) \\
q_2 &= z - \frac{L_c}{2} + \frac{b\sin(\psi)}{2} \\
&\quad - L\sin\left(\left|\operatorname{acos}\left(\frac{x + \frac{L_{\mathrm{EE}}\sin(z)}{2} + L_{\mathrm{grip}}\sin(z)}{L}\right)\right|\right) \\
q_3 &= z - \frac{L_c}{2} \\
&\quad - L\sin\left(\left|\operatorname{acos}\left(\frac{x + \frac{L_{\mathrm{EE}}\sin(z)}{2} - L_{\mathrm{grip}}\sin\left(z + \frac{3\pi}{2}\right) + \frac{\sin(z)\left(b - y - \frac{L_{\mathrm{EE}}\cos(z)}{2} + L_{\mathrm{grip}}\cos\left(z + \frac{3\pi}{2}\right)\right)}{\cos(z)}}{L}\right)\right|\right) \\
&\quad - \frac{b\sin(\psi)}{2} \\
q_4 &= z + L\sin\left(\left|\operatorname{acos}\left(\frac{x + \frac{L_{\mathrm{EE}}\sin(z)}{2} - L_{\mathrm{grip}}\sin\left(z + \frac{3\pi}{2}\right) + \frac{\sin(z)\left(b - y - \frac{L_{\mathrm{EE}}\cos(z)}{2} + L_{\mathrm{grip}}\cos\left(z + \frac{3\pi}{2}\right)\right)}{\cos(z)}}{L}\right)\right|\right) \\
&\quad - \frac{b\sin(\psi)}{2}
\end{aligned}
\tag{C.1}
$$

Where X, Y, Z, and ψ are the input values measured from the world frame.

- L_{EE} = length of the end-effector
- b = between
- L = length of the leg
- L_c = length of the coupler
- L_{grip} = length of the gripper

C.3. Simulink Model

C.3.1. Overview

Simscape is used to build a model in Simulink. Simscape uses physical objects that can be linked in order to collaborate. Every block can be translated and rotated individually, using a rigid transform (C.3b). Rigid bricks (C.3e) are used to simulate beams and other rigid parts. These parts are linked using joint relations. Revolute (C.3d) and prismatic (C.3c) joints (both 1DOF) are used.
Simulink uses a backward Euler solver configuration with a sample time of 0.001 s. In the mechanism configuration block (C.3a), gravity is simulated uniformly as $[0, 0, -9.81 m/s^2]$. From the world frame, a transformation (translation and rotation) is made to the base frame, which forms the basis of the mechanism. One slider is placed here, the second is translated along the Y-axis.
Figure C.2 gives a followable version of the Simulink model in figure C.4.
Matlab calculates the positions using the transformation from section 5.3.2. And combines it with a time column. The prismatic actuator takes this array, which is given by q_1, q_2, q_3 and q_4 as an input. The physical relations then transform the input into a positional output at the platform. The prismatic actuators carry sensors that measure position, velocity, acceleration, total force, and torque at the actuator. At the end-effector, sensors are placed to measure the Cartesian position and the rotation around the X-axis. The measured output values are then converted to 1D arrays, which are returned to the Matlab workspace.

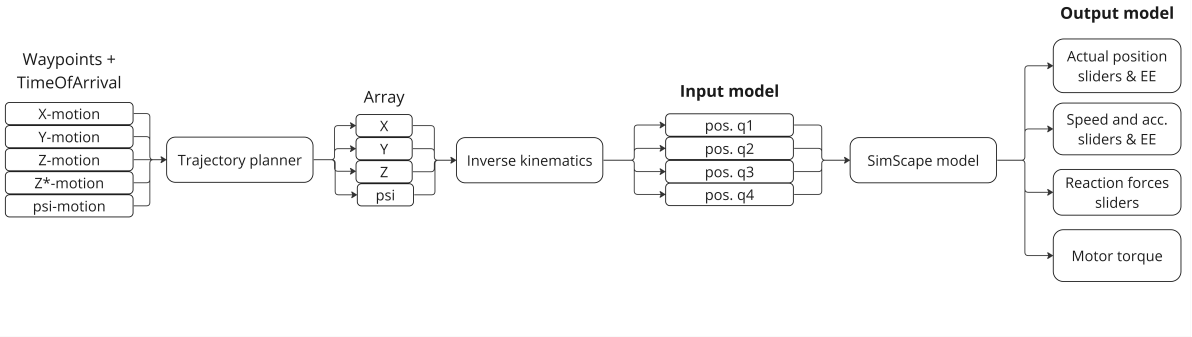
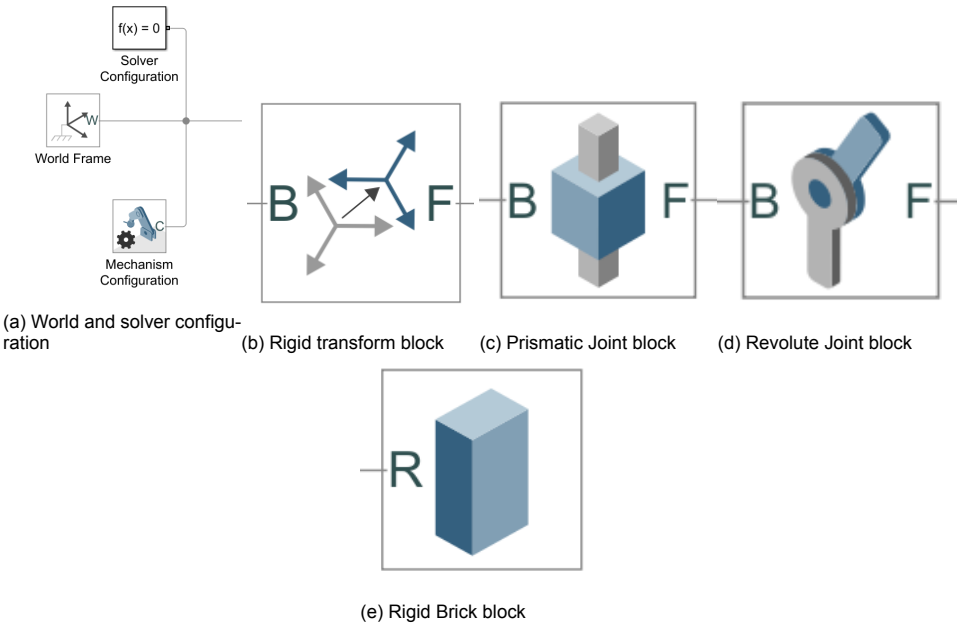


Figure C.2: Flowchart from waypoints to simulation results



(a) World and solver configuration (b) Rigid transform block (c) Prismatic Joint block (d) Revolute Joint block

