

Fakultät für Maschinenwesen
Lehrstuhl für Angewandte Mechanik

Agricultural Manipulators

Simulation, Design and Motion Planning

Jörg Thomas Baur

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der
Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. dr. ir. Daniel Rixen

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. habil. Heinz Ulbrich (i.R.)
2. Univ.-Prof. Dr.-Ing. Gunther Reinhart

Die Dissertation wurde am 27. Oktober 2014 bei der Technischen Universität
München eingereicht und durch die Fakultät für Maschinenwesen am
9. März 2015 angenommen.

Abstract

This PhD thesis covers several topics that are essential for the design of a modular manipulator for the agricultural tasks of selective fruit harvesting and precision spraying of pesticides. Besides the kinematic design and its evaluation, the modeling and simulation of the dynamics with experimental parameter identification has been performed. Moreover, a motion planning strategy is proposed, comprising a workspace planner and an efficient inverse kinematics algorithm with redundancy resolution, suitable for on-line operation. Two manipulator generations were designed and three prototypes were realized in total. All the implemented methods were verified on the developed manipulator prototypes in laboratory and field experiments for spraying of grapevine and harvesting of sweet pepper, apples and grapes.

Danksagung

Die vorliegende Arbeit ist in meiner Zeit als wissenschaftlicher Mitarbeiter an dem Lehrstuhl für Angewandte Mechanik der Technischen Universität München entstanden.

An erster Stelle möchte ich mich herzlich bei meinem Doktorvater Prof. HEINZ ULBRICH bedanken. Durch sein Vertrauen hatte ich die einzigartige Möglichkeit an diesem spannenden Thema zu forschen. Insbesondere seine andauernde Unterstützung und die vielfältigen Möglichkeiten des Lehrstuhls haben entscheidend zum Gelingen der Dissertation beigetragen. Prof. DANIEL RIXEN danke ich für das Interesse an meiner Forschungsarbeit und der Übernahme des Prüfungsvorsitzs. Außerdem danke ich Prof. GUNTHER REINHART für seine Anregungen und die Erstellung des Zweitgutachtens.

Für die vielen Diskussionen, guten Ideen und dem positiven Arbeitsklima bin ich allen Kollegen des Instituts sehr dankbar. JULIAN PFAFF hat durch seine mechanische Konstruktion der Manipulator-Prototypen entscheidend zum Erfolg dieser Arbeit beigetragen. Bei CHRISTOPH SCHÜTZ bedanke ich mich an dieser Stelle für die tatkräftige Unterstützung in den unterschiedlichsten Bereichen. Die intensive und produktive Zusammenarbeit in unserem Team werde ich stets in positiver Erinnerung behalten. Für die Einarbeitung zu Beginn meiner Lehrstuhlzeit und Übernahme zahlreicher organisatorischer Aufgaben bin ich THOMAS VILLGRATNER überaus dankbar. Ohne die Erfahrung von GEORG MAYR und seiner unermüdlichen Arbeit an der Elektronik für die Roboter wäre diese Arbeit unmöglich gewesen. Genauso danke ich SIMON GERER, PHILIP SCHNEIDER, TOBIAS SCHMID und GEORG KÖNIG für die mechanische Fertigung der Manipulator-Prototypen. Für die exzellente Koordination der Werkstätten und weitreichende Unterstützung in unterschiedlichsten Belangen bin ich THOMAS THÜMMEL zu großem Dank verpflichtet.

Die enge Zusammenarbeit mit der Forschungsgruppe für humanoide Roboter des Lehrstuhls hatte einen entscheidenden Einfluss auf meine Arbeit. Die zahlreichen Ideen und Anregungen von THOMAS BUSCHMANN waren für mich eine große Hilfestellung. Außerdem bedanke ich mich bei MARKUS SCHWIENBACHER für seine intensive Unterstützung insbesondere bei der Erstellung des Kollisionsmodells für die Manipulatoren.

Die studentischen Arbeiten von CHRISTOPH SCHÜTZ (pneumatischer Muskel), THOMAS SCHILDHAUER (Modellierung) und SEBASTIAN DENDORFER (modularer Prüfstand) sind außerdem ein wichtiger Bestandteil für den Erfolg dieser Arbeit.

Bei meiner Freundin CLAUDIA bedanke ich mich für die durchgängige und verständnisvolle Begleitung während meiner gesamten Promotion.

Abschließend ist die gute Zusammenarbeit innerhalb des internationalen Konsortiums des Forschungsprojekts CROPS, sowie die finanzielle Förderung durch die Europäische Union hervorzuheben.

Contents

1	Introduction	1
1.1	Literature Review and Related Work	2
1.2	Outline of this Thesis	5
2	The European Project CROPS	7
2.1	Project Objectives	7
2.2	Overview and Organization	7
2.3	Applications	9
2.3.1	Selective Harvesting	9
2.3.2	Selective Spraying of Grapes	14
3	Manipulator Design	15
3.1	Introduction	15
3.2	Kinematic Design	16
3.2.1	The Manipulability Measure	19
3.2.2	Analysis of Manipulator Workspace	22
3.3	Selection of the Actuation Principle	24
3.3.1	Pneumatic Artificial Muscles	25
3.3.2	Electrical Drives	37
3.3.3	Summary and Conclusion	37
3.4	Modeling and Simulation	38
3.4.1	Direct Kinematics	38
3.4.2	Dynamics	39
3.4.3	Joint Friction Modeling	42
3.4.4	Self-Collision Model	44
3.5	Hardware and Control Architecture	46
3.5.1	Hardware Overview	47
3.5.2	Decentralized Joint Control	49
3.5.3	Improvements with the Final Manipulator	50
3.6	Software Architecture	51
3.6.1	Real-Time System	52
3.6.2	Further Remarks on the Implementation	55
3.6.3	Host Interface	55
3.7	The Agricultural Manipulator Prototypes	59
3.8	Chapter Summary	59
4	Modular Test Bed for Robot Drives	61
4.1	Introduction	61
4.2	Related Work	62
4.3	Dynamic Modeling of Robot Drives	63
4.4	Test Procedure and Requirements	65
4.4.1	Test Procedure	65

4.4.2	Requirements and Specifications	65
4.5	Design and System Architecture	66
4.6	Experiments and Results	68
4.6.1	Friction Torque Measurements	71
4.6.2	Efficiency Measurements	75
4.7	Chapter Summary	78
5	Motion Planning	79
5.1	Introduction	79
5.2	Overview and Related Work	79
5.2.1	Sampling Based Planning	81
5.2.2	The Planning Approach	83
5.3	Planning in the Manipulator Workspace	84
5.3.1	Task Space Definition	84
5.3.2	A Heuristic Approach	85
5.3.3	Navigation Potential Function	90
5.3.4	Results and Discussion	93
5.4	Inverse Kinematics	95
5.4.1	Redundant Manipulators	97
5.4.2	Local Optimization of Kinematic Redundancy	97
5.4.3	Secondary Objective Functions	99
5.4.4	Singularity Robust Inverse Kinematics	106
5.5	Chapter Summary	108
6	Field Experiments and Results	109
6.1	Selective Harvesting	109
6.1.1	Sweet Pepper Harvesting	109
6.1.2	Apple and Grape Harvesting	111
6.2	Selective Spraying of Grapevine	113
7	Conclusion and Outlook	115
7.1	Summary	115
7.2	Discussion and Recommendations for Future Work	117
A	Basic Technical Data	119
B	Manipulator Kinematics	121
C	Gear Friction of Harmonic Drives	123
D	Experimental Friction Identification	127

1 Introduction

Since the establishment of robot manipulators for industrial automation, starting in the 1960s, a lot of research has been performed to extend the field of applications for robots in several areas. A very challenging domain in robotics is the automation to selectively harvest fruits or vegetables. Many researchers have worked on this multidisciplinary problem, facing highly unstructured and rough environments as well as inherent variations of biological products. But so far only very few commercially available agricultural robots for selective harvesting applications exist. Out of the following reasons, automation in agriculture is of great relevance.

According to an estimation of the POPULATION DIVISION from the United Nations (UN)¹, the world population will increase from 7.2 billion in 2013 to 8.1 billion in 2025, while in 2050 a population of 9.6 billion is expected. While the population is mostly increasing in developing countries, in the European Union (EU), the population is more or less becoming stagnant. However, the European Commission for AGRICULTURE AND RURAL DEVELOPMENT reports a reduction of 4.8 million full-time jobs in the field of agriculture in the period from 2000-2012². While the growing population leads to the requirement of an increased agricultural efficiency in the food production process, the decreasing labor force is indicating the progress of automation. Actually, this tendency is no recent development. According to Sistler [1987] throughout history, agricultural production has improved, initially by the replacement of manpower with animal power and later on by the introduction of steam engines. Around the year 1930, large-scale farming machines were more and more applied. However, compared to other industries, robotic manipulation is still less common in agriculture. In general, this can be explained by the following major reasons. First, there are many different tasks, like planting, pruning or harvesting, making it difficult to design a machine that can be operated at full capacity during the whole season. Furthermore, the robot must be suitable for the rough environmental conditions, like high or low temperatures, dust and mud, or humidity, commonly encountered in agriculture. Finally, the rather cheap agricultural products have inherent variations and a successful and reliable automation is therefore very complex and hence expensive. But the intensifying of automation in agricultural production processes is of great relevance today and in the near future due to increasing labor costs, particularly in industrial countries, as well as the unhealthy working conditions, especially in greenhouses. Furthermore, in completely automated greenhouses, the environmental conditions could be

1 United Nations, Department of Economic and Social Affairs, Population Division (2013). *World Population Prospects: The 2012 Revision, Highlights and Advance Tables*. Working Paper No. ESA/P/WP.228.

2 The report is available online at http://ec.europa.eu/agriculture/rural-area-economics/briefs/pdf/08_en.pdf, visited on October 2014.

adapted, for example by rising the amount of carbon dioxide in the air, towards more favorable conditions for the plants, resulting in higher yields and hence productivity (Wittwer and Robb [1964]).

This thesis deals with the simulation, design and the implementation of algorithms for motion planning of modular manipulators that are suitable for several different agricultural tasks, like selective harvesting of single fruits or precision spraying of pesticides. The results were obtained within a collaborative research project with the acronym CROPS (cf. Chapter 2) and the work was partly funded by the EU. Other topics relevant for the operation of an agricultural robot related to the mechanical design, object recognition as well as localization, and end-effector design are beyond the scope of this work.

1.1 Literature Review and Related Work

Within this Section, an overview on reported systems for automation in agriculture will be given. The goal is to provide the reader an impression on available systems without going into many technical details. Relevant literature for other topics discussed in this thesis are referenced in the corresponding Chapters.

The automation of agricultural tasks with robot manipulators is closely related to the introduction of manipulators for industrial production. Therefore, the literature review will begin with a brief history on the introduction of robots in industry. A very detailed overview on major breakthroughs in this field is provided in the textbook of Siciliano and Khatib [2008] and by a report from the INTERNATIONAL FEDERATION OF ROBOTICS³.

The patent that led to industrial robots has been submitted by GEORGE DEVOL in 1954 and was accepted in 1961 (Devol [1961]). The first manipulator with hydraulic actuation, called UNIMATE was produced in 1959 and had been sold by the first robot company UNIMATION. It has been applied for spot-welding and workpiece handling tasks. In the 1970s, robots were mainly used in the car production industry and the company ASEA (nowadays ABB) designed the first robot completely actuated by electrical drives. This robot, which is called IRB-6, was capable of moving along a continuous path, making it suitable for arc-welding procedures. In 1978, HIROSHI MAKINO and his team developed the so-called Selective Compliance Assembly Robot Arm (SCARA) with four Degrees of Freedom (DoFs). Due to the kinematics of the robot, this design is well-suited for assembly tasks as well as fast pick-and-place operations. Because of the high accuracy and repeatability capabilities of robots, typical applications were (and still are): car assembly, painting, machining, workpiece handling and suchlike.

By MOTOMAN, a trademark of the Japanese company YASKAWA, the first commercially available two handed robot was introduced in 2005. With the dual-arm design, inspired by human torsos with similar reachability and dexterity, such

3 The report is available online at http://www.ifr.org/fileadmin/user_upload/downloads/forms___info/History_of_Industrial_Robots_online_brochure_by_IFR_2012.pdf, visited on May 15, 2015.

systems can replace human labor with a minimum amount of required modifications to the production process. Since 2006, the company KUKA is selling and developing a lightweight seven DoFs manipulator. The first version had a payload of 7 kg with the remarkable low weight of 16 kg. This robot was originally developed by the German Aerospace Center (DLR) and each joint is equipped with torque sensors (Hirzinger et al. [2002]). The recent advances in robotic hardware components, like increased sensing capabilities, lightweight structures, and improved dexterity make manipulators more suitable for collaborative human-robot interactions. In comparison to the traditional automation with robots, i.e. strict separation of human labor and robot manipulators out of safety considerations, this tendency allows for new automation concepts in a cage-free production environment. The robot BAXTER, a dual-arm humanoid robot from the company RETHINK ROBOTICS, is a recent example for the automation in modern production, suitable for the close cooperation with human beings.

Although a lot of research and development has been performed in the last decades to increase the autonomy of manipulators, most of the robots today are still utilized for large-scale and highly standardized production tasks performing repetitive motions in a well-defined environment.

Contrary to industrial manipulators, that are very well defined by a standard specification (ISO 8373), the definition of an agricultural or bioproduction robot is not available (Kondo and Ting [1998a]). This can be explained by the huge variety of different tasks with a wide range of requirements. Typical tasks in agriculture are related to: soil preparation, planting, cultivation, trimming, watering, spraying, harvesting, milking, sheep shearing, slaughtering, and so on. For some of the aforementioned tasks, automated solutions already exist (e.g. automatic milking).

Schertz and Brown [1968], although far from a complete technical solution, already discussed possibilities to mechanize the citrus fruit harvesting process. They considered mass (or bulk) and individual-fruit (or selective) harvesting, which is a commonly made distinction when discussing automated harvesting. Bulk harvesting systems, based on tree shaking mechanisms are already commercially available. An efficient and labor-saving system is applied for grape harvesting, sold for example by the company NEW HOLLAND AGRICULTURE. Similar harvesters exist for citrus fruits (OXBO), apples (Peterson et al. [1999]) or for cherry trees (MUNCKHOF).

Although the bulk harvesting process is very efficient, mechanical damage to the product is hard to avoid and therefore this approach can not be applied to all agricultural products (Sarig [1993]). In particular for high value and high quality crops, the design of a selective harvesting machine is a worthwhile consideration. A lot of research has been performed, starting around the late 1980s, on selective harvesting machines. According to Kondo and Ting [1998a], the earliest robot of this kind has been developed by KAWAMURA and others from the KYOTO UNIVERSITY in Japan in 1982 for tomato harvesting. The complete system consisted of a five DoFs manipulator, a stereo vision system, an end-effector and a mobile platform. In Japan, many additional systems for selective fruit harvesting have been developed in the 1990s for tomato, cherry tomato, cucumber and strawberries by Kondo et al. [1995], Kondo et al. [1996], Kondo and Ting [1998b], and Arima

and Kondo [1999]. Besides mechanisms for the harvesting end-effector tool, the kinematics design, based on the manipulability measure is discussed in the work of NAOSHI KONDO and his team. A good overview on developed agricultural robots around that time is given in the textbook of Kondo and Ting [1998a]. For a cultivation system of grapevine with hanging berries, Monta et al. [1995] developed a robot for grape production. According to Sarig [1993] and Tillett [1993], an apple harvesting robot, called MAGALI, had been developed by a french research team. The first version, initially actuated by hydraulics, utilized a manipulator with three DoFs which was mounted at the end of an elevating arm. Tillett [1993] shared his view of the requirements on an agricultural robot in his review article. He claimed that an accurate sensing system is of particular importance. Furthermore, he proposed the application of pneumatic actuators and pointed out that pneumatic muscles might be worth an investigation for agricultural robots (see Section 3.3.1).

From the year 2000 until today, many researchers contributed to the development of selective harvesting machines for various applications. Reed et al. [2001] developed a laboratory as well as a field demonstrator for mushroom harvesting. According to his study, a Cartesian space manipulator was sufficient for this application. Within two independent projects, Edan et al. [2000] and Sakai et al. [2002, 2008] developed a harvesting robot for watermelons. In the work of Sakai et al. [2002], the focus was on the mechanism and the mechanical design as well as on the evaluation of the kinematics, with respect to reachable workspace and manipulability. Finally, they proposed the application of parallel kinematics. Hayashi et al. [2002] designed a harvesting robot for eggplants, with focus on the development of a vision system, feedback control algorithm for the positioning of the robot and an end-effector tool. An industrial manipulator with six DoFs has been mounted on a linear slide for a cucumber harvesting robot by van Henten et al. [2002]. Many prototypes for autonomous harvesting of oranges were developed, for example by Muscato et al. [2005] or Sivaraman and Burks [2007]. A detailed description of the mechanical design of the manipulator for orange picking is provided in the PhD thesis of Sivaraman [2006]. The combination of an industrial manipulator with an orchard platform has been utilized for apple harvesting by Baeten et al. [2008] while Guo et al. [2010] also designed an apple harvesting robot which has been evaluated in laboratory experiments. A very fast system with four custom designed picking manipulators (1 s picking time per fruit and arm) for harvesting kiwi fruits was designed by Scarfe et al. [2009]. With particular focus on the vision system Kitamura and Oka [2005, 2006] worked on a sweet pepper harvesting robot and more recently, Bachche [2013] presented his work also for a sweet pepper harvesting system with considerations on the localization and characterization as well as the grasping and detaching of the fruits. Other systems for harvesting were reported in literature for cherries (Tanigaki et al. [2008]), radicchio (Foglia and Reina [2006]), dates (Aljanobi et al. [2010]), asparagus (Irie and Taguchi [2014]; Irie et al. [2009]), and strawberries (Han et al. [2012]; Hayashi et al. [2010]; Rajendra et al. [2011]). Remarkably, for the strawberry harvesting, the cultivation system was adapted to make it more suitable for automation and the Japanese company SHIBUYA SEIKI Co.LTD is planning the commercialization of

the robot. Another system for strawberry harvesting is already available, sold by the company AGROBOT⁴, which has been founded in 2009. A very recent review article on selective harvesting robots was published by Bac et al. [2014b].

Using robot manipulators in agriculture for the spraying of pesticides is less common. If not the whole canopy must be sprayed but only infected areas should be targeted, there is a high potential to reduce the amount of required fluids. A platform, equipped with a detection system and several spray nozzles was reported by Gillis et al. [2001]. When moving along the canopy, the infected areas are detected and only the required spray nozzles are operated. Other spraying systems are emphasizing on the autonomous greenhouse navigation (Mandow et al. [1996]; Sammons et al. [2005]). Shapiro et al. [2009] developed a spraying robot for palm trees and performed experiments with a scaled down prototype. An automatic spraying device has been developed by Li et al. [2009] with a simple robot for the positioning of the nozzle.

1.2 Outline of this Thesis

The first objective of this thesis is to provide the basis for the mechatronic design of agricultural manipulators for specific harvesting and spraying tasks. Therefore, a framework for kinematic and dynamic simulations has been implemented. Another major objective was the development of suitable motion planning algorithms for the agricultural robots. Within this project three dexterous manipulator prototypes were developed for the application in agriculture at the Institute of Applied Mechanics (AM). The first two prototypes, developed from the very beginning, have no significant differences and were designed, manufactured and brought to service in a remarkable short time of one year and three months by JULIAN PFAFF and myself in close cooperation with the institute's internal mechanics and electronics machine shops as well as the support of THOMAS VILLGRATTNER. Throughout this thesis, no distinction is made between these first two prototypes and they are referred to as the *first manipulator prototype*. The experiences from field and laboratory experiments with the first manipulator prototype contributed to the deeply revised design of a following manipulator, referred to as the *final manipulator prototype*. The improvements with respect to the mechanics of the final manipulator will be reported in the PhD thesis of JULIAN PFAFF while other enhancements, in particular with respect to the electronics architecture are provided in this document. The main contributions of the thesis are:

- Mechatronic design of manipulators:
 - Kinematic design and evaluation
 - Evaluation of Pneumatic Artificial Muscles (PAMs) for the actuation
 - Electronics and software architectures
- Dynamic modeling and simulation, partially including experimentally identified friction models.

⁴ <http://www.agrobot.com/>, visited on June 5, 2015.

- Motion planning and inverse kinematics for the redundant manipulators including the definition of problem-specific secondary objective functions.
- Verification of the manipulator hardware and algorithms in field experiments that were carried out by project partners.

The thesis is organized as follows. In Chapter 2, a brief introduction to the research project CROPS with its main objectives and organization will be given. Furthermore, the intended applications of the designed agricultural manipulators are described and basic requirements on the manipulator workspace are formulated.

Chapter 3 provides a detailed description of the mechatronic manipulator design. Based on the derivation of the kinematics, alternative actuator concepts are evaluated, and the equations for dynamics simulation as well as a simplified self-collision model⁵ are applied. Besides the illustration of the hardware and control concept, this Chapter also focuses on a detailed description on the real-time software architecture. Finally, the developed manipulator prototypes are presented.

The Chapter 4 deals with the design of a modular test bed for experimental friction identification in the robot drive modules of different sizes, developed by JULIAN PFAFF for the final manipulator prototype. The curve fitting results of measurements to different friction laws are presented and discussed.

Initially, in Chapter 5, a brief overview on available motion planning strategies will be given. The selected planning approach for the harvesting and precision spraying tasks, especially designed for this agricultural robot is introduced. Afterwards, particular attention is drawn to the inverse kinematics algorithms of kinematically redundant manipulators and its suitability for on-line computation.

Chapter 6 describes the field experiments that were carried out with the developed manipulators and algorithms. Besides the listing of major hardware components, like sensors, end-effectors and so forth utilized for the experiments, key data obtained during each precision spraying and selective harvesting measurements is reported. Another intention of this Chapter is to illustrate photographs of the integrated agricultural robot to provide the reader an impression of the overall system in the field.

5 The self-collision model is based on the framework developed by Schwenbacher [2013].

2 The European Project CROPS

The collaborative research project CROPS with the full title „Intelligent sensing and manipulation for sustainable production and harvesting of high value crops, clever robots for crops“ was mainly funded by the EU as part of the *Seventh Framework Programme* in the theme „nanosciences, nanotechnologies, materials & new production technologies“ within the call „automation and robotics for sustainable crop and forestry management“. The project number is 246252 and its call identifier FP7-NMP-2009-LARGE-3.

2.1 Project Objectives

CROPS aimed at the development of agricultural machinery and its algorithms for high value crops. Contrary to automation with bulk harvesting machines, CROPS followed the concept of selectively pick, spray or evaluate the ripeness of single fruits. By utilizing a modular design, the purpose was the design of a highly configurable system which is able to adapt to several tasks and conditions. Field tests of the developed demonstrators were conducted in greenhouse experiments for sweet pepper harvesting and precision spraying of grapes as well as for apple harvesting and grapes in orchards. The detection and classification of obstacles and other objects in forests for autonomous navigation were also considered. However, they played only a minor role in the project. Besides the development of prototypes, economic aspects, like the economic feasibility, were analyzed.

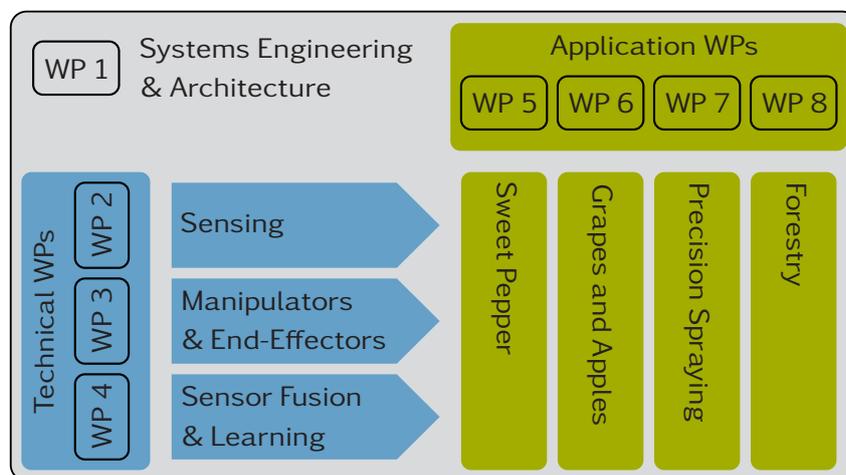
2.2 Overview and Organization

The project started in October 2010 and was scheduled for a duration of four years until September 2014 with an overall contribution from the EU of 7.64 million Euro. The consortium, composed of participants from research facilities as well as from industry, had fourteen partners in total, mostly from Europe, but also from Israel and Chile. The project was organized in Work Packages (WPs) and an overview of the thirteen WPs is given in Table 2.1. WP one, responsible for the systems engineering and architecture, was coordinating the partners from a technical perspective. Besides, it was devoted to the gathering of system specifications, the development of a software framework and monitoring of the integration process of system components. The so-called application WPs (five to eight) were lead by experts from the field of agriculture and were amongst others responsible for the derivation of requirements. Furthermore, they worked on the practical integration and field experiments while providing input and feedback according to their expertise in the field of agriculture and forestry to the technical experts

Table 2.1: List of WPs in the CROPS project.

WP	Title
1	System Engineering and Architecture
2	Sensing
3	Manipulators and End-Effectors
4	Intelligent Sensor Fusion and Learning Algorithms
5	Sweet Pepper – Protected Cultivation
6	Harvesting Systems in Orchards: Grapes and Apples
7	Precision Spraying
8	Forestry
9	Training
10	Dissemination
11	Final Demonstration
12	Economics, Social Aspects, Sustainability, and Exploitation
13	Coordination

from the WPs two to four. The other WPs (nine, ten and twelve) addressed important aspects like public relations, training of the participants, and economics. An overview of the project organization and interaction of the WPs is given in Figure 2.1.

**Figure 2.1:** Interaction of the WPs within the CROPS project.

The AM was responsible for the WPs one and three with its main contribution in the design and development of three manipulator prototypes and its motion planning algorithms. Additionally, extensive support during the integration process and the field experiments was provided. In the course of WP one and by regularly monitoring the integration progress, recommendations for future actions towards the development of successful demonstrators were proposed.

2.3 Applications

The manipulator prototypes were designed for the automation of agricultural tasks like selective harvesting and spraying. The robot system is intended for harvesting of sweet pepper, apples and grapes as well as precision spraying of grapevine. For an appropriate system design, commonly used cultivation systems and environmental conditions for the different tasks are of interest. Therefore, a brief description of the different applications with special focus on the growing systems will be given in this Section. The selected system has direct influence on the yield, it can avoid the spreading of diseases, or it can be more or less suitable for the mechanization of agricultural tasks. However, depending on the environmental conditions and personal experience of farmers, there exist many different variations in the production of agricultural products. The upcoming Sections will provide an overview on the greenhouse and orchard cultivation which were used during the CROPS project. If possible, modifications of the cultivation system to increase the suitability for automation are made. However, innovations in the field of agriculture must be considered carefully and the effects on the production must be evaluated over many years. From the tasks considered in this thesis, the automation of sweet pepper harvesting has the highest requirements on the manipulator design and is therefore discussed in particular.

2.3.1 Selective Harvesting

Sweet Pepper Cultivation in Greenhouses

Nowadays, in Europe sweet pepper plants are usually cultivated in greenhouses. Plants with fruits in the colors red, yellow, green and orange are commonly used for commercial production. Before the fruits get their final color, they are always green. The color is an indicator for the fruit ripeness. The greenhouse production is based on a full year cycle and the plants are cultivated on substrate. It is common to bring the crop in the greenhouse around February and they continuously carry fruits starting from March till the end of the season around September. When the season is over the remaining greenery is removed. The plants grow from about 300 mm up to 4000 mm relative to the greenhouse floor. However, ripe fruits usually are located at the upper part, about 500 mm near the top, of the plant. In the CROPS project, harvesting experiments were carried out in the Netherlands. Therefore, only the typical cultivation system in those greenhouses will be presented here and a picture with sweet pepper plants is shown in Figure 2.2. Early in the season, either two or three stems of a single plant are connected to a wire system. These wires run vertically and support the stems during plant growth. The plants are cultivated in rows with a certain distance between two rows. During the season the canopy of the plant grows and thus the available space in between the rows is decreasing. In the corridors between the plant rows are two pipes for heating the greenhouse. These pipes are used as a rail system and a cart, equipped with an elevating mechanism and an actuator to move forward in the row, can be placed there. It is state of the art to use this cart to transport human workers in the corridors for Plant Maintenance Operations (PMOs). Two schematic side views to



Figure 2.2: Sweet pepper cultivation in a greenhouse.

illustrate commonly used cultivation systems are shown in Figure 2.3. No leaves or fruits are drawn. In Figure 2.3a the side view along the rows is shown. The two stems cultivation system is illustrated on the left hand side while the three stem cultivation system is illustrated on the right hand side. In between the plants is the corridor with the heating pipes. Figure 2.3b shows the cultivation system perpendicular to the plant rows. Again on the left hand side the system with two stems is drawn. In this case, only one stem is facing the corridor and therefore visible. On the right hand side, the three stem cultivation system is illustrated. Compared to the system with two stems, the average distance between stems is less and therefore the canopy is more dense. The wires are indicated in gray. The vertical wires are knotted to a top wire which is horizontal and in parallel to the row. In a dutch greenhouse, the positions of 60 stems and 165 fruits of one plant

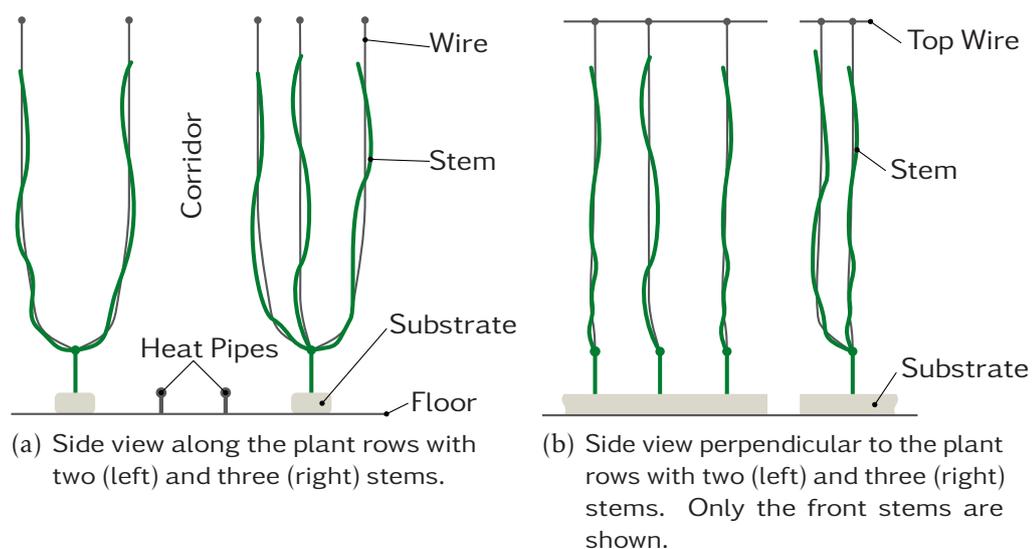


Figure 2.3: Side view of sweet bell pepper cultivation systems.

row were measured at the beginning of the season¹. Although growing systems vary from farmer to farmer and dimensions change during the season, the data gives an idea on the dimensions and its variations in sweet pepper cultivation. Figure 2.4 shows a scheme plot of the cultivation system according to the two stems cultivation system in the top view. The plant row on the left hand side

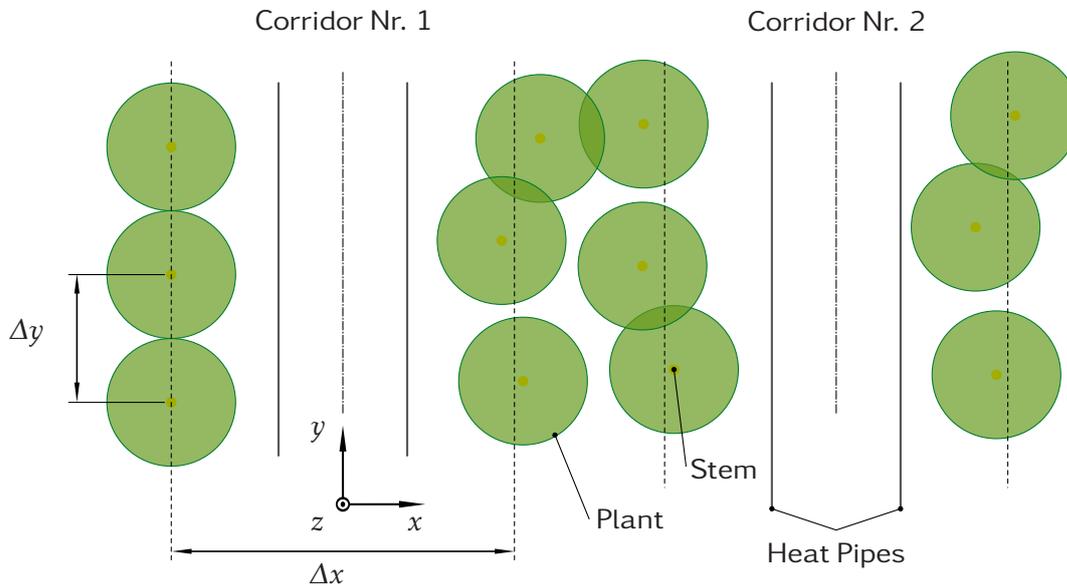


Figure 2.4: Top view of sweet pepper cultivation system.

of the left corridor shows the nominal distances. The plant area around a stem, including leaves, branches and fruits is approximated by a green circle. Furthermore, basic dimensions are indicated in the Figure and its nominal values as well as the standard deviations are given in Table 2.2. Remarkably, the plant spacing

Table 2.2: Basic dimensions and its standard deviations of sweet pepper cultivation in a dutch greenhouse.

Parameter	Nom. Val. [mm]	Std. Dev. [mm]	Min. [mm]	Max. [mm]
Row-to-row Δx	862	50	750	1020
Stem-to-stem Δy	192	60	40	350
Fruit distribution in z	–	220	0	1650
Stem-to-fruit	38	6.9	15	96

with a nominal distance of 192 mm is very narrow and there are strong variations in the distance of neighboring plants, which is reflected in the considerably high standard deviation. Also the big differences between the minimum and maximum stem-to-stem distance indicate a very irregular and unstructured environment. Besides the stem arrangement, the radial distance and angle of fruit center positions with respect to the corresponding stem center positions were measured

¹ The measurements were performed by WOUTER BAC from the research institute WAGENINGEN UR, Netherlands.

to get information on the fruit distribution relative to the plant stems. All the measured fruit positions are plotted in Figure 2.5 with respect to the stem center in one graph.

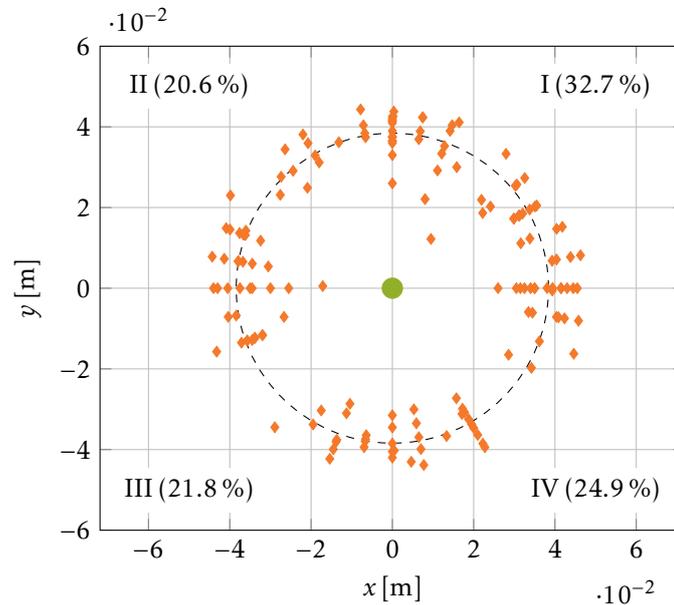


Figure 2.5: Fruit distribution with respect to the stem (green disc). The fruit positions, illustrated by diamonds, were measured on a plant row and the amount of fruits in each quadrant is provided. The dashed circle indicates the average stem-to-fruit distance.