(e) Rigid Brick block

Figure C.3: Used physical Simscape blocks

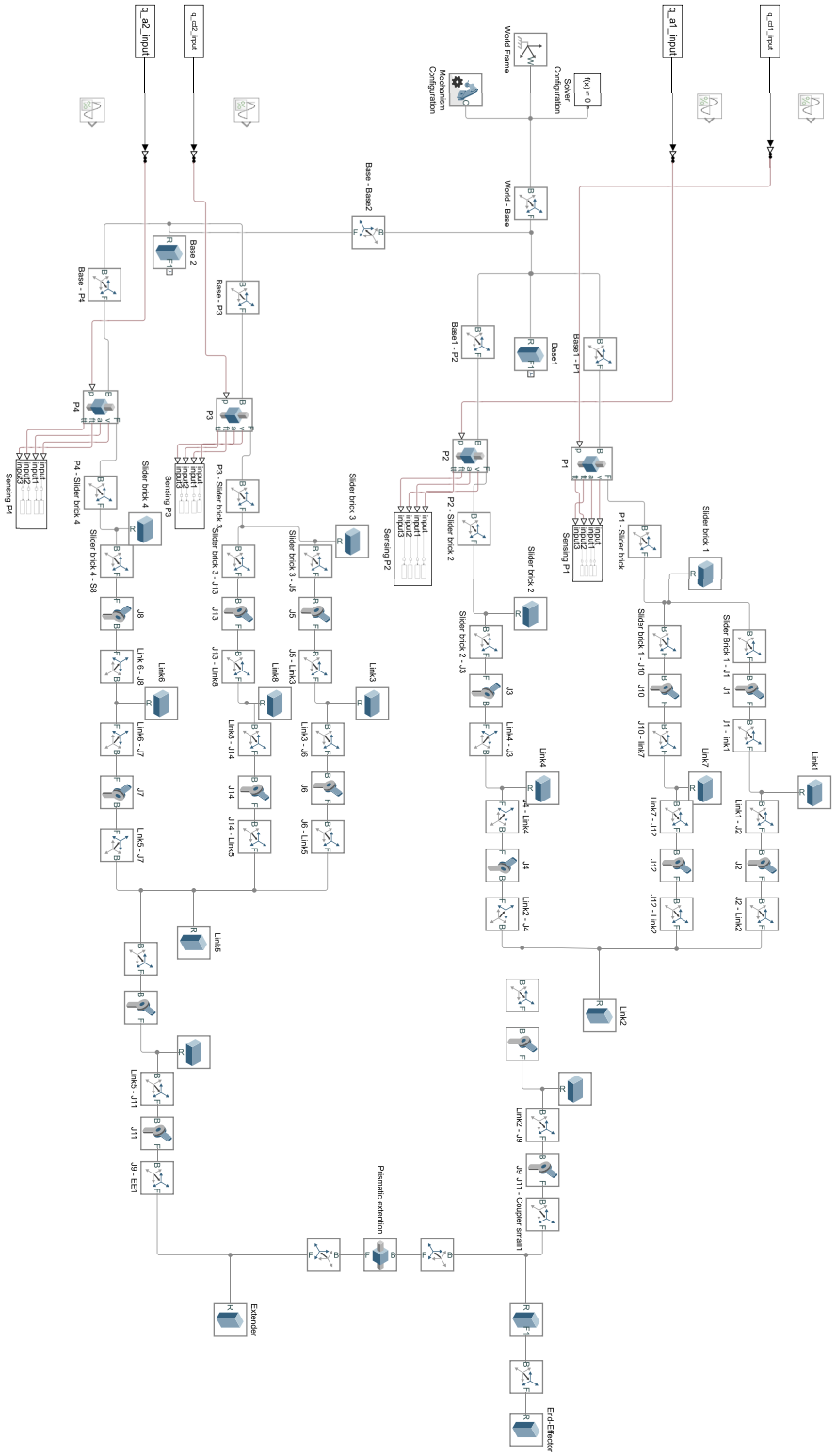
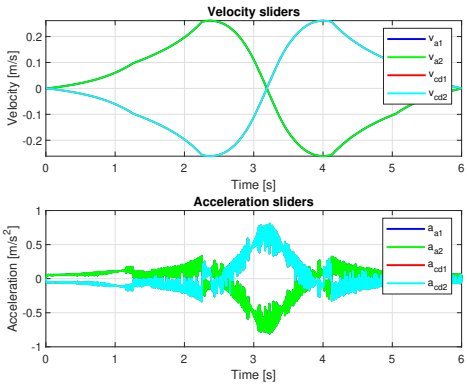


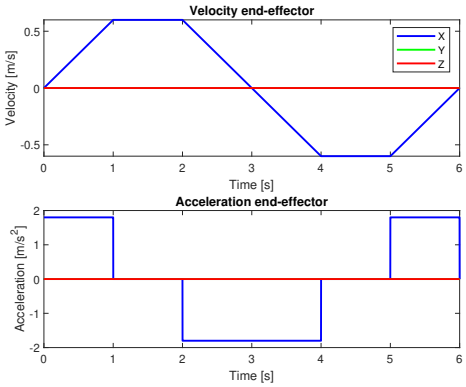
Figure C.4: Full simulink model

C.3.2. Results

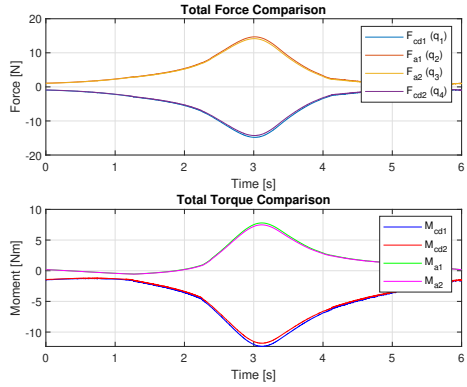
These are the figures representing further results of the simulation, with the resultant speed and acceleration of the sliders in figure C.5a. The position, speed, and acceleration of the end-effector are found in figure C.5d and figure C.5b. The reaction forces F_x and M_y are found in figure C.5c. These figures are repeated for all simulated motions.



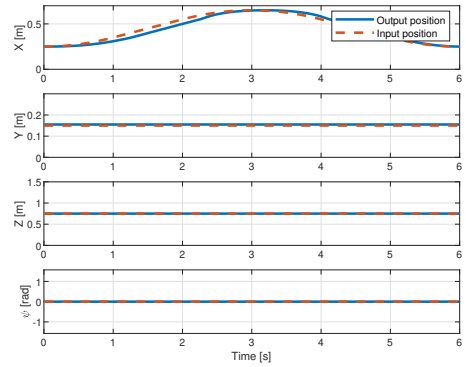
(a) Velocity and acceleration of X-motion



(b) Velocity and acceleration of the end-effector of X-motion

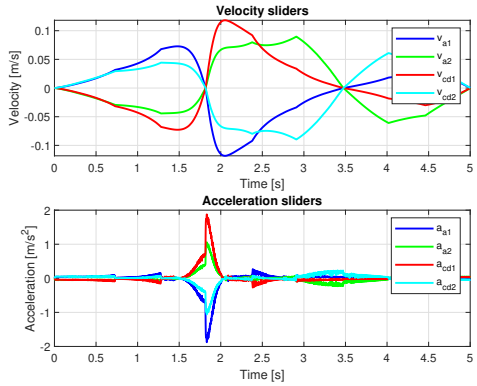


(c) Reaction forces of the X-motion

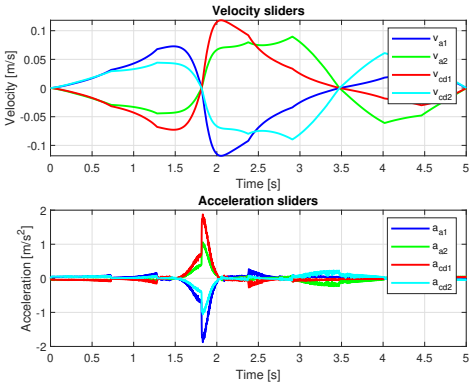


(d) Position of the end-effector of the X-motion

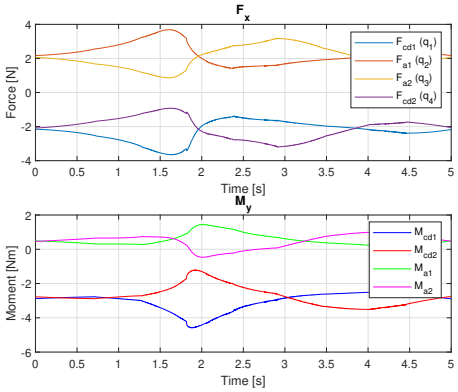
Figure C.5: Results for X-motion



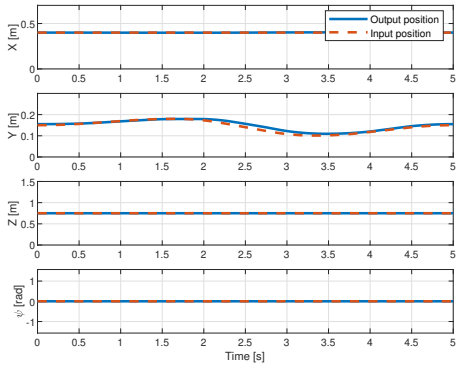
(a) Velocity and acceleration of Y-motion



(b) Velocity and acceleration of the end-effector of Y-motion

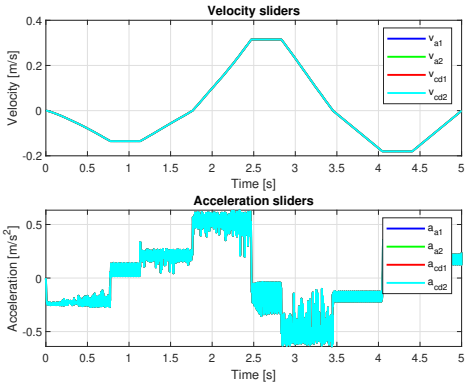


(c) Reaction forces of the Y-motion

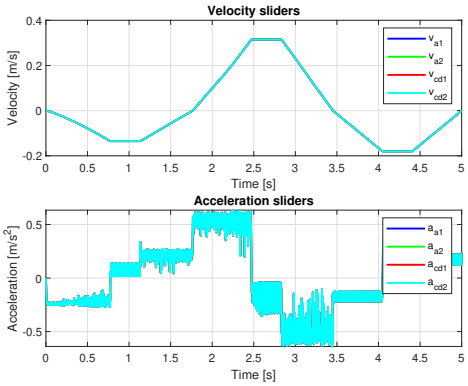


(d) Position of the end-effector of the Y-motion

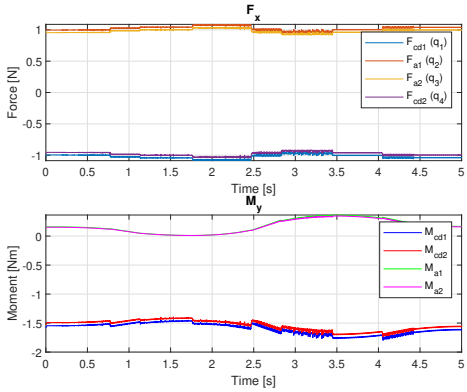
Figure C.6: Results for Y-motion



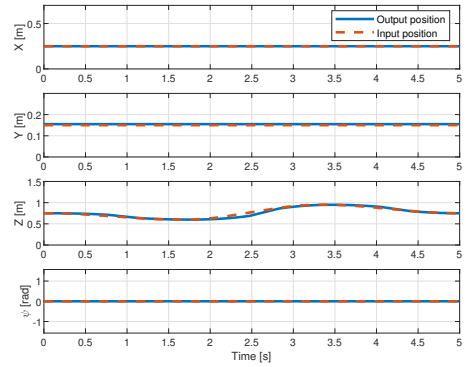
(a) Velocity and acceleration of Z-motion