Apple Cultivation in Orchards

According to a statistical report on agriculture of the EU (Coyette and Schenk [2013]), the 28 member states (as of July 1, 2013) produced 11.7 million tonnes of apples in 2012. Apples are produced in nearly all member states and require regions with temperate climate conditions. The apple trees are usually grown for several years while carrying fruits each year. Ripe fruits are selectively harvested around autumn. A very high amount of different varieties exist but will not be addressed in this thesis. However, for the manipulator design and in particular for the design of the kinematics, the planting system in orchards is of relevance. Several different cultivation methods are reported in the literature. Refer to Ferree and Warrington [2003] for a detailed discussion and a broad overview on the botany and production of apples. Certainly not all systems are suitable for automation. In the CROPS project the harvesting experiments were carried out in Belgium and a picture of an orchard is provided in Figure 2.6. Furthermore, it has been decided to focus on so-called *flat planar canopy systems*. This kind of growing systems, developed in the 1950s by Italian growers, were called palmette training systems. They restrict the canopy in a two dimensional plane. This cultivation systems allow for the application of carrier platforms providing the basis for very efficient PMOs, like thinning, pruning or fruit picking. The trees are planted



Figure 2.6: Commercial apple cultivation in a Belgian orchard.

in an espalier system with a spacing of 3.5 to 4 m in the row and the distance between two rows is about 4-5 m. In the original palmette system the branches of a central tree trunk are trained in the plane either in a horizontal way or with an inclination of 30 to 45°. The branches which are growing out of the plane are removed or connected to a trellises system with four to six wires. In Figure 2.7 the horizontal-palmette growing system is illustrated for a single tree in the side view.

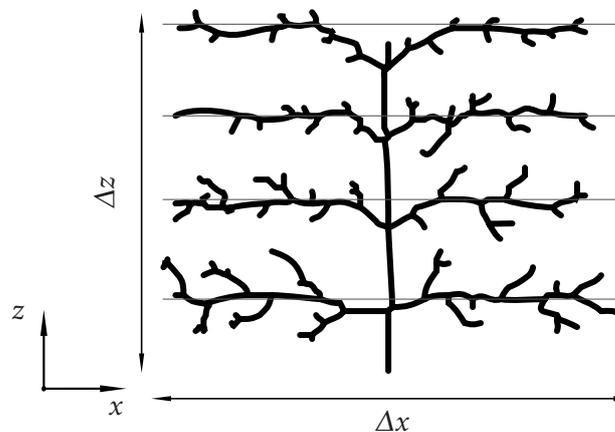


Figure 2.7: The horizontal-palmette growing system with the branches attached to the trellis system as reported by Ferree and Warrington [2003] with $\Delta x \approx \Delta z \approx 2.5$ m.

Grapevine Cultivation in Orchards

In 2012, the EU produced 22.8 million tonnes of grapes, which accounts for about two thirds of the world wide production. According to Coyette and Schenk [2013], the largest part of the grapes (91 %) were used for the production of wine. Similar to the other harvesting applications, a great amount of different cultivation

systems are commonly used. A detailed discussion on grape growing is beyond the scope of this thesis. Please refer to Creasy and Creasy [2009] for further information. In Figure 2.8 the growing system which was considered to be the most suitable for automated harvesting within the CROPS project is shown. The



Figure 2.8: Cultivation of grapevine.

vine is grown in an espalier system with an average tree spacing of about 1.1 m and a row-to-row distance of about 2.1 m. Grape bunches are hanging down and are easily accessible from the side. By removing the leaves of the vine (cf. Figure 2.8b) for the field experiments, the visibility of the grapes as well as the accessibility is improved a lot.

2.3.2 Selective Spraying of Grapes

Besides the selective harvesting task, the modular agricultural manipulator was also intended to be used for the application of precision spraying of grapes. The basic idea of selective spraying is to reduce the amount of pesticides used in the field. This can be achieved by autonomously applying pesticides only on infected areas instead of spraying the whole canopy. Therefore, the manipulator must position the air-carried spray of an end-effector in front of the canopy at a distance of about 0.4-0.6 m. However, the requirements on the position accuracy for this application in the range of several centimeters is rather low for up-to-date commercially available industrial manipulators. To spray areas at inner layers of the canopy positioning in different directions might be advantageous.

3 Manipulator Design

3.1 Introduction

This Chapter discusses the major steps leading to the final mechatronic design of the agricultural manipulator prototypes which have been realized in the course of this thesis. The overall design process with respect to the technical implementation of the robot is illustrated in Figure 3.1. Based on the systems requirements, the

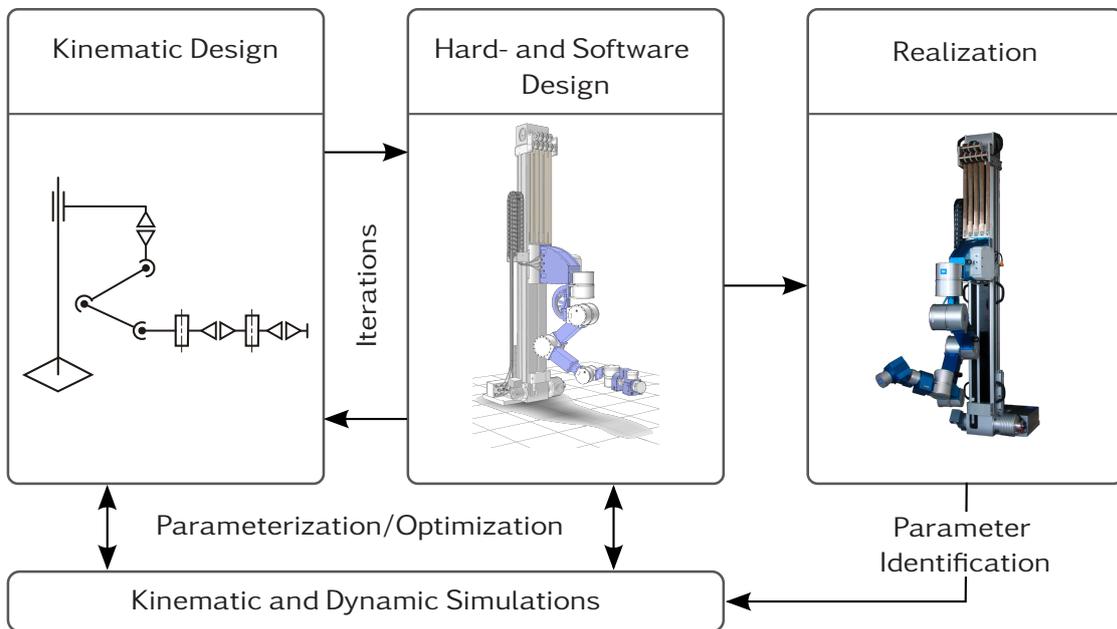


Figure 3.1: The iterative design process of the agricultural manipulator.

design is an iterative process, starting with several kinematic designs and their evaluation. Then based on the initial evaluation, one or more kinematic models are selected and the dimensions of the mechanical parts can be estimated. This initial dimensioning provides the parameter for the dynamic simulations and from the results, a more detailed analysis of the mechanical structure can be performed and suitable actuators are selected. With more and more detailed simulations, the systems behavior and its limitations can be estimated. According to the constraints, like limited available design space or actuation power, it might be necessary to reconsider and adapt the initial kinematic designs. When the mechanical design becomes more definite, the estimated dimensions, like inertia parameter, become more and more accurate, hence improving the predicted results of the dynamic model step by step and the mechanical parts can be further optimized. At a certain point of the process, the most suitable design must be selected and finalized while the manufacturing as well as the assembly begins. During the start-up of system

components and with the final system itself, parameter identification experiments can be performed and the models are further improved and validated.

This Chapter will begin with the introduction on the kinematics design, followed by a detailed study on PAMs as actuator alternatives for the manipulator prototypes in Section 3.3. The equations and assumptions to model the kinematics and dynamics of the robot, will be introduced in Section 3.4. Another important aspect of nearly any mechatronic design is the selection of an appropriate software and electronics architecture. Therefore, a brief introduction of the electronics architecture will be given in Section 3.5 and a detailed description of the software framework, developed for the manipulator will be addressed in Section 3.6.

Many contents of the work presented in the following have already been published in scientific papers by the author of this thesis and by the co-workers involved in this project. Therefore, clear references will be given in the corresponding Sections.

3.2 Kinematic Design

During the development of a manipulator, one of the first things which has to be considered is the kinematic structure of the system. Influencing variables on the kinematic design are the selected approach to fulfill a certain task, the required workspace as well as the end-effector tool that will be applied. Usually, by an accurate description of the task, it is possible to design very specialized and optimized kinematics. However, during the design of the here described agricultural prototypes, two major issues had to be faced. On the one hand, the manipulator had to deal with agricultural products. Compared to applications in automation industry with well defined tasks, it is not possible for agricultural duties to provide a task description with similar accuracy. Furthermore, cultivation systems are in general not standardized and farming methods can differ from one farmer to another. On the other hand, the robot is intended to be used for different tasks which require, amongst others, different workspace dimensions (see Section 2.3) while different task-specific end-effector tools are used. To provide a solution to these demands, it has been decided to design a modular manipulator, meaning that it should be easily possible to reconfigure the kinematics of the prototypes for the needs of the different applications. Nevertheless, a decision has to be made on the basic structure of the kinematics and it would obviously be of great benefit, if one kinematic set-up could be utilized for many different applications. With this in mind, it was chosen to design a kinematic configuration that is as flexible as possible with a high amount of dexterity to be able to deal with the different demands.

Initially, in this Section, the kinematic designs of the first and final developed prototype generations are presented. Later on, the methods used during the design and for validation are introduced and briefly discussed.

A schematic illustration of the kinematics of the first prototype is given in Figure 3.2. Rotational joints are indicated by cylinders and prismatic joints by cuboids. A disk is illustrating the interface for the end-effectors. The orientation

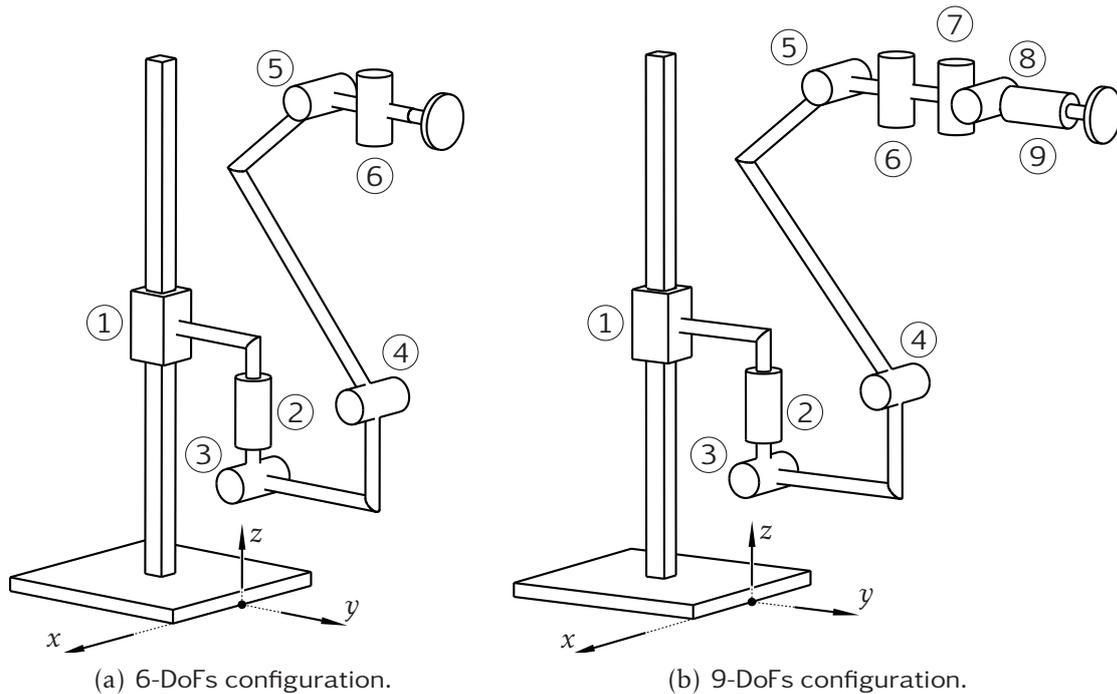


Figure 3.2: The kinematic design of the first manipulator prototype in the two supported configurations.

of the coordinate base frame is given in each Figure. A detailed explanation on the selected design as well as the discussion of alternatives was presented by Baur et al. [2012]. Both kinematic configurations are equipped with a prismatic joint. Therefore, the robot is able to operate at different heights required for all applications. The 6-DoFs configuration (cf. Figure 3.2a) has been designed and successfully applied for the selective spraying application of grape vine (see Section 6.2). For this application, the task space definition does not require a fully defined pose in three dimensional space. In Figure 3.2b, the 9-DoFs configuration is shown. This set-up was realized and applied for the harvesting applications and the results are presented in Section 6.1. Remarkably, the manipulator for the harvesting tasks combines the ability to cover a large workspace with a high dexterity but it also has the property to obtain compact configurations, suitable for the working area with the limited space in the narrow rows. Similar to industrial manipulators, three rotational joints with consecutively perpendicular axes are added after the last joint of the 6-DoFs configuration. This three joints are commonly called wrist in robotics (due to the kinematic similarity to the human wrist). The manipulator in the 9-DoFs configuration is capable to obtain any three dimensional pose in the dexterous workspace. The dimensions of the manipulator are provided in Table 3.1. The Table describes the relative poses of one body fixed frame after another. The first joint is referenced with respect to the base frame and the z -axis of each body fixed frame is set along the corresponding joint axis. The relative positions are provided in the coordinates of the previously defined frame with the x , y , and z positions as well as the α , β , and γ rotations, i.e. fixed angle rotations around the x - y - z axes. The parameter q_i indicates the DoF of the

Table 3.1: Kinematics parameter of the first manipulator prototype.

Joint	x_i [m]	y_i [m]	z_i [m]	α_i [rad]	β_i [rad]	γ_i [rad]
1	0	0	$0.75+q_1$	0	0	0
2	0	0.1	-0.1	0	0	$\pi/2 + q_2$
3	0	0	0	$\pi/2$	0	q_3
4	0.23	0.15	0	0	0	q_4
5	0.4	0.2	0	0	0	q_5
6	0.1	0	0	$-\pi/2$	0	q_6
7	0.2	0	0	0	0	q_7
8	0.045	0	0	$\pi/2$	0	q_8
9	0.05	0	0	0	$\pi/2$	q_9

joint i .

In a preliminary study by De Hoog [2013] the manipulator design of the first prototype has been evaluated in simulation and experiments and the general suitability for the sweet pepper harvesting application, which is considered the most challenging one, has been confirmed. Experiments showed that the robot is able to reach 91.8 % of the colored fruits, using a rather basic motion planning algorithm. Therefore, only minor changes with respect to the kinematics were made during the design of the final manipulator prototype. An overview of the kinematic design of the final manipulator is illustrated in Figure 3.3. The 9-DoFs configuration of the final prototype (cf. Figure 3.3b) is similar to the 9-DoFs configuration of the first prototype. The main difference is in the arrangement of the wrist joint axes. This change had been made out of mechanical design considerations. Due to the design with integrated robot drive modules of the final prototype (Pffaff

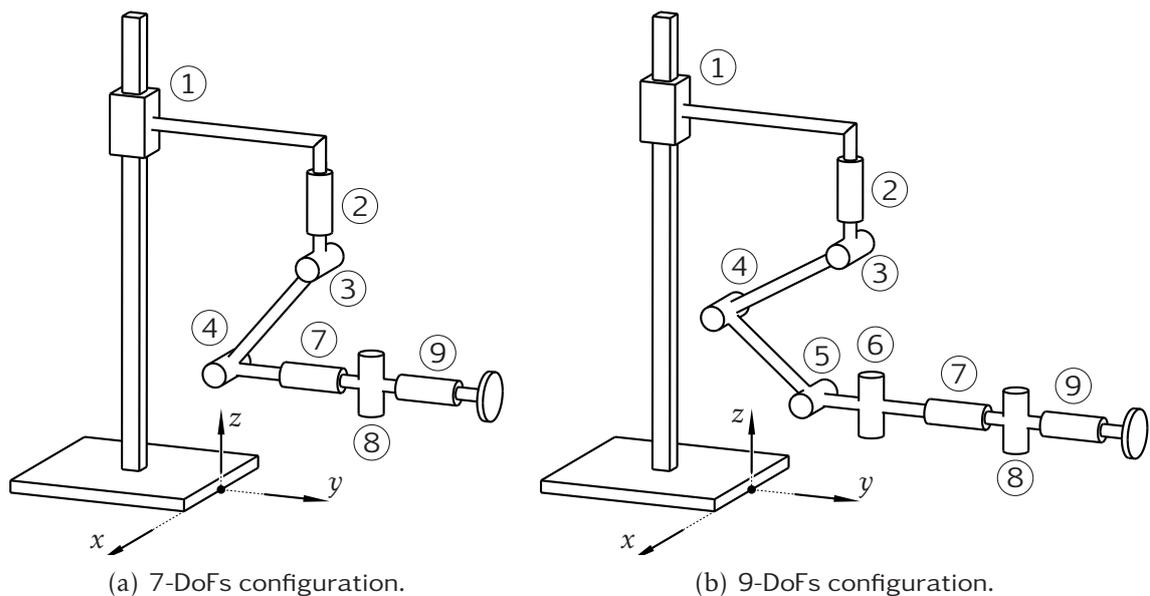


Figure 3.3: The kinematic design of the final manipulator prototype in the two supported configurations.

et al. [2014]), the modularity of this prototype is strongly enhanced. It is possible to remove or reconfigure the robot joints easily in a few minutes. Therefore, it has been decided to support a 7-DoFs configuration (see Figure 3.3a) following the standard industrial manipulator kinematics structure (Craig [1986]) with an additional prismatic joint. To obtain the 7-DoFs configuration from the 9-DoFs configuration, the joints five and six have to be removed. Both kinematics shown in Figure 3.3 are suitable for the selective harvesting applications and have been investigated for the sweet pepper application. The parameter of the 9-DoFs kinematics are summarized in Table 3.2 and the different parameter of the 7-DoFs configuration are highlighted in Table 3.3.

Table 3.2: Kinematics parameter of the final manipulator prototype in 9-DoFs configuration.

Joint	x_i [m]	y_i [m]	z_i [m]	α_i [rad]	β_i [rad]	γ_i [rad]
1	0.115	-0.017	$0.796+q_1$	0	0	0
2	-0.115	0.48	0.0	0	0	$\pi/2 + q_2$
3	0	0	-0.2	$\pi/2$	0	q_3
4	0.35	0	0	π	0	q_4
5	0.35	0	0	π	0	q_5
6	0.18	0	0	$-\pi/2$	0	q_6
7	0.15	0	0	0	$\pi/2$	q_7
8	0	0	0.1	0	$-\pi/2$	q_8
9	0.065	0	0	$\pi/2$	0	q_9

Table 3.3: Modification of the kinematics parameter of the final prototype for the 7-DoFs configuration.

Joint	x_i [m]	y_i [m]	z_i [m]	α_i [rad]	β_i [rad]	γ_i [rad]
4 \rightarrow 7	0.4	-0.0295	-0.043	$\pi/2$	$\pi/2$	q_7

As already pointed out at the beginning of this Chapter, an accurate task definition for all applications is not available and therefore, the definition of a well defined optimization problem to derive a kinematic scheme is impossible. The presented designs have been derived on the basis of engineering experience, mechanical design constraints and kinematic simulations. In the next Sections, the tools and methods for evaluation of the designs are presented.

3.2.1 The Manipulability Measure

Because of the vague task description one has to face for the here considered agricultural applications, the analysis with the so-called manipulability measure seems to be an adequate method and has been performed exemplarily with the first

manipulator prototype to validate the proposed design. First of all, based on the available knowledge, a task description will be given and the approach to investigate the manipulability of the manipulator is explained. From the investigated applications, the selective harvesting of sweet pepper has been identified as the application with the highest requirements on the manipulator kinematic design. Figure 3.4 shows an illustration of the top view of a sweet pepper cultivation system, similar to Figure 2.4. In addition, an end-effector is represented schematically.

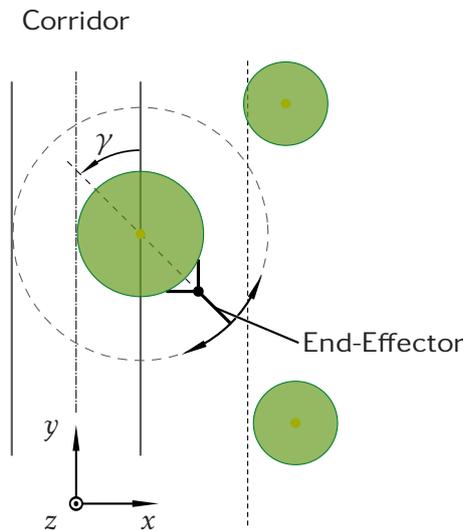


Figure 3.4: Abstraction of the task for sweet pepper harvesting (Baur et al. [2012]).

A path is illustrated by a dashed circle and the end-effector location on the path shall be described by a rotation around the plant middle point (stem center) with the angle γ between the vertical line and the extension of the end-effector's middle axis. The three plants are indicated in green and at the middle one a harvesting operation should be performed while the location of the robot base can be depicted by the base frame. With the fruit distribution, reported in Figure 2.5, it is clear that the end-effector must reach any position around the plant to harvest all fruits. Furthermore, in all likelihood, the fruit picking might be most successful if the plant is approached from a radial direction, so that the middle axis of the end-effector is in a line with the fruit center and the stem center. Several end-effector positions around the plant were investigated and a measure that indicates the ability of the robot to move along certain directions in a given joint configuration is provided by the manipulability measure, introduced by Yoshikawa [1985]. Please refer to Section 5.4.3 of this thesis for a definition of the measure, based on the Jacobian matrix of the Tool-Center-Point (TCP), as well as further explanations and detailed information on the inverse kinematics algorithm applied here. In Figure 3.5 the manipulability ellipsoids, projected into the (x,y) -plane are drawn for various positions of the TCP distributed on a half circle around the plant, while the end-effector is always pointing towards the stem center according to Figure 3.4. In this simulation, the stem center is located at $(x = 0.225\text{ m}, y = 0.7\text{ m})$ in the plane and the height of the TCP is at $z = 0.9\text{ m}$ with respect to the robot base frame. Due to the kinematic redundancy of the manipulator (cf. Section 5.4.1),

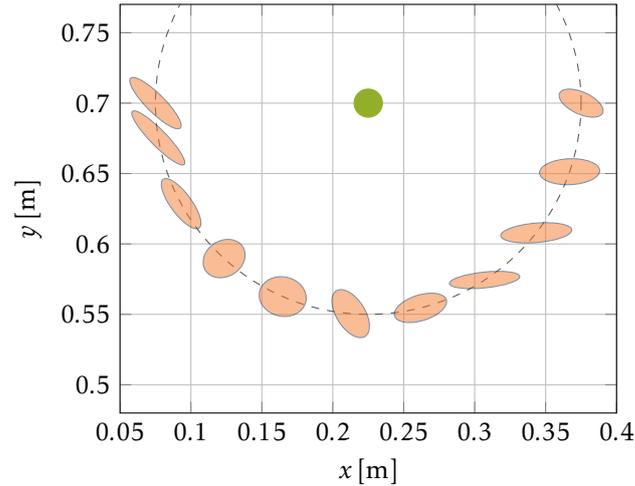


Figure 3.5: Manipulability ellipsoids for several TCP positions distributed in a half circle (dashed line) around a sweet pepper stem (green circle) according to Figure 3.4 (Baur et al. [2012]). For clarity the ellipsoids have been scaled to an appropriate size.

an end-effector pose can in general be obtained by an infinite amount of joint configurations. The joint configuration can be more or less advantageous with respect to a certain task. The selection of the optimized configuration for the applied simulation will be discussed in more detail in Section 5.4.3. According to the manipulability measure, evaluated for this height and the given stem position, the kinematic design is appropriate. A wider range of discrete positions in the manipulator workspace has been investigated and is presented in the following. The manipulability measure according to (5.59) for end-effector heights in the range $0\text{ m} \leq z \leq 1.5\text{ m}$ and orientations from $-\pi/2 \leq \gamma \leq \pi/2$, for the stem position previously defined, is plotted in Figure 3.6. The manipulability plot has been normalized with the minimum value of all investigated points. The numerical value „1“ corresponds to the highest manipulability found and the manipulability decreases with smaller numbers. Remarkably, the manipulability is lowest at a height of about $z = 0.8\text{ m}$ and an angle of $\gamma = \pi/2$. The operation of the manipulator at these positions should be avoided, if possible. Remarkably, only one half of the plant can be approached with the constraint on the end-effector orientation according to Figure 3.4. For the other half, the robot has to be turned around in the corridor. Further discussion is provided in Baur et al. [2012].

3.2.2 Analysis of Manipulator Workspace

One of the basic requirements on the manipulator kinematic design is to cover the workspace described in the specifications. In general, the workspace of a manipulator can be described as the volume, the robot can reach with at least one orientation. In robotic literature, for example in Siciliano et al. [2009], this workspace definition is commonly referred to as the *reachable* workspace and can be computed with the direct kinematics equations or graphically constructed, while taking into account the mechanical joint limits and the geometry of the robot. Because the direct kinematics function is continuous, the volume can be

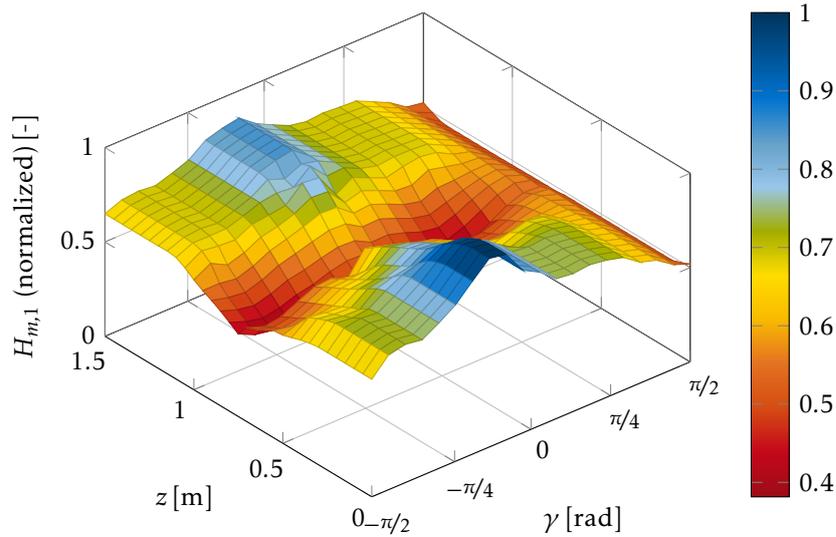


Figure 3.6: The normalized manipulability measure of the manipulator for several positions of the TCP with local optimization of the redundant DoFs around a sweet pepper stem (Baur et al. [2012]).

defined by an enclosing surface. Notably, an operation of the manipulator at the workspace limit should in general be avoided, because the freedom to move the end-effector in any direction degenerates. The points in the robot workspace that can be reached in different orientations belong to the so-called *dexterous* workspace. If for a certain task the set of required TCP positions is known, a more detailed analysis can be performed.

For the harvesting applications several not clearly defined positions must be reached in various poses. This makes a workspace analysis, based on the manipulability measure more suitable, as described in the previous Section. However, for the selective spraying application, the desired positions and the orientations are very well defined. With this information, the designer can very accurately provide recommendations concerning the capabilities and limitations of the system to the user. Exemplary, a workspace analysis of the first manipulator prototype in the 6-DoFs configuration for the spraying application is performed. A brief description of the workspace requirements is given in Section 2.3.2 and a more detailed illustration including the schematics of the precision spraying end-effector is shown in Figure 3.7. The spraying end-effector must be positioned in a distance of 0.4 m-0.6 m from the canopy at a height of about 1 m to 2 m. With the prismatic joint of the manipulator in an operating range close to 1 m, the height positioning is not an issue. Additionally, to improve the spray coverage, it is desirable to apply the pesticides from different angles γ (see Figure 3.7). This leads to the following appropriate description of the operational space w for the spraying task (refer to Section 5.3.1 for more details on the task space definition):

$$w := (x, y, z, \alpha, \gamma)^T \quad (3.1)$$

Given a set of desired poses in the task space coordinates, it can be checked with

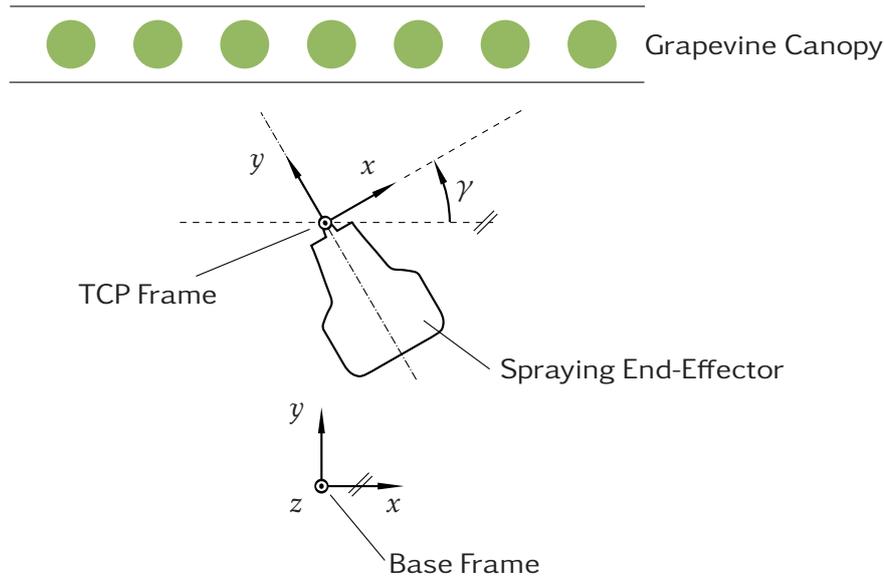


Figure 3.7: Schematic illustration of the precision spraying task.

an appropriate inverse kinematics algorithm, if one joint configuration \mathbf{q} exists that provides a solution to the inverse kinematics (see Section 5.4 for further information on the inverse kinematics problem). For a desired pose \mathbf{w}_d and with the direct kinematics function $\mathbf{f}(\mathbf{q})$ a nonlinear constrained optimization problem can be formulated:

$$\|\Delta\mathbf{w}\| := \|\mathbf{w}_d - \mathbf{f}(\mathbf{q})\| \rightarrow \min!_{\mathbf{q} \in \mathcal{C}} \quad (3.2)$$

If the derivation of the desired pose to the pose computed by the direct kinematics function $\mathbf{f}(\mathbf{q})$ is below a certain threshold ϵ , i.e. $\|\Delta\mathbf{w}\| < \epsilon$, the desired pose can be marked as reachable. The optimization problem (3.2) can be solved numerically, for example with the Matlab function `fmincon`. To visualize feasible end-effector poses a grid of discrete points can be defined with the task space vector (3.1). Exemplary, the following grid has been applied:

$$-0.78 \text{ m} \leq x \leq 0.78 \text{ m} \quad \text{with} \quad \Delta x = 0.01 \text{ m} \quad (3.3a)$$

$$0.3 \text{ m} \leq y \leq 1.2 \text{ m} \quad \text{with} \quad \Delta y = 0.01 \text{ m} \quad (3.3b)$$

$$z = 1.2 \text{ m} \quad (3.3c)$$

$$\alpha = 0^\circ \quad (3.3d)$$

$$\gamma = \pm 40^\circ \quad (3.3e)$$

and the resulting areas of the reachable end-effector poses are illustrated in Figure 3.8. With the visualization of the reachable workspace for desired poses, a verification of the requirements can be performed and the optimal positioning of the robot with respect to the canopy can be derived. An analysis of the reachable workspace for $\alpha = \gamma = 0^\circ$ revealed that the manipulator is able to cover a range from 0 m-1.8 m in the z -direction, from 0.3 m-1.2 m in the y -direction and from -0.6 m to 0.6 m in the x -direction. Therefore, the manipulator in the proposed 6-

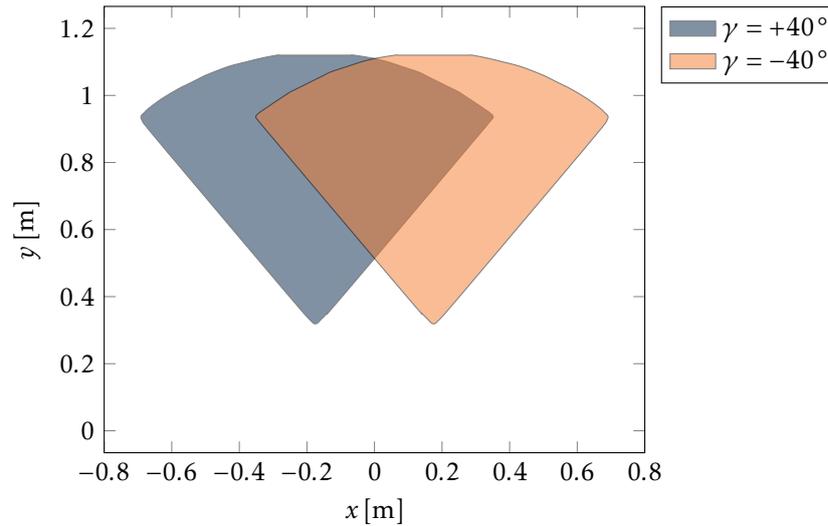


Figure 3.8: Areas of reachable end-effector poses in the robot workspace for $\gamma = \pm 40^\circ$ according to the grid specified in (3.3).

DoFs configuration can cover the required workspace for the spraying application. The experimental results are reported in Section 6.2.

3.3 Selection of the Actuation Principle

This Section deals with the choice of appropriate actuators for the agricultural manipulator. Because actuators are the influencing components in any automation process, a thoroughly analysis of different possibilities is worthwhile. According to Isermann [2005], actuators can be divided in the following categories, based on the power source they use:

- Electromechanical actuators
- Fluid actuators
- Other, unconventional actuators

In each category itself, many different kinds exist, respectively. From the type of the conventional principles, electromechanical and pneumatic actuators have been identified as the most promising concepts for the manipulator prototypes. Hydraulic actuation has not been considered, because it is more suitable for systems which are in the need of high actuation forces. The investigated agricultural applications require moderate forces. In Section 3.3.1, PAMs are investigated in detail as actuator candidates. Additionally, the design of a testbed and several control strategies are presented. Most of the work on the investigation of the PAM has already been published in Baur et al. [2014c] and was supported by CHRISTOPH SCHÜTZ in the course of his student thesis. At the Section's end, the advantages and disadvantages of electrical drives are very briefly discussed and the final actuation concept for the agricultural manipulators is presented.

3.3.1 Pneumatic Artificial Muscles

In the article of Davis et al. [2003] a very good description on the history of PAMs can be found and a brief overview of the development of the actuator will be given in the following. Pierce [1936] patented the first structure with similar behavior to nowadays used PAMs, but he only suggested to make use of the radial expansion to break coal. The first patent as pulling actor has been made by De Haven [1949]. Afterwards it has been more and more studied (Gaylord [1958]) and Joseph McKibben suggested its application for prosthetics (Tondu and Lopez [2000]). The similarity of the PAM characteristics to biological muscles made the actuator very interesting for the application in bionics. In the 1980s, the company Bridgestone started the commercialization of the PAM, which they called *Rubbertuators*. But despite their efforts, the product was taken off the market in the 1990s. Later on, the German company Festo developed a PAM pulling actuator for commercial use, which is still available today.

The well-known advantages of pneumatic actuators are (Isermann [2005]):

- Good power-to-weight ratio
- High reliability
- Wide operating range at different temperatures

with the following major drawbacks:

- Compressed air is required
- Control to achieve accurate positioning can be challenging
- Big dimensions

Further advantages of PAM are the high robustness, because the actuator does not possess any moving parts itself and they can be applied in unclean areas. Additionally, the passive elasticity (Chou and Hannaford [1996]) with its inherent compliance might be a favorable property for the design of robot joints due to the prospect of increased safety in case of impacts. As any linear actuator the PAM has the inherent disadvantage that rotational motion is not directly available, but must be obtained, for example, by the combination of two actuators in an antagonistic set-up (cf. Figure 3.11).

Applications in Robotics

This Section briefly presents some applications of the PAM in the field of robotics. The first manipulators driven by *Rubbertuators* have been developed by the Japanese tire manufacturer Bridgestone and were called *RASC* and *SOFT ARM*. According to Inoue [1988] the design is explosion proof and he suggested applications like painting or coating. Two models of the *SOFT ARM* were manufactured, one with four DoFs and another with five DoFs and the robots were able to deal with a payload of 1 kg and 3 kg, respectively. Not many systems have been developed in the meantime until Tondu et al. [2005] presented an arm with seven

revolute joints, driven by McKibben artificial muscle pairs. Many details on the modeling and mechanical design of revolute joints can be found in the article. The prototype has been used for a telerobotics application. More recently the automation company FESTO has presented a humanoid torso with two arms and hands, actuated by PAMs (Boblan and Schulz [2010]). Due to the similar characteristics of PAMs to biological muscles, the application of the actuators for several humanoid robot projects has been reported in the literature (Mizuuchi et al. [2012]; Niiyama et al. [2010]; Vanderborght et al. [2005]).

Structure and Functioning

The PAM has a cylindrical shape and is a combination of a flexible rubber tube which is encircled by a braided shell (Chou and Hannaford [1996]). By applying pressure to the internal tube (e.g. via compressed air) it tends to expand its volume. Due to the structure of the braided shell the PAM only expands radially while it is shortened at the same time (cf. Figure 3.9a). In case it is connected to a load it will apply a pulling force in longitudinal direction. According to Daerden and Lefebvre [2002], the following basic properties of a PAM are:

- Upon contracting, the volume of the PAM is increasing.
- The PAM contracts if the pressure increases (assuming a constant load).
- The contraction has an upper limit. At the limit the volume is maximal and no pulling force is available.