(b) Velocity and acceleration of the end-effector of Z-motion

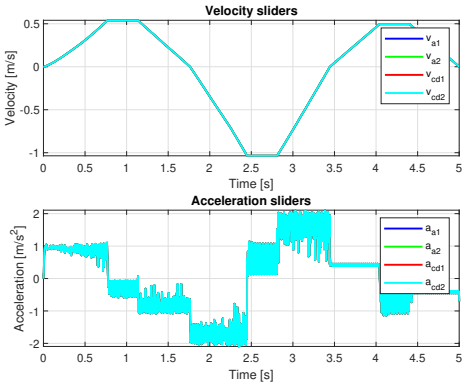


(c) Reaction forces of the Z-motion

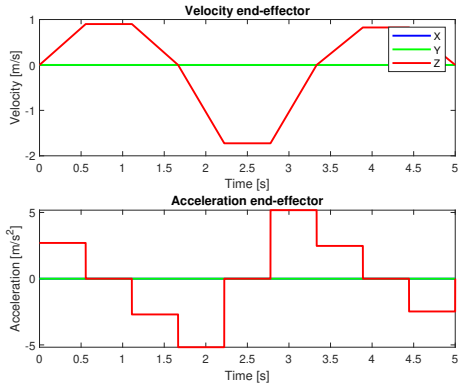


(d) Position of the end-effector of the Z-motion

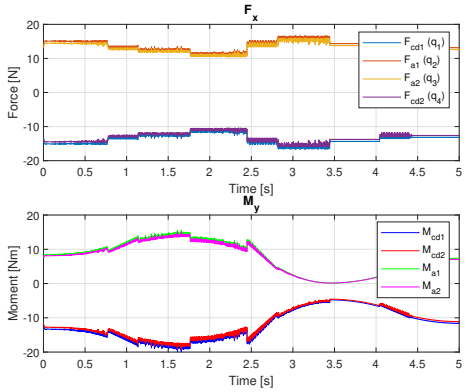
Figure C.7: Results for Z-motion



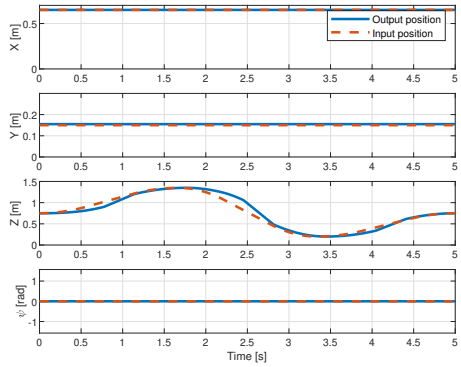
(a) Velocity and acceleration of Z*-motion



(b) Velocity and acceleration of the end-effector of Z*-motion

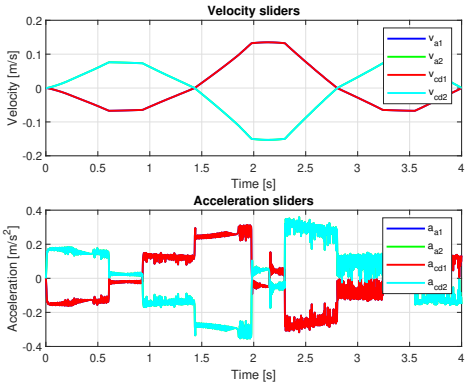


(c) Reaction forces of the Z*-motion

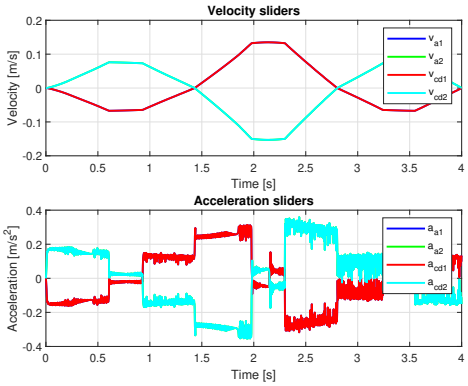


(d) Position of the end-effector of the Z*-motion

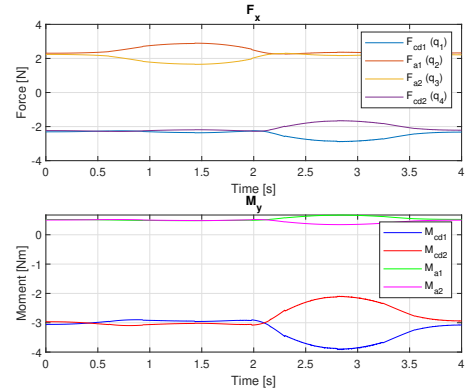
Figure C.8: Results for Z*-motion



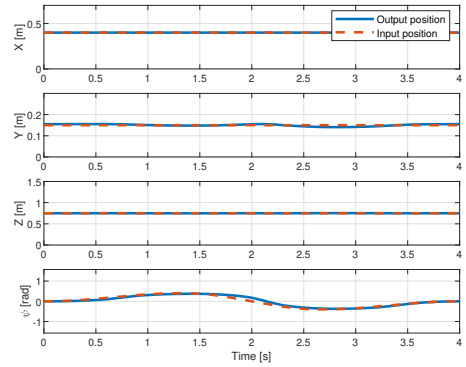
(a) Velocity and acceleration of ψ-motion



(b) SVelocity and acceleration of the end-effector of ψ-motion



(c) Reaction forces of the ψ-motion



(d) Position of the end-effector of the ψ-motion

Figure C.9: Results for ψ-motion

D

Implementation

D.1. Design choices

D.1.1. Base

Multiple considerations are made in forming the base. It is important the base is rigid, long enough, and accounts for the functional requirements for use in horticulture. The main concept consists of a linear guide rail, equipped with two belt-driven carts. The guide rail constraints all torque at the end-effector, so it has to withstand a certain threshold. Festo produces these rails, but only applicable for one motor per rail. Since these guide rails are tested and guaranteed to withstand high force and torque, this is the safest choice for implementation. Other practical problems like slip between the motor and belt, high friction, and mismatching outlines between components are therefore avoided. An axial connection kit provides the link to the stepper motor. Festo's Electric Motion Sizing is a tool in which components can be combined and simulated, with the target position, payload, external force, speed, acceleration, deceleration, and travel time as input. It then calculates the approximate load on the selected components. For this example, a situation is simulated where 1 cart travels a distance from 0 to 1500mm, with $a = 2m/s^2$, $v = 0.5m/s$, $m = 0.5$, and $t = 5s$. The results are shown in figure D.1. A more detailed representation is shown in table D.1.

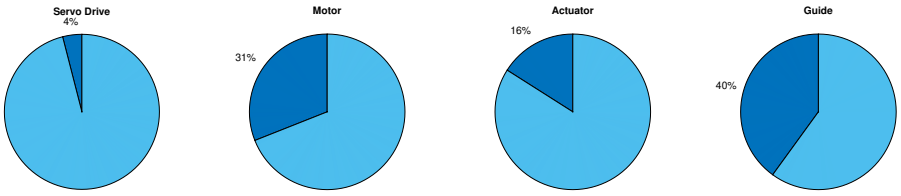


Figure D.1: Capability usage over maximum in percentage for s = [0, 1500]mm

Figure D.2 displays the velocity and acceleration profile the cart makes. Next to this, the motor characteristics are plotted. The dots representing the situation 0 to 1500mm and back, also shows that there is a significant safety factor included. This stepper motor has higher torques at low rpm, which is ideal for this application.

Linear guide rails

Festo provides a toothed belt axis (ELGC-TB-KF-45-1500) [38], which is an electromechanical linear axis. Basically, this is a linear actuator driven by an external servo or stepper motor. The toothed belt allows a stretch of 0.187% over the total length. The guide itself is guided by recirculating call bearings.