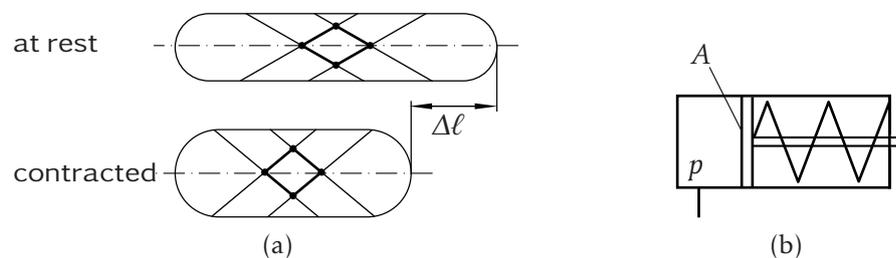


Figure 3.9: Schematic construction of a PAM: In (a) the muscle with the braided shell is at rest (nominal length). At the inflated state it is shortened by $\Delta\ell$; In (b) a single acting piston (area of the piston is depicted by A) with a spring is shown. The PAM has a similar dynamic behavior.

Figure 3.10 shows several isobars to characterize the physical behavior of PAMs at different lengths and pressure levels. The fluid muscle force F is plotted against the contraction $\kappa = \Delta\ell/\ell_0$, with the nominal muscle length ℓ_0 and the change in length $\Delta\ell$. For $\kappa > 0$ the muscle is contracted. Exemplary, the data provided by the automation company FESTO for the commercially available muscle DMSP-20-200N¹ is used. According to Figure 3.10 the force is depending on the contraction and the internal pressure. For an increasing contraction, the available pulling force decreases.

¹ type-diameter-nominal length

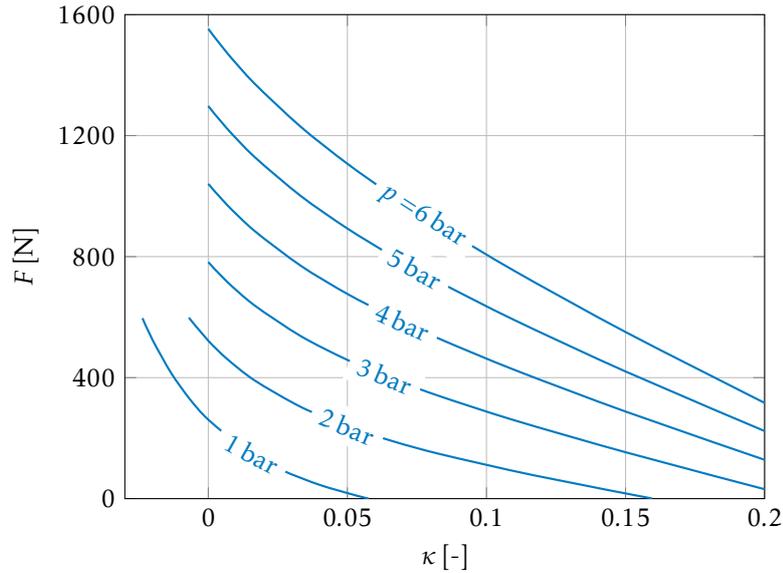


Figure 3.10: Force-to-contraction characteristic map of the PAM DMSP-20-200N.

Test Bed with an Antagonistic Muscle Set-up

The test bed was designed to evaluate the suitability of PAM actuators for robot joints as the drive mechanism of the developed agricultural manipulators. Since the muscles can only provide pulling forces, a revolute joint was realized with two PAMs in an antagonistic setup (see Figure 3.11). At the left hand side, the muscles

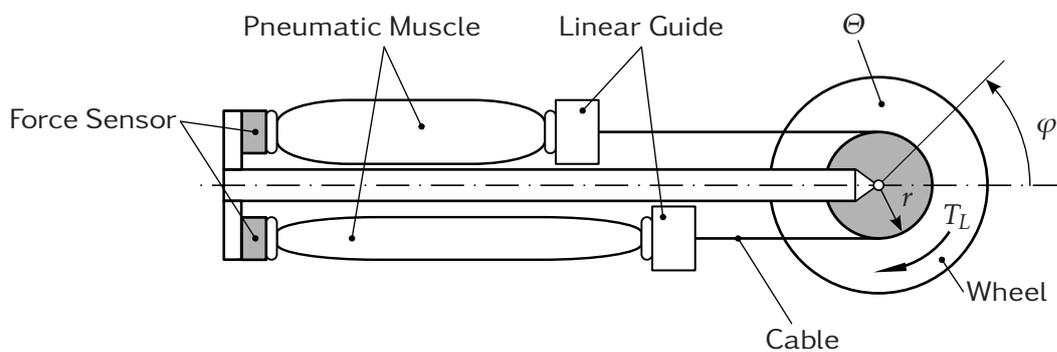


Figure 3.11: Scheme of the test bed with two PAMs in an antagonistic set-up.

are fixated on a rigid base via a single-axle force sensor. These sensors were used only during the identification of the muscle characteristics. On the right hand side, the muscles are connected to linear slides which are linked to the rotational joint by cables. The pressure level in the PAMs is regulated by two proportional valve pressure regulators. The data processing and the real-time control is handled with a controller board from the company dSPACE. An overview of the most relevant specifications of major system components is summarized in Table 3.4.

Table 3.4: Major specifications of the system components.

Component	Manufacturer (classification)	Specification
Fluid Muscle	FESTO (DMSP-20-300N-RM-CM)	$F_{max} = 1.5 \text{ kN @ 6 bar}$
Pressure Regulator	FESTO (VPPM-6L-L-1-G18-0L6H-V1P-S1)	Nominal Air Flow: 15 l/s
Incremental Encoder	Heidenhain (ROD426)	3600 ticks per revolution
Force Sensor	HBM (HBM-U9B)	Meas. Range: $\pm 2 \text{ kN}$
Controller Board	dSPACE (ds1103)	–

Modeling and Identification

An accurate modeling of the system components, especially due to the nonlinear characteristics of the PAM, is very important for the design of an efficient position control strategy. In the model, the components considered for the description of the pneumatic behavior are the PAM and the proportional valve. This Section explains the muscle model first. The pulling force of a fluid muscle can be modeled as the composition of a static force F_s , as a function of the relative pressure p and the contraction κ , with a friction term F_f :

$$F(p, \kappa, \dot{\kappa}) = F_s(p, \kappa) + F_f(\dot{\kappa}) \quad (3.4)$$

The friction F_f is modeled by a static and viscous friction term and is experimentally identified, including the overall friction of the test bed and its bearings. In the work of Chou and Hannaford [1996], an approximation of the force F_s is obtained by calculating the change of mechanical energy $dW_m = F dx$ and the energy of the fluid $dW_g = p dV$, with the change in the muscle length dx and inner volume dV , respectively. The conservation of energy $dW_g \stackrel{!}{=} dW_m$ yields to:

$$F = p \frac{dV}{dx} \quad (3.5)$$

According to Chou and Hannaford [1996], the change in muscle volume only depends on the muscle length. Because a description of the muscle volume is not easily derived, Hildebrandt et al. [2002, 2005] suggested to model the PAM as a one way pneumatic piston cylinder with variable piston area $A(\kappa)$ and an elasticity of the membrane, represented by the force F_m (cf. Figure 3.9b):

$$F_s(\kappa, p) = p A(\kappa) - F_m(\kappa) \quad (3.6)$$

With the following approximations (Hildebrandt [2009])

$$A(\kappa) = \sum_{i=0}^2 a_i \kappa^i \quad (3.7a)$$

$$F_m(\kappa) = \sum_{i=0}^3 b_i \kappa^i + b_4 \kappa^{\frac{2}{3}} \quad (3.7b)$$

the static force of the PAM can be obtained by fitting (3.6) and (3.7) to measurements or catalog data. Figure 3.12 shows the result of the least-squares curve-fitting problem, with respect to the seven parameter a_i ($i = \{0,1,2\}$) and b_j ($j = \{0,1,2,3\}$) for N measured data points:

$$\sum_{n=1}^N (F_s(\kappa_n, p_n) - F_n)^2 \rightarrow \min \quad (3.8)$$

The optimization problem (3.8) was solved with the function `lsqcurvefit` from Matlab, using the force measurements F_n from the test bed force sensors. The average of the absolute fitting error (i.e. $F_s(\kappa_n, p_n) - F_n$) is less than 6 N and the maximum error is below 33 N. Out of several investigated models from the lit-

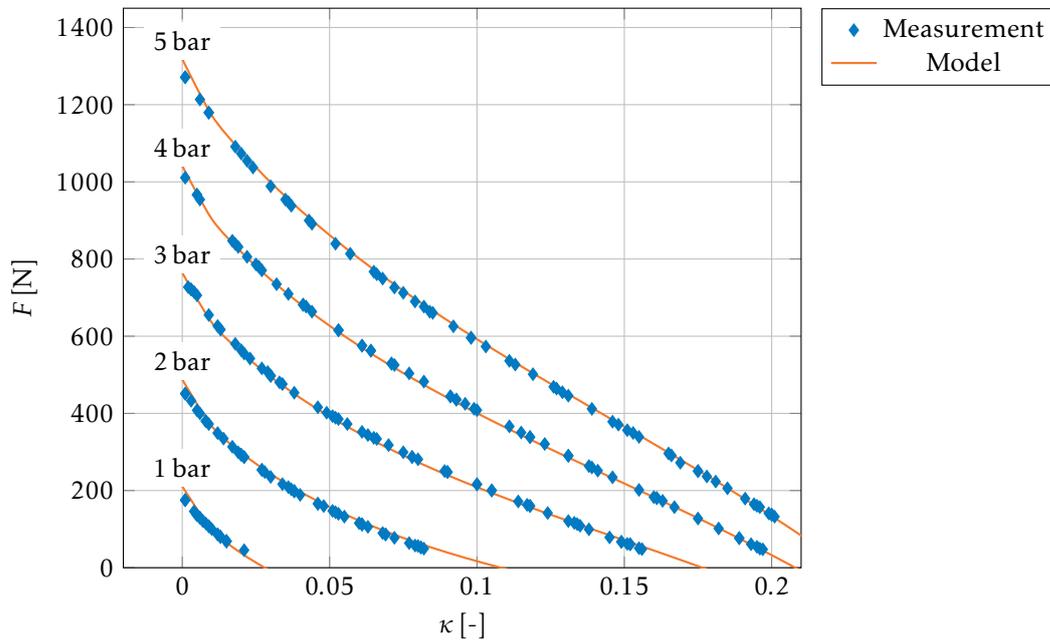


Figure 3.12: Measurement and curve fit with model (3.6) of isobars from 1–5 bar.

erature (Chou and Hannaford [1996]; Kerscher et al. [2006]; Tondu and Lopez [2000]), the model described above fitted the measurements best and will be used for further investigation. Due to the second term in (3.7b) with the cube root of κ , the model is only valid for $\kappa > 0$. However, this does not impose a restriction on the performed experimental study since the PAMs on the test bed will only be operated in a contracted state.

As mentioned earlier, the internal pressure of the muscle is controlled by the pressure regulator valves from FESTO. To experimentally identify the pressure characteristics, dynamic responses of this component were measured for several individual step responses in the range from 1 bar up to 5 bar. During the experiments, the muscle has been mechanically fixed at its nominal length, i.e. $\kappa = 0$. The measured pressure characteristics at a step time of 0.1 s, are shown in Figure 3.13. A first-order lag element in combination with a dead time and a rate limiter can be very accurately fitted to the measurements. The dead time is

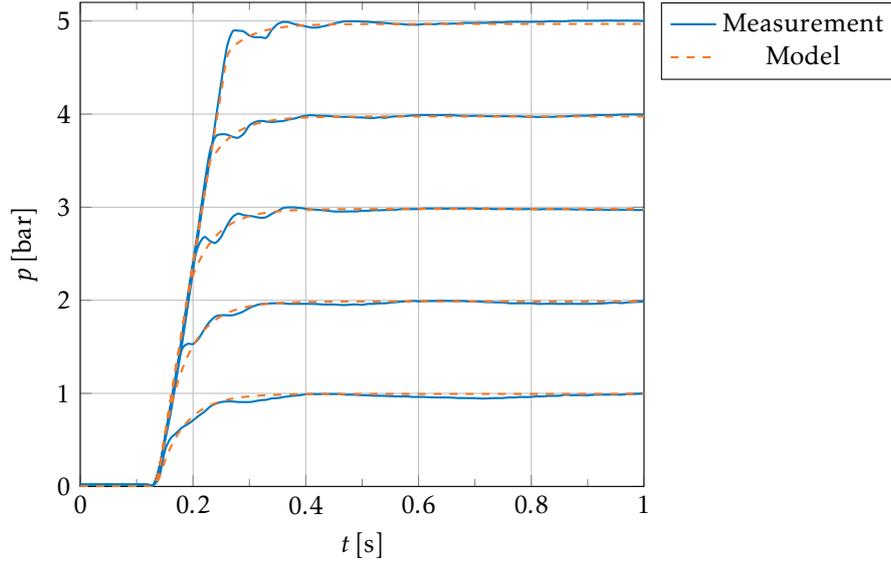


Figure 3.13: Pressure step commands for the pressure regulator at 0.1 s and the responses in simulation and measurement.

explained by the latency of the pneumatic tubes and the rate limiter represents the limited mass flow through the valve. However, further experiments showed that the dead time is decreasing for higher initial pressures and the limited mass flow is only significant for large steps ($\Delta p > 2$ bar). In the antagonistic setup, the PAMs are operated at an adjustable middle pressure p_0 , so the pressure steps are in general not higher than 2 bar. Therefore, given a desired pressure p_d , it is admissible to model the pressure characteristics of the pressure regulator as a linear first-order lag element with the time constant T and the static gain K :

$$T \dot{p} + p = K p_d \quad (3.9)$$

According to Figure 3.11 the principle of angular momentum can be applied on the wheel with the joint angle φ , the moment of inertia Θ , the load torque T_L , and the radius r at which the PAM is connected to the wheel. With (3.6) the joint dynamics are given by:

$$\Theta \ddot{\varphi} = (F_{s,1}(\kappa_1, p_1) - F_{s,2}(\kappa_2, p_2)) r - T_L - T_f(\dot{\varphi}) \quad (3.10a)$$

$$T_f = \text{sgn}(\dot{\varphi}) T_{f,0} + b \dot{\varphi} \quad (3.10b)$$

$$\kappa_{1,2} = \kappa_0 \pm \frac{\varphi r}{\ell_0} \quad (3.10c)$$

The friction term T_f is described according to (3.10b) by the static friction ($T_{f,0}$) and a viscous friction term ($b \dot{\varphi}$). The design parameter κ_0 defines the contraction of both muscles at zero joint displacement, i.e. $\varphi = 0$.

Controller Design

In this study, three different control strategies are analyzed and evaluated according to their performance. These are:

- A linear proportional-integral controller (PI)
- A PI-controller with feed forward (PI-FF)
- An input-output linearization (I/O)

For all control schemes the actuating variable u , chosen as the pressure difference Δp with respect to the mean (or initial) pressure p_0 is applied (Daerden [1999]; Inoue [1988]; Schröder et al. [2003])

$$u = \Delta p \quad (3.11a)$$

and thus the pressures in the two PAMs are given by

$$p_1 = p_0 + \Delta p \quad (3.11b)$$

$$p_2 = p_0 - \Delta p \quad (3.11c)$$

With the valve dynamics (3.9), the definition of the system input (3.11), and the state vector defined by

$$\mathbf{x} = (\varphi, \dot{\varphi}, p_1, p_2)^T \quad (3.12)$$

the nonlinear state space model, based on (3.10), is derived as follows:

$$\dot{\mathbf{x}} = \underbrace{\begin{pmatrix} \dot{\varphi} \\ 1/\Theta [(F_{s,1} - F_{s,2})r - T_L - T_f] \\ 1/T (K p_0 - p_1) \\ 1/T (K p_0 - p_2) \end{pmatrix}}_{=:a(\mathbf{x})} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ K/T \\ -K/T \end{pmatrix}}_{=:b(\mathbf{x})} u \quad (3.13a)$$

$$y = c(\mathbf{x}) = \varphi \quad (3.13b)$$

The systems output y is the joint angle φ . Notably, (3.13) is linear in the input u and has one input and one output variable. In Figure 3.14 the control architecture with the linear PI-control scheme is shown. It illustrates the antagonistic set-up and the controller acting on the system's input Δp . This control scheme will be used as a reference system for the comparison with more advanced model based controller designs. Since the muscle characteristics have been identified, one straightforward extension of the linear PI-control is obtained by a model based feed forward compensation. In this scheme the desired pressure difference Δp is computed by solving the static equilibrium of (3.10a) with the desired contractions κ_1 and κ_2 . The pressure Δp is then added to the PI-controller output.

With the nonlinear state space model (3.13) a controller based on the principle of input-output linearization will be designed in the following. The design steps

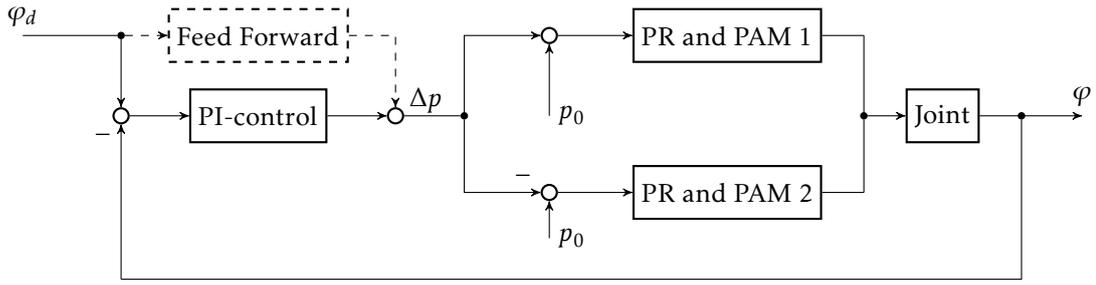


Figure 3.14: Overall view of the PI-control scheme with optional feed forward compensation (dashed) in the antagonistic muscle set-up with the pressure regulators (PR).

are based on Slotine and Li [1991]. With the Lie derivative $L_f h(\mathbf{x})$ of a scalar function $h(\mathbf{x})$ and the vector field $\mathbf{f}(\mathbf{x})$ defined by

$$L_f h(\mathbf{x}) = \left(\frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \right)^T \mathbf{f}(\mathbf{x}), \quad (3.14)$$

the relative degree δ of the system (3.13) is obtained as the first Lie derivative $L_b L_a^i c(\mathbf{x})$ unequal to zero (for $i = 0, \dots, \dim(\mathbf{x})$):

$$L_b L_a^i c(\mathbf{x}) = 0, \quad \text{for } i = \{0, 1\} \quad (3.15a)$$

$$L_b L_a^2 c(\mathbf{x}) = 2 \frac{rK}{\Theta T} \left(a_0 + a_1 \kappa_0 + a_2 \left(\kappa_0^2 + \frac{r^2}{\ell_0^2} x_1^2 \right) \right) \quad (3.15b)$$

$$\neq 0, \quad \text{for } x_1 \in \mathbb{R} \setminus \left\{ x_1 = \pm \sqrt{\frac{\ell_0^2}{r^2} \left(-\frac{a_0 + a_1 \kappa_0}{a_2} - \kappa_0^2 \right)} \right\} \quad (3.15c)$$

According to (3.15) the relative degree of the system is $\delta = 3$. Since $\delta < \dim(\mathbf{x})$, the system has unobservable internal dynamics. To design a stable controller, it must be shown that the internal behavior is stable. Please refer to Baur et al. [2014c] for a stability analysis of the internal dynamics. The control law, based on the system states \mathbf{x} is formally obtained by:

$$u = -r(\mathbf{x}) + v(\mathbf{x}) \varphi_d \quad (3.16a)$$

$$r(\mathbf{x}) = \frac{L_a^3 c(\mathbf{x}) + \alpha_2 L_a^2 c(\mathbf{x}) + \alpha_1 L_a c(\mathbf{x}) + \alpha_0 c(\mathbf{x})}{L_b L_a^2 c(\mathbf{x})} \quad (3.16b)$$

$$v(\mathbf{x}) = \frac{1}{L_b L_a^2 c(\mathbf{x})} \quad (3.16c)$$

An integral term can be added in (3.16b) to compensate for modeling errors or external disturbances at steady state. With the control law (3.16) the external dynamics of the closed loop system are described by the third order linear differential equation:

$$\ddot{\varphi} + \alpha_2 \dot{\varphi} + \alpha_1 \varphi + \alpha_0 \varphi = \varphi_d \quad (3.17)$$

By an appropriate selection of α_i ($i = \{0,1,2\}$) the desired dynamic behavior is achieved. If the desired trajectory $\varphi_d(t)$ is smooth and continuously differentiable up to its third derivative, exact tracking can be achieved by choosing the following control law:

$$u = -r(x) + \frac{\ddot{\varphi}_d + \alpha_2 \dot{\varphi}_d + \alpha_1 \varphi_d + \alpha_0 \varphi_d}{L_b L_a^2 c(x)} \quad (3.18)$$

With (3.18) the transfer function $G(s)$, describing the relation from the input φ_d to the output φ , of the closed loop system is obtained by the Laplace transformation:

$$G(s) = \frac{s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0}{s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0} \equiv 1 \quad (3.19)$$

For the realization of the controller, the states must be known. The variables φ , p_1 and p_2 are measured by the incremental encoder and the pressure regulators, respectively. To obtain the velocity $\dot{\varphi}$ from the position measurement a high gain observer has been implemented for the velocity estimation. Figure 3.15 gives an overview of the input-output linearization control scheme in form of a block diagram.

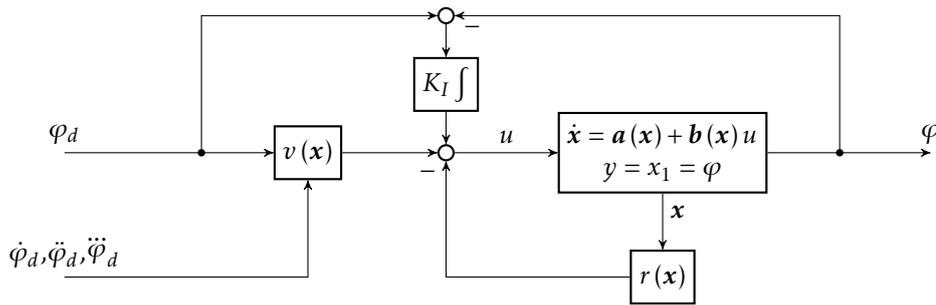


Figure 3.15: Block diagram of the input-output linearization control scheme.

Results

In this Section the performance of the investigated controller will be discussed, evaluated and compared. At first, all parameter used for the simulation and controller design are listed. In the Tables 3.5, the parameter resulting from the optimization problem according to (3.8) for the PAM model and the parameter describing the pressure regulator as well as the test bed parameter are given. Furthermore, Table 3.6 is listing the parameters implemented in the developed control algorithms. Experiments were carried out with a periodic rectangle profile and a staircase signal. To make the third time derivative available for the input-output linearization, the target position was filtered by a third order low pass filter. In Figure 3.16 the joint angle measurements for a rectangle profile with a period of 4s and an amplitude of 0.6 rad are shown for the PI-control with feed forward and the input-output linearization. For clarity, the results of the

Table 3.5: Estimated parameter for the PAM model (3.6) and the pressure regulator (3.9) are given in Table (a). In Table (b) the test bed parameter required for the dynamic model according to (3.13) are listed.

(a)			(b)		
Parameter	Value	Unit	Parameter	Value	Unit
a_0	$2.79 \cdot 10^{-3}$	m^2	r	$2.50 \cdot 10^{-2}$	m
a_1	$-7.32 \cdot 10^{-3}$	m^2	Θ	$3.20 \cdot 10^{-3}$	kgm^2
a_2	$-1.47 \cdot 10^{-2}$	m^2	$T_{f,0}$	$1.12 \cdot 10^{-1}$	N m
b_0	$3.38 \cdot 10^1$	N	b	$3.68 \cdot 10^{-1}$	N m rad^{-1}
b_1	$1.64 \cdot 10^3$	N	T_L	0	N m
b_2	$-3.46 \cdot 10^4$	N	ℓ_0	$3.00 \cdot 10^{-1}$	m
b_3	$6.96 \cdot 10^4$	N	κ_0	$1.06 \cdot 10^{-1}$	-
b_4	$2.08 \cdot 10^3$	N			
K	$9.94 \cdot 10^{-1}$	N m^{-2}			
T	$4.52 \cdot 10^{-2}$	s			

standard PI control scheme are not plotted in Figure 3.16. The control based on the input-output linearization is following the desired trajectory in the beginning of the step very accurately, while the desired final value is reached later compared to the PI-FF control scheme. This behavior can be explained by an inaccurate dynamic modeling of the PAM and therefore, the final value can only be reached by the additional integral term of the I/O control scheme. The corresponding tracking error

$$e := \varphi_d - \varphi \quad (3.20)$$

for the duration of one period is shown in Figure 3.17, including the tracking error results of the PI-control scheme.

Figure 3.18 compares the tracking results applied to the different control schemes with a desired position provided in a staircase profile ranging from -0.8 rad to 0.8 rad. In correspondence to the previously observed result, the accurate following behavior of the I/O controller is apparent, while the final value is

Table 3.6: Parameter of the controller.

Parameter	Value	Unit
p_0	$3.00 \cdot 10^5$	N m^{-2}
α_0	$1.47 \cdot 10^5$	$\text{m}^{-2} \text{s}^{-3}$
α_1	$8.39 \cdot 10^3$	$\text{m}^{-2} \text{s}^{-2}$
α_2	$1.59 \cdot 10^2$	$\text{m}^{-2} \text{s}^{-1}$
K_P (PI)	1.80	N m^{-2}
K_I (PI)	$1.89 \cdot 10^1$	$\text{N m}^{-2} \text{s}^{-1}$
K_P (PI-FF)	1.20	N m^{-2}
K_I (PI-FF)	4.50	$\text{N m}^{-2} \text{s}^{-1}$
K_I (I/O)	2.00	$\text{N m}^{-2} \text{s}^{-1}$

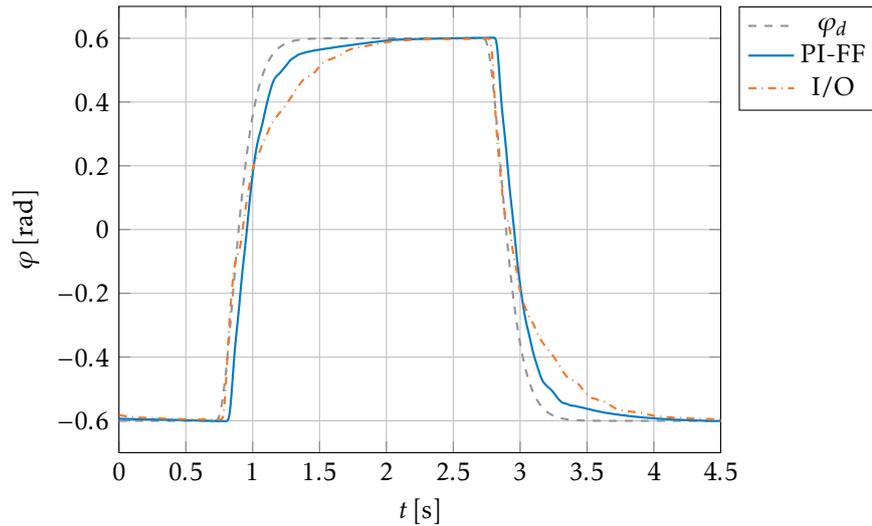


Figure 3.16: Comparison of the position following behavior of a PI-control with feed forward compensation and a controller based on input-output linearization for a rectangle profile.

reached not as fast as with the PI-FF control. The corresponding tracking error of the investigated control algorithms is plotted in Figure 3.19.

For a quantitative comparison of the experimental results, a performance index L is defined as follows:

$$L := \frac{1}{\bar{T}} \int_0^{\bar{T}} |e(t)| dt \quad (3.21)$$

The integral is evaluated for one period \bar{T} of the desired position value. In Table 3.7 the performance index L as well as the maximum values of the absolute tracking errors are summarized. Of the three investigated controllers, the PI-control in combination with the feed forward compensation performed best with respect

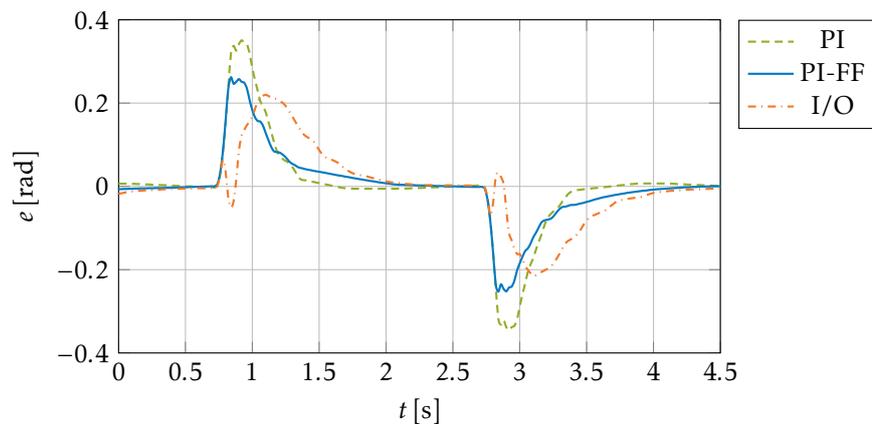


Figure 3.17: Comparison of the control error of the PI-control, the PI-control with feed forward compensation and the controller based on input-output linearization for the rectangle profile.

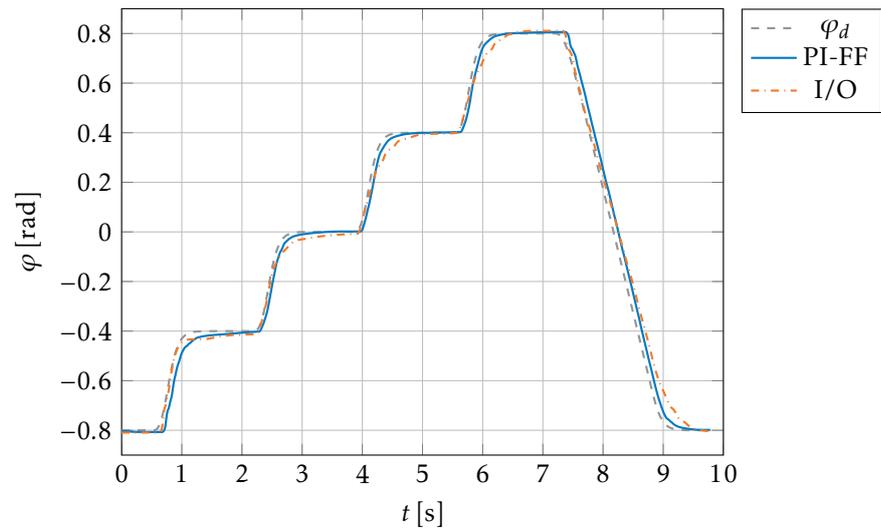


Figure 3.18: Comparison of the position following behavior of a PI-control with feed forward and a controller based on input-output linearization for a staircase profile.

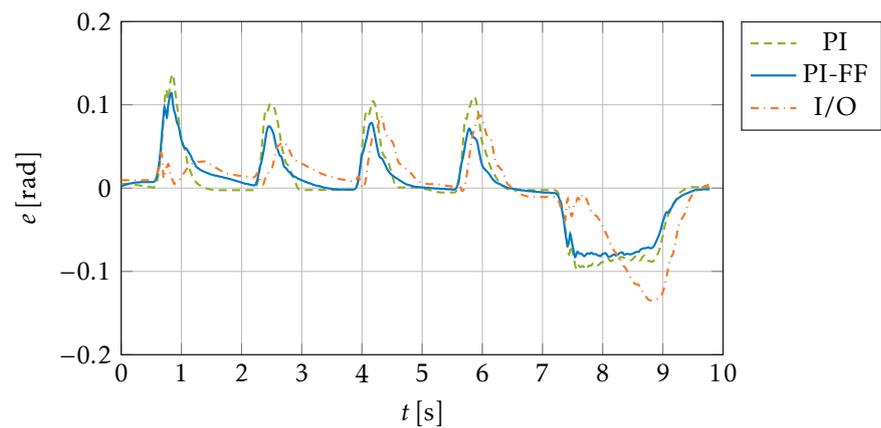


Figure 3.19: Comparison of the control error of the PI-control, the PI-control with feed forward and a controller based on input-output linearization for the staircase profile.

Table 3.7: Summary of the control quality of the investigated schemes.

Profile	Staircase		Rectangle	
Control Quality	L [rad]	$\max(e)$ [rad]	L [rad]	$\max(e)$ [rad]
PI	0.0321	0.1364	0.0575	0.3506
PI-FF	0.0281	0.1150	0.0515	0.2618
I/O	0.0321	0.1348	0.0636	0.2198

to the performance index L . The nonlinear controller with the input-output linearization has the smallest absolute error for the rectangle profile. As already indicated by the qualitative observations, the non-model based PI-control scheme has the poorest overall performance.

3.3.2 Electrical Drives

Electrical drives as actuators of mechatronic systems are widely used and a lot of experience from other robotics projects is available at the AM (Gienger et al. [2001]; Lohmeier et al. [2009]; Pfeiffer [2007]). Because there exists a great amount of different types, only a certain class, i.e. the Brushless Direct Current Electric Motor (BLDC) actuators will be discussed here. According to Siciliano and Khatib [2008], these type of actuators are a typical choice for the application in the field of robotics. They require a motor driver to provide the electrical commutation and the advantages especially for applications in robotics are (Isermann [2005]):

- High dynamics and overall efficiency
- Accurate positioning
- Good power-to-weight ratio
- Maintenance-free and high reliability
- Provide high peak torques for short periods

Although the actuator itself is not particular expensive, the disadvantages are that a motor driver as well as several sensors are required, hence making the overall costs of these actuators rather expensive.

3.3.3 Summary and Conclusion

In this Section two different concepts of actuation principles based on pneumatic and electric power sources have been discussed for the installation in the designed manipulators. Although a considerable amount of research has been performed on the integration of PAM actuators in manipulators, it is still today an innovative and far less studied drive concept compared to well-known electrical drives. Therefore, the design of a joint with PAMs actuators in an antagonistic set-up has been evaluated in detail and a test bed was built providing satisfactory results of the investigated controller concerning the position accuracy and variable stiffness control under laboratory conditions. In the end, however, the popular actuation mechanism for industrial manipulators, i.e. an electrical drive in combination with a high reduction ratio gear has been the preferred choice out of several reasons, being explained in the following. First of all, as the analysis for the kinematics design revealed, the available workspace is very limited, but a high dexterity is required to reach the goal poses and simultaneously avoid collisions with obstacles. To obtain this dexterity, many joints must be arranged on a very limited space with wide operation ranges. Due to the limited contraction capabilities of PAM,

wide joint ranges are impossible to obtain in an antagonistic set-up at reasonable transmission ratios. Another disadvantage of PAMs is that the dimensions of the robot would increase, because two actuators are required for one revolute joint. The advantage of PAM, namely the inherent compliance, might be unnecessary for agricultural applications, because most of the objects, like plants and leaves, in the environment are soft and therefore compliant. Finally, great efforts in the controller design would have been necessary to achieve high dynamics and accurate positioning with high reliability for the different applications while facing rough environmental conditions, like a wide temperature range.

3.4 Modeling and Simulation

The modeling of the dynamics of the developed manipulator prototypes is a very important aspect during the overall design and the operation of the robot. The dynamic model has been used during the system design at an early stage of the project for the dimensioning of the mechatronic components. The parameter for the first dynamic model were mainly taken out of catalog data. However, it is desirable for the prototypes as research platforms to describe the dynamics very accurately. Therefore, parameter identification experiments were performed with single components and the overall system.

The manipulator is assembled out of aluminum parts and the robot links are assumed to be rigid. On the basis of this assumption, major kinematic equations are derived in the Subsection 3.4.1. In the then following Subsections, the modeling of the dynamics and the gear friction is explained. Finally, the self-collision model applied for fast and efficient collision checking as well as the calculation of the minimum distance between collision pairs is introduced in Subsection 3.4.4.

3.4.1 Direct Kinematics

When working with robot manipulators, the equations for the description of the kinematics are of fundamental importance. According to the illustration of the developed manipulator kinematics in Figure 3.2 and Figure 3.3 all designed manipulators have an open-chain kinematic structure. Based on a reasonable assumption, the robots consist of several rigid bodies, also referred to as links or segments, which are connected with each other via joints. Because the agricultural manipulator has only prismatic and revolute joints, the relative position between two consecutive links is sufficiently described by one variable. The set of all joint position variables \mathbf{q} provides a minimal representation of the manipulator kinematics. The dimension of the joint vector \mathbf{q} specifies the amount of DoFs. To each link, a body fixed frame of references is attached. Similar to the convention introduced by Denavit and Hartenberg [1955], the z -axis of body fixed frames points along the joint axis to enable a straightforward implementation of the relative kinematics. Figure 3.20 shows several bodies and coordinate frames with focus on a prismatic joint (see Figure 3.20a) and a revolute joint (cf. Figure 3.20b) in a kinematic chain. Each body, except the first and last, has one parent and one child link.