Parameter	Value	Parameter	Value
Total cycle time	6.91 s	Mass moment of inertia ratio	7.99
Max. motor revolution	499.96 rpm	Servo drive workload - effective	2%
Max. motor torque	0.15 Nm	Servo drive workload - maximum	4%
Root mean square (RMS) of motor torque	0.09 Nm	Motor workload - effective	11%
Max. current	0.57 A	Motor workload - maximum	31%
Root mean square (RMS) of current	0.35 A	Actuator workload - effective	8%
Max. power	10.75 W	Actuator workload - maximum	16%
Root mean square (RMS) of power	4.43 W	Guide workload - effective	33%
Deceleration for emergency stop	15 m/s^2	Guide workload - maximum	40%
Displacement during emergency stop	8.33 mm		
Current for emergency stop	1.8 A		

Table D.1: System Specifications for s = [0, 1500] mm.

Therefore, it can achieve accelerations of $15m/s^2$ and a maximum speed of $1.2m/s$. Its maximum feed force is $F_x = 75N$. Also, it has a radial maximum force of $F_y = 300N$ and $F_z = 600N$. Next to that, the maximum allowable torque is $M_x = 5.5Nm$, $M_y = 4.7Nm$, and $M_z = 4.7Nm$. Higher torques do not reduce the performance, but they reduce the theoretical lifespan of the guide rail. The motor is allowed to convey a maximum of 0.716 Nm torque.

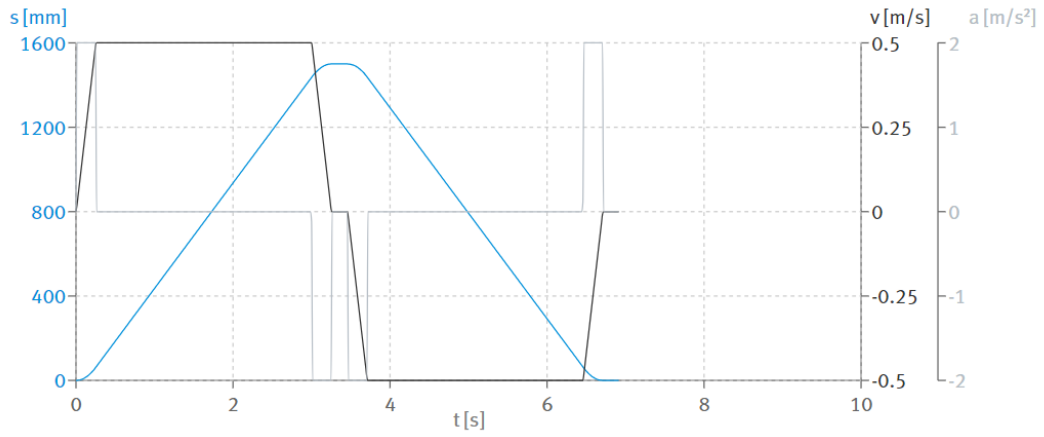
Stepper motor

Because the system works at relatively low speeds, a stepper motor is selected. Stepper motors provide a high pole count, which means that stepper motors move incrementally with a consistent pulse. Festo provides a stepper motor (EMMS-ST-42-S-SE-G2) [35]. This motor has 200 poles, each being 1.8^o. The maximum rotational speed is $1740rpm$ and the holding torque is $0.5Nm$ at $U = 48V$ and $I = 1.8A$.

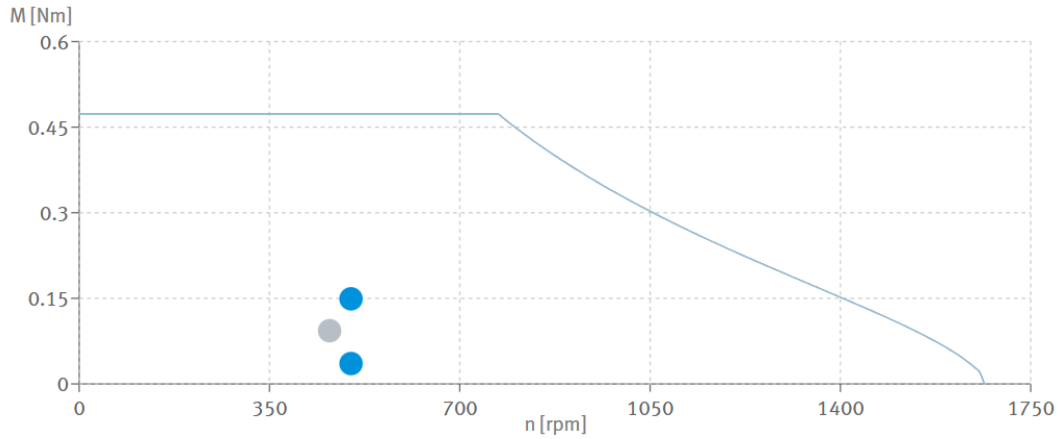
D.2. Final design implementation

D.2.1. Control panel

The control panel stands as the primary unit, governing the motor current and ensuring a safe power-down during emergencies. The control configuration's block diagram can be seen in Figure D.3. The system uses dual power supplies: 24V for the PLC and controllers, and 48V dedicated to the motors. A fuse follows these power supplies to safeguard the power distribution. Between the controllers and the PLC, a safety relais is placed. This ensures a delayed power cut-off to the motors in case of use of the emergency button, in order to prevent potential damage to the motors. After which the system loses its power. The system then becomes powerless. Power is reinstated once the emergency button is disengaged. Upon power restoration, the PLC activates the servo drives, each responsible for a specific motor. Using an axial fixation kit, the motor shaft is connected to a belt-driven linear guide. This eventually drives a cart which is allowed free motion over the rail. The parallel manipulator is then attached as depicted in figure D.6, with each cart supporting a quarter of the entire manipulator. The kinematic transformation from section 5.3.2 is programmed into the PLC, so that the inputs X, Y, Z, ψ are transformed into slider positions.