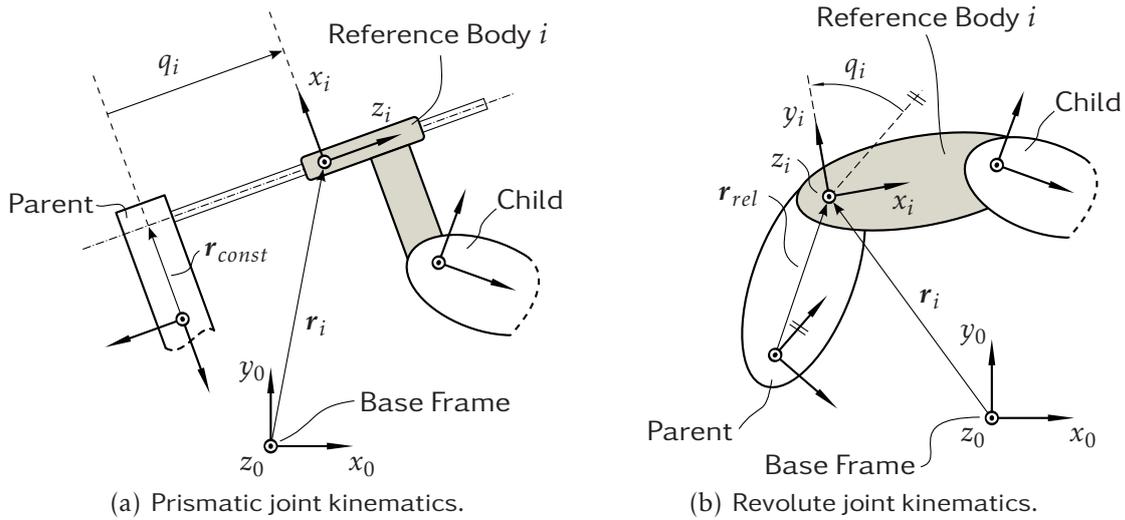


Figure 3.20: Serial manipulator kinematics.

Kinematic quantities of particular interest are:

- Position, velocity and acceleration of the origin O_i of the body fixed frame of link i : \mathbf{r}_i , $\dot{\mathbf{r}}_i$, and $\ddot{\mathbf{r}}_i$.
- Angular velocity and acceleration of link i : $\boldsymbol{\omega}_i$, $\dot{\boldsymbol{\omega}}_i$.
- Orientation of the body fixed frame with respect to the base frame in form of the transformation matrix \mathbf{A}_{0i} .
- Jacobian matrix of rotation $\mathbf{J}_{R,i}$ of link i : $\mathbf{J}_{R,i} := \frac{\partial \boldsymbol{\omega}_i}{\partial \dot{\mathbf{q}}}$
- Jacobian matrix of translation $\mathbf{J}_{TO,i}$ of the origin O_i of the body fixed frame of link i : $\mathbf{J}_{TO,i} := \frac{\partial \dot{\mathbf{r}}_i}{\partial \dot{\mathbf{q}}}$.

Starting from the base frame, all kinematic quantities can be calculated in a recursive fashion (Buschmann et al. [2006]; Siciliano et al. [2009]). Please refer to the Appendix B for the equations to recursively calculate the most relevant kinematic measures.

3.4.2 Dynamics

The dynamic modeling of the agricultural manipulator has already been discussed in a former publication of the author (Baur et al. [2013]) and is presented here for the sake of completeness. In Figure 3.21 the components with relevance for the modeling of the dynamics are illustrated. A few remarks on the modeling of elastic joints is made in Section 4.3 for the developed robot drive modules of the final manipulator. The belt is modeled as a rigid transmission. Initially, the mechanic modeling of the multibody dynamics is briefly described in the next Subsection. Afterwards, modeling of the actuator dynamics is explained and in Section 3.4.3 the applied gear friction models are introduced.

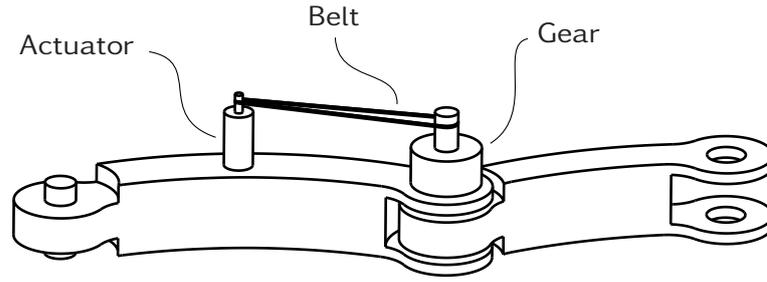


Figure 3.21: Major components of the drive chain (revolute joint) of the first manipulator prototype (Baur et al. [2013]).

Mechanical Modeling

The modeling of the mechanics is based on the classic Newton-Euler dynamic equations. Many textbooks are available and provide a good introduction on the modeling of dynamics in technical mechanics (Gross et al. [2008]; Pfeiffer and Schindler [2014]; Ulbrich [1996]). In the following, basic equations are summarized for convenience. Based on the set-up of a typical drive chain (cf. Figure 3.21), the free-body diagram of one link element i is drawn in Figure 3.22. In the

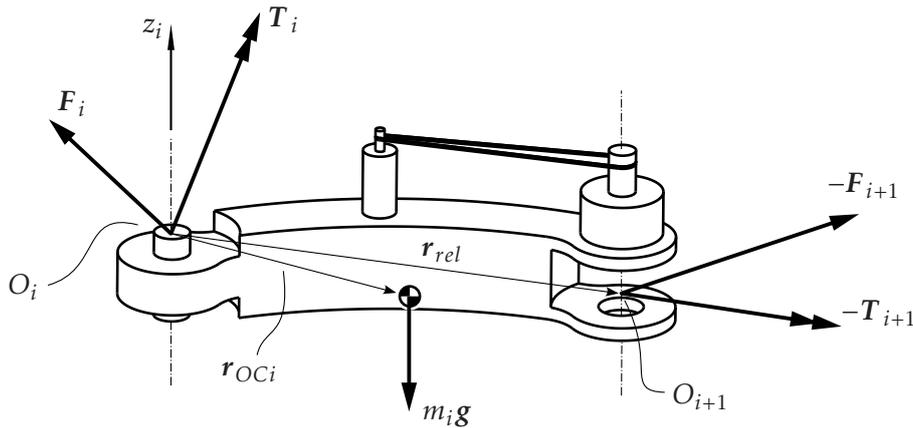


Figure 3.22: Free-body diagram of a robot link (Siciliano et al. [2009]).

free-body diagram, the forces (torques) F_i (T_i) and F_{i+1} (T_{i+1}) are imposed on the link i at the interfaces to neighboring bodies. In the center of mass acts the resulting force exerted on the body by gravity. Principles of linear and angular momentum theory according to Figure 3.22 can be applied to the link i :

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \mathbf{F}_{i+1} + m_i \mathbf{g} \quad (3.22a)$$

$$\dot{\mathbf{L}}_i^{\langle O_i \rangle} + m_i \tilde{\mathbf{r}}_{OCi} \ddot{\mathbf{r}}_i = \mathbf{T}_i - \mathbf{T}_{i+1} + \mathbf{T}_{G_i}^{\langle O_i \rangle} - \mathbf{T}_{F_{i+1}}^{\langle O_i \rangle} \quad (3.22b)$$

The forces F_{i+1} and the load due to gravity $m_i \mathbf{g}$, impose the following torques with respect to the origin O_i :

$$\mathbf{T}_{F_{i+1}}^{\langle O_i \rangle} = \tilde{\mathbf{r}}_{rel} \mathbf{F}_{i+1} \quad (3.23a)$$

$$\mathbf{T}_{G_i}^{\langle O_i \rangle} = m_i \tilde{\mathbf{r}}_{OCi} \mathbf{g} \quad (3.23b)$$

The time derivatives of linear momentum \mathbf{p}_i and the angular momentum $\mathbf{L}_i^{(O_i)}$ with respect to O_i are given by:

$$\dot{\mathbf{p}}_i = m_i \ddot{\mathbf{r}}_{Ci} \quad (3.24a)$$

$$\dot{\mathbf{L}}_i^{(O_i)} = \mathbf{I}_i^{(O_i)} \dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i \mathbf{I}_i^{(O_i)} \boldsymbol{\omega}_i \quad (3.24b)$$

The body mass is indicated by m_i , the inertia tensor of link i , including the rotary inertia of the gear and motor shaft, with respect to O_i by $\mathbf{I}_i^{O_i}$ and the absolute acceleration of the center of gravity $\ddot{\mathbf{r}}_{Ci}$ is calculated according to:

$$\ddot{\mathbf{r}}_{Ci} = \ddot{\mathbf{r}}_i + (\dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i \boldsymbol{\omega}_i) \mathbf{r}_{OCi} \quad (3.25)$$

The generalized force Q_i , acting in the free motion direction of joint i is obtained with the unit vector in z-direction \mathbf{e}_z by:

$$Q_i = \begin{cases} \mathbf{T}_i^T \mathbf{e}_z & \text{for revolute joints} \\ \mathbf{F}_i^T \mathbf{e}_z & \text{for prismatic joints} \end{cases} \quad (3.26)$$

By consideration of the friction torque (force) T_{fi} (F_{fi}), the motor torque T_{ai} , acting on the input shaft is calculated with the reduction ratios of the belt and the gear, N_{Bi} and N_{Gi} as follows:

$$T_{ai} = \begin{cases} \frac{1}{N_i} Q_i + T_{fi} & \text{for revolute joints} \\ \frac{1}{N_i} (Q_i + F_{fi}) & \text{for prismatic joints} \end{cases} \quad (3.27a)$$

$$N_i = N_{Bi} N_{Gi} \quad (3.27b)$$

For the dimensioning and design of the parts, the forces and torques acting on the links at a certain motion of the manipulator are of interest. For a given joint trajectory $\mathbf{q}(t)$ and its first and second order time derivative, the kinematic measures can be calculated in a forward recursion (see Appendix B) and the time derivatives of the linear and angular momentum are obtained by (3.24). Afterwards the forces and torques are computed in a backward recursion, starting from the end-effector with known (in this case usually zero) external forces and solving the six equations (3.22) for \mathbf{F}_i and \mathbf{T}_i for each link. Notably, the equations for the dynamics are not provided in closed-form. However, they can be written in the compact and closed-form for the multibody system:

$$\mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \bar{\mathbf{Q}} \quad (3.28)$$

In (3.28), \mathbf{M} is the mass matrix, the vector \mathbf{h} contains the Coriolis, centrifugal, and gravitational forces while $\bar{\mathbf{Q}}$ is the vector of generalized torques including motor and friction torques. In the form of (3.28) the equations of motion are more suitable for controller design.

Electrical Modeling

The applied actuators for the developed manipulators are BLDCs electrical motors (see Section 3.3). They require a controller for the electrical commutation, usually implemented on the motor driver. The three-phase system of the actuator coil with three alternating currents can be modeled with the direct quadrature zero transformation initially described by Park [1929] with two direct current quantities in a rotor fixed frame. These quantities are the active current I_q and the reactive current I_d . Usually the reactive current is controlled to zero by the motor driver and the motor torque is proportional to the active current with the torque constant k_m . It is a commonly used approximation to model the BLDC actuator similar to the direct current motor (Isermann [2005]; Siciliano et al. [2009]) according to:

$$L\dot{I} + RI + k_m\dot{\varphi} = U \quad (3.29)$$

The scalar parameter L describes the motor inductance, the electrical resistance is given by R , and the term $k_m\dot{\varphi}$ is representing the electromotive force, induced by the speed of the rotor $\dot{\varphi}$ in the magnetic field generated by the actuator coil. The variables $I = I_q$ and U indicate the armature active current and voltage.

3.4.3 Joint Friction Modeling

In this Section the modeling of the friction in the manipulator joints is described. For the revolute joints, gears with a high reduction ratio from the company Harmonic Drive were installed and the design of a testbed for experimental friction identification is extensively described in Chapter 4. However, in the next Subsection, the friction models used during the design and the adapted model after the detailed experimental analysis are introduced. For the prismatic joints, the friction has been experimentally identified. These measurements and the applied model are described afterwards.

Harmonic Drive Gears

One way of describing the friction term T_f of the gear, according to Buschmann [2010], is:

$$T_f = -\text{sgn}(\dot{\varphi})\left(T_{f,0} + \mu|T_l|\right) - (b + \gamma|T_l|)\dot{\varphi} \quad (3.30)$$

This is a classic approach with a static friction term ($T_{f,0}$), the Coulomb friction ($\mu|T_l|$), viscous friction ($b\dot{\varphi}$) and a coupled load and speed dependent term ($\gamma|T_l|\dot{\varphi}$). In the catalog of the gear manufacturer, experimental results on the efficiency η of the product are given for varying load, speed and temperature cases. During the design phase, this data can be used to compute the parameter $T_{f,0}$, μ , b and γ of (3.30), by solving the following constrained optimization problem according

to Buschmann [2010]:

$$\sum_i (\eta_{i,\text{catalog}} - \eta_{i,\text{model}})^2 \rightarrow \min \quad (3.31)$$

$$T_{f,0} \geq 0, \quad \mu \geq 0, \quad b \geq 0 \quad (3.32)$$

The results of the parameter estimation of the installed Harmonic Drive gears are summarized in the Appendix C. The curve fit with (3.30) on the catalog data is very accurate and sufficient at the design state. However, experiments with robot modules of the final manipulator prototype showed considerable differences between the curve fit of the friction model (3.30) to the measurements (Baur et al. [2014a]). Therefore, a friction law, similar to one suggested by HESS AND SOOM (referred to in Armstrong-Hélouvry et al. [1994]), is proposed for a more accurate friction model description based on measurements:

$$T_f = -\text{sgn}(\dot{\varphi}) \left(\bar{T}_{f,0} + \bar{\mu} |T_l| \right) - \bar{b} \dot{\varphi} - \frac{T_{f,S} - \bar{T}_{f,0}}{1 + \left(\frac{\dot{\varphi}}{\dot{\varphi}_S} \right)^2} \quad (3.33)$$

The first three terms have the same meaning as in (3.30). However, the bar indicates different values of these variables. The fraction term is describing the Stribeck Effect and is parametrized by $T_{f,S}$ and $\dot{\varphi}_S$. A detailed discussion on the experiments and the parameter identification is given in Chapter 4.

Friction Modeling of the Prismatic Joints

For the modeling of the friction in the linear bearings of the manipulator prototypes, experimental parameter identifications have been performed. The prismatic joint of the first prototype consists of three round guidance rails arranged in a triangle while the carriage (first robot element) is equipped with six linear roller bearings of type KS-30PP² from the company Ina. A toothed belt is connected to the carriage for the transmission of the motor torque. Experiments were performed with the completely assembled manipulator. To identify the viscous and static friction term, the prismatic joint has been moved from the minimum to the maximum joint position limit with the following motion profile: constant acceleration, no acceleration, and constant deceleration. During the experiment, the motor active current and the motor shaft position have been measured and the velocity was estimated. Notably, this experiment identifies all friction effects in the drive chain of the prismatic joint at once. The center of mass with respect to the base frame of the manipulator during the experiments was at $r_{CoM} = (200\text{mm}, 58\text{mm})^T$. During a post-processing evaluation, the part with constant velocity has been identified and the average velocity and current during this range were calculated. Figure 3.23 presents the measurements of the friction force F_f plotted against the joint velocity \dot{q}_1 . Since the system was imposed to the gravity g , the constant gravitational force $F_G = mg$ has been compensated in the

² inner diameter: 30 mm; outer diameter: 47 mm; length: 68 mm

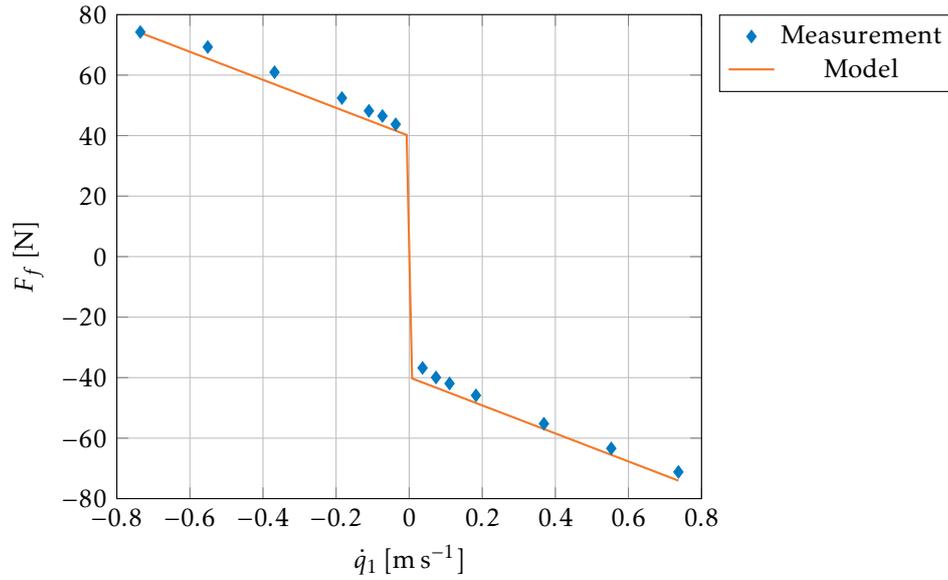


Figure 3.23: Friction measurement with the prismatic joint of the first manipulator prototype (version 2).

results. With the current measurement data I , the torque constant k_m , and the diameter d of the cog wheel, the friction force is computed by:

$$F_f = \frac{2k_m I}{d} - F_G \quad (3.34)$$

The joint velocity has been calculated under the assumption of a rigid transmission of the belt and the cog wheel diameter according to the data sheet.

The following friction law with the static friction $F_{f,0}$ and the viscous friction b has been fitted to the measurements:

$$F_f = -\text{sgn}(\dot{q}_1)F_{f,0} - b\dot{q}_1 \quad (3.35)$$

For the curve fit, the Matlab function `lsqcurvefit` with the implemented `trust-region-reflective` algorithm has been applied. Relevant parameter and results of the experiment are listed in Table 3.8. As expected due to Coulomb friction,

Table 3.8: Major parameters and curve fitting results for the friction identification experiments of the prismatic joint (first prototype, second version).

m [kg]	k_m [Nm A ⁻¹]	d [mm]	$F_{f,0}$ [N]	b [Ns m ⁻¹]
19.32	0.56	70	39.91	46.32

measurements performed in different robot configurations, i.e. with a different center of masses, provided slightly different results. However, further investigation has not been performed, since the obtained results were graded sufficient for the system design.

For the final manipulator, the linear axle of type EGC-TB from the company F&S TO

has been utilized. The experiments were carried out similar to the measurements with the first prototype. However, the manipulator was not mounted on the carriage and the slide has been oriented horizontally. The friction model according to (3.35) has been fitted to the measurements. Major parameters and results of the experiments are summarized in Table 3.9.

Table 3.9: Major parameters and curve fitting results for the friction identification experiments of the prismatic joint (final manipulator).

k_m [N m A ⁻¹]	d [mm]	$F_{f,0}$ [N]	b [N s m ⁻¹]
0.56	73.85	67.41	22.97

3.4.4 Self-Collision Model

For the developed manipulators, a motion planning and inverse kinematics algorithm, suitable for real-time application has been developed (see Chapter 5). To avoid hardware damage during operation, an algorithm to perform collision checks of manipulator parts in one sample interval is mandatory for an on-line planning scheme. Additionally, for the efficient avoidance of manipulator configurations in self-collision, the minimum distance between two collision pairs is of particular relevance (cf. Section 5.4.3). A recent and very detailed overview on available methods and software frameworks for distance computation and collision checking is provided by Schwienbacher [2013].

During the development of the agricultural manipulators, the framework implemented by MARKUS SCHWIENBACHER for the humanoid robot LOLA has been adapted and applied to the manipulator prototypes. The basic idea is to approximate the robot link geometries with several simple volumes while each of these volumes can be described by a few parameters. These volumes are called Swept Sphere Volumes (SSVs) and allow for a very efficient distance computation. To ensure collision avoidance, the geometry is conservatively approximated, i.e. the model completely encloses the robot volume given by the Computer-Aided Design (CAD) design. The available shape primitives of the framework are shown in Figure 3.24 and listed in the following:

- Point SSV with the parameter point and radius.
- Line SSV with the parameter two points and radius.
- Triangle SSV with the parameter three points and radius.

With a higher amount of SSVs utilized to describe the robot's geometry, the more accurate the geometry can be approximated. During the system design, the SSV elements used to describe a given object were parametrized based on a three dimensional visualization of the geometry with a modeling tool described in Schwienbacher [2013]. For the first prototype in the 9-DoFs configuration, a total amount of 74 SSVs has been used to approximate the geometry. Because of the

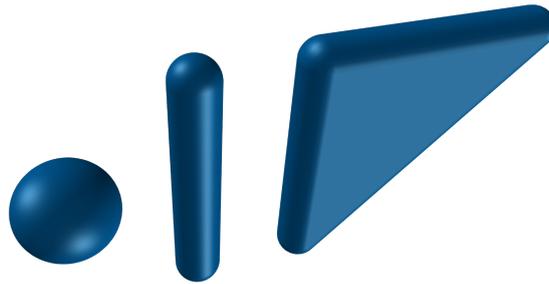


Figure 3.24: The basic point, line and triangle SSVs (Schwienbacher et al. [2011]).

smooth surface and the more protected mechanical design of the final manipulator prototype 28 SSVs elements were sufficient to approximate its geometry (9-DoFs configuration). The SSVs representations of the first and the final manipulator prototype are illustrated in Figure 3.25.

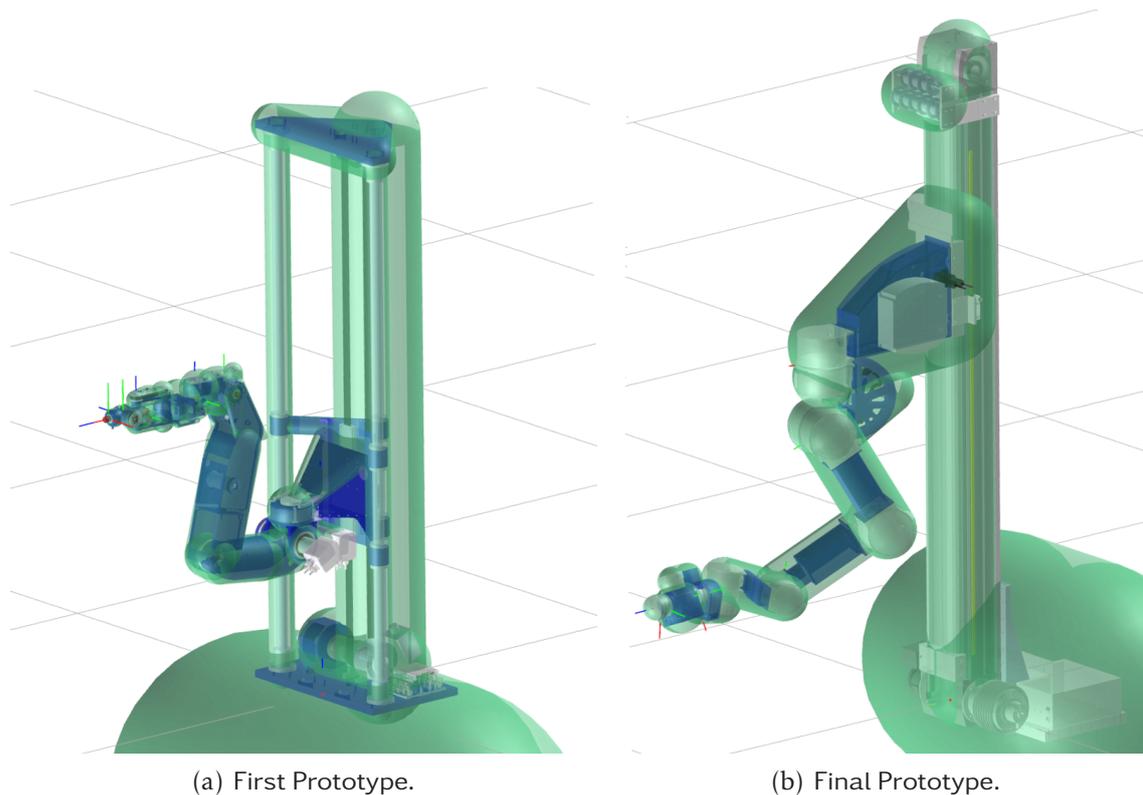


Figure 3.25: Self-collision models with SSVs of the developed prototypes (both in 9-DoFs configuration).

The average computation time of the cost functional for collision avoidance according to (5.55) took less than $80 \mu\text{s}$ on an Intel i5-661 processor with 3.33 GHz for the first manipulator prototype. The integration of the self-collisions model into the inverse kinematics algorithm is explained in Section 5.4.3.

3.5 Hardware and Control Architecture

In this Section, an overview on the systems electronics architecture of the two generations of developed manipulators will be given. The selected electronic components and their interaction with each other are presented. First, the hardware components used for the first generation of prototypes are introduced and at the end of this Section, the improvements with respect to the hardware design of the final manipulator are highlighted. The design of the manipulator electronic hardware components was strongly supported by GEORG MAYR. Besides, CHRISTOPH SCHÜTZ contributed to the system architecture, in particular by the integration of the EtherCAT communication bus for the final manipulator prototype and the end-effector I/O interface.

3.5.1 Hardware Overview

An overview of the manipulator hardware architecture with its most significant electrical components is illustrated in Figure 3.26 including nine robot joints of the first ① and the final ② manipulator. An explanation of each component and its interaction with the other system modules is given in the following.

Real-Time Control Unit

This control unit is running an operating system for real-time applications. Every manipulator related component is connected to this computer and it can therefore be considered as the central part of the manipulator electronics. The computer is equipped with two PCI-CAN boards from the company SOFTING (type: CAN-AC2-PCI). Each board has two CAN bus channels, respectively. To achieve low sample times, one bus is connected to three motor drivers at most. Hence, the manipulator in the 9-DoFs configuration requires three out of the four available bus interfaces. The fourth CAN bus is the interface to the end-effector. With the digital I/O's provided by the computer's parallel port, the real-time application is communicating with a microcontroller which monitors and controls the main power supply. Furthermore, a watchdog signal is transmitted to the aforementioned microcontroller for safety purposes. Via an Ethernet interface, the real-time control unit is linked to a host computer provided for user interaction. A detailed description of the real-time software application can be found in Section 3.6.1.

Rack-Mounted Electronics

In a standard 19-inch rack, the main and auxiliary power supplies as well as the line voltage for the real-time control unit and the host computer were mounted. The main power with a voltage level of 48 V is the power source for the electrical actuators, while the auxiliary power with 24 V is utilized to power other electrical components³, in particular sensors, the end-effector and microprocessors installed in the motor drivers. By a digital output from a microcontroller of the company ATMEL, a relay is controlled to turn off the main power supply.

3 The auxiliary voltage is transformed to lower voltage levels for some sensors.

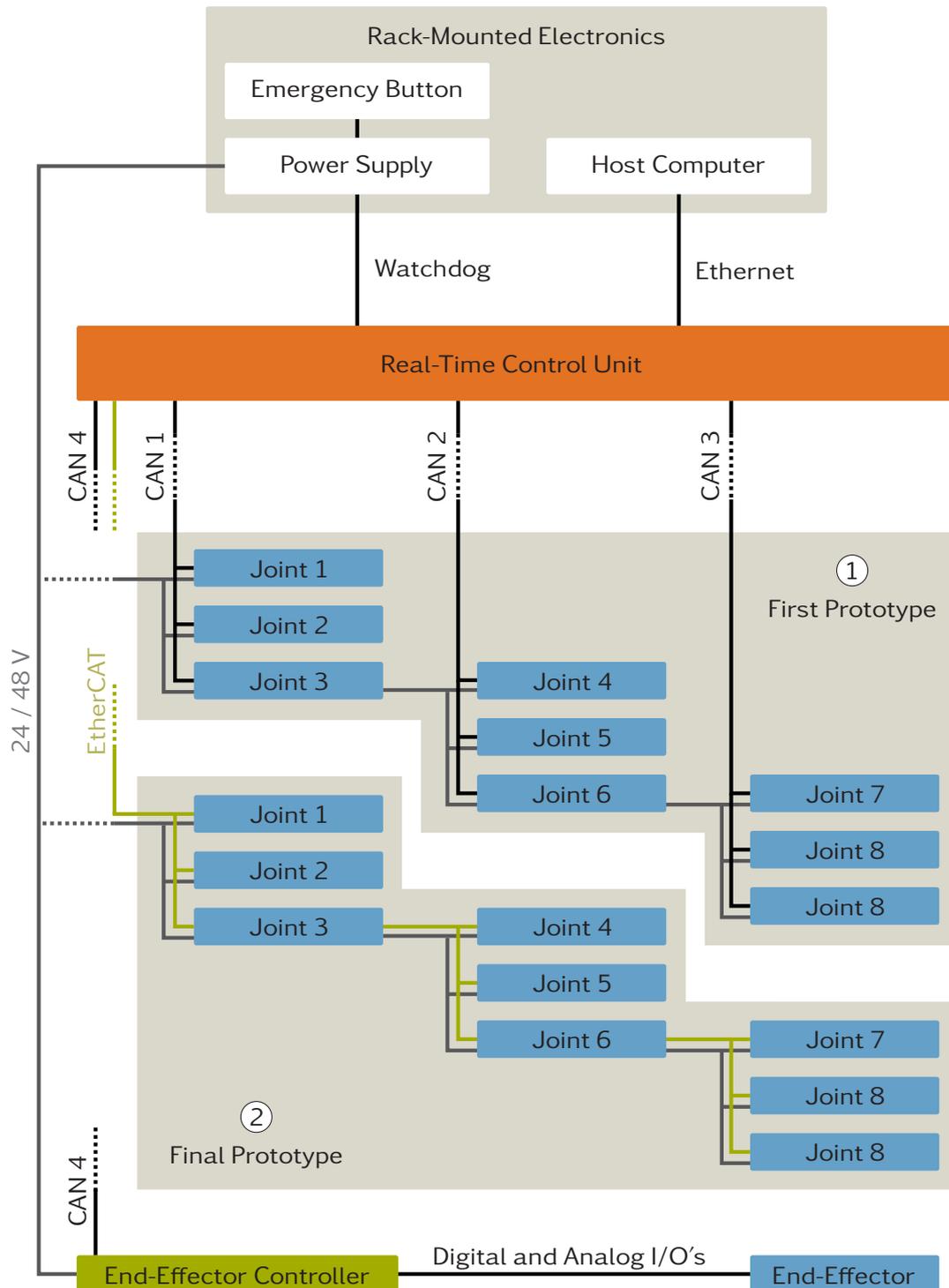


Figure 3.26: Electronics architecture of the developed manipulators (9-DoFs configuration). The communication of the real-time control unit with the motor drivers is obtained via CAN for the first and via EtherCAT for the final prototype.

Host Computer

This computer is connected via a cross-over Ethernet cable with the real-time unit and is the interface to control and monitor the manipulator. The software running on this machine is explained in Section 3.6.3.

Robot Joints

Almost all developed robot joints include electronic components of the same type. In Figure 3.27 the major components that are connected to the motor driver are illustrated. All motor drivers have been selected from the company

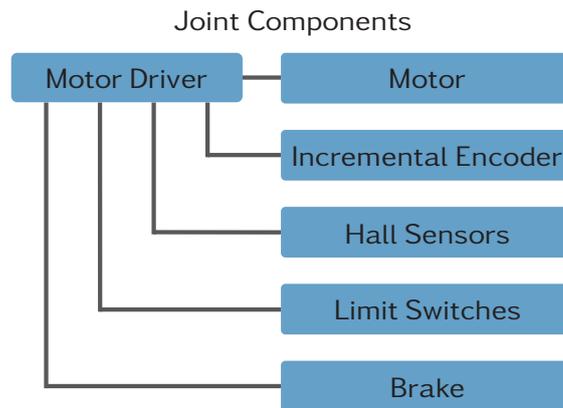


Figure 3.27: Controller, actuators and sensors of the robot joints.

ELMO MOTION CONTROL and are of the product line *SimpleIQ*. For the revolute joints, the model *Solo Whistles* and for the prismatic joint the *Drum* servo drive controller model has been chosen. The motor driver of a robot joint is connected to the BLDC motor and the main feedback source for the position control is an incremental encoder attached to the motor shaft on the input side of the joint. The hall sensors are used during start-up to measure the position of the stator with respect to the motor shaft so that the motor driver can set-up the electrical commutation. After the initial electrical field angle is known, only the feedback of the encoder is used to keep up the electrical commutation. Since only incremental encoders were installed in the first prototypes, limit switches are utilized to obtain the reference position after the start-up of the system. An automated sequence to acquire the reference positions of all joints has been implemented. Finally, a mechanical brake⁴ is linked to the motor shaft to hold the manipulator in its configuration when it is not switched on.

End-Effector Controller

This controller board is the interface between the real-time control unit and the end-effectors. The module is connected via CAN to the real-time unit and provides digital and analog I/O's for the different end-effectors developed in the project.

⁴ Only the joints that are imposed to gravitational loads were equipped with brakes.

Typical input commands could be „grip“ or „release“, and output could be the feedback from sensors on the end-effector reporting success or failure of a gripping process. For the precision spraying end-effector an analog output is utilized to control the fan speed and a digital output controls the nozzle.

3.5.2 Decentralized Joint Control

For robot manipulators a broad variety of control strategies are very well studied and presented in fundamental literature on control and robotics (Siciliano et al. [2009]; Slotine and Li [1991]). Typically, control objectives could be defined as the tracking of a joint trajectory on the position level, the force control of an end-effector tool or a combination of both. Other aspects, like the interaction with the environment might be considered as well. In the case of the here developed manipulators, the first objective is the position control of the TCP along a certain trajectory and in particular with the highest accuracy close to the final position. For this, joint control with synchronized motion is performed and a joint trajectory generation is required (see Chapter 5). The requirements on the position accuracy for the agricultural applications are rather low, compared to the capabilities of standard industrial manipulators or the here developed prototypes. Therefore, it has been decided to select a basic decentralized joint control scheme. This scheme considers each joint controller as a Single-Input and Single-Output (SISO) system while the coupling effects, like configuration dependent load and inertia or other effects from the multibody dynamics, are treated as disturbances. This approach is reasonable, especially for manipulators utilizing gears with high reduction ratios, since the transmission has a decoupling effect on the multibody dynamics. However, in contrast to the actuation with direct drives, the gear friction becomes more significant (see Chapter 4).

An efficient and well-established control strategy for servo motors is a classical cascade control scheme. It is implemented on the commercial motor drivers, already mentioned in Section 3.5.1. To compensate for a steady state error the control scheme of the closed loop system should provide integral behavior. For the inner loop a PI-control structure is present. It is controlling the current in the actuator coil and is running at a sampling rate of 90 μ s. The desired variable is the current (torque) command and the actuation variable is the voltage in the motor coils generated by a Pulse-Width Modulation (PWM) signal while the feedback is based on a current measurement. The PI-velocity controller is wrapped around the inner control loop with twice the sampling interval of the current controller. The velocity feedback is estimated based on the motor shaft position measurement provided by an incremental encoder. Finally, a proportional gain controller is forming the outer position control loop. The sample rate of the position controller is four times the sample rate of the inner control loop. When an integral controller is involved, an anti wind-up algorithm is added. The controller were tuned consecutively, starting from the inner to the outer loop with a software tool provided by the manufacturer of the motor driver. In identification experiments, the open and closed loop characteristic function for the transfer behavior have been measured. Occasionally, a manual tuning has been performed

afterwards to optimize the controller accuracy and robustness.

The desired joint positions for certain time instances are sent via the CAN bus interface from the real-time control unit to the motor driver as Process Data Objects (PDOs) every 5 ms. The motor driver then performs a linear interpolation between two consecutive supporting points and tracks the joint trajectory.

3.5.3 Improvements with the Final Manipulator

The improvements with respect to the hardware components of the final manipulator prototype compared to the first two prototypes will be, in essence, highlighted in this Section. To support the concept of modularity, integrated drive modules in different sizes have been designed, so that a reconfiguration in various desired kinematic configurations is possible (Pfaff et al. [2014]). All integrated drive modules contain the same hardware components as the robot joints introduced in Section 3.5.1 (cf. Figure 3.27) including brakes and in addition absolute position encoders on the joint output side. With the absolute position measurements, no position referencing is required upon start-up of the system and a more accurate control could be implemented. Furthermore, the next generation of servo drives, i.e. the *Gold Line* from the manufacturer ELMO MOTION CONTROL were built-in the modules. On the one hand, the sampling time of the cascade control loop is reduced (inner current control loop is running at 50 μ s) and on the other hand, they support the Ethernet based real-time protocol EtherCAT, developed by the company BECKHOFF AUTOMATION. With the EtherCAT bus standard, the real-time control unit is operated as master and the motor drivers are connected as slaves. Due to the much higher bandwidth more data can be exchanged and the latency is reduced. Therefore, the sample rate of the real-time control unit is reduced from 5 ms to 1 ms improving the control performance.

3.6 Software Architecture

Although programming efforts should be reduced to a minimum the software development remains an integral part of the manipulator design. It is required not only for the implementation of the algorithms presented in this thesis, but also to dictate the overall behavior of the autonomous system. The purpose of this Section is the description of the developed software components as well as their interaction and integration with other system components. Besides the suitability for efficient and fast programming, there are several aspects and prevailing conditions which have to be considered during the software design and the selection of development tools. First of all, the manipulator software application must provide low level communication interfaces to the robot hardware components, like motor controllers or sensors. Furthermore, the application should be computationally efficient and capable of real-time operation to handle feedback control and on-line interaction. Of course, and especially since the robot has been used by project partners during field and laboratory experiments, the software application must be fail-safe with a good error handling to avoid and trace hardware damages and failures. For efficient testing of other software modules and their interaction with

the manipulator application a simulation environment should be provided so that the robot hardware is not necessarily required. The software must also be very robust and easy to use. Finally, the code must provide data logging capabilities and debugging operations to easily trace programming errors. The manipulator program has to be interfaced with the main application, sensors and other algorithms developed for the different applications. Figure 3.28 gives an overview of the software structure and the integration of the manipulator application and its interfaces within the CROPS project. The manipulator application for the

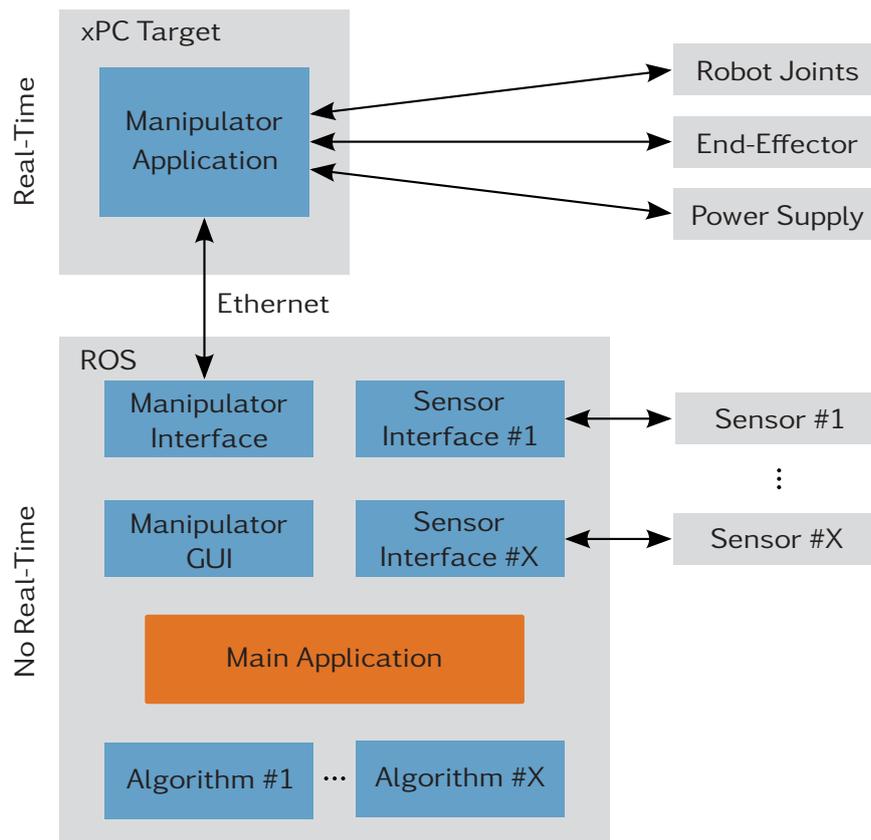


Figure 3.28: Overview of the software framework for the manipulators and its integration in the overall framework.

real-time operating system xPC Target from the company MATHWORKS has been programmed in Matlab/Simulink. The dedicated computer running the manipulator application is interfaced via Ethernet connection to an application that provides an interface to a middleware. Within the CROPS project, the Robot Operating System (ROS) (Quigley et al. [2009]) was chosen as the middleware and is supposed to be the common interface for all soft- and hardware components. Inside this software framework, all developed applications can communicate with each other based on a publisher/subscriber concept. A more detailed overview on all software modules, developed in the CROPS project, has been published in a joint publication by Barth et al. [2014]. In the upcoming Subsections, the components of the real-time manipulator application and the interface to the ROS middleware will be explained in more detail.

3.6.1 Real-Time System

The real-time application is the core element of the software to control the manipulator. For a fast and efficient development, the xPC Target system from MATHWORKS has been selected. On a host computer, a model developed with Matlab/Simulink can be build and uploaded to a target computer hardware. This approach has the major advantage, that drivers for many commercially available hardware boards, providing analog or digital inputs and outputs as well as communications protocols (e.g. CAN bus) are supported. This saves time during the software development since no hardware drivers have to be implemented. Furthermore, the xPC Target application has an interface to Matlab and data logging and signal monitoring can be done easily. The overall manipulator application has been divided into several smaller modules with defined interfaces. This allows for an efficient programming and debugging. The modules were stored as Matlab/Simulink libraries supporting code reuse. The major software components for the real-time application and the relevance for the manipulator are illustrated in Figure 3.29. A brief description of the modules will be given in the following.

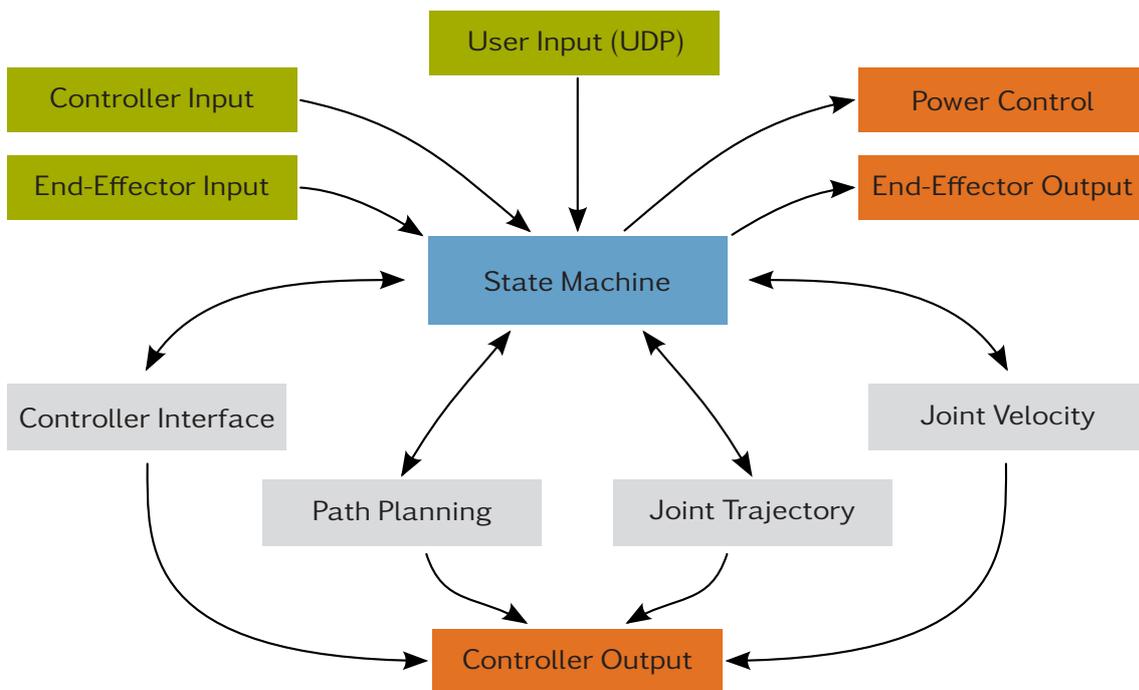


Figure 3.29: Main components of the manipulator (real-time) application.

State Machine

The finite state machine is responsible for the manipulator behavior and is of particular importance for any operation. The user input, which is sent via User Datagram Protocol (UDP) messages to the real-time application is processed by the state machine that triggers the desired actions in correct order. It is monitoring the status of the system and reports success or the reason of failure for a certain

task. In case of failure it will turn off the main power supply and switch into the FAULT state. Figure 3.30 illustrates the possible states and transitions for a point-to-point motion of the manipulator.

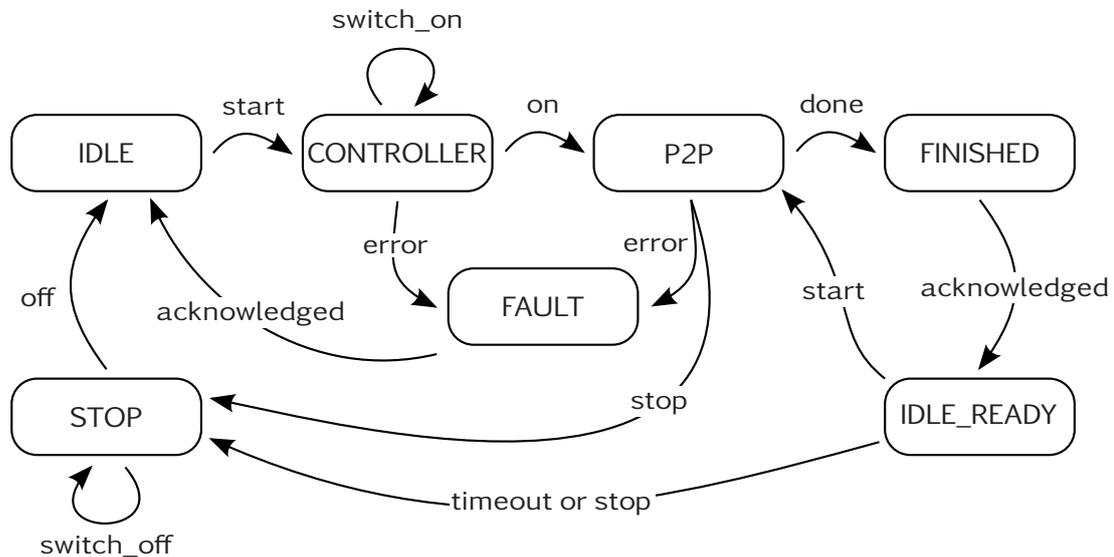


Figure 3.30: State transitions for a point-to-point motion.

Controller Interface

Each joint is equipped with an actuator and several sensors required for proper operation. The sensor data is read in by commercial motor controllers which are connected to the bus system. The controller interface is responsible to handle the communication between the motor controller and the real-time application. It is sending the required transitions to switch the motor controller into certain modes of operation (e.g. position control mode, quick stop, and so on). Furthermore, it monitors status messages of the controller devices and reports to the state machine. Similar to the state machine, this part of the application is event driven.

Path Planning

The path planning software module comprises the workspace planning and the inverse kinematics algorithm according to Figure 5.3 using kinematic models for all developed manipulator prototypes. Furthermore, it is calculating the direct kinematics, monitors the joint and TCP tracking and performs collision checking during manipulator motion. Besides the data input described in Section 5.3, the module is capable of processing a TCP velocity vector while constantly computing the inverse kinematics. Reference joint position and velocities at each sample time interval are sent to the motor controller and via UDP to the host interface for monitoring. This module is performing the highest amount of computations and is therefore the significant factor (besides the latency of I/O modules) for the selection of the lowest possible sample time of the overall real-time application.

Joint Trajectory

This module is an interface to handle offline generated joint trajectories and forward them to the joint controller. The supporting points with a fixed time interval of a generated trajectory can be send to this module using the UDP interface. If the interval is not constant, a cubic spline interpolation is performed automatically. Furthermore, the smoothness of the trajectory as well as position limits and self-collisions are checked. With this interface, offline planners can be integrated which are running on a separate computer.

Joint Velocity

With this part of the manipulator program, desired joint velocities can be forwarded from the user interface to the motor controller. The maximum allowed velocities, joint limit violations and self-collisions are monitored. The motor controller are operated in a velocity control mode. This interface allows for the implementation of a position control loop via the UDP interface on the host computer.

Other Modules

Other software modules not shown in Figure 3.29 are for dynamic simulation and control purposes. These are an implementation of the dynamic equations discussed in the Subsection 3.4.2 and joint cascade controller. Especially during the simulations for the system design, they have been of particular importance.

3.6.2 Further Remarks on the Implementation

All core algorithms for the real-time application have been implemented in C/C++ making use of the object-oriented capabilities as well as the high flexibility and efficiency provided by this programming language. The implemented C/C++ classes were sorted and compiled as libraries. For example, one library, labeled *models* contains all the kinematic and dynamic parameter of the manipulator prototypes and inherits algorithms for kinematic and dynamic computations for the multibody system from another library and classes. A library supporting very efficiently common operations in the field of linear algebra has been developed at the AM for other robotic projects and has been combined with the code. To link C/C++ code with the Matlab/Simulink framework, MATHWORKS provides a standard interface with so called S-Functions. The S-Function blocks are compiled to Matlab executable files and have some predefined callbacks methods which are invoked by the Simulink engine. For these S-Functions, C/C++ wrapper classes have been implemented for all software modules, linking the pointers of input and output data from the Simulink block to the custom generated C/C++ code. This approach is illustrated in Figure 3.31.

Figure 3.29 illustrates the software framework for the real-time target computer and the usage with the manipulator hardware. However, in many cases, for example when the physical hardware is not available or for testing purposes, a

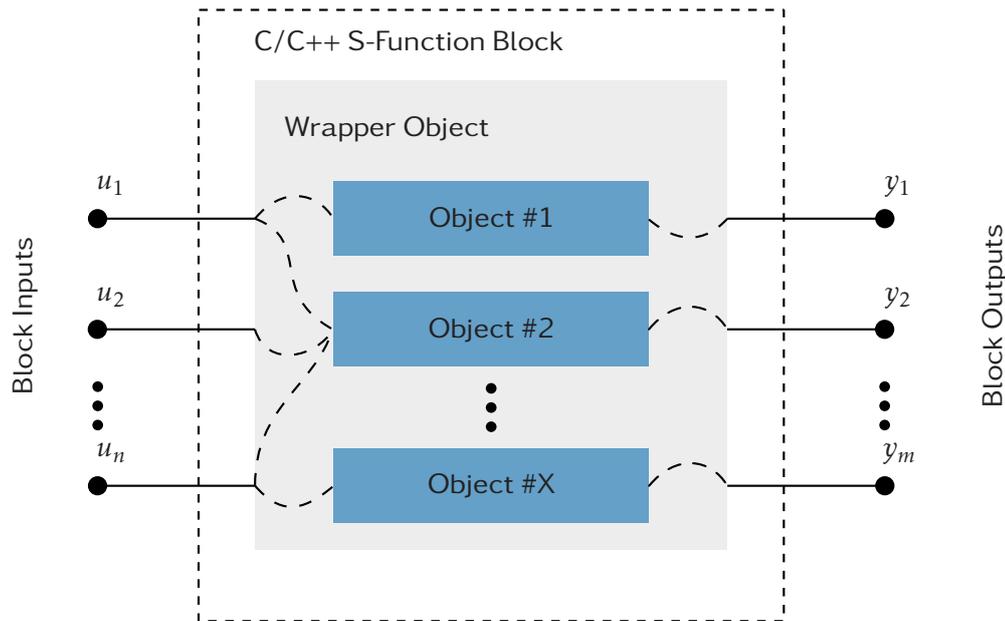


Figure 3.31: Integration of C/C++ code with Matlab/Simulink using S-Functions and wrapper objects.

simulation environment is required. For the simulator model application the input (green) and output (orange) elements of the model shown in Figure 3.29 are replaced by simulated elements (i.e. the controller in- and outputs are replaced by the interfaces to the dynamic simulation model).

3.6.3 Host Interface

Besides the manipulator, in the interdisciplinary CROPS project many different hard- and software components, like sensors, grippers and perception algorithms must be merged together. Depending on the usage of the robot system for one of the selective harvesting or the precision spraying tasks, problem-specific applications had to be implemented and specialized sensor hardware had to be applied, respectively. Therefore, a software framework which enables modular software development was required so that basic algorithms can be shared while specific hard- and software components are easily integrated. The Microsoft Robotics Developer Studio (MRDS) (Jackson [2007]) and ROS (Quigley et al. [2009]) have been considered as a basic framework for the software development. Due to its wide-spread application in the robotics research community (Cousins [2011]) and the support of many third party software libraries, the ROS framework has been selected. ROS is suitable for collaborative software development among several distributed groups. Applications, called *nodes*, are linked together with *messages*, based on a publisher/subscriber concept. Since these *messages* are making use of the TCP/IP protocol, the nodes can run on several distributed machines and communicate via a network connection. Besides simple *messages*, the implementation of *services*, which include a request and a response are also supported. The ROS middleware has been used to control, monitor and visualize manipulator motion and it pro-

vides a well-defined interface to the soft- and hardware components implemented by other project partners. In Figure 3.32, the nodes directly related to the interaction with the manipulator are illustrated. However, for clarity, not all messages

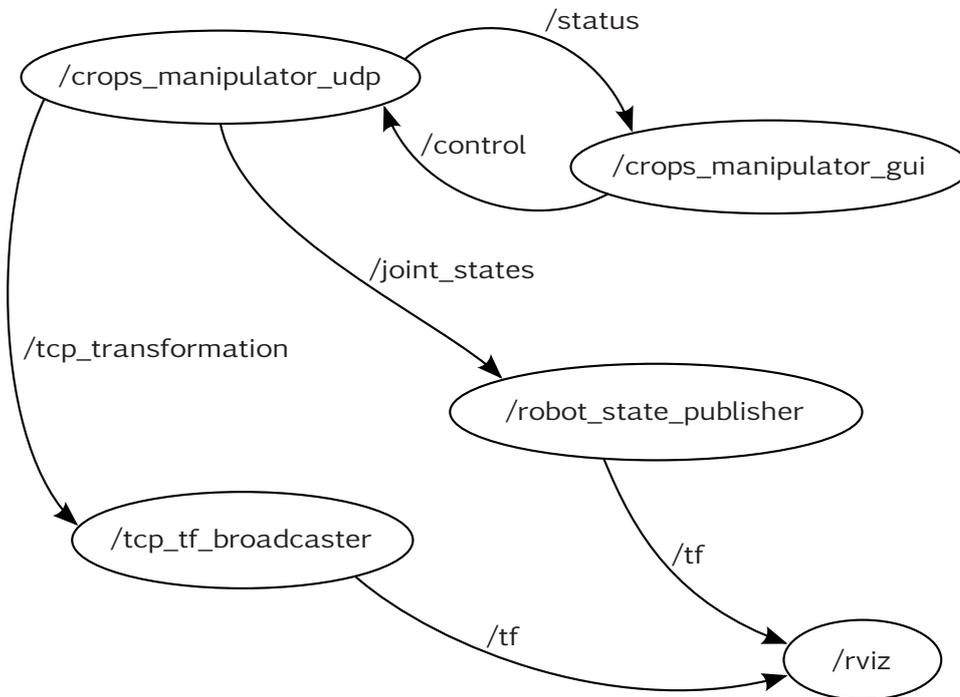
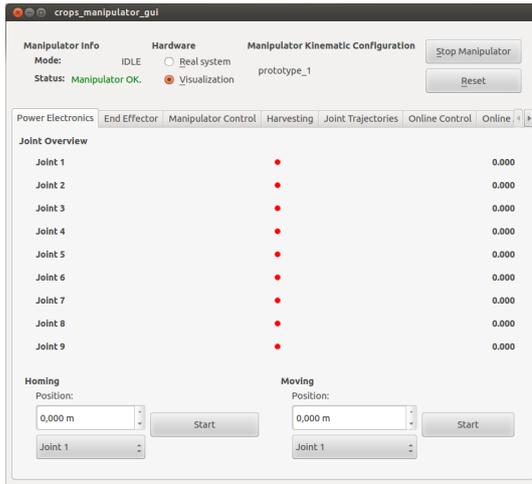
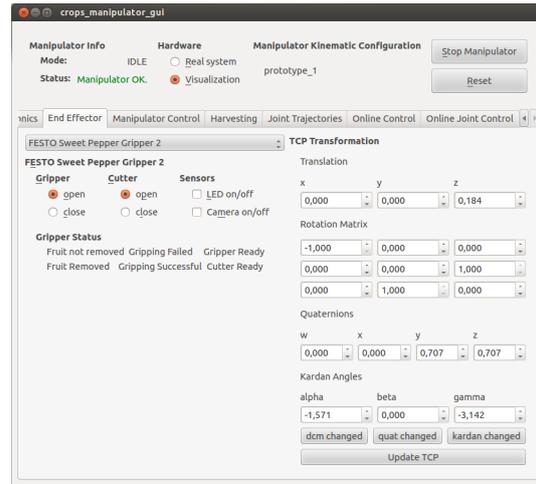


Figure 3.32: Nodes (ellipsoids) and messages of the ROS interface for the manipulators.

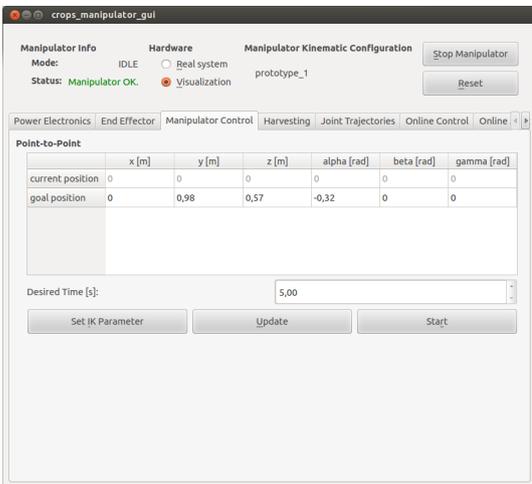
send and received by the applications are shown. The node `crops_manipulator_udp` is handling the communication with the manipulator application (cf. Figure 3.28) using the UDP protocol with custom defined message contents. Due to the importance of this node, a watchdog has been implemented, sending a signal in a regular interval to the real-time application and if the node crashes or a communication error occurs, the manipulator state machine switches into the FAULT state. The interface node constantly publishes sensor data, like joint states, and it accepts and forwards motion commands and monitors the status of the hardware and returns success or failure as feedback to requested tasks. Another important, but not required, application is the `crops_manipulator_gui` node. It is a Graphical User Interface (GUI) providing an interface to monitor and control most of the manipulator functions. Since the manipulator software was constantly developed and new functions were added over time, the GUI is organized in tabs so that it can easily be expanded while a clear structure is maintained. Figure 3.33 shows six out of several more developed tabs. Above the tab field, general information, like the current state of the manipulator is displayed. Furthermore, the robot can be stopped any time and a selection if the real system is to be used or a simulation should be performed can be set. While the tab illustrated in Figure 3.33a provides status information on the motor controller of each joint and basic functions to set-up the manipulator for operation are offered, the tab on the upper right hand side (cf. Figure 3.33b) is an interface to control and monitor the end-effectors as



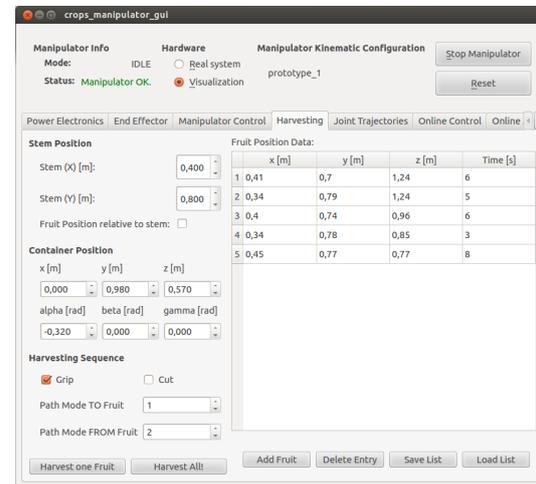
(a) Tab to display basic information and control of the joint controller.



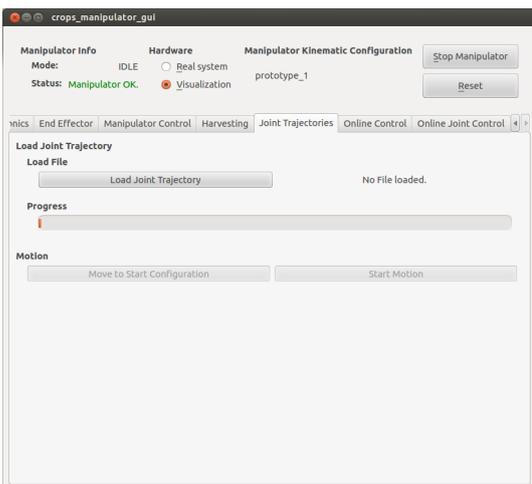
(b) Tab to set-up end-effector options.



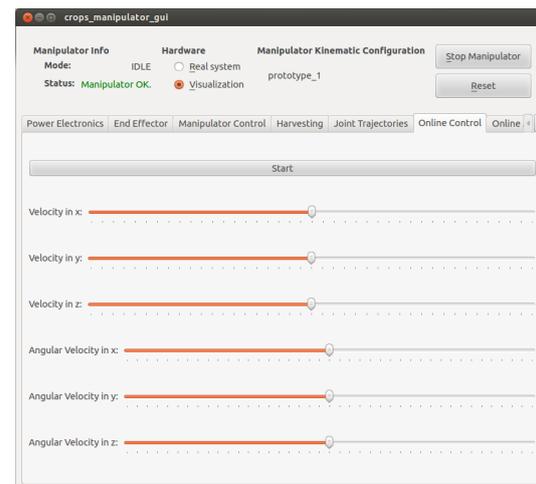
(c) Basic point-to-point motion tab.



(d) Tab for harvesting of a list of fruits.



(e) Offline joint trajectory tab.



(f) Tab to control the TCP velocity.

Figure 3.33: The manipulator GUI based on tabs supporting the modular software development.

well as set-up the geometrical transformation from the last robot link to the TCP. An interface for point-to-point motion on a straight line is shown in Figure 3.33c and a harvesting sequence according to a list of fruits can be performed with the tab depicted in Figure 3.33d. A text file with the supporting points of a joint trajectory can be uploaded via the tab illustrated in Figure 3.33e to the real-time control unit. This is very useful to playback offline generated or recorded joint trajectories. With the tab depicted in Figure 3.33f, the user can control the TCP velocity along Cartesian coordinates in the manipulator task space. The three linear and three rotatory velocities are entered via six sliders. With another tab, not shown here, the joint velocities of the manipulator are controlled.

Based on the joint states and a kinematic model of the manipulator, the ROS node *robot_state_publisher* is publishing the three dimensional pose of body fixed coordinate frames of robot links using a ROS standard tool to handle coordinate transformations, referred to as *tf*. The robot geometry and other parameter are stored in an Extensible Markup Language (XML) file, based on the Unified Robot Description Format (URDF). With the pose information of each link and the 3D model of each part, the robot can be visualized with the ROS viewer *rviz* (see Figure 3.34). Depending on the end-effector, the coordinate transformation between the last robot link and the TCP can be adapted during run-time of the manipulator application and the node *tcp_tf_broadcaster* is constantly updating the actual *tf* of the TCP frame and forwarding it to *rviz* for visualization.

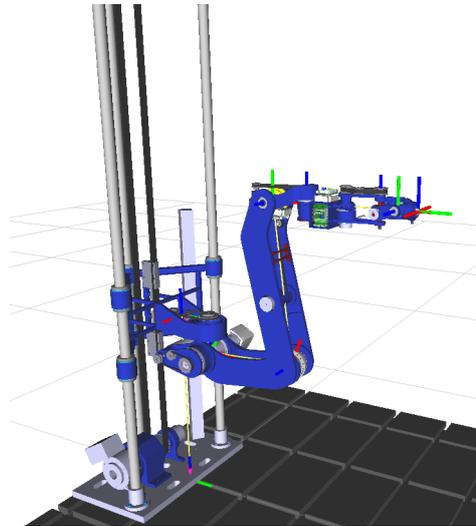


Figure 3.34: Visualization of the first manipulator prototype with ROS/*rviz*.

3.7 The Agricultural Manipulator Prototypes

Within this Section, the agricultural manipulators, which have been designed by JULIAN PFAFF and myself are presented. In Figure 3.35a the first manipulator prototype in the 9-DoFs configuration is shown. This prototype has been tested in field experiments for the sweet pepper, apples and grapes harvesting as well

as for the precision spraying application (cf. Chapter 6). Another version of this prototype (not shown) was built and used for software and hardware development in the laboratory of the AM. The final manipulator, assembled out of robot drive modules, has an increased robustness and modularity (see Figure 3.35b). It has been tested in sweet pepper harvesting experiments.



(a) First manipulator prototype.

(b) Final manipulator prototype.

Figure 3.35: The developed prototypes.

3.8 Chapter Summary

This Chapter presented the major steps, leading to the design of the modular agricultural manipulator prototypes. Starting from the description and evaluation of the manipulator kinematics in a 6-DoFs and 9-DoFs configuration for the first prototype and a 7-DoFs and 9-DoFs set-up for the final manipulator an actuation concept based on PAMs has been evaluated in detail. However, due to some disadvantages, an actuation of the robots with electrical actuators, i.e. BLDC motors, has been preferred. Afterwards, the modeling of the kinematics,

multibody dynamics, gear friction and the actuator dynamics were addressed. Complete models for all prototypes have been implemented and were applied during simulations to support the design. The basic electronics architecture, and its integration and communication has been addressed. Furthermore, the software framework with the real-time control application for the manipulator and its interface to the ROS middleware were discussed and described in this Chapter.

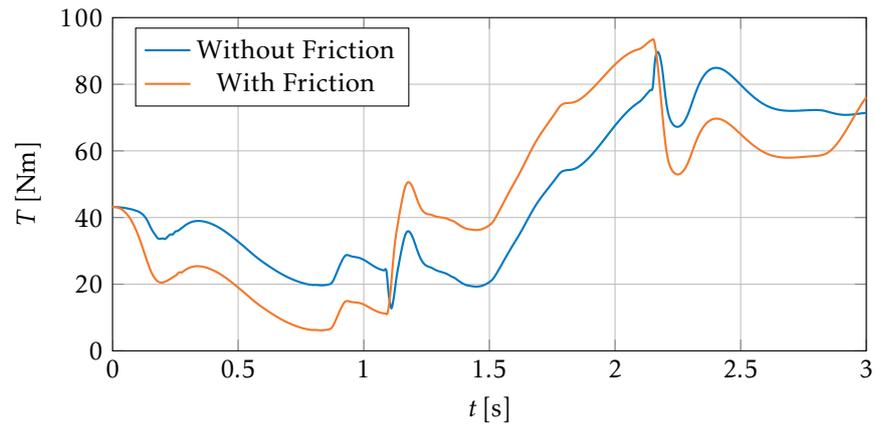
4 Modular Test Bed for Robot Drives

4.1 Introduction

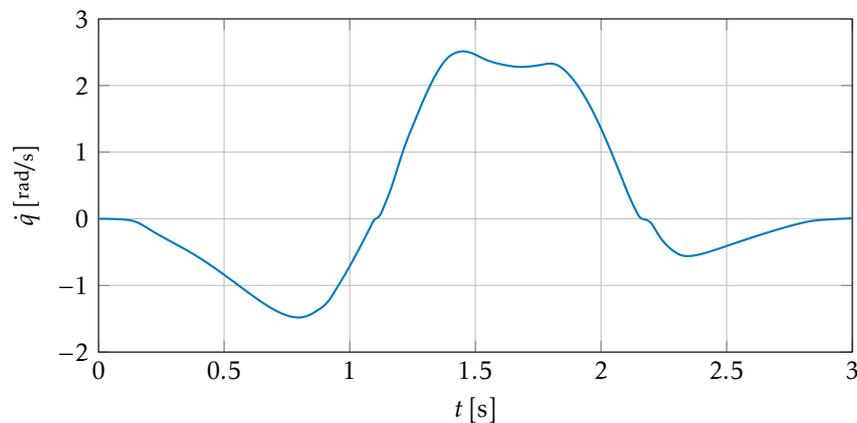
The drive mechanism of many robot joints are composed of an electrical actuator and a gear transmission. Besides multibody dynamics, actuator dynamics and gear elasticity, consideration of friction effects are of particular importance for an accurate modeling of the system. Therefore, a modular test bed has been designed for experimental parameter identification in robot drives and is described in this Chapter. The main objective of the test bed is the measurement of the friction torque at different operation conditions. With the test bed, robot drive modules designed for the final agricultural manipulator (cf. Figure 3.35*b*), as well as drive modules for the humanoid robot LOLA, designed by Lohmeier et al. [2009] were investigated. The drives are a combination of an electrical actuator and a gear transmission from the company Harmonic Drive with reduction ratios typically in the range of 50:1 up to 160:1.

Especially for robotic applications, the gear friction must be taken into account and modeled for an efficient system and control design. To illustrate the influence of friction on the actuation torque of robot joints, the results of a dynamics simulation for a typical joint trajectory of the final manipulator are shown in Figure 4.1. The torque T required for the joint actuation is plotted in Figure 4.1*a* with friction modeling and under the assumption of a frictionless joint. In Figure 4.1*b* the corresponding joint velocity \dot{q} is provided for reference. Depending on the load and speed of the joint, the torque difference with and without friction modeling is up to 50 % for this exemplary case. As already described in Section 3.4.2, at an early stage of the system development, the designer has to rely on modeling assumptions or on the specifications provided by the data sheet of the manufacturer. Of course, in many engineering applications and under conservative assumptions a successful design of the system components is possible. Nevertheless, for further improvement of the dynamic model and for evaluation of the designed drive models in general, parameter identification on the real system is required.

An overview on the test bed design and experimental results were already described in Baur et al. [2014a] and many Sections of this Chapter are based on this publication. However, not all experiments with the robot drive modules of the agricultural manipulator were carried out at that time. Besides, a detailed description of the test bed is given and main experimental results will be presented. In particular at the design, the selection of components and during the start-up, SEBASTIAN DENDORFER contributed to the realization of the test bed in the course of his student thesis.



(a) Generalized torque of the final manipulator (joint 3) with and without considering the influence of friction (the friction model is based on catalog data).



(b) Corresponding joint speed (slow side).

Figure 4.1: Simulation result displaying the influence of friction in robot drives.

4.2 Related Work

For a long time now, friction phenomena in machinery were studied and analyzed by many researchers. In nearly all engineering applications, friction plays a more or less significant role and accurate models can be of great benefit. However, especially for Harmonic Drive gears, due to the nonlinear characteristics and many factors involved, modeling of friction solely based on catalog data or simple experiments will not lead to accurate results (Tuttle and Seering [1996]). Many set-ups to experimentally identify friction effects in bearings are reported in literature (for example refer to Harnoy et al. [2008, 1994]). According to Seyfferth et al. [1995] or Tuttle and Seering [1996], the basic test set-up for friction identification in Harmonic Drive gears consists of an electrical actuator connected to the input (fast) side of the gear while it is possible to fixate the output shaft (slow side). A test bed for unrestrained motion experiments with an inertia load on the output side was reported in Taghirad and Belanger [1996, 1998]. This set-up, however, does not allow to investigate arbitrary load cases. Although, experimental identification on

single components has many advantages, in many cases, an identification on single drive units or gears is impossible (e.g. when working with a bought industrial manipulator). In such a case, the parameter identification must be performed on the complete system (Hamon et al. [2010]; Kennedy and Desai [2005]). For the most part, the friction losses in the investigated robot drive units occur in the built-in Harmonic Drive gears (refer to Figure 4.2 for an overview of the gear's structure). So they are of particular interest for friction modeling. According to Taghirad and Belanger [1998], the tooth-meshing of the circular and flexspline is responsible for a large part of friction loss in the gear. Other losses, due to viscous damping, occur in the bearings of the wave generator and the flexspline. Additionally, the structural damping of the deformable flexspline is an important factor. In Tuttle and Seering [1996] a detailed analysis of Harmonic Drives is given with models considering position dependent friction, as well as influences of resonance vibrations, due to high torques in the tooth-meshing. More details on Harmonic Drive modeling and parameter identification based on experiments were carried out by Tuttle [1992].