(a) Trajectories of distance, speed, and acceleration



(b) Motor characteristics: moment against RPM

Figure D.2: System characteristics using Festo system design software

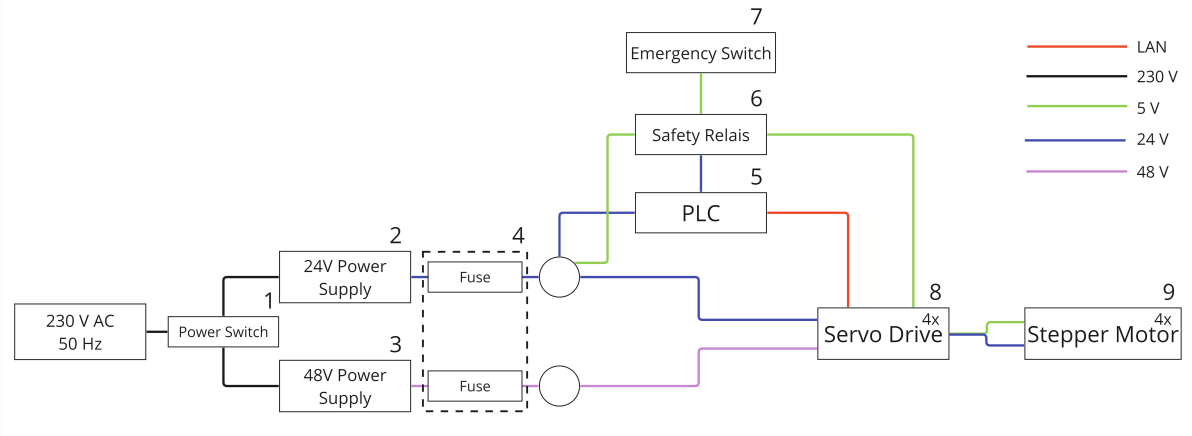


Figure D.3: Block scheme of the control set-up with reference to figure D.4

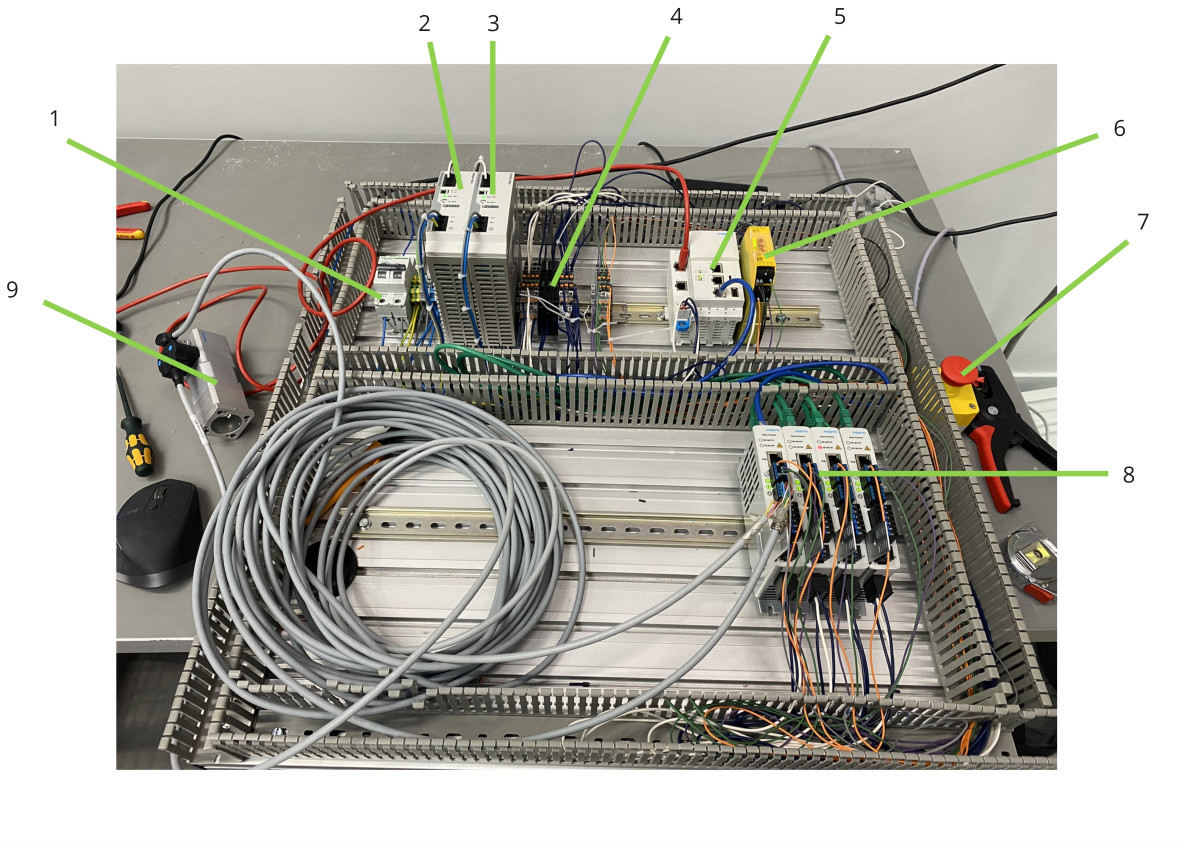


Figure D.4: Control panel with reference numbers

D.2.2. Manipulator

The manipulator is built upon the carts using solid 3D-printed brackets, which are able to withstand large amounts of force compared to infill printers. From these brackets, carbon rods are able to rotate and are combined at the end, closing the 4-bar mechanism. To compensate for small inaccuracies, self-aligning joints are used between the brackets and the rods. These are attached with SLS-printed plugs which clamp from within the carbon rod. Between these 4-bar mechanisms, a prismatic joint is placed which holds the end-effector tip. This prismatic joint is able to rotate in 2DOF at each side, which allows it to translate the intended motion from the sliders to the end-effector. Then the whole setup is attached to an aluminum profile wall at an angle of 90 degrees, with the motors being at the bottom facing forward. The assembly, made in SolidWorks, gives the alignment of the guides towards each other.

D.2.3. Codesys

In order to communicate with the PLC, Codesys was used. Codesys is a Structured Text (ST) PLC programming software that uses function blocks and a logic step approach to execute actions. Different function blocks are used to move actuators, read out data, and calculate output values. For simplification reasons the controllers are set up in a point-to-point configuration, meaning they are individually actuated, instead of in an axis pool. Thus, there is no 'knowledge' about where the other sliders are. In order to read out the data, a so-called 'trace' is installed, which reads out the 'actualPosition', 'actualVelocity', and 'actualTorque' from the controllers over time. Actuation is done using the 'moveAbsolute' function block. The system then waits until it receives a message back, saying the movement has ended and proceeds with the code. The kinematic model is found in function block 'InverseKinematics', which takes the Cartesian coordinates as an input and returns the position of the sliders. A basic flowchart of this operation is shown in figure D.7. Variations in the motion tasks are applied to conduct certain experiments.

Figure D.5: Real setup

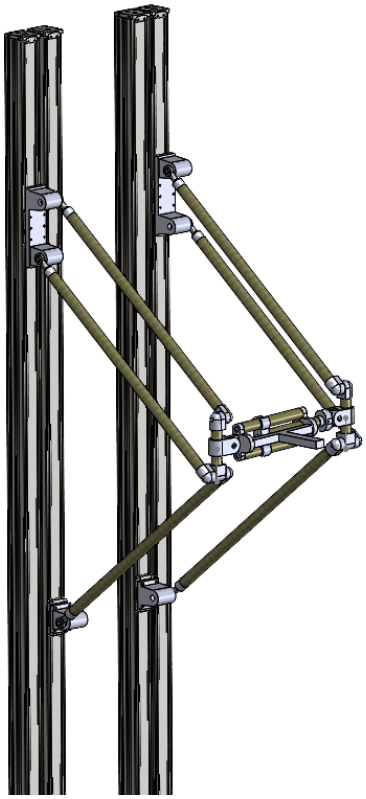


Figure D.6: CAD representation of the setup

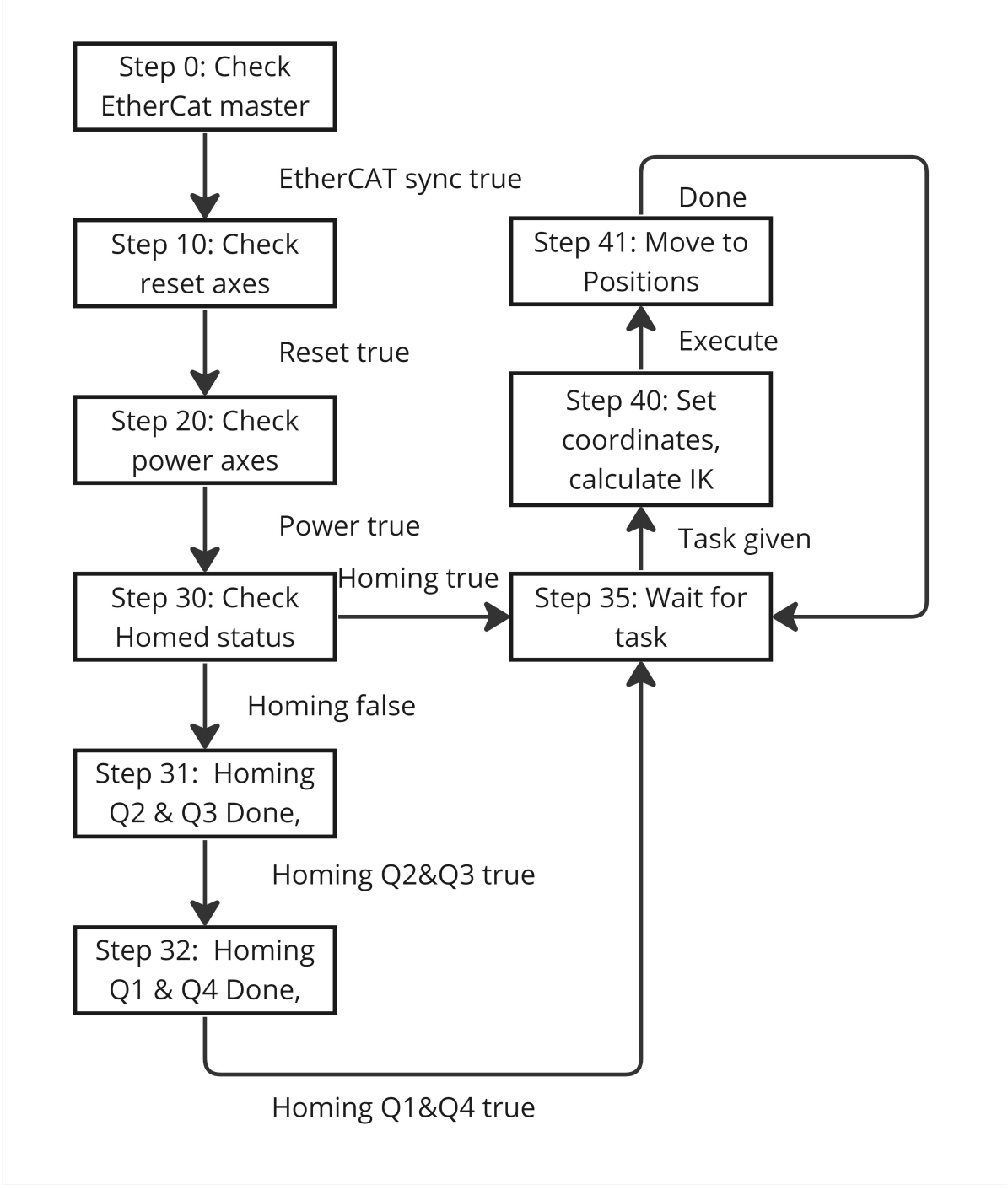


Figure D.7: General flowchart of the logic operation in Codesys

E

Testing and Validation

Other important figures are shown in this section. Here, the prototype positions are plotted on top of the simulation and the kinematic model output. Important to see is that all mediums reach the required points and that the model is properly followed. The difference is in the path controller. As seen, the simulation and the model make use of the same path trajectory generator, which is not the same in Codesys. Codesys uses a linear path planner, which does not calculate the ideal velocity trajectory. Each slider is controlled individually and is given a position to go to. Further research must be done on the implementation of a controller which calculates the ideal trajectory for every slider.