4.3 Dynamic Modeling of Robot Drives

For the friction modeling of the robot drive units, the Harmonic Drive gear is of particular importance. In Figure 4.2 a model of the gear with its major components is illustrated. The circular spline ① is a solid steel ring with internal teeth, the flexspline ② is a deformable steel cylinder with external teeth and the wave generator ③ is a ball bearing with an elliptical shape on the outside (Har [2011]). There are several ways to assemble the gears. In the here investigated robot

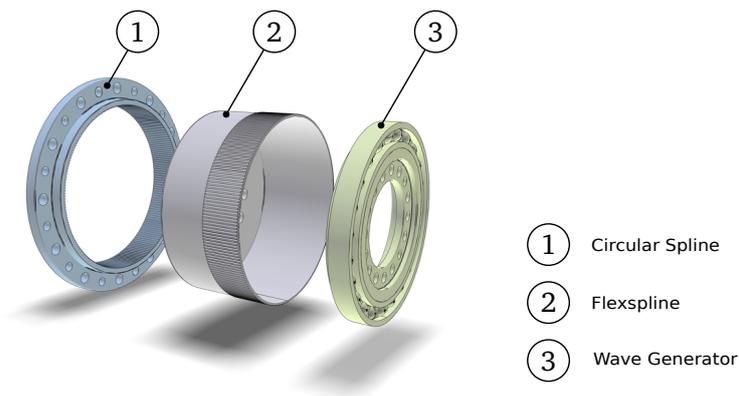


Figure 4.2: Major components of Harmonic Drive gears.

drive units the gear is used to reduce the motor shaft speed while increasing the actuator's torque. Therefore, the circular spline is connected to the housing on the input side (fast side) and the flexspline is connected to the shaft of the output side (slow side), while the motor shaft is driving the wave generator. For more details on the mechanical assembly refer to Lohmeier [2010] (humanoid LOLA) or Pfaff et al. [2014] (CROPS manipulator). Based on the lumped component

model of relevant mechanic elements (cf. Figure 4.3) a model to describe the module dynamics can be derived. The gear's inertia is added to the rotational

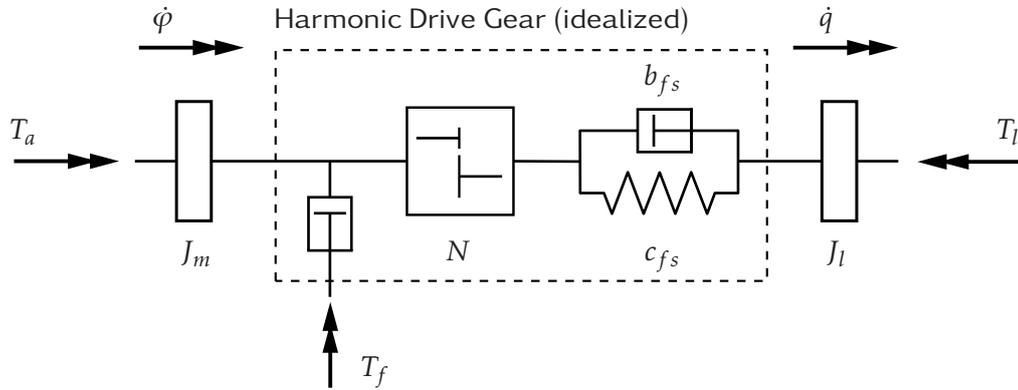


Figure 4.3: Schematic diagram of the module mechanics.

inertia J_m of the motor shaft. The speed of the actuator is depicted by $\dot{\phi}$ and the motor torque T_a is acting on the input shaft. The friction loss of the bearings on the in- and output side is added to the gear's friction. The Harmonic Drive gear is described with a nonlinear friction term T_f , the transmission ratio N and a (nonlinear) stiffness c_{fs} and damping b_{fs} on the gear's output side. The load position and velocity is described by q and \dot{q} and it has the inertia J_l while the load torque T_l is applied on the output side. According to the free-body diagram in Figure 4.4, the angular momentum theorem can be applied and the equations

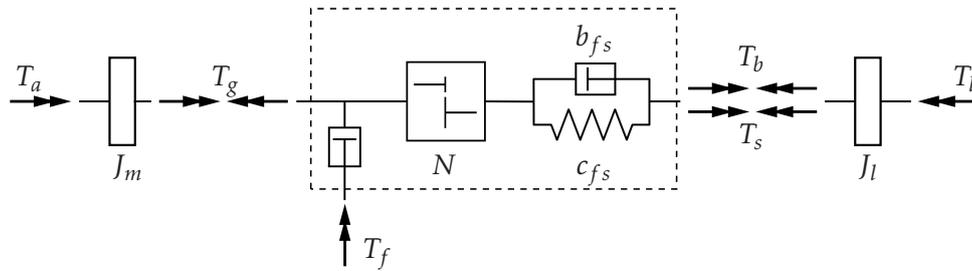


Figure 4.4: Free-body diagram of the lumped element model of the robot drive.

of motion for the system are easily derived with the torque T_g acting on the gear's input shaft:

$$\ddot{\phi} = \frac{1}{J_m} (T_a + T_g) \quad (4.1a)$$

$$\ddot{q} = \frac{1}{J_l} (-T_l - T_s - T_b) \quad (4.1b)$$

$$T_g = T_f + \frac{1}{N} (T_s + T_b) \quad (4.1c)$$

The elasticity and damping torques T_s and T_b of the gear can be modeled as follows:

$$T_s = \left(q - \frac{\varphi}{N} \right) c_{fs} \quad (4.2a)$$

$$T_b = \left(\dot{q} - \frac{\dot{\varphi}}{N} \right) b_{fs} \quad (4.2b)$$

4.4 Test Procedure and Requirements

In this Section, the test procedure to identify the desired parameter will be described. Mainly based on that, the requirements for the test bed are derived.

4.4.1 Test Procedure

The purpose of the experiments is the identification of the characteristic curve of the drive modules frictional behavior. Besides temperature, major influences on the friction are the velocity and the load torque. The temperature influence is not investigated by the experiments performed on the test bed. However, a temperature sensor is attached to the housing of the modules and all experiments were carried out after a warm-up period. For the friction identification, the measurements were conducted with varying angular velocities and load torques at the unbounded system at steady-state velocity and constant load torque. With the equations (4.1), a torque balance at the input (motor shaft) and the gear's output shaft, the friction torque is obtained to:

$$T_f = \frac{1}{N} T_l - T_a \quad (4.3)$$

Note, that the friction is directly measured by the effective motor current and the load torque. The drive modules efficiency η is calculated by:

$$\eta = \frac{T_l}{N T_a} \quad (4.4)$$

Based on (4.3) or (4.4), the main quantities for the measurements are identified. This leads, amongst other, to a set of major system requirements on the test bed, which are described in the next Section.

4.4.2 Requirements and Specifications

For proper parameter identification, in particular identification of friction effects, the following quantities must be measured or estimated:

1. The torque of the actuator T_a on the input shaft.
2. The position φ and velocity $\dot{\varphi}$ of the input shaft.
3. The load torque T_l acting on the output shaft.

4. The position q and velocity \dot{q} of the output shaft.

Further requirements are:

5. It must be possible to change the load torque T_l in a close range from values close to zero up to the nominal torque of the most powerful investigated module.
6. It must be possible to reach the maximum velocity of the fastest module for any load case.
7. Experiments must be conducted without modifying the modules.
8. Modular design to allow for testing of different drive modules.

For a direct measurement of the input torque, the motor or wave generator shaft must be modified. This is no option, since this would change the drive module's characteristics. Instead, the effective current I is measured by the motor controller and the actuating torque T_a is calculated using the torque constant from the data sheet provided by the manufacturer, based on the following equation:

$$T_a = k_m I \quad (4.5)$$

For operation of the motor below the nominal torque, (4.5) is a reasonable assumption. Since the torque constant is not experimentally verified, one must assume that the computation of T_a is imperfect. For control purposes, however, the actuating variable is the effective current and thus the aforementioned inaccuracy is compensated. The position of the input shaft is measured with the incremental encoder of the modules. Note, that the velocity is estimated based on the position measurement. The estimation algorithm is implemented on the motor controller and unknown to the author. However, since the encoder resolution is very high and quasi-static point of operations were investigated, the velocity estimation is assumed to be sufficiently accurate for the identification experiments.

4.5 Design and System Architecture

With the requirements described in Section 4.4.2 the test bed was designed and a schematic side view of the result is shown in Figure 4.5. Important design steps will be explained in the following. One of the major decision on the test bed design has to be made on the method to apply and measure (or estimate) the load torque. Two possibilities seemed to be the obvious choice. The first one is the application of an electrical actuator and the second one is the usage of a brake. In any case, an accurate measurement of the load torque is important. Therefore, a torque sensor ③ from the company BURSTER with two calibrated measuring ranges from 0 - 20 Nm and from 20 - 200 Nm was chosen to ensure accurate results at low and high loads. With a properly designed motor, the load torque could be controlled in a wide range and varying loads could be applied. However, torque ripples might be an issue at lower load torques, while the overall system

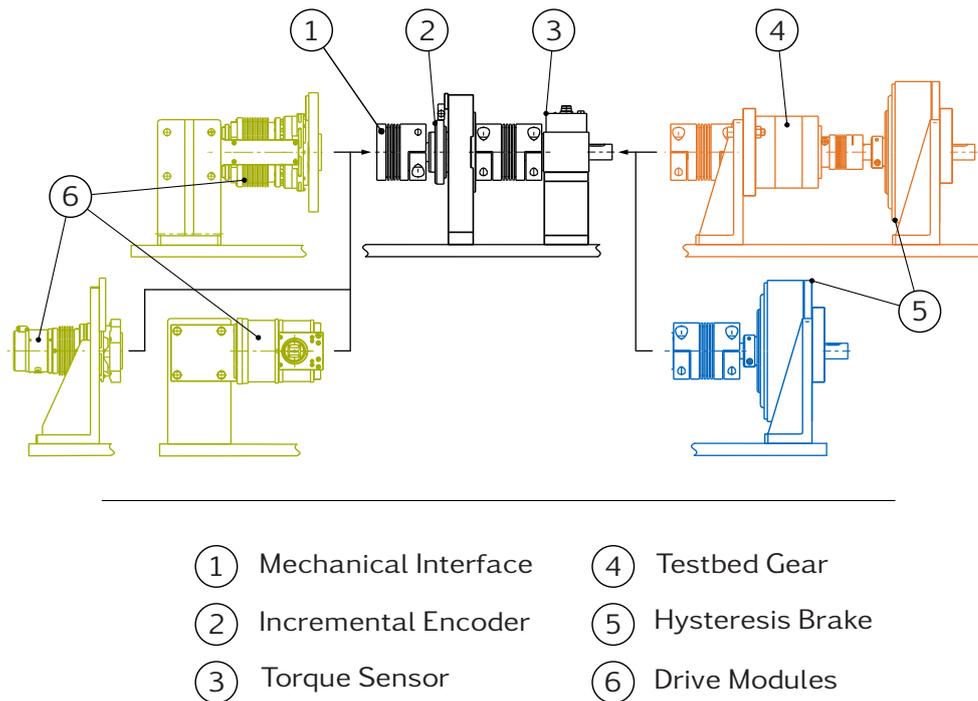


Figure 4.5: Schematic overview of the test bed and its components. Two different configurations for the investigation of high load torques (indicated in orange) or low load torques (blue) are possible.

design is more complex and more expensive compared to the application of brakes. Therefore, it was decided to use a current controlled hysteresis brake (⑤) to enable different load cases. Due to the working principle of these kind of brakes, the torque is independent of the angular velocity and, similar to electrical actuators, the torque is applied contact-free. So the lowest possible torque is only limited by the damping in the bearings of the brake. Furthermore, the brake torque can be continuously adjusted. The control of the brake is rather simple, especially reversing speeds are handled without further efforts. A brake from the company MOBAC with a minimum torque of about 0.092 Nm and a maximum torque of 17 Nm was selected. To cover the complete range of desired load torques, the test bed can be assembled in two different configurations. In the first configuration, for the low torque range, the brake is directly connected to the torque sensor. Without the gear transmission (④), the back driving torque of the gear (i.e. the minimal required torque to move the gear from the slow side with no load on the faster side of the gear) is disconnected from the system. This allows the examination of very low load torques across speeds in different directions. In the second configuration the test bed gear transmission can be added between the brake and the torque sensor. The planetary test bed gear is from the company NEUGART with a transmission ratio of 5:1. This allows for experiments with load torques up to 85 Nm. Finally, an incremental encoder is added to the test bed (②) to measure the position of the output shaft. The components so far can be lined up on a test bed shaft, which is to be connected to the drive module's output shaft. To allow for the required modularity of testing different modules a flange

is provided. Combined with a grooved panel, as basis for the test bed, it is easy to design various adaptations, each suitable for a specific test module, respectively. In Table 4.1 the specifications of the major system components are summarized. The motor controller for the drives are from the company ELMO MOTION CONTROL

Table 4.1: Major Specifications of the test bed components.

Component	Manufacturer (classification)	Specification
Incremental Encoder	ASM (PMIR5/PMIS4)	131072 ticks / rev.
Torque Sensor	Burster (8661-5200-V1202)	Meas. Range: 0-20 Nm 0-200 Nm
Brake	Mobac (HB-1750M-2DS)	Brake Torque: 0-15 Nm*
Gear	Neugart (PLE-120-5-OP01)	Ratio: 5

*Possible dissipation is 350 W continuously and 2400 W non continuously.

and have a CAN bus communication interface. The brake torque is controlled using standard PI-feedback including an anti-wind up element. The current of the hysteresis brake is the manipulated variable, while the torque sensor measurement is the input. Figure 4.6 shows the result of the brake control without the test bed gear for desired torque step commands at a constant rotational velocity. The

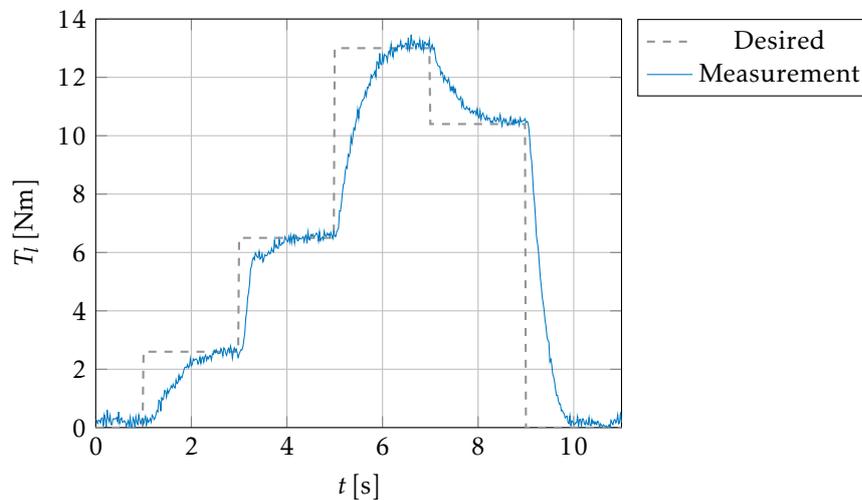


Figure 4.6: Desired load torque on the test bed at 10rpm without the test bed gear.

brake control is very stable for different velocities. The settling time of around 1 s is maintainable for the application. The data acquisition and the CAN bus communication is handled by a dSPACE board (ds1103). The software for the test bed is implemented in MATLAB/SIMULINK. Figure 4.7 shows a picture of the designed test bed.

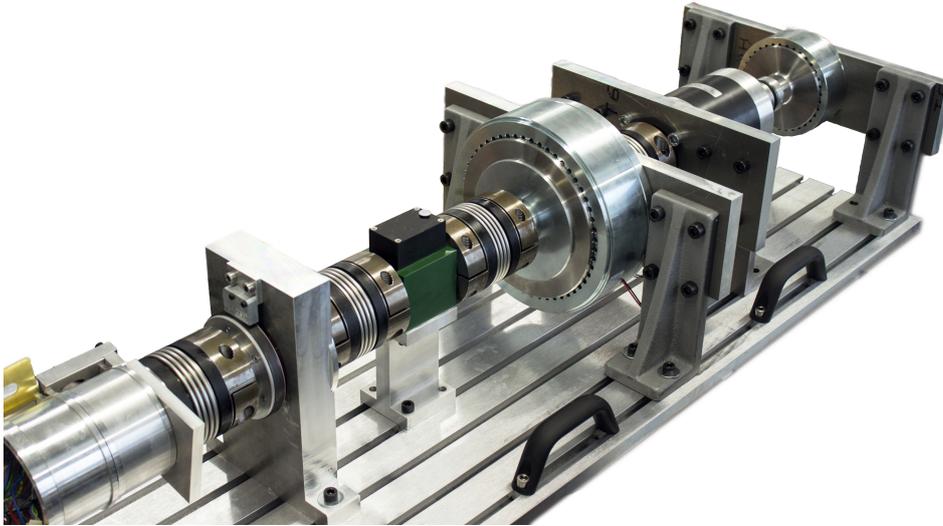


Figure 4.7: Picture of the modular test bed.

4.6 Experiments and Results

Measurements for friction identification were performed on two different module types (A and B), a test module for the final manipulator (B.0), as well as for two joints of the humanoid robot LOLA (hip adduction and flexion). The naming and main specifications of the modules used for the final CROPS manipulator prototype are summarized in Table A.2 of the Appendix. Figure 4.8 illustrates each type of the investigated modules. Note that several modules of each type were built.



Figure 4.8: Investigated robot drive modules.

When the modules were placed in the test bed and the set-up was successfully tested, the identification was performed automatically after a warm-up phase. With the chosen approach, quasi-static points of operations, i.e. discrete combinations of speed and load torque, were investigated. Each point of operation is held for several seconds so that the quasi-static state is obtained. In the automatic mode, a constant drive speed is set while the load torque is subsequently increased

until a maximum value is reached. Then the next speed is set automatically and the process is repeated until the maximum speed and load torque is reached. The data of the test series is saved and analyzed afterwards. The major steps of the data analysis are described in the following. First of all, the measurements of the effective current I , the load torque T_l and the velocity $\dot{\phi}$ are filtered by a low-pass. The time windows of quasi-static states are detected by comparison of the actual values with the desired values of speeds and load torques, respectively. For each data point in the quasi-static state, the friction T_f and the efficiency η is determined via (4.3) and (4.4). Finally, the average values and its standard deviations of the speed, efficiency and friction torque for the quasi-static time windows are calculated. In the final manipulator prototype, three modules of type A were installed (see Table A.2). Notably, for this drive module type, gears with a different reduction ratio were used. The module A.4 is assembled with the Harmonic Drive gear CPL-32-50 (reduction ratio: 50) and the modules A.2 and A.3 with the gear CPL-32-100 (reduction ratio: 100). The key figures of the experiments performed with the drive modules of type A are summarized in the Table 4.2. The average temperature of the housing after the warm-up phase was

Table 4.2: Summary of the measurements performed on the modules of type A with the range of the nominal load torque T_l and the nominal speed $\dot{\phi}$.

Module	$\dot{\phi}$ [rpm]		T_l [Nm]		No. of test series
	min	max	min	max	
A.3	100	1000	19.5	45.5	4
A.3	500	3500	13	45.5	6
A.4	100	1000	6.5	40	3
A.4	500	3500	6.5	40	3

around 40 °C. Two modules of type B are built-in the final manipulator prototype and both of them were investigated on the test bed. Table 4.3 gives an overview on the conducted measurements and main parameter are listed. The average temper-

Table 4.3: Summary of the measurements performed on the modules of type B with the range of the nominal load torque T_l and the nominal speed $\dot{\phi}$.

Module	$\dot{\phi}$ [rpm]		T_l [Nm]		No. of test series
	min	max	min	max	
B.0	500	3500	1.3	11.7	2
B.1	500	3500	6.5	17.5	3
B.1	100	1000	6.5	17.5	3
B.2	500	3500	6.5	17	3
B.2	100	1000	6.5	17	3

ature of the housing during the experiments on the modules of type B was around 30 °C. Significant experimental results for the robot drive modules A.3, A.4 and B.1 will be presented in the upcoming Sections. Results of measurements on the

modules B.0, B.2 as well as the measurements on the modules for the humanoid robot LOLA are provided in the Appendix D. The focus of the presented results lies on the following two points:

1. Friction torque measurements and a comparison of two separate test series of the modules. Furthermore, the curve fit of (3.30) and (3.33) to the friction torque measurements is shown and discussed.
2. A comparison of the efficiency measurement of the modules with the efficiency based on catalog data of the gear manufacturer.

4.6.1 Friction Torque Measurements

This Section presents the average values of the friction torque measurements of the quasi-static points of operation. For the investigated modules, a comparison of two test series performed on the same module is shown to illustrate and discuss the repeatability of the experiment. Additionally, the resulting curve fits of the friction laws, proposed in Section 3.4.3, are provided. In Figure 4.9 and Figure 4.10 measurements of the friction torque plotted against the speed of the motor shaft for three different load torques with two separate test series of the module A.3 and A.4 are shown in each case. The error bars indicate the standard deviation of the

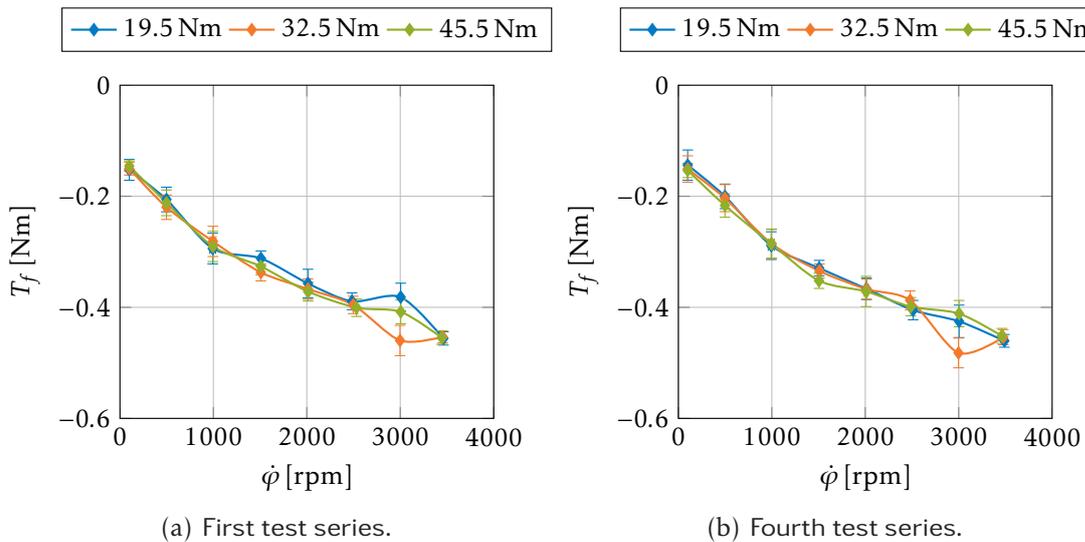


Figure 4.9: Friction torque measurements plotted against the input speed and varying load torques of the robot drive module A.3. Two separate series of measurements are shown to discuss the repeatability of the experiment.

mean values. Since both measurements in the respective Figures illustrate very similar results the repeatability of the experiment is discernible. According to the catalog data, the efficiency of the gear in module A.3 (CPL-32-100, cf. Figure C.3c) is in general slightly less than the efficiency of the gear in module A.4 (CPL-32-50, cf. Figure C.3b). However, for states with comparable loads and speeds the friction torque of the module A.4 (cf. Figure 4.10) has significantly higher absolute

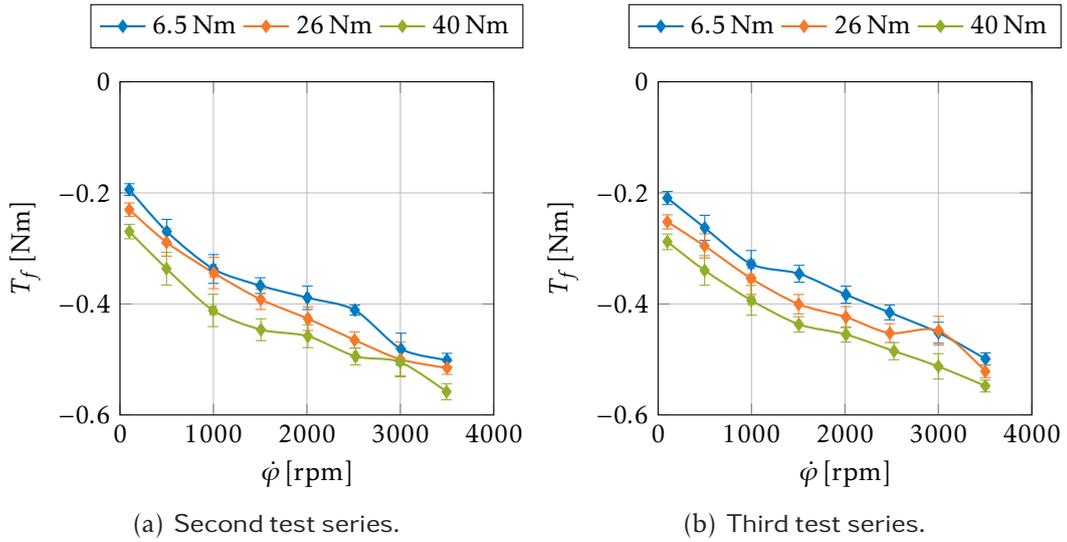


Figure 4.10: Friction torque measurements plotted against the input speed and varying load torques of the robot drive module A.4. Two separate series of measurements are shown to discuss the repeatability of the experiment.

values than the friction occurring in the module A.3 (cf. Figure 4.9). In agreement with the conservation of energy, this is an expected result since the reduction ratio of the gear in module A.4 compared to module A.3 is different by the factor two, while the gear's efficiency according to catalog data only has a rather small difference.

In Figure 4.11 the results of the friction measurements are plotted against the drive speeds for the three different load torques applied to the module B.1. Again, for a discussion on the repeatability of the experiments, test series 1 (cf.

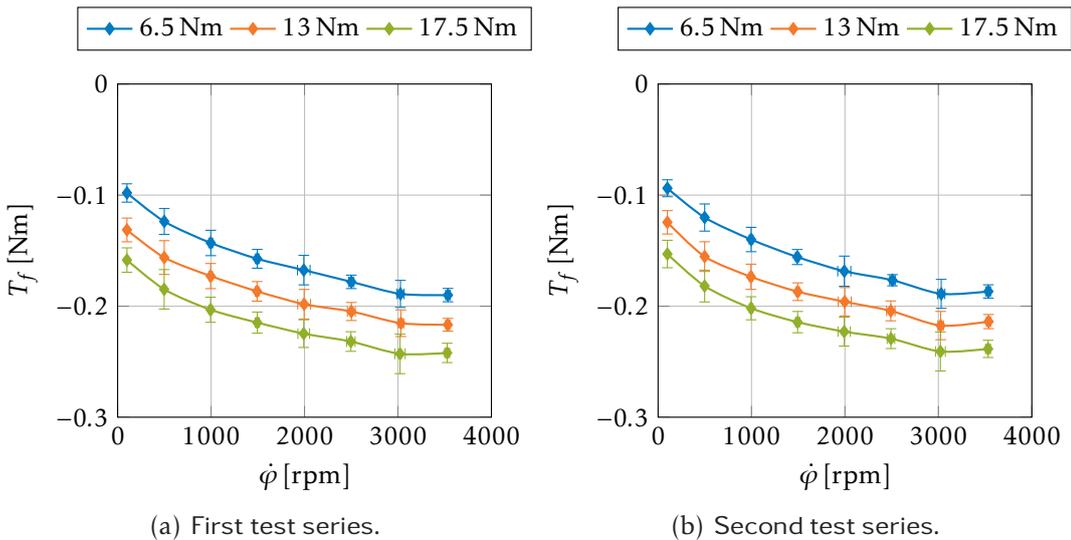


Figure 4.11: Friction torque measurements plotted against the input speed and varying load torques of the robot drive module B.1. Two separate series of measurements are shown to illustrate the repeatability of the experiment.

Figure 4.11a) is compared with the test series 2 (Figure 4.11b). The two measurements display the same behavior and the friction torque results are within the expected accuracy indicated by the standard deviation. Therefore, only one measurement will be used for the further evaluation of the experiment. As expected, the friction torque's absolute values of all investigated modules is increasing with higher speeds. However, for the module A.3 the variations of the load seem to have only a very small effect on the friction (see Figure 4.9). As the experiments with the other modules demonstrated, this behavior is an exceptional case for the modules investigated in this study (see for example Figure 4.10 or Figure 4.11). The friction torques of the drive modules of type A compared to the modules of type B for similar speeds and loads are higher. This is a reasonable result, since the gear's size of the modules B have smaller dimensions than the gears in the modules of type A.

For an improvement of the initial friction model that is based on the curve fit to catalog data, the friction laws (3.30) and (3.33) are fitted to the measurements. This was achieved by solving the constraint quadratic optimization problem (3.31) using the results of the efficiency measurements instead of the catalog data with the MATLAB function `FMINCON` and the Sequential Quadratic Programming (SQP) algorithm. Since the measurements indicate a nonlinear characteristic of the friction torque with respect to the motor angular velocity, the friction model based on (3.33) fits the measurements more accurate than model (3.30). The curve fitting results of the friction laws with a comparison to the friction torque measurements are shown in Figure 4.12, Figure 4.13 and Figure 4.14 for the modules A.3, A.4 and B.1. For clarity only two load torque variations are shown

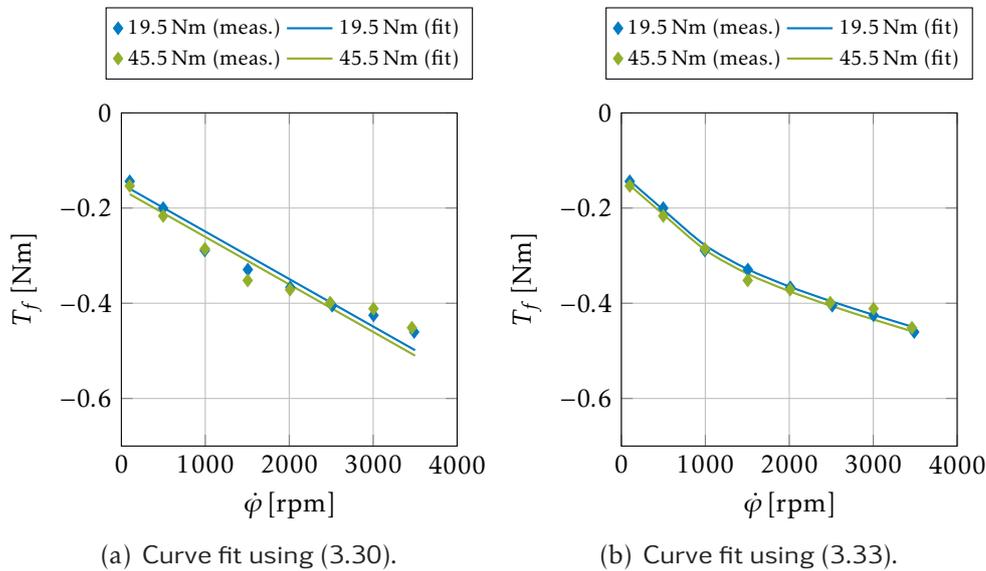


Figure 4.12: Curve fit to the measurements (fourth test series) on module A.3.

in the curve fitting of module A.3 in Figure 4.12. Clearly, for all cases the friction law (3.33) provides a more accurate fit, compared to the fit of the friction law (3.30). Table 4.4 summarizes the parameter identification results of the friction law according to (3.33) while Table 4.5 presents the results from the curve fit

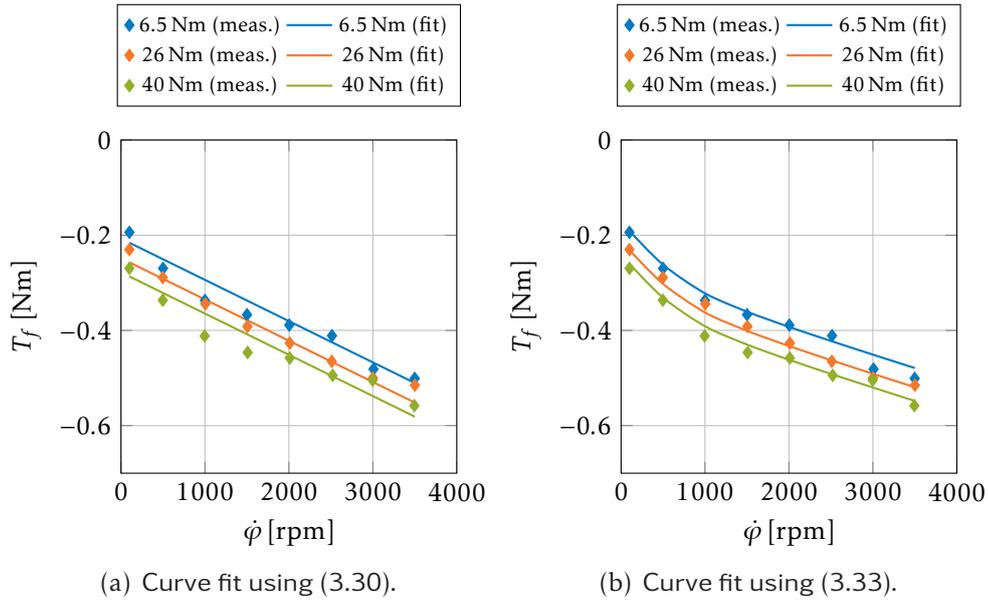


Figure 4.13: Curve fit to the measurements (second test series) on module A.4.

using (3.30). Additionally, the Root Mean Square Error (RMSE) is given and will be discussed later in this Section. The parameters in both Tables are fitted to the test series shown in the Figure 4.12, 4.13, and 4.14. For all investigated modules, the curve fit with the model (3.30) revealed that the parameter γ is estimated to zero which is also recognizable by the parallel straight lines of the fitting results in Figure 4.12a, Figure 4.13a and Figure 4.14a. This means that the corresponding term (coupled load and speed dependency) is not represented by the measurements, although this term has improved the curve fit to catalog data.

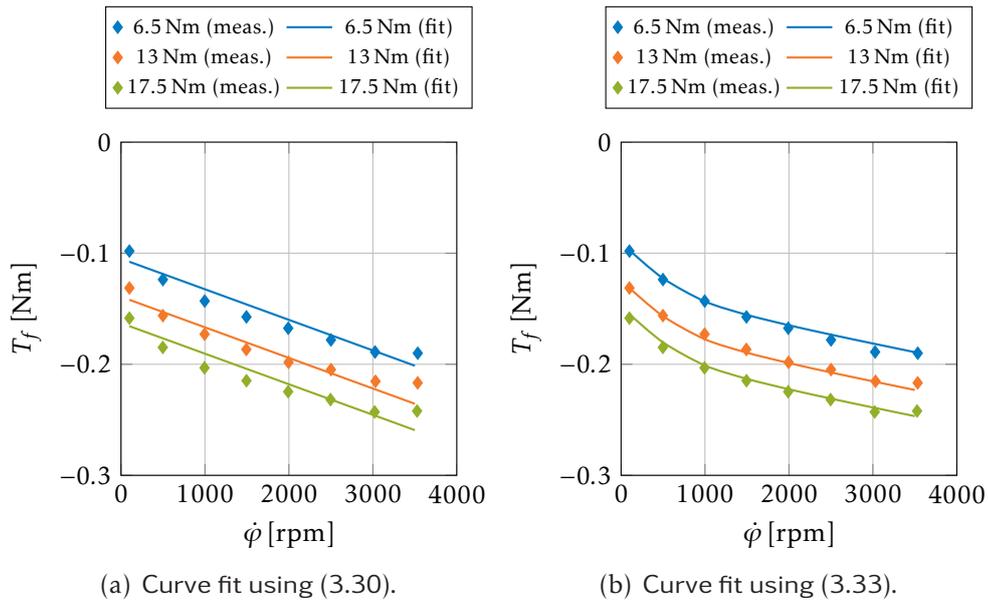


Figure 4.14: Curve fit to the measurements (first test series) on module B.1.

Table 4.4: Parameter estimation result of the friction model (3.33) based on the efficiency measurements for the modules of type A and B. The fitting results correspond to the Figure 4.12b, 4.13b, and 4.14b.

Module	$\bar{T}_{f,0}$ [Nm]	$\bar{\mu}$ [-]	\bar{b} [Nm s/rad]	$T_{f,s}$ [Nm]	$\dot{\varphi}_S$ [rad/s]	RMSE
A.3	$2.79 \cdot 10^{-1}$	$3.75 \cdot 10^{-4}$	$4.68 \cdot 10^{-4}$	$1.28 \cdot 10^{-1}$	$8.17 \cdot 10^1$	$1.27 \cdot 10^{-2}$
A.4	$2.76 \cdot 10^{-1}$	$2.10 \cdot 10^{-3}$	$5.24 \cdot 10^{-4}$	$1.68 \cdot 10^{-1}$	$5.33 \cdot 10^1$	$1.23 \cdot 10^{-2}$
B.0	$1.12 \cdot 10^{-1}$	$4.40 \cdot 10^{-3}$	$9.06 \cdot 10^{-5}$	$5.70 \cdot 10^{-2}$	$9.99 \cdot 10^1$	$3.00 \cdot 10^{-3}$
B.1	$1.04 \cdot 10^{-1}$	$5.20 \cdot 10^{-3}$	$1.42 \cdot 10^{-4}$	$6.03 \cdot 10^{-2}$	$5.45 \cdot 10^1$	$3.30 \cdot 10^{-3}$
B.2	$1.36 \cdot 10^{-1}$	$4.40 \cdot 10^{-3}$	$1.95 \cdot 10^{-5}$	$4.77 \cdot 10^{-2}$	$6.52 \cdot 10^1$	$3.20 \cdot 10^{-3}$

Furthermore, for quantitative evaluation and for comparison of the fit based on the aforementioned friction laws, the RMSE is provided in the Table 4.4 and Table 4.5. For the N data points used for the curve fit, the RMSE is defined by:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_{f,\text{model},i} - T_{f,\text{measurement},i})^2} \quad (4.6)$$

When comparing the RMSE in the Table 4.4 as well as Table 4.5, the qualitative observation, that the fitting to measurement data with (3.33) is more accurate than the fitting with (3.30) is clearly confirmed. Depending on the module, the RMSE of the fit with (3.30) compared to (3.33) is reduced by a factor in the range from around two to five.

Finally, a comparison of the friction torque measurements of the two modules B.1 and B.2 is shown in Figure 4.15. Although the components in both modules are identical, the absolute values of the friction torques in the module B.1 are higher than the friction in module B.2. Therefore, this clearly indicates that it makes sense to perform measurements and experimental parameter identification on each module, even though they are identically designed, in order to obtain accurate friction models.

Table 4.5: Parameter estimation result of the friction model (3.30) based on the efficiency measurements for the modules of type A and B. The fitting results correspond to the Figure 4.12a, 4.13a, and 4.14a.

Module	$T_{f,0}$ [Nm]	μ [-]	b [Nm s/rad]	γ [s/rad]	RMSE
A.3	$1.41 \cdot 10^{-1}$	$4.36 \cdot 10^{-4}$	$9.53 \cdot 10^{-4}$	0	$2.79 \cdot 10^{-2}$
A.4	$1.93 \cdot 10^{-1}$	$2.10 \cdot 10^{-3}$	$8.28 \cdot 10^{-4}$	0	$2.34 \cdot 10^{-2}$
B.0	$6.70 \cdot 10^{-2}$	$4.40 \cdot 10^{-3}$	$2.26 \cdot 10^{-4}$	0	$6.40 \cdot 10^{-3}$
B.1	$7.07 \cdot 10^{-2}$	$5.30 \cdot 10^{-3}$	$2.63 \cdot 10^{-4}$	0	$8.90 \cdot 10^{-3}$
B.2	$6.31 \cdot 10^{-2}$	$4.50 \cdot 10^{-3}$	$2.75 \cdot 10^{-4}$	0	$1.59 \cdot 10^{-2}$

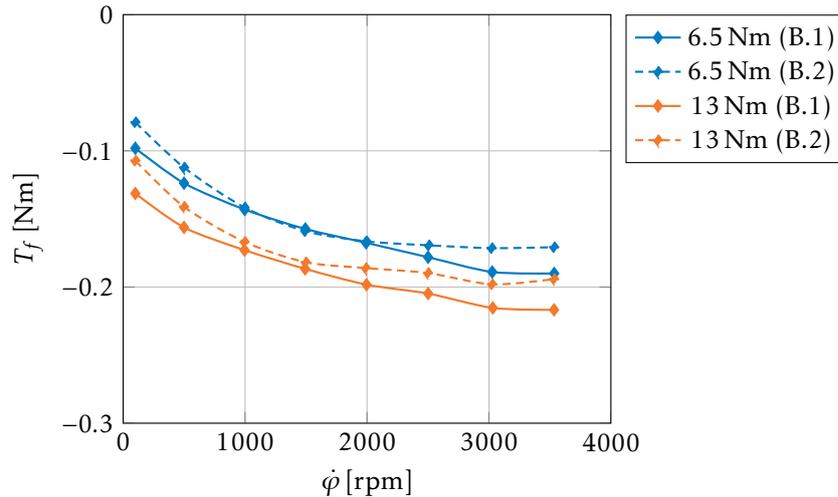


Figure 4.15: Comparison of the friction torque measurements (test series 1) of the two modules B.1 and B.2.

4.6.2 Efficiency Measurements

In this Subsection, the average values of the efficiency plotted against the load torque measurements for the input side velocities of 500 rpm and 3500 rpm, calculated according to (4.4) for the modules A.3, A.4 and B.1 are illustrated and discussed. The efficiency according to the catalog data of the gear which is built-in the module is provided in each efficiency plot. Note that a quantitative comparison is inadmissible out of the following reasons. Firstly, the average temperature during the experimental investigation was not the same as the temperature corresponding to the efficiency values provided by the catalog data of the gear manufacturer. Although the efficiency values for a certain temperature range are provided in the catalog, the process of the temperature measurement is unknown to the author of this thesis, but it surely can not be compared to the temperature measurement during the experiments with the modular test bed, since the temperature was measured at a certain point on the housing of the robot module and not on the gear itself. Secondly, since the input torque is not measured directly, but calculated based on the torque constant provided by the motor manufacturer, an unavoidable error in the efficiency computation is introduced. And thirdly, as the efficiency measurements on the modules take into account losses on all relevant components (e.g. bearings of the motor shaft) a comparison with the efficiency of the gear alone is questionable. Nevertheless, as a reference and for a qualitative discussion on the experimental results, the gear's efficiency according to catalog data can be used because the gear is the main source for friction loss. Figure 4.16 shows the efficiency plot of the module A.3 with the efficiency plot of the gear CPL-32-100 while Figure 4.17 provides the corresponding efficiency plot of module A.4 and gear CPL-32-50. Since the efficiency measurements of the modules take into account all losses in the robot module one would expect the measured efficiency to be below the efficiency of the gear alone. However, the efficiency in Figure 4.16 for 500 rpm is above the

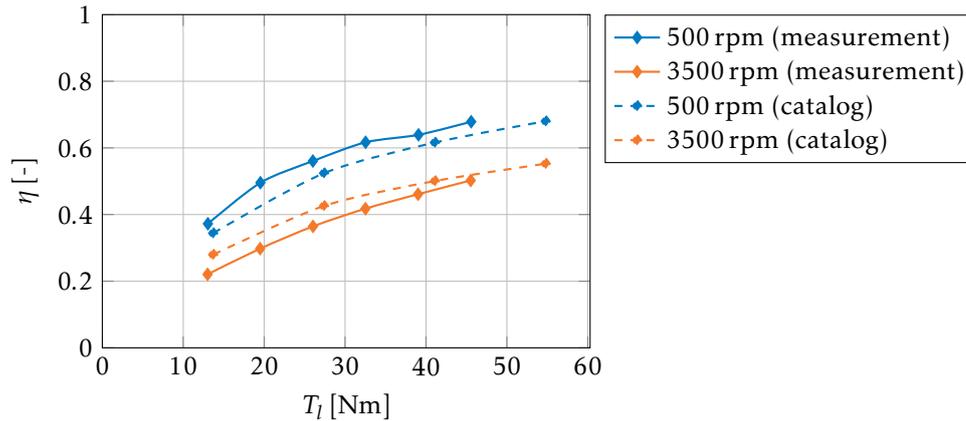


Figure 4.16: Comparison of the efficiency measurement of module A.3 (test series 4) with the catalog data of the Harmonic Drive gear CPL-32-100.

corresponding efficiency of the catalog data. A possible explanation could be that the average temperature of the housing during the experiments on module A was around 40 °C while the catalog data corresponds to temperatures of 20 °C. Since the efficiency is in general increasing with ascending temperatures, these deviations might be comprehensible. The efficiency of the module A.3 (cf. Figure 4.16) for comparable loads and speeds is less than the efficiency of module A.4 (cf. Figure 4.17). This result is in accordance with the efficiency tables provided by the gear manufacturer and thus seems to be a reasonable result.

Finally, in Figure 4.18 the result of the efficiency measurement for the module B.1 is shown together with the efficiency of the Harmonic Drive gear CSD-20-100. Comparing the efficiency measurements of this robot module to the catalog data, there is a clear difference between the absolute efficiency values. In this case, as expected, the efficiency measurements of the modules are below the gear efficiency according to catalog data. One possible explanation might be that the average temperature during the experiments of about 30 °C is closer to the corresponding temperature of the efficiency taken from the catalog data.

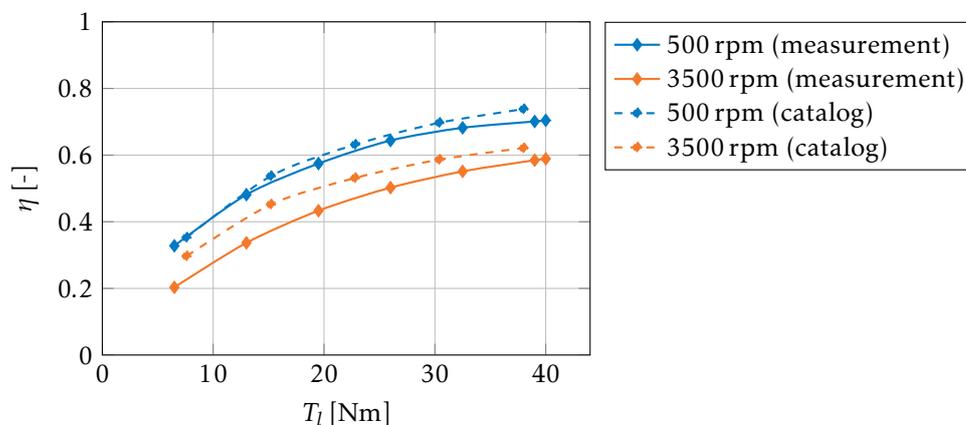


Figure 4.17: Comparison of the efficiency measurement of module A.4 (test series 2) with the catalog data of the Harmonic Drive gear CPL-32-50.

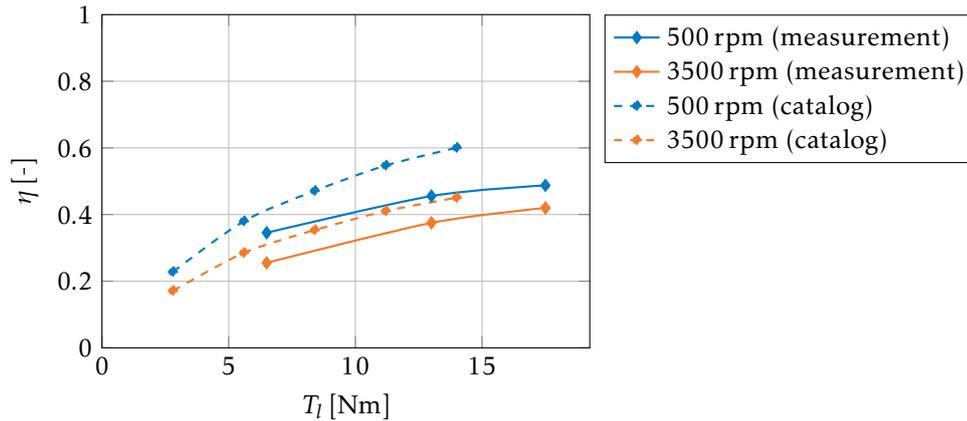


Figure 4.18: Comparison of the efficiency measurement of module B.1 (test series 1) with the catalog data of the Harmonic Drive gear CSD-20-100.

4.7 Chapter Summary

This Chapter presented the design of a modular test bed for experimental parameter identification of robot drive modules. The identification results allow an improvement of friction and dynamic models. In addition, an automatic test procedure to investigate and model steady-state friction effects has been described and was implemented. Experiments have been conducted with several robotic drive modules with Harmonic Drive gear transmissions. The modules were developed for the humanoid LOLA and the final CROPS manipulator. Although, the main source of friction is the gear transmission, the experiments showed that friction modeling based on measurements is significantly improved compared to modeling solely based on the catalog data of the gear's manufacturer.

5 Motion Planning

5.1 Introduction

Robot motion planning is one of the central parts in the design of autonomously acting manipulators. The term *motion planning* incorporates a multitude of different aspects, like finding a collision free path, handling the interaction of cooperating robots, impedance control, developing of grasping strategies, dealing with inaccurate or incomplete sensor information and much more. Additionally, while fulfilling the aforementioned primary objectives, motion planning algorithms have to handle several physical constraints inherent to any technical system, like speed or power limitations of actuators. Even tasks that can be considered rather simple for human beings, like grasping an object, already include a high level of perception and require complex mathematical algorithms for an autonomous robot. For an interaction with the environment a large number of sensors is needed while the sensor data must be evaluated and processed into useful information for the robotic system. Human beings combine effortlessly the input of many different sensors, like visual information from the eyes or tactile feedback from contact of the body with the environment, to build up a world model and take dozens of decisions in a few seconds based on that information.

This Chapter is organized as follows. First of all, a brief survey of relevant motion planning literature is given and basic concepts are illustrated in next Section. The planning approach, chosen by the author will be presented and discussed in Section 5.2.2. Based on the definition of the manipulator task space, the investigated methods for the planning in the workspace are introduced in Section 5.3. In Section 5.4 the inverse kinematics algorithm for kinematically redundant manipulators with a local optimization for real-time applications is presented.

5.2 Overview and Related Work

In this Section, basic concepts and algorithms to solve a certain class of motion planning problems will be discussed. However, a lot of research has been done in this field in the last decades and therefore only a brief overview will be given since a complete survey is far beyond the scope of this thesis. Finally, the approach chosen for the agricultural manipulator will be motivated. A good overview on the topic of motion planning in general is given in the textbooks of Latombe [1991] or Laumond [1998] and more recently in Choset et al. [2005], Siciliano and Khatib [2008] or Siciliano et al. [2009]. A very detailed discussion, in particular on sampling based planners can be found in LaValle [2006].

To begin with, the fundamental concept of *configuration space* will be introduced as a basis for a proper definition of the motion planning problem. A first systematic analysis of this concept has been performed by Lozano-Perez [1983]. The location of articulated manipulators in the workspace is usually provided by the positions of each joint. Since this set of positions, summarized in the vector $\mathbf{q} \in \mathbb{R}^n$, can be directly measured and controlled, it is often considered as the most natural description of a manipulator. The set of all feasible configurations is called the configuration space \mathcal{C} with the dimension n . Usually, the dimension corresponds to the DoFs of the manipulator. The robot's geometry \mathcal{A} , represented by a kinematic chain of rigid bodies, is moving in the Euclidean space $\mathcal{W} \in \mathbb{R}^{2 \text{ or } 3}$ with the p rigid obstacles $\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_p$ as a subset in \mathcal{W} . These obstacles \mathcal{O}_i , $i = 1, \dots, p$ can be mapped into the configuration space \mathcal{C} , the so-called \mathcal{C} -obstacles $\mathcal{C}\mathcal{O}_i$. The union of all obstacles $\bigcup_{i=1}^p \mathcal{C}\mathcal{O}_i$ is the \mathcal{C} -obstacle region while the free configuration space \mathcal{C}_{free} is defined as the complement of the \mathcal{C} -obstacle region:

$$\mathcal{C}_{free} = \mathcal{C} \setminus \bigcup_{i=1}^p \mathcal{C}\mathcal{O}_i \quad (5.1)$$

For a given start and goal configuration $\mathbf{q}_{init} \in \mathcal{C}_{free}$ and $\mathbf{q}_{goal} \in \mathcal{C}_{free}$, the motion planning problem is kinematically solved, if a path, described by a continuous function τ , can be found in the free configuration space from the initial to the goal position:

$$\tau : [0, 1] \rightarrow \mathcal{C}_{free} \quad \text{with} \quad \tau(0) = \mathbf{q}_{init} \quad \text{and} \quad \tau(1) = \mathbf{q}_{goal} \quad (5.2)$$

The path is usually parametrized with a scalar function as a function of time by the path parameter s . In Figure 5.1 the basic idea which is behind the representation of motion planning problems in the configuration space is demonstrated with an illustrative example. Figure 5.1a shows the two dimensional workspace with three sphere obstacles, indicated in gray, and the kinematic model of a two link planar arm in its initial (green) and desired goal (orange) configuration. By construction of the free configuration space according to Figure 5.1b it is obvious that there exists a collision free path from \mathbf{q}_{init} to \mathbf{q}_{goal} and therefore a solution to the path planning problem (5.2). When the free configuration space has been constructed, as in the example shown in Figure 5.1b, the following classical approaches to solve the path planning problem are reported in literature: roadmaps, cell decomposition, and artificial potential fields. Please refer to the literature (e.g. Choset et al. [2005]; Latombe [1991]) for more details on those planning algorithms. Since this planning methods rely on a more or less accurate and in general time consuming construction of the free configuration space \mathcal{C}_{free} they become unsuitable for planning problems in higher dimensions.

Another well-known approach for motion planning is provided by the class of sampling based planners. These planners avoid the explicit construction of \mathcal{C}_{free} , enabling the solution for many practical scenarios in a reasonable amount of time. They imply the generation of collision free samples in the configuration space and strategies to connect these samples to a collision free path. They make use of

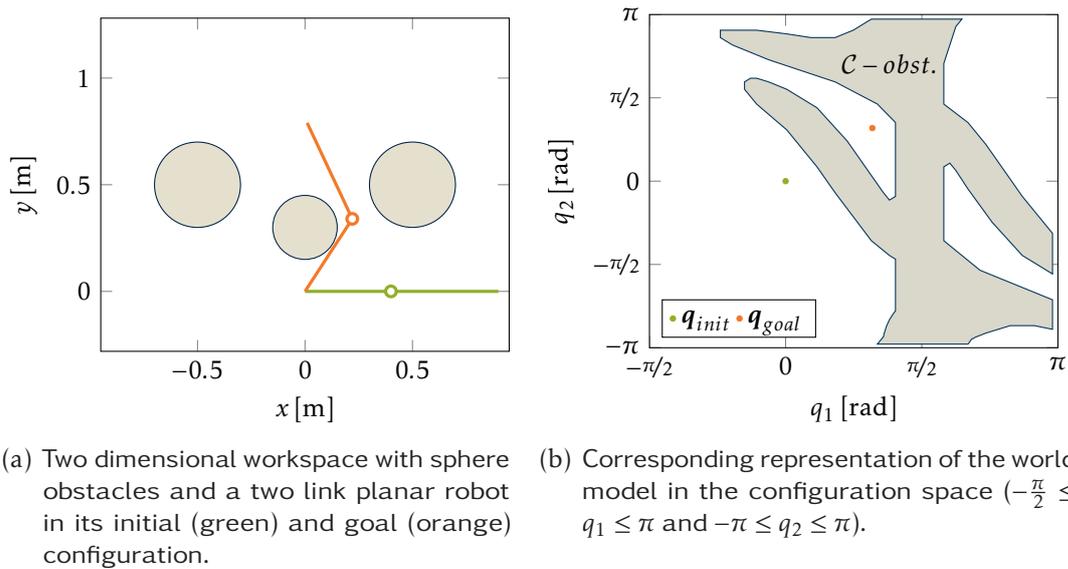


Figure 5.1: A planar motion planning problem represented in the work- and configuration space.

the fact that collision checking of single configurations is computationally very efficient and fast. Depending on the sampling strategy, they can be categorized into single- and multiple-query algorithms (Choset et al. [2005]). Multiple-query planners, like the Probabilistic Road-Map Planner (PRM), are suitable to solve many planning problems with different start and goal configurations while the free configuration space remains unchanged. Usually, they do invest more time in the preprocessing for efficient solving later on. Single-query planners are applied when the planning scene is changing in each run and a typical example of this type, the Rapidly-Exploring Random Trees (RRT), is introduced in the upcoming Section.

5.2.1 Sampling Based Planning

The RRT is a well-known representative of single-query planners as stated by Kuffner and LaValle [2000]. Due to its wide-spread usage in many applications, the basic algorithm will be briefly explained in the following. The original form of the RRT was introduced by LaValle [1998]. Basically, the RRT algorithm samples the free configuration space and builds up one tree \mathcal{T}_1 rooted at the initial configuration q_{init} . By randomly exploring the free configuration space \mathcal{C}_{free} , the tree is grown as will be explained in the following. In Algorithm 5.1, the pseudo-code to build up this single tree is illustrated. Based on a random sample q_{rand} , the nearest neighbor q_{near} according to a certain distance metric is searched in the tree. New configurations q_{new} are added as vertexes to the tree only if they can be connected collision free to the existing tree by sampling the configuration space from q_{near} in the direction to q_{rand} with an appropriate step size and for a predefined length. The edges of the tree are representing the collision free path from q_{near} to q_{new} . For a faster success, the selection of these random configurations can regularly be

Algorithm 5.1 Basic RRT Algorithm (LaValle [1998]).

```

1: function RRT( $\mathcal{T}, \mathbf{q}_{init}, \mathbf{q}_{goal}, N$ )
2:    $\mathcal{T}.init(\mathbf{q}_{init})$ 
3:    $k \leftarrow 0$ 
4:   while  $k < N$  do                                     ▶ Max amount of iterations N
5:      $\mathbf{q}_{rand} \leftarrow \text{random}()$                  ▶ Obtain a random configuration
6:      $\mathbf{q}_{near} \leftarrow \text{nearest\_neighbor}(\mathcal{T}, \mathbf{q}_{rand})$  ▶ Search nearest neighbor of  $\mathbf{q}_{rand}$  in  $\mathcal{T}$ 
7:      $\mathbf{q}_{new} \leftarrow \text{extend}(\mathbf{q}_{near}, \mathbf{q}_{rand}, \Delta q)$  ▶ Extend with step size  $\Delta q$ 
8:      $\mathcal{T}.addNode(\mathbf{q}_{new})$ 
9:      $\mathcal{T}.addVertex(\mathbf{q}_{near}, \mathbf{q}_{new})$ 
10:     $k \leftarrow k + 1$ 
11:  end while
12:  return  $\mathcal{T}$ 
13: end function

```

replaced by choosing the goal configuration \mathbf{q}_{goal} (goal bias). A solution to the path planning problem is found when the goal configuration is added to the tree \mathcal{T}_1 . Usually, the path is further optimized in a post-processing step. In Figure 5.2 the bidirectional RRT algorithm is applied to the exemplary planning problem according to Figure 5.1. In this variation of the basic RRT algorithm, two trees \mathcal{T}_1

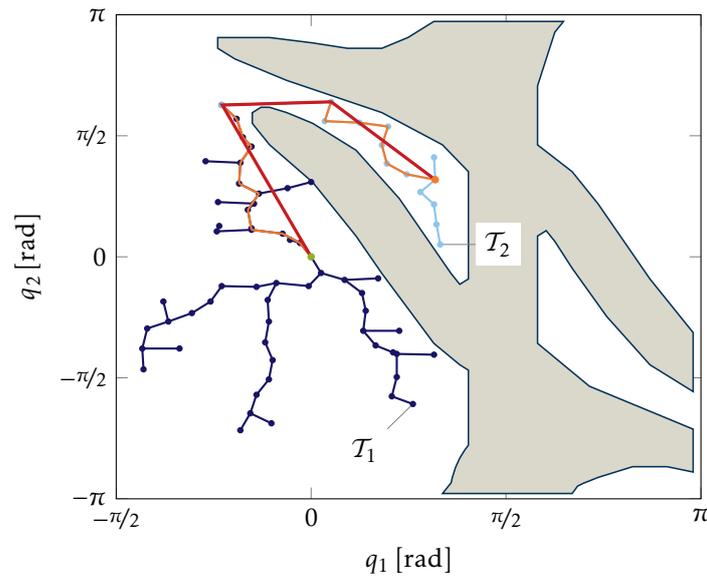


Figure 5.2: RRT algorithm applied to the two link planar arm and the planning scene, represented in the configuration space, according to Figure 5.1.

and \mathcal{T}_2 are grown until they can be collision free merged. The connection from \mathbf{q}_{init} to \mathbf{q}_{goal} along the trees is highlighted in orange and can easily be shortened to the final path, which is indicated in red. The simple shortening algorithm starts at the first node of the path and after skipping the second node it tries to connect the first and the third node along a straight line. If successful, the third node is skipped and the straight connection between the first and fourth is tried. If this fails, the path from the first to the third node is saved and the procedure is

repeated again starting from the third node instead of the first node until the final one is reached. Although this planning approach returns a collision free path, the resulting joint trajectories are not smooth and further optimization of the path is required for the application with a real manipulator. It is worth mentioning that the RRT is not deterministic and another run would provide a different result.

Sampling based planners, like the RRT, are very powerful and many variations exist. Based on the software framework MoveIt! (Sucan and Chitta), they have been evaluated with the here developed manipulators for an apple harvesting scenario within the CROPS project by Nguyen et al. [2013].

5.2.2 The Planning Approach

In Baur et al. [2014b], the author proposed a different approach for the path planning of the agricultural manipulator as will be described in the following. The overall algorithm consists of the combination of a workspace planner, providing a trajectory in the task space and an efficient inverse kinematics algorithm for redundant manipulators to obtain the joint space trajectory. The general scheme is illustrated in Figure 5.3. This procedure was selected out of several reasons

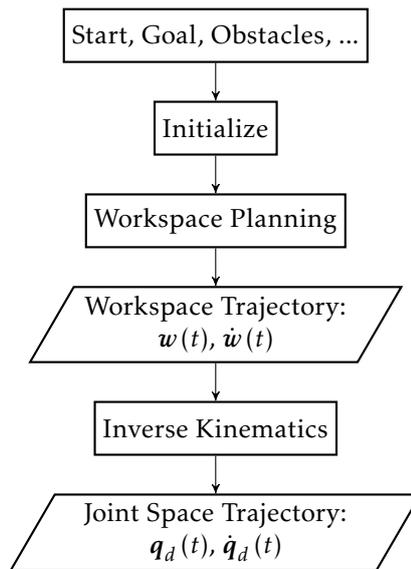


Figure 5.3: Overview of the joint trajectory generation.

which will be presented next. Of all the applications considered within the CROPS project (cf. Section 2.3), the selective harvesting has the highest requirements on the motion planning and must therefore be considered in the first place. Plants and trees in commercial fruit production are grown very dense. Hence, even skilled human worker do not avoid collisions when picking fruits. On the one hand, it might be impossible due to the limited free workspace and on the other hand it is usually unnecessary due to the compliant properties of the obstacles, like leaves or branches, one has to deal with during harvesting. In the CROPS project, the developed end-effector tools for the harvesting applications have bigger dimensions compared to human hands and therefore autonomous harvesting

with complete obstacles avoidance is impossible for many cases. However, by observation of human workers, the picking strategies can be studied and adapted to the workspace planning for the manipulator. In addition, for the planning of the workspace trajectory, known facts of the environment can easily be exploited. Furthermore, the proposed planning algorithm is suitable for real-time application. Therefore, if sensor data is available on-line, the motion of the manipulator could be updated constantly. In particular for the application in agriculture, feedback from tactile sensing could be very beneficial. Especially, since occlusion is a very big issue for vision based detection and contact with the environment is unavoidable, feedback from tactile sensors could be integrated into the proposed planning scheme, for example to limit the contact forces of the robot with the environment. Another advantage of a real-time planning approach is the prospect of low cycle times and thus the higher productivity of the overall system. This is of particular importance when dealing with low cost products like fruits or vegetables and the fact that each path must be individually planned. And finally, as will be discussed in Section 5.4 many secondary objectives can be integrated into the inverse kinematics calculation and are locally optimized by exploiting the redundant DoFs of the manipulator.

5.3 Planning in the Manipulator Workspace

5.3.1 Task Space Definition

For any given task, specific constraints must be considered and a certain amount of DoFs are necessary. The variables required for the task at hand are described by the task space (or sometimes referred to as the operational space) vector $w \in \mathbb{R}^m$. For the harvesting application the full pose of the end-effector in the three dimensional space must be defined. This is easily achieved with the Cartesian coordinates x , y and z as well as the Cardan angles α , β and γ :

$$w := (x, y, z, \alpha, \beta, \gamma)^T \quad (5.3)$$

Other methods, to describe the end-effector orientation are commonly used and reported in literature (Craig [1986]; Siciliano et al. [2009]). With the definitions

$$w_p := (x, y, z)^T \quad \text{and} \quad w_o := (\alpha, \beta, \gamma)^T \quad (5.4)$$

the direct kinematics function f , depending on the joint position vector q , can be written as:

$$w = f(q) = \begin{pmatrix} w_p(q) \\ w_o(q) \end{pmatrix} \quad (5.5)$$

By derivation of (5.5) with respect to time, the end-effector velocity \dot{w}_p and the time derivative of the Cartesian coordinates \dot{w}_o are formally obtained to

$$\dot{w}_p = \frac{\partial w_p}{\partial \mathbf{q}} \dot{\mathbf{q}} = \mathbf{J}_{T,w} \dot{\mathbf{q}} \quad (5.6a)$$

$$\dot{w}_o = \frac{\partial w_o}{\partial \mathbf{q}} \dot{\mathbf{q}} = \bar{\mathbf{J}}_{R,w} \dot{\mathbf{q}} \quad (5.6b)$$

with the Jacobian matrices of translation $\mathbf{J}_{T,w}$ and rotation $\bar{\mathbf{J}}_{R,w}$ for the end-effector. In literature $\bar{\mathbf{J}}_{R,w}$ is commonly referred to as the analytical Jacobian and is not directly available from the recursive kinematics calculations (Siciliano et al. [2009]). Nevertheless, with the Jacobian matrix of rotation \mathbf{J}_R , the analytical Jacobian $\bar{\mathbf{J}}_{R,w}$ is easily obtained, as will be shown in the following. With the transformation matrix \mathbf{T}_A , the time derivative of the Cartesian coordinates \dot{w}_o is related to the angular velocity ω of the end-effector by

$$\omega = \mathbf{T}_A \dot{w}_o \quad (5.7)$$

and with the relation of joint to angular velocities using the Jacobian matrix of rotation \mathbf{J}_R

$$\omega = \mathbf{J}_R \dot{\mathbf{q}} \quad (5.8)$$

the relation between \mathbf{J}_R and $\bar{\mathbf{J}}_{R,w}$ is obtained

$$\bar{\mathbf{J}}_{R,w} = \mathbf{T}_A^{-1} \mathbf{J}_R \quad (5.9)$$

while the matrix inverse of \mathbf{T}_A is given by:

$$\mathbf{T}_A^{-1} = \frac{1}{\cos \beta} \begin{pmatrix} \cos \beta & \sin \beta \sin \alpha & -\sin \beta \cos \alpha \\ 0 & \cos \beta \cos \alpha & \cos \beta \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \quad (5.10)$$

Finally, the task space Jacobian \mathbf{J}_w can be written down to:

$$\mathbf{J}_w = \begin{pmatrix} \mathbf{J}_{T,w} \\ \bar{\mathbf{J}}_{R,w} \end{pmatrix} \quad (5.11)$$

Mentionable, by the selection of Cardan angles to represent the end-effector orientation, one must be aware that due to the matrix inversion in (5.9) a representative singularity occurs at $\cos \beta = 0$.

5.3.2 A Heuristic Approach

In this Section the planning of a trajectory for the harvesting applications, introduced in Section 2.3, will be presented. By exploiting known facts of the environment, a heuristic algorithm for the trajectory generation was designed. Although this approach does not provide a general solution, it will lead to suc-

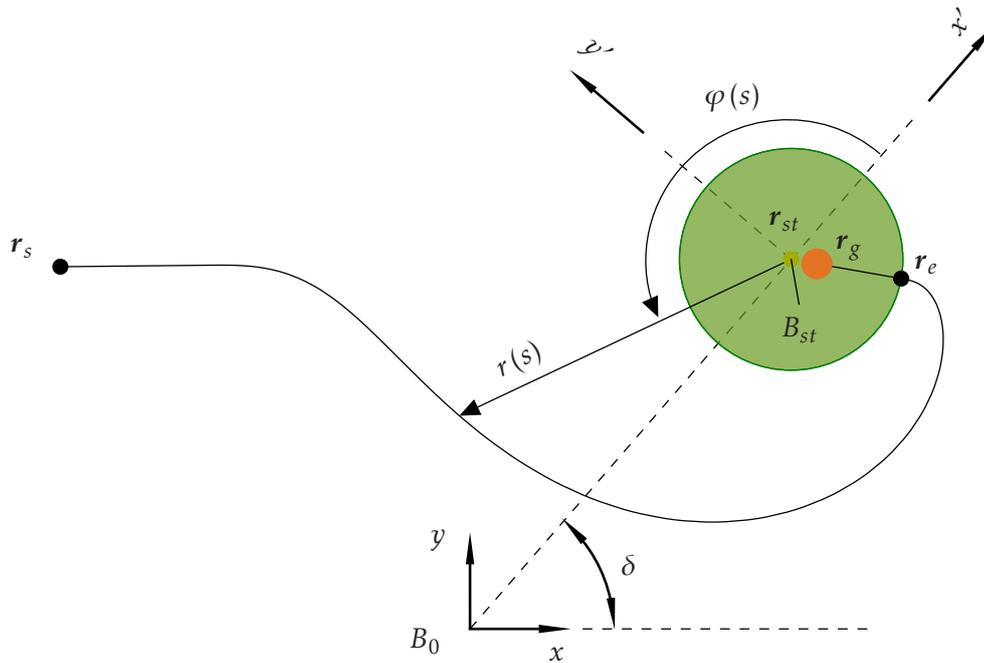


Figure 5.4: Top view of a plant (green) with fruit (red), stem center point (dark green) and a path in the robot workspace.

successful results in many practical scenarios. Furthermore, this method requires minimal information of the environment and thus has low requirements on the sensor system. The heuristic algorithm will be explained for the sweet pepper application but can be directly applied to the other harvesting tasks. In Figure 2.4 the common structure for fruit cultivation is shown with areas indicated by green circles (diameter d) containing leaves, fruits and branches of plants. When reaching for the fruit, a collision of the manipulator with the main stem must be avoided while the target must be approachable. In all likelihood a radial approach, i.e. an approach on the extension of the straight line from the fruit center towards the stem center, seems the most promising in avoiding this collision. Figure 5.4 reduces the planning task into the (x, y) -plane and shows the top view of one free standing plant and a path with the location vectors of the start position $\mathbf{r}_s \in \mathbb{R}^2$, the fruit center $\mathbf{r}_g \in \mathbb{R}^2$ and the main stem position $\mathbf{r}_{st} \in \mathbb{R}^2$ in Cartesian coordinates. Expanding the planning to the three dimensional case is straight forward and will be shown at the end of the Section. Since the stem is principally growing straight upwards, it can be described by two (x and y) coordinates. The major goals for the heuristic planner is to find the trajectory in task space coordinates $\mathbf{w}(t)$ (cf. (5.3)) that brings the end-effector from the start position \mathbf{r}_s to the fruit (goal) position \mathbf{r}_g . Furthermore, the following characteristics must be satisfied:

- The only inputs for the planner are the start position of the end-effector in the manipulator workspace \mathbf{r}_s , the stem coordinates ${}^0\mathbf{r}_{st} = (x_{st}, y_{st})^T$ and the fruit (goal) coordinates ${}^0\mathbf{r}_g = (x_g, y_g)^T$. The vectors are given in the coordinates of the base frame reference system B_0 .

- The goal orientation γ_g of the task space coordinates is computed as a function of the stem and fruit coordinates.
- The path must pass through the entry point $\mathbf{r}_e \in \mathbb{R}^2$ and the circular plant area, defined by the diameter d and the stem center \mathbf{r}_{st} , must be avoided until the entry point is reached.
- At the entry point \mathbf{r}_e the goal orientation must be reached and the end-effector must move on a straight line from \mathbf{r}_e to \mathbf{r}_g .
- The trajectory $w_d(t)$ must be at least C^2 - smooth with zero velocity at the beginning and the end.