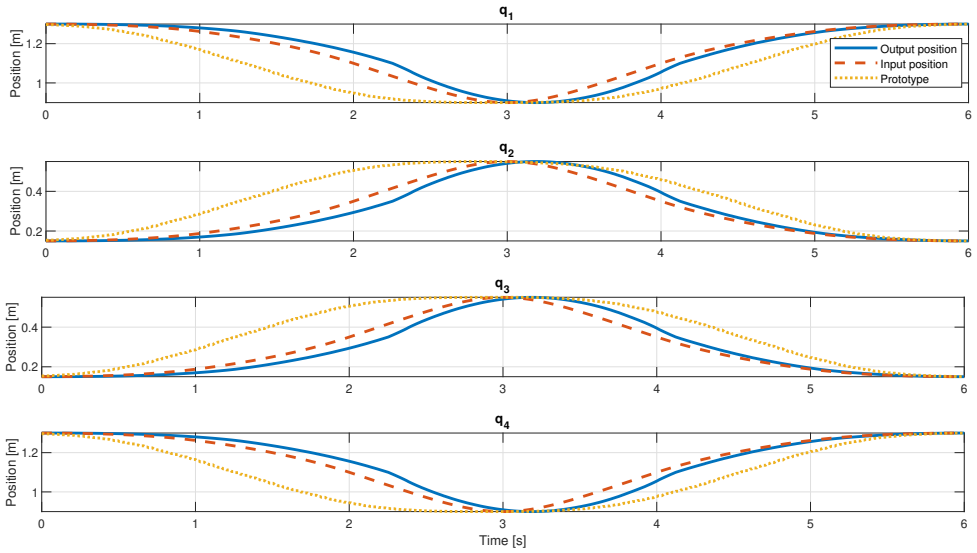


Figure E.1: Slider position for X-motion

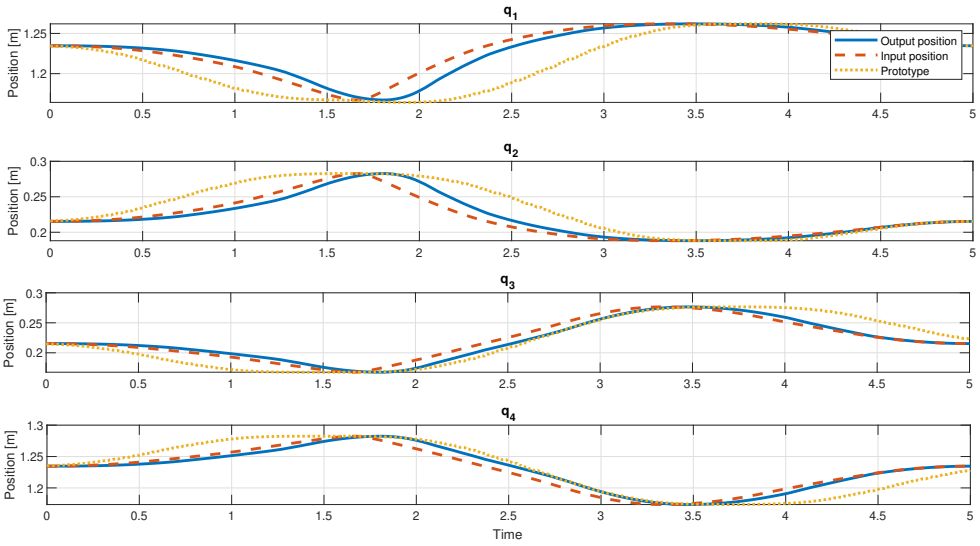


Figure E.2: Slider position for Y-motion

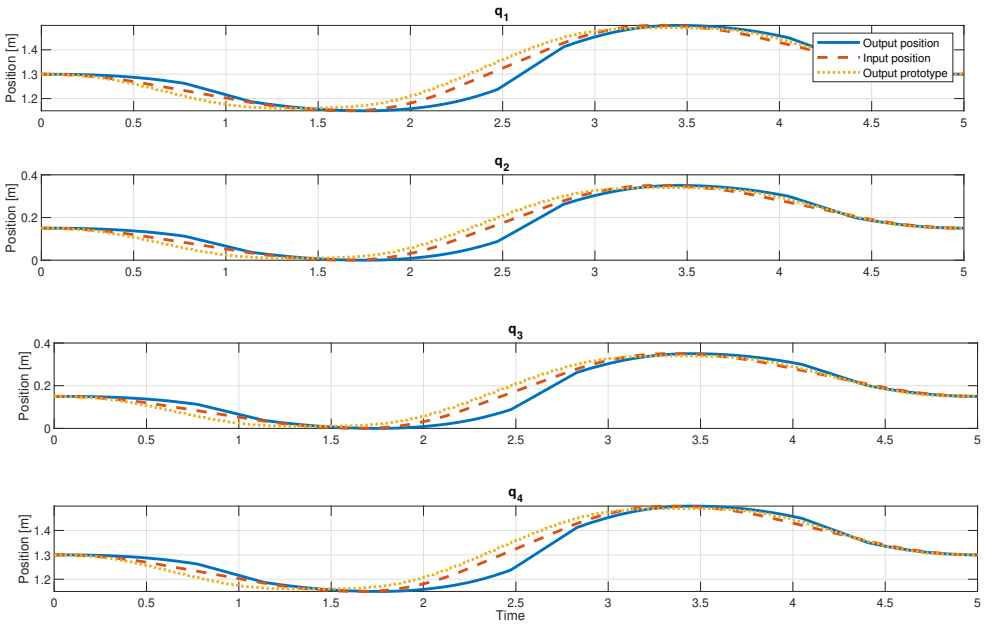


Figure E.3: Slider position for Z-motion

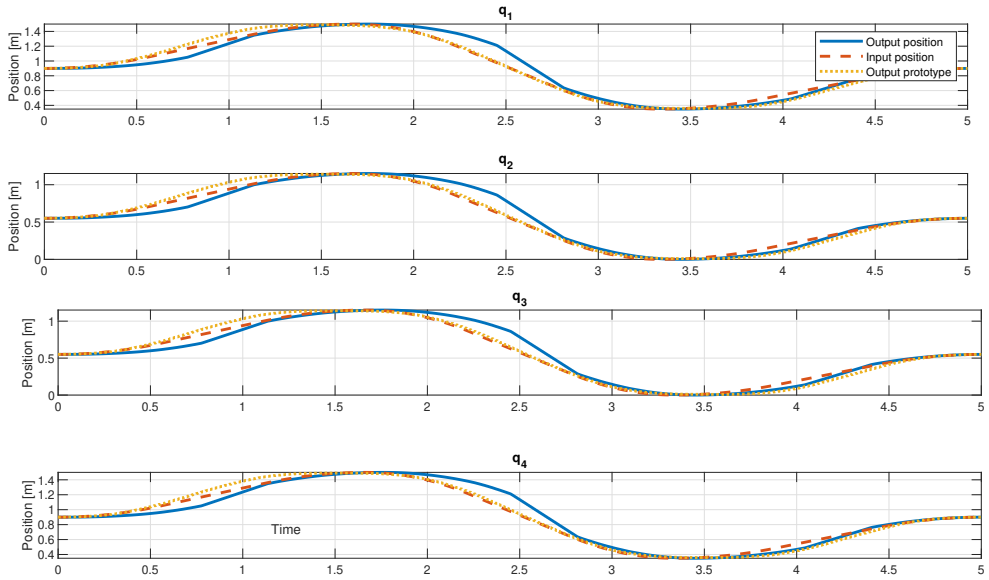


Figure E.4: Slider position for Z*-motion

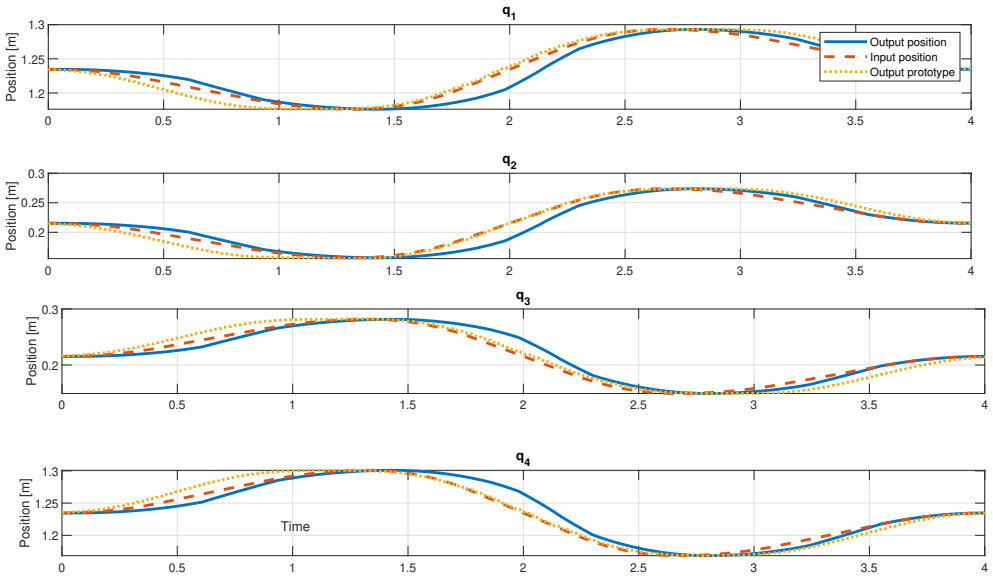


Figure E.5: Slider position for ψ-motion

F

Matlab code

F.1. Simulation parameters

Listing F.1: MATLAB Code for Dimensional Parameters

```
%Dimensional parameters link 1
L1 = 1.5; %m
W1 = 0.01; %m
H1 = 0.01; %m
rho = 2700; %kg/m^3

%Dimensional parameters slider block
L2 = 0.02; %m
H2 = 0.02; %m
W2 = 0.05; %m
rho2 = 2500; %kg/m^3

%Dimensional parameters link
L3 = 0.57; %m
H3 = 0.01; %m
W3 = 0.01; %m
rho3 = 2200; %kg/m^3 Carbon Rod

%Dimensional parameters link between
L4 = 0.1; %m
H4 = 0.01; %m
W4 = 0.01; %m
rho4 = 2200; %kg/m^3 Carbon

%Dimensional parameters link between
L5 = 0.31; %m
H5 = 0.01; %m
W5 = 0.01; %m
rho4 = 2200; %kg/m^3

L6 = 0.025; %m
H6 = H5; %m
W6 = W5; %m

L7 = L5 + 0.01;

```

```
rgb = [0.1 0.2 0.1]; %[R G B]
rgb1 = [0.5 0.5 0.5]; %[R G B]
rgb2 = [0.2 0.2 1.0];
rgb3 = [0.0 1.0 0.0];
```

F.2. Inverse kinematics

Listing F.2: MATLAB Function for Inverse Kinematics

```
function [q_a1, q_cd1, q_a2, q_cd2, d_x1, d_x2, phi, S_extend, alpha1,
    alpha2] = inverseKinematicsSYM(L_EE, L_grip, L, L_c, b, x, y, zz, psi)

    % Define symbols
    syms d_x1 d_x2 gammas phi  S_extend q_a1 q_cd1 q_a2 q_cd2 alphas1
        alphas2
    % d_x1 = output distance in X-direction of the legs on the left [m]
    % d_x2 = output distance in X-direction of the legs on the right [m]
    % gammas = angle of the coupler link [rad]
    % q_a = distance of the slider a [m]
    % q_cd = distance of the slider cd [m]
    % alphas = angle of the legs wrt the slider [rad]

    q = [x y zz psi]';

    % Vector loop equations
    eq1 = d_x1 - (1/2)*L_EE*sin(phi) - (L_grip)*sin((3/2)*pi + phi) == x;
    eq2 = (1/2)*L_EE*cos(phi) + (L_grip)*cos((3/2)*pi + phi) == y;
    eq3 = d_x2 - (S_extend + (1/2)*L_EE)*sin(pi + phi) - (L_grip)*sin
        ((3/2)*pi + phi) == x;
    eq4 = 2*b+(S_extend + (1/2)*L_EE)*cos(phi + pi) + (L_grip)*cos((3/2)*
        pi + phi) == y;

    % Define and solve for d_x1, dx_2, gamma
    eq = [eq1, eq2, eq3, eq4];

    % Solve for closed-vector loop equations and make sure the solutions
        are real
    S = solve(eq, [d_x1, d_x2, phi, S_extend], 'Real', true);

    phi = (S.phi(end));
    S_extend = (S.S_extend(end));
    d_x1 = (S.d_x1(end));
    d_x2 = (S.d_x2(end));

    % Use previous answer for finding the angle alpha
    x1 = L*cos(alphas1) == d_x1;
    x2 = L*cos(alphas2) == d_x2;

    alpha1 = solve(x1, alphas1);
    alpha2 = solve(x2, alphas2);

    alpha1 = abs(alpha1(end));
    alpha2 = abs(alpha2(end));

    % Set equations for the height (z-axis) of the end-effector
    z1 = q_a1 + L*sin(alpha1) + (L_c/2) + L_EE/2*sin(psi) == zz;
```

```
    z2 = q_cd1 - L*sin(alpha1) + L_EE/2*sin(psi) == zz;

    z3 = q_a2 + L*sin(alpha2) + (L_c/2) + (S_extend+L_EE/2)*sin(pi + psi)
       == zz;
    z4 = q_cd2 - L*sin(alpha2) + (S_extend+L_EE/2)*sin((pi + psi)) == zz;

    % Calculate the required position of the sliders
    [q_a1, q_a2, q_cd1, q_cd2] = solve([z1, z2, z3, z4], [q_a1, q_a2,
       q_cd1, q_cd2]);
end
```

F.3. Jacobian and singularity analysis

Listing F.3: MATLAB Code for Kinematics and Jacobian Calculations

```
% Define parameters
run('RobotParameters.m')
syms x y z psi q_a1 q_cd1 q_a2 q_cd2

L = 0.57;
b = 0.16;
L_grip = 0.1;
L_EE = 0.3;
L_c = 0.1;

% Calculate symbolic equations
[q_a1_sym, q_cd1_sym, q_a2_sym, q_cd2_sym, ~, ~, ~, ~, ~, ~] =
   inverseKinematicsSYM(L_EE, L_grip, L, L_c, b, x, y, z, psi);

% Calculate the Jacobian symbolically
J_sym = jacobian([q_a1_sym, q_cd1_sym, q_a2_sym, q_cd2_sym], [x, y, z, psi
   ]);

% Convert the symbolic Jacobian to a numeric function
J_fun = matlabFunction(J_sym, 'Vars', [x, y, z, psi]);

vars = [x, y, z, psi];

% Define the ranges for each variable
x_range = linspace(0.1, 0.75, 20);
y_range = linspace(0.11, 0.18, 20);
z_value = 0.75;
psi_value = 0; % Assuming psi is constant as per your initial guesses

% Create a meshgrid of initial guesses
[X, Y] = meshgrid(x_range, y_range);
initial_guesses = [X(:), Y(:), repmat(z_value, numel(X), 1), repmat(
   psi_value, numel(X), 1)];

% Find solution for every initial guess
solutions = [];
options = optimoptions('fsolve','OptimalityTolerance',1e-12,'StepTolerance
   ',1e-12);
for i = 1:size(initial_guesses, 1)
    x0 = initial_guesses(i, :);

```

```
    [x,fval,exitflag,output] = fsolve(@(vars) equation(vars, J_fun), x0,
        options);

    % Check if solution is unique (not close to already found solutions)
    if isempty(solutions) || min_distance(solutions, x) > 1e-4
         solutions = [solutions; real(x)];
    end
end

% Define constraints for the found solutions
solutions = solutions(solutions(:,1) >= 0 & solutions(:,1) <= 0.75 &
    solutions(:,2) >= 0 & solutions(:,2) < 0.23, :);
solutions_indirect = solutions;
disp('Found solutions:');
disp(solutions);

% Function for setting up the equations
function F = equation(vars, J_fun)
    x = vars(1);
    y = vars(2);
    z = vars(3);
    psi = vars(4);

    L = 0.57;
    b = 0.16;
    L_grip = 0.1;
    L_EE = 0.3;
    L_c = 0.1;

    J = J_fun(x, y, z, psi);
    detJ = det(J);
    invJ = (det(inv(J)));
    inv_detJ = invJ;

    F(1) = detJ;
    F(2) = inv_detJ;
    F(3) = min(svd(J));
end

% Function for not repeating the same points
function minDist = min_distance(solutions, x)
    minDist = Inf;

    for i = 1:size(solutions, 1)
        dist = norm(solutions(i, :) - x); % Compute Euclidean distance
        if dist < minDist
            minDist = dist;
        end
    end
end
```

F.4. Check feasibility of the points in the reachable workspace

Listing F.4: MATLAB Script for Feasibility Analysis and 3D Plotting

```
%% Parameters
```

```
run('RobotParameters.m');

syms xx yy zz psii
L = 0.57;
L_c = 0.1;
S = L6;
L_EE = 0.3;
b = 0.16;
L_grip = 0.1;
n = 40 ; % Number of samples

% Define the range and step size for X, Y, and Z coordinates of the end-effector
xRange = linspace(0,0.7,n);
yRange = linspace(0.101,0.2,n);
zRange = linspace(0,1.5,n);
psiRange = [0, 0.4];
feasiblePoints = zeros(n^4,4);

index = 1; % Initialize an index variable to keep track of the feasible points

tic
% Find symbolic relation for inverse-kinematics
[q_a1_sym, q_cd1_sym, q_a2_sym, q_cd2_sym, ~, ~, ~, ~] = forwardKinematicsSyms(L_EE, L_grip, L, L_c, b, xx, yy, zz, psii);
%%
for psi = psiRange
    for x = xRange
        for y = yRange
            for z = zRange
                endEffectorPos = [x, y, z, psi];
                q_a1 = double(subs(q_a1_sym, {xx, yy, zz, psii}, {endEffectorPos(1), endEffectorPos(2), endEffectorPos(3), endEffectorPos(4)}));
                q_cd1 = double(subs(q_cd1_sym, {xx, yy, zz, psii}, {endEffectorPos(1), endEffectorPos(2), endEffectorPos(3), endEffectorPos(4)}));
                q_a2 = double(subs(q_a2_sym, {xx, yy, zz, psii}, {endEffectorPos(1), endEffectorPos(2), endEffectorPos(3), endEffectorPos(4)}));
                q_cd2 = double(subs(q_cd2_sym, {xx, yy, zz, psii}, {endEffectorPos(1), endEffectorPos(2), endEffectorPos(3), endEffectorPos(4)}));
                sliders = [q_a1, q_cd1, q_a2, q_cd2];
                % Check if the solution is feasible
                if actualCheck(sliders, b, psi)  % Implement checkFeasibility function
                    feasiblePoints(index, :) = [x, y, z, psi];
                    index = index + 1;
                end
            end
        end
    end
end
toc
```

```
46
47 % Trim the extra zeros from the preallocated matrix (if necessary)
48 feasiblePoints = feasiblePoints(1:index-1, :);
49
50 %%
51 % Calculate the convex hull
52 psiCondition = feasiblePoints(:,4);
53 K = convhull(feasiblePoints(:,1:3));
54 Kpsi = convhull(feasiblePoints(psiCondition ~= 0,1:3));
55
56 % Extract the vertices of the convex hull
57 vertices = feasiblePoints(unique(K(:)), 1:3);
58 verticesPsi = feasiblePoints(unique(Kpsi(:)), 1:3);
59
60 % Calculate the volume of the convex hull
61 reachableVolume = convexHullVolume(vertices);
62 reachableVolumePsi = convexHullVolume(verticesPsi);
63
64 disp(['Reachable Volume: ', num2str(reachableVolume), ' m^2']);
65
66 % Extract the x, y, and z coordinates from the feasiblePoints matrix
67 x_coords = feasiblePoints(:, 1);
68 y_coords = feasiblePoints(:, 2);
69 z_coords = feasiblePoints(:, 3);
70
71
72 %Extract the points which are accessible if the EE is rotated
73 x_coords_psi = feasiblePoints(psiCondition ~= 0, 1);
74 y_coords_psi = feasiblePoints(psiCondition ~= 0, 2);
75 z_coords_psi = feasiblePoints(psiCondition ~= 0, 3);
76
77 %% plot
78 % Create a 3D scatter plot
79 figure;
80 scatter3(x_coords, y_coords, z_coords, 'filled', 'MarkerFaceColor', 'blue'
       ); hold on;
81 scatter3(x_coords_psi, y_coords_psi, z_coords_psi, 'filled', '
       MarkerFaceColor', 'red');
82 xlabel('X');
83 ylabel('Y');
84 zlabel('Z');
85 title('Feasible Points in 3D Space');
86 legend('\psi = 0', '\psi = pi/3')
87 grid on;
88
89 % Set xlim, ylim, and zlim to match the ranges
90 xlim([min(xRange), max(xRange)]);
91 ylim([-0.5, 0.5]);
92 zlim([min(zRange), max(zRange)]);
93
94 plotcube([0.45, 0.15, 1.5], [0, 0, 0], 0.1, [1 0 0]); hold on;
95 scatter3(0, 0, 0, 'filled', 'MarkerFaceColor', 'g', 'SizeData', 100);
96 legend('Origin (Green Dot)');
97
98 % Plot outer boundary
99 k = boundary(feasiblePoints(:,1:3));
```

```
100
101 figure();
102 trisurf(k, feasiblePoints(:,1), feasiblePoints(:,2), feasiblePoints(:,3),
        ...
103     'Facecolor', 'red', 'FaceAlpha', 0.5);
104 xlabel('X-axis'); ylabel('Y-axis'); zlabel('Z-axis');
105 title('Refined 3D Surface Plot');
106 grid on;
107 set(gca, 'LineWidth', 1.5, 'Color', [0.95 0.95 0.95], 'FontSize', 12);
108
109 %% Function for adding constraints to the found solutions
110 function feasible = actualCheck(sliders, b, psi)
111     feasible =  all(sliders >= 0) && ...                % All sliders
            above pos=0m
112                 all(sliders <= 1.5) && ...              % All sliders
                        below pos=1.5m
113                 (sliders(2)-sliders(1)) > 0.2 && ...    % Minimal distance
                        between two same-side sliders >0.2
114                 (sliders(4)-sliders(3)) > 0.2 && ...
115                 (sliders(2) - sliders(1)) < 1.25 && ... % Maximum distance
                        between two same-side sliders >1.25
116                 (sliders(4) - sliders(3)) < 1.25 && ...
117                 (sliders(1) < sliders(2)) && ...
118                 (sliders(3) < sliders(4)) && ...
119                 (abs(mean([sliders(1),sliders(2)]) - mean([sliders(3),
                        sliders(4)])) <= 2*b*tan(psi));   %Check if angle psi
                        is correct implemented
```

G

Other

G.1. FlexCraft project

The Flexcraft project stands as a beacon of innovation in the realm of horticultural automation, focusing on an approach that encompasses world perception, control, gripping mechanisms, and vision systems. This ambitious endeavor aims to merge these diverse fields to accelerate research and development in automated horticulture, a sector poised for significant growth. From the insightful presentations witnessed, it becomes apparent that the project's central focus is not on the robotic manipulators themselves. Intriguingly, the manipulators currently in use within the project appear to be not entirely optimized for the tasks at hand. Despite this, the overarching goal is clear: to ensure these robotic systems have complete accessibility to the crops they tend. This requirement poses a unique challenge, as the manipulators must operate in constrained spaces, typical of horticultural environments, where maneuverability is limited. A critical issue identified is the need for reconfiguration due to singularities, a problem arising from employing generic, or manipulators with strong overcapacity for specialized tasks. This challenge underscores the necessity for more tailored robotic solutions in horticulture, where precision and adaptability are key. The talks with experts highlight an important aspect of the Flexcraft project. While significant strides have been made in integrating automation into horticulture, there remains a crucial need for thoughtful consideration in the development of the perfect manipulator. Such a manipulator would not only have to navigate the spatial constraints of agricultural environments but also perform tasks with the precision and delicacy required in handling crops.

G.2. Greentech 2023

Greentech 2023 was a hub of innovation, especially in the field of horticulture. This year, the spotlight was on harvesting automation, a sector crucial to the future of agriculture and food production. Notably, the presence of three pioneering companies - Dogtooth, Ridder (Grow), and Priva - highlighted this theme, demonstrating the specialized nature and challenges of this emerging field. Conversations with representatives from these companies revealed the intricacies and challenges of integrating high-tech solutions into horticulture. The complexity of these technologies and their application in diverse agricultural contexts underscore the need for specialized knowledge and continuous innovation in this sector. Despite the challenges, the presence of these companies at Greentech 2023 was a testament to the sector's potential. The event was not only a display of current technological capabilities but also a platform for discussing the future needs and directions of horticultural automation. It underscored the critical role that automation could play in addressing one of the most pressing issues of our time: sustainably feeding a growing global population.