Planar Path Generation

First of all, the generation of the path in the (x, y) -plane will be addressed. For the planning in the plane, the manipulator task space is initially defined by

$$\mathbf{w} = (x, y)^T \quad (5.12)$$

with the Cartesian coordinates x and y . According to Figure 5.4 the planar path can be described by the polar coordinates

$$\boldsymbol{\phi}(s) = \begin{pmatrix} r(s) \\ \varphi(s) \end{pmatrix} \quad (5.13)$$

with respect to the stem coordinate frame B_{st} and the path parameter $s(t) \in [0, 1]$. The transformation of any Cartesian coordinates (ξ, η) , provided in the stem frame coordinate system, into the polar coordinates r and $\varphi \in [0, 2\pi)^1$ is given by:

$$\boldsymbol{\phi} = \begin{pmatrix} \sqrt{(\xi^2 + \eta^2)} \\ \text{atan2}(\eta, \xi) \end{pmatrix} \quad (5.14)$$

With the angle $\delta = \text{atan2}(y_{st}, x_{st})$ an arbitrary location vector in base coordinates ${}^0\mathbf{r}_P$ pointing from the origin of B_0 to P is easily transformed with the transformation matrix $A_{st,0}$ into the vector ${}_{st}\mathbf{r}_{st,P}$ in stem coordinates pointing from the origin of B_{st} to P by:

$${}_{st}\mathbf{r}_{st,P} = A_{st,0} ({}^0\mathbf{r}_P - {}^0\mathbf{r}_{st}) \quad (5.15a)$$

$$A_{st,0} = \begin{pmatrix} \cos \delta & \sin \delta \\ -\sin \delta & \cos \delta \end{pmatrix} \quad (5.15b)$$

Applying the transformations (5.14) and (5.15) to \mathbf{r}_s and \mathbf{r}_g , the start and goal positions in polar coordinates $\boldsymbol{\phi}_s = (r_s, \varphi_s)^T$ and $\boldsymbol{\phi}_g = (r_g, \varphi_g)^T$ are obtained. With the plant diameter d , the entry point is given by $\boldsymbol{\phi}_e = (r_e = d/2, \varphi_g)^T$. According to

1 The range $(-\pi, \pi]$ returned by the atan2 function is mapped into the range $(0, 2\pi]$ by adding 2π to negative values.

the requirements on the planner, the following constraints on the path must be considered:

$$\phi(s=0) \stackrel{!}{=} \phi_s \quad (5.16a)$$

$$\phi(s=s_e) \stackrel{!}{=} \phi_e \quad (5.16b)$$

$$\phi(s=1) \stackrel{!}{=} \phi_g \quad (5.16c)$$

With the path parameter at $s = s_e$ ($0 < s_e < 1$) the entry of the TCP into the plant area is defined. To gain an appropriate workspace path, one can shape the functions $r(s)$ and $\varphi(s)$ (cf. (5.13)), while satisfying the constraints (5.16). One promising parametrization is given by:

$$r(s) = \begin{cases} e^{-as} + b & \text{for } 0 \leq s \leq s_e \\ \bar{r}(s) = \sum_{i=0}^5 r_i s^i & \text{for } s_e < s \leq 1 \end{cases} \quad (5.17)$$

and

$$\varphi(s) = \begin{cases} \varphi_s & \text{for } 0 \leq s \leq s_a \\ \bar{\varphi}(s) = \sum_{i=0}^5 \varphi_i s^i & \text{for } s_a < s \leq s_e \\ \varphi_g & \text{for } s_e < s \leq 1 \end{cases} \quad (5.18)$$

The additional parameter $s_a \in [0, s_e]$ is utilized as an application specific tuning variable for the path. While the parameter a and b are easily obtained with the position constraints (5.16) to

$$a = -\frac{1}{s_e} \ln(r_e - b) \quad (5.19a)$$

$$b = r_s - 1 \quad (5.19b)$$

and the coefficients of the fifth order polynomials $\bar{r}(s)$ and $\bar{\varphi}(s)$ are calculated with the following boundary conditions:

$$\bar{r}(s_e) \stackrel{!}{=} r_e \quad \bar{r}(1) \stackrel{!}{=} r_g \quad (5.20a)$$

$$\frac{d}{ds} \bar{r}(s_e) = \bar{r}'(s_e) \stackrel{!}{=} -a e^{-as_e} \quad \bar{r}'(1) \stackrel{!}{=} 0 \quad (5.20b)$$

$$\bar{r}''(s_e) \stackrel{!}{=} a^2 e^{-as_e} \quad \bar{r}''(1) \stackrel{!}{=} 0 \quad (5.20c)$$

and

$$\bar{\varphi}(s_a) \stackrel{!}{=} \varphi_s \quad \bar{\varphi}(s_e) \stackrel{!}{=} \varphi_g \quad (5.20d)$$

$$\bar{\varphi}'(s_a) = \bar{\varphi}''(s_a) \stackrel{!}{=} 0 \quad \bar{\varphi}'(s_e) = \bar{\varphi}''(s_e) \stackrel{!}{=} 0 \quad (5.20e)$$

The resulting curves for r and φ are plotted across s in Figure 5.5. The polar coordinates (5.13) are transformed back into the Cartesian coordinates of the base

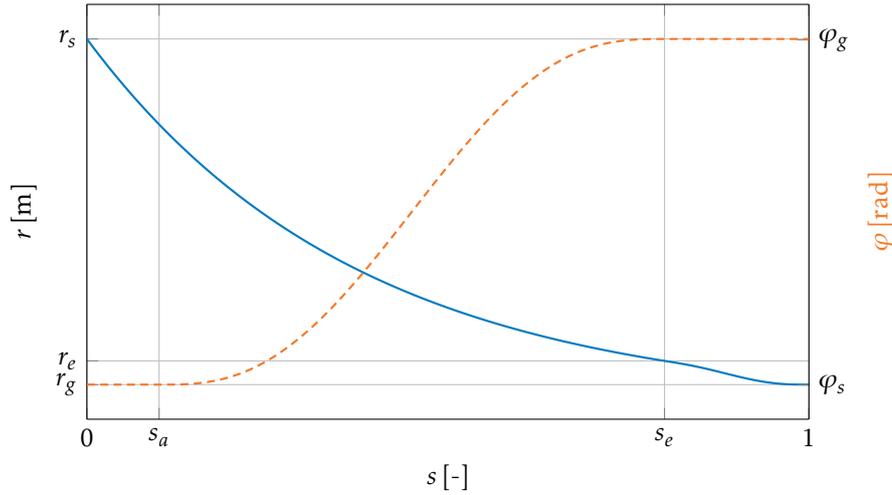


Figure 5.5: The polar coordinates r and φ to shape the path in the manipulator workspace.

frame by:

$$\mathbf{w}(s) = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r(s) \cos \varphi(s) \\ r(s) \sin \varphi(s) \end{pmatrix} + {}_0\mathbf{r}_{st} \quad (5.21)$$

The resulting workspace paths with the heuristic method in the two dimensional plane are presented and discussed in Section 5.3.4.

Time Parametrization

The time parametrization is achieved with the path parameter $s(t)$. For a trajectory with the duration T , a fifth order polynomial with zero value of the first and second order time derivative at the beginning ($t = 0$) and end ($t = T$) was implemented. By differentiating (5.21) with respect to time, the trajectory's velocity $\dot{\mathbf{w}}$ is simply calculated:

$$\dot{\mathbf{w}} = \begin{pmatrix} r' \dot{s} \cos \varphi - r \varphi' \dot{s} \sin \varphi \\ r' \dot{s} \sin \varphi + r \varphi' \dot{s} \cos \varphi \end{pmatrix} \quad (5.22)$$

Since $r(s)$, $\varphi(s)$ and $s(t)$ are C^2 - smooth and $\dot{s}(0) = \dot{s}(T) = 0$, the trajectory \mathbf{w} is C^2 -smooth with zero velocity at the boundaries.

Trajectory for the General Three Dimensional Case

For a gripping tool, usually, the complete pose of the end-effector, i.e. the position in three dimensions and the orientation must be defined at the target location. According to (5.3) this can be done with three Cartesian coordinates and Cardan angles. Based on the start and the fruit center position, provided in three dimensions, the start and goal height, z_s and z_g , are obtained. Since the fruits are usually hanging down straight, the goal orientation can be defined to $\alpha_g = \beta_g = 0$ if no further information is available while the angle γ_g is easily calculated with

φ_g and δ . The goal height and orientation must be reached at the entry point, parametrized by s_e with the point in time t_e , corresponding to the path parameter function ($s(t_e) = s_e$). The trajectory for the z coordinate can be described by

$$z(t) = \begin{cases} \bar{z}(t) & \text{for } 0 \leq t \leq t_e \\ z_g & \text{for } t_e < t \leq T \end{cases} \quad (5.23)$$

with the constraints:

$$z(0) \stackrel{!}{=} z_s \qquad z(t_e) \stackrel{!}{=} z_g \quad (5.24a)$$

$$\dot{z}(0) \stackrel{!}{=} 0 \qquad \dot{z}(t_e) \stackrel{!}{=} 0 \quad (5.24b)$$

$$\ddot{z}(0) \stackrel{!}{=} 0 \qquad \ddot{z}(t_e) \stackrel{!}{=} 0 \quad (5.24c)$$

A trajectory $\bar{z}(t)$, fulfilling (5.24) and being C^2 -smooth, was presented by Ewald et al. [2012] and is applied here. Furthermore, the proposed interpolation method avoids overshoot while minimizing the cost function on the time interval of the trajectory:

$$\phi = \int \ddot{z}(t) dt \rightarrow \min \quad (5.25)$$

The aforementioned approach is effortlessly transferred in a similar way to generate the trajectories for $\alpha(t)$ and $\beta(t)$. With (5.17), (5.18), the transformation (5.21), and (5.23), the complete workspace trajectory $w(t)$ is defined.

5.3.3 Navigation Potential Function

Motion planning with potential functions was at first introduced by Khatib [1986] where he presented a framework for real-time collision avoidance. However, a well known issue of artificial potential functions applied for motion planning problems is the occurrence of local minimum and in robotics literature many varieties are discussed to reduce its number and an overview on state-of-the-art methods can be found in the textbooks of Latombe [1991] or Siciliano and Khatib [2008]. A function with only one global minimum at the goal configuration is called (global) navigation function and was introduced by Rimon and Koditschek [1989]. In this Section, the concept and relevant equations for this planning approach are presented. The potential navigation function ϕ , defined in a bounded sphere space according to Choset et al. [2005] for any task or configuration space denoted by $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ and the target $\mathbf{x}_g \in \mathbb{R}^n$ is described by:

$$\phi(\mathbf{x}) = \frac{d(\mathbf{x}, \mathbf{x}_g)^2}{\left[d(\mathbf{x}, \mathbf{x}_g)^{2\kappa} + \beta(\mathbf{x}) \right]^{1/\kappa}} \quad (5.26)$$

With an attractive potential β_0 of the bounding sphere centered at \mathbf{x}_0 and radius r_0 and the repulsive potentials β_i for $i = 1, \dots, p$ of the spherical obstacles (center \mathbf{x}_i and radius r_i):

$$\beta(\mathbf{x}) = \prod_{i=0}^p \beta_i(\mathbf{x}) \quad (5.27)$$

$$\beta_0(\mathbf{x}) = -d(\mathbf{x}, \mathbf{x}_0)^2 + r_0^2 \quad (5.28)$$

$$\beta_i(\mathbf{x}) = d(\mathbf{x}, \mathbf{x}_i)^2 - r_i^2 \quad (5.29)$$

The function $d(\mathbf{x}_1, \mathbf{x}_2)$ calculates the Euclidean distance of two arbitrary vectors $\mathbf{x}_1 \in \mathbb{R}^n$ and $\mathbf{x}_2 \in \mathbb{R}^n$:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\|\mathbf{x}_1 - \mathbf{x}_2\|} = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2)} \quad (5.30)$$

With an appropriate selection of the tuning parameter κ , it can be guaranteed that (5.26) has only one minimum of zero value at the target \mathbf{x}_g while on the radius of obstacles and the bounding sphere, it has the maximum value of one. Please refer to Koditschek and Rimon [1990] for further details. Starting from any configuration $\mathbf{x}_s \in \mathbb{R}^n$ within the bounding sphere, a collision free path is obtained by moving along the negative gradient of the potential function.

Gradient of the Navigation Potential Function

The gradient of the navigation potential function

$$\text{grad}(\phi) = \nabla\phi = \left(\frac{\partial\phi}{\partial x_1}, \dots, \frac{\partial\phi}{\partial x_n} \right)^T \quad (5.31)$$

is obtained by applying the quotient rule to (5.26) (the index $k = \{1, \dots, n\}$ denotes the k -th element of the vectors \mathbf{x} , \mathbf{x}_i , \mathbf{x}_0 or \mathbf{x}_g):

$$\frac{\partial\phi}{\partial x_k}(\mathbf{x}) = \frac{u'v - uv'}{v^2} \quad (5.32)$$

$$u := d(\mathbf{x}, \mathbf{x}_g)^2 \quad (5.33)$$

$$v := \left[d(\mathbf{x}, \mathbf{x}_g)^{2\kappa} + \beta(\mathbf{x}) \right]^{1/\kappa} \quad (5.34)$$

$$(5.35)$$

For better clarity, the following abbreviations are applied from now on (if appropriate): $d = d(\mathbf{x}, \mathbf{x}_g)$ and $\beta = \beta(\mathbf{x})$. The partial derivatives u' and v' are obtained to:

$$u' = \frac{\partial u}{\partial x_k} = 2d \frac{\partial d}{\partial x_k} \quad (5.36)$$

$$v' = \frac{\partial v}{\partial x_k} = \frac{1}{\kappa} [d^{2\kappa} + \beta]^{1/\kappa-1} \left[2\kappa d^{2\kappa-1} \frac{\partial d}{\partial x_k} + \frac{\partial \beta}{\partial x_k} \right] \quad (5.37)$$

And the partial derivatives of the Euclidean distance (5.30) and the potential function (5.27) are given by:

$$\frac{\partial d(\mathbf{x}, \mathbf{x}_g)}{\partial x_k} = \frac{x_k - x_{g,k}}{d(\mathbf{x}, \mathbf{x}_g)} \quad (5.38)$$

$$\frac{\partial \beta(\mathbf{x})}{\partial x_k} = \sum_{i=0}^p \prod_{j=0}^p \bar{\beta}_j \quad (5.39)$$

with

$$\bar{\beta}_j = \begin{cases} \frac{\partial \beta_j(\mathbf{x})}{\partial x_k} & \text{if } i = j \\ \beta_j(\mathbf{x}) & \text{else} \end{cases} \quad (5.40)$$

$$\frac{\partial \beta_j(\mathbf{x})}{\partial x_k} = \begin{cases} -2(x_k - x_{0,k}) & \text{if } j = 0 \\ 2(x_k - x_{i,k}) & \text{else} \end{cases} \quad (5.41)$$

Gradient Descent

To find a suitable path with the artificial potential field, the commonly used gradient descent algorithm to search for the minimum can be applied and the algorithm is illustrated in pseudo-code in Algorithm 5.2. The configurations (or a subset

Algorithm 5.2 Optimization with gradient descent

```

1:  $i = 0$ 
2:  $\mathbf{x}(0) = \mathbf{x}_s$ 
3: while  $\nabla \phi(\mathbf{x}) \neq 0$  do ▷ Other criteria to exit the while loop are possible
4:    $\mathbf{x}(i+1) \leftarrow \mathbf{x}(i) + \alpha(i) \nabla \phi(\mathbf{x}(i))$ 
5:    $i \leftarrow i + 1$ 
6: end while
7: return  $\mathbf{x}$ 

```

of them), by iteratively solving the minimization problem are the supporting points for the path. By interpolation of the supporting points, for example with cubic splines, a smooth trajectory is found. In literature on optimization, several methods to control the step size α at the iteration i for improved convergence are described and discussed in Bertsekas [1999].

Example Application

Unlike the application of the potential navigation function in configuration space, as it is reported in literature, the planning method is utilized for path calculation in the robot workspace. Similar to the heuristic approach (cf. Section 5.3.2) the planning scene is reduced into the two dimensional plane with the identical

procedure for the extension into the general three dimensional case. For an illustration of the planning scheme, the method is applied to a world scene with three spherical obstacles according to the numerical values provided in Table 5.1. With reference to the harvesting application, the obstacles could be

Table 5.1: The sphere obstacles with center at (x, y) and radius r of the world model.

Obstacles	x [m]	y [m]	r [m]
1	0	0.8	0.02
2	0.43	1.2	0.05
3	0.43	0.3	0.05

interpreted as stems or other fruits. The start position is given by the location vector $\mathbf{r}_s = (-0.59\text{m}, 0.77\text{m})^T$ and the goal position is $\mathbf{r}_g = (0.43\text{m}, 0.75\text{m})^T$. The center of the bounding sphere is defined by the midpoint in-between the start and goal position with a radius of $r_0 = 1.2d(\mathbf{r}_s, \mathbf{r}_g)$, while $d(\cdot)$ denotes the distance function according to (5.30). The navigation potential function (5.26) with the scalar $\kappa = 3$ is plotted for this exemplary scenario in Figure 5.6. At the boundary

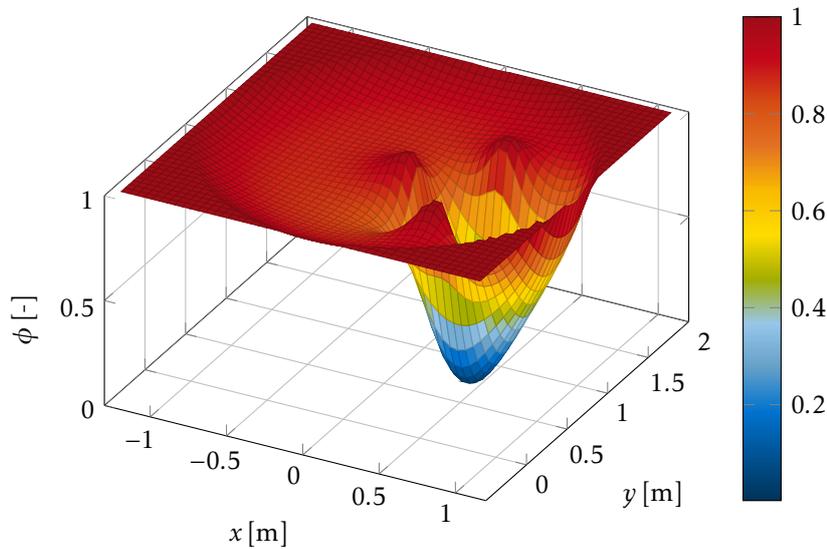


Figure 5.6: Three dimensional plot of the navigation potential function with the three sphere obstacles.

of the obstacles the value of the potential function approaches $\phi \rightarrow 1$, while at the goal position, the potential has its minimum with $\phi \rightarrow 0$. Remarkably, the potential function has a gentle gradient close to the start position, while at the proximity of the target a very steep gradient can be observed. With Algorithm 5.2 the optimized path in the workspace can be found and is illustrated in the contour plot in Figure 5.7.

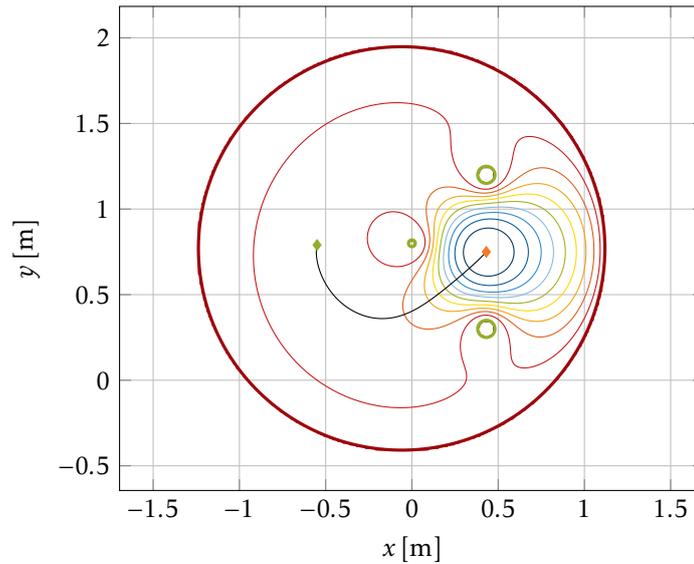


Figure 5.7: Contour plot of the potential navigation function with the optimal path, resulting from the iterations with the gradient descent algorithm. The start and goal configuration is indicated by one green and red diamond while the obstacles are highlighted by green circles.

5.3.4 Results and Discussion

The results of the workspace planning methods based on the heuristic and the potential navigation function will be presented here. The planar paths of the two methods are illustrated for a scene corresponding to the sweet pepper selective harvesting application. Several fruit positions, located around a stem with the coordinates ${}^0r_{st} = (0.43\text{ m}, 0.8\text{ m})^T$ are approached from the start position ${}^0r_s = (0.55\text{ m}, 0.79\text{ m})^T$. Figure 5.8 shows the paths, with a circle of radius r_e drawn in green around the stem for the heuristic method. From the entry radius to the fruit, the paths are given by a straight line. The same planning scene has been investigated with the concept of navigation potential fields while the only obstacle was the stem center position. The resulting paths from the start to the fruits are depicted in Figure 5.9. On the one hand, the results indicate that with the heuristic planning (cf. Figure 5.8), the plant area is clearly avoided until the fruit is reached. However, according to Figure 5.9 this is not the case for the potential field approach. On the other hand, with the heuristic planning, the path in the workspace does cover a rather big area compared to the planning with the navigation function. Accordingly, when applying the heuristic approach, the manipulator will sweep through a larger volume increasing the risk of collisions with the parts of the robot and other plants or fruits. Although the planning with the potential field approach can deal with more general cases, it has some major drawbacks for the agricultural applications. First of all, saddle points can still occur in the potential function. Furthermore, the obstacle regions must be disjoint which is not always the case and especially for the harvesting applications, obstacles can be very close to the goal configuration. This results in very high gradients and numerical issues when searching for a path with the gradient descent algo-

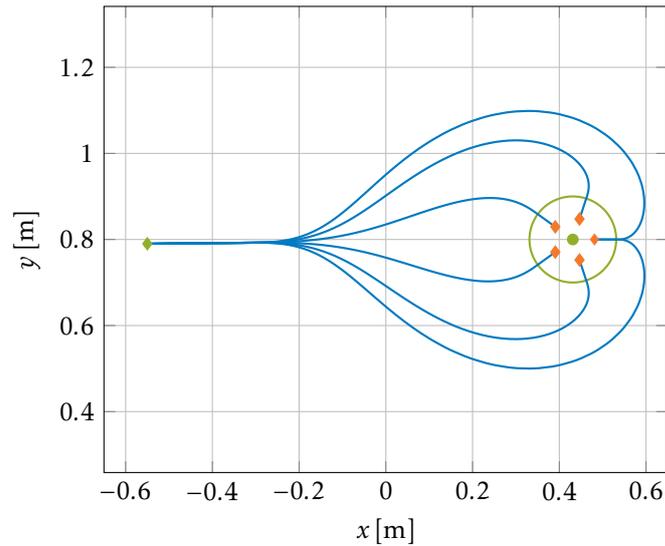


Figure 5.8: Resulting paths in the (x, y) -plane by applying the heuristic planner for several fruits distributed around a stem ($r_e = 0.1$ m, $s_a = 0.1$ and $s_e = 0.8$).

rithm. Therefore, the heuristic planning was chosen for the implementation in the laboratory demonstration and suggested for the application in field experiments. Since it does provide a high robustness and it also ensures, due to the definition of the entry radius, that the gripper approaches the fruit without crossing the plant area as long as possible and therefore reducing the risk of collisions with main stem and manipulator.

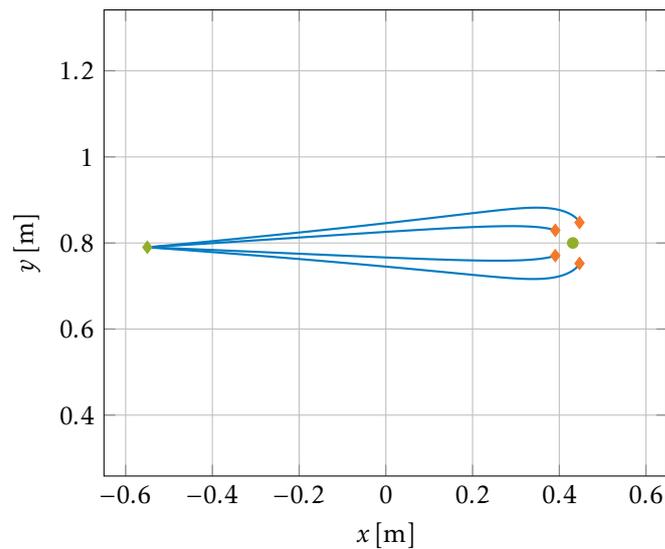


Figure 5.9: Resulting paths of the planning with the navigation potential function for several fruits distributed around a stem.

5.4 Inverse Kinematics

This Section will present the inverse kinematics algorithm implemented for the developed agricultural robot (cf. Chapter 3). As an introduction, the definition of the inverse kinematics problem will be given and a very brief overview on conventional solution strategies is made. For serial-chain manipulators with given geometrical dimensions the inverse kinematics solution is defined as a set of joint positions that corresponds to a given end-effector position and orientation (Craig [1986]). Or in other words, for a desired end-effector pose w_d , the joint configuration q_d that satisfies the direct kinematics equation

$$w_d = f(q = q_d) \quad (5.42)$$

is the solution of the inverse kinematics problem. Algorithms to solve the inverse kinematics are of particular importance in robotics, because in most applications the robot has to operate in the workspace while it is controlled in the joint space. Since (5.42) is nonlinear in q , the computation of the inverse kinematics on the position level requires to solve a set of nonlinear equations, which in general has no closed-form solution and a numerical computation is required. Among the iterative numerical algorithms, the most fundamental one is probably the Newton-Raphson method with the disadvantage that its convergence strongly depends on the initial guess. Depending on the kinematics of manipulators a closed-form solution might exist. For example, according to Pieper [1968] and Siciliano and Khatib [2008], the analytical solution for commonly used industrial manipulators with six DoF can always be obtained if three consecutive axis are in parallel and the other three consecutive axis intersect in one point. Further solutions of simple kinematic structures can be found in Siciliano et al. [2009]. A closed-form solution to the inverse kinematics has the great advantage that it is computationally fast and if multiple solutions exist, it usually provides all of them. Unfortunately, in general these solutions can only be found for a certain class of robots. Besides dealing with the inverse kinematics on the position level with the nonlinear equation of the direct kinematics (5.42), it can be advantageous to consider the differential kinematics equation which is the time derivative of (5.42):

$$\dot{w} = \frac{\partial f}{\partial q} \dot{q} = J_w(q) \dot{q} \quad (5.43)$$

The Jacobian matrix J_w represents the gradient of the direct kinematics function with respect to the joint positions. The investigation of differential kinematics of robot manipulators is called resolved motion rate control and was introduced by Whitney [1969]. Equation (5.43) provides a linear mapping of the joint speeds in the configuration space and the end-effector velocities in the workspace. For a given end-effector trajectory \dot{w}_d and if the Jacobian J_w is non-singular, the inverse kinematics on the velocity level is easily obtained by solving the system of linear

equations²:

$$\dot{\mathbf{q}}_d = \mathbf{J}_w(\mathbf{q})^{-1} \dot{\mathbf{w}}_d \quad (5.44)$$

Based on (5.44) and the initial condition for $\mathbf{q}(0) = \mathbf{q}_0$ the joint positions are obtained by time integration. Usually, this integration is implemented numerically and algorithms to compensate for the numerical drift are commonly used (cf. Figure 5.10). The aforementioned algorithm is only suitable for kinematically non-redundant manipulators, since the Jacobian matrix must be invertible, i.e. at least the dimensions of the configuration space $\mathbf{q} \in \mathbb{R}^n$ and the workspace $\mathbf{w} \in \mathbb{R}^m$ must be the same ($m = n$), so that the Jacobian is square. Basic properties of kinematic redundant manipulators are discussed in the next Subsection.

5.4.1 Redundant Manipulators

Manipulators with more DoFs than they require to fulfill a certain task are termed kinematically redundant. The dimension of the joint space n is higher than the dimension of the workspace m , i.e. $n > m$. The degree of redundancy is defined by $r = n - m$. For manipulators with kinematic redundancy it is possible to change the joint configuration without changing the end-effector position and orientation. Or in other words, for one single end-effector pose infinite solutions to the inverse kinematics problem exist. Besides providing a high dexterity for the robot, the redundancy can be, amongst others, exploited to avoid obstacles, joint limits or singular configurations. Redundancy and its mathematical properties were very well analyzed and presented by Nakamura [1991] and Siciliano et al. [2009]. The analysis is mainly based on the differential kinematics equation (5.43) and the linear mapping of joint velocities to end-effector velocities by the Jacobian $\mathbf{J}_w(\mathbf{q})$. For a given manipulator configuration, the end-effector velocities which can be generated by the joint velocities are defined by the range space of \mathbf{J}_w , while the null space of \mathbf{J}_w is the subspace of joint velocities that does not result in any end-effector velocity. The next Subsection will present a well-known technique to solve the inverse kinematics while exploiting the redundant DoFs.

5.4.2 Local Optimization of Kinematic Redundancy

In the previous Subsection, redundant robots were introduced. They provide more DoFs than the minimum number required to fulfill a certain task. An approach to systematically exploit these redundant DoFs has been proposed by Liégeois [1977] and will be explained here. A quadratic optimization problem with the cost functional ϕ and the constraint (5.45b) can be formulated as follows:

$$\phi = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{W} \dot{\mathbf{q}} + \alpha \frac{\partial H}{\partial \mathbf{q}} \dot{\mathbf{q}} \rightarrow \min! \quad (5.45a)$$

$$\mathbf{0} = \dot{\mathbf{w}}_d - \mathbf{J}_w \dot{\mathbf{q}} \quad (5.45b)$$

² The matrix inverse is usually avoided and the more efficient Gaussian elimination is applied.

The first term on the right hand side of the cost functional (5.45a) minimizes the quadratic sum of joint velocities with the usually constant and diagonal weighting matrix \mathbf{W} . Besides establishing the relation between prismatic and revolute joints, the matrix \mathbf{W} can be used to penalize individual joint movement according to a specific task. The second term, with the scaling parameter α is utilized to minimize the configuration dependent cost function $H(\mathbf{q})$. This constraint optimization problem can be solved for $\dot{\mathbf{q}}$ with the method of Lagrange multipliers. The closed-form solution of the inverse kinematics yields to:

$$\dot{\mathbf{q}} = \mathbf{J}_w^\# \dot{\mathbf{w}}_d - \underbrace{\alpha (\mathbf{I} - \mathbf{J}_w^\# \mathbf{J}_w)}_{=: \mathbf{N}} \left(\frac{\partial H}{\partial \mathbf{q}} \right)^T \quad (5.46)$$

The identity matrix with appropriate dimensions is indicated by \mathbf{I} and the generalized inverse $\mathbf{J}_w^\#$ with the weighting matrix \mathbf{W} is given by:

$$\mathbf{J}_w^\# = \mathbf{W}^{-1} \mathbf{J}_w^T (\mathbf{J}_w \mathbf{W}^{-1} \mathbf{J}_w^T)^{-1} \quad (5.47)$$

Remarkably, an arbitrary joint space velocity $\mathbf{x} \in \mathbb{R}^n$ multiplied with the null space matrix \mathbf{N} will not result in any velocity of the end-effector, since:

$$\mathbf{J}_w \mathbf{N} \mathbf{x} = \mathbf{0} \quad (5.48)$$

Based on the joint velocities, a straightforward way to obtain the joint positions is by numerical integration with the Euler method. The block diagram scheme of the inverse kinematics algorithm is given in Figure 5.10. Assuming that the joint

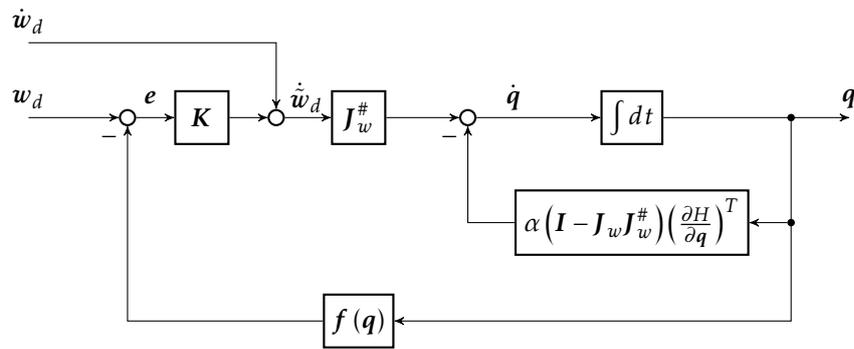


Figure 5.10: Block diagram of the inverse kinematics algorithm.

positions $\mathbf{q}(t_i)$ and velocities $\dot{\mathbf{q}}(t_i)$ are known at a certain time t_i , then the joint positions at $t_{i+1} = t_i + \Delta t$ are approximated as follows:

$$\mathbf{q}(t_{i+1}) = \mathbf{q}(t_i) + \dot{\mathbf{q}}(t_i) \Delta t \quad (5.49)$$

According to Siciliano et al. [2009], for numerical drift compensation during integration with (5.49) the desired end-effector velocity $\dot{\mathbf{w}}_d$ in (5.46) can be replaced

by an effective velocity $\dot{\boldsymbol{w}}_d$ as follows:

$$\dot{\boldsymbol{w}}_d := \dot{\boldsymbol{w}}_d + \boldsymbol{K} \boldsymbol{e} \quad (5.50a)$$

$$\boldsymbol{e} := \boldsymbol{w}_d - \boldsymbol{f}(\boldsymbol{q}) \quad (5.50b)$$

The error vector $\boldsymbol{e} \in \mathbb{R}^m$ is computed with the direct kinematics function $\boldsymbol{f}(\boldsymbol{q})$ and \boldsymbol{K} is a diagonal gain matrix.

A detailed discussion on the computational complexity of different inverse kinematics solutions is given in Nakamura [1991]. For an efficient numerical computation of (5.46), Nakamura [1991] referred to KLEIN AND HUANG (1983), who pointed out that the matrix inversion in (5.47) can be avoided using a Gaussian elimination. The efficient computation is illustrated with pseudo-code in Algorithm 5.3.

Algorithm 5.3 Efficient inverse kinematics computation

- 1: $\boldsymbol{p} \leftarrow \dot{\boldsymbol{w}}_d + \alpha \boldsymbol{J}_w \boldsymbol{W}^{-1} \left(\frac{\partial H}{\partial \boldsymbol{q}} \right)^T$
 - 2: $\boldsymbol{B} \leftarrow \boldsymbol{J}_w \boldsymbol{W}^{-1} \boldsymbol{J}_w^T$
 - 3: Solve: $\boldsymbol{B} \boldsymbol{\gamma} = \boldsymbol{p}$ ▷ By efficient Gaussian elimination
 - 4: Compute: $\dot{\boldsymbol{q}} = \boldsymbol{W}^{-1} \left(\boldsymbol{J}_w^T \boldsymbol{\gamma} - \alpha \left(\frac{\partial H}{\partial \boldsymbol{q}} \right)^T \right)$
-

5.4.3 Secondary Objective Functions

For the developed agricultural manipulator, the following separate secondary objectives were investigated:

- H_j for joint limit avoidance
- H_c for (self)-collision avoidance
- H_m for the maximization of manipulability

The weighted sum

$$H(\boldsymbol{q}) = w_j H_j + w_c H_c + w_m H_m \quad (5.51)$$

with the weighting parameter w_j , w_c and w_m is the overall secondary objective function $H(\boldsymbol{q})$. This Subsection will provide an overview on the mathematical definition of the objective functions and its gradients which are required for the inverse kinematics computation according to (5.46).

Joint Limit Avoidance

An objective function for avoidance of configurations close to mechanical joint limits was originally introduced by Liégeois [1977]. An adapted version was

already presented in Baur et al. [2012] and is provided here for convenience:

$$H_j = \begin{cases} \sum_{i=1}^n \frac{(q_i - \bar{q}_{M,i})^2}{(q_{M,i} - \bar{q}_{M,i})^2} & \text{for } q_i > \bar{q}_{M,i} \\ \sum_{i=1}^n \frac{(q_i - \bar{q}_{m,i})^2}{(q_{m,i} - \bar{q}_{m,i})^2} & \text{for } q_i < \bar{q}_{m,i} \\ 0 & \text{else.} \end{cases} \quad (5.52)$$

In the equation above, the index i denotes a single joint and n is the total amount of DoFs. Maximum (minimum) position limits for each joint are defined by $q_{M,i}$ ($q_{m,i}$). An application specific threshold $\bar{q}_{M,i}$ ($\bar{q}_{m,i}$) is applied as a tuning parameter for the joint limit avoidance. For joint positions q_i in-between the thresholds, i.e. $\bar{q}_{m,i} < q_i < \bar{q}_{M,i}$ the objective function has zero value and gradient, thus no influence on the joint motion. However, further joint movement towards the limits is penalized with a quadratic function when either threshold is exceeded. The maximum value of one summand of H_j is normalized to one and Figure 5.11 shows the dependency of the cost function to the position of a single joint variable q_i . Other cost functions consider favorable configurations, for example the

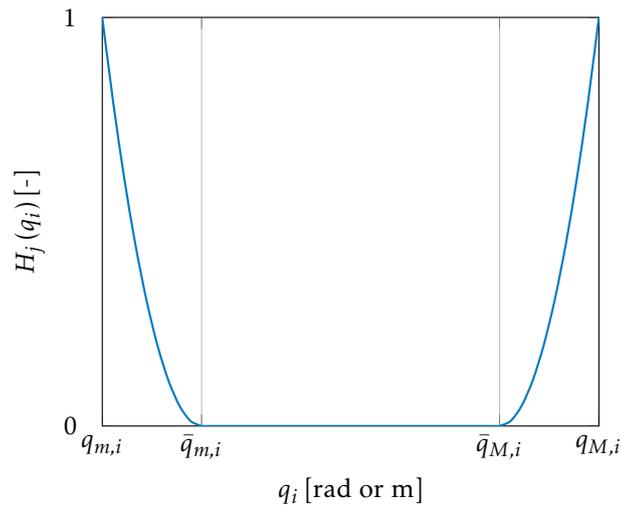


Figure 5.11: The objective function H_j for joint limit avoidance of a single joint.

middle position in-between joint limits (Siciliano et al. [2009]). The application of the joint limit avoidance algorithm to the 7-DoF agricultural manipulator is very illustrative demonstrated in Figure 5.12. While the task space includes the Cartesian coordinates of the TCP in three dimensional space, the TCP position is set as a straight line path from the lower part to the upper part of the reachable workspace (cf. red line in Figure 5.12a). When the joint limit avoidance is active, the manipulator successfully avoids the upper limit of the first joint by utilizing its redundant DoFs while tracking the workspace trajectory. If the joint limit avoidance algorithm is inactive, the upper joint limit is violated and the motion is aborted (cf. Figure 5.12b).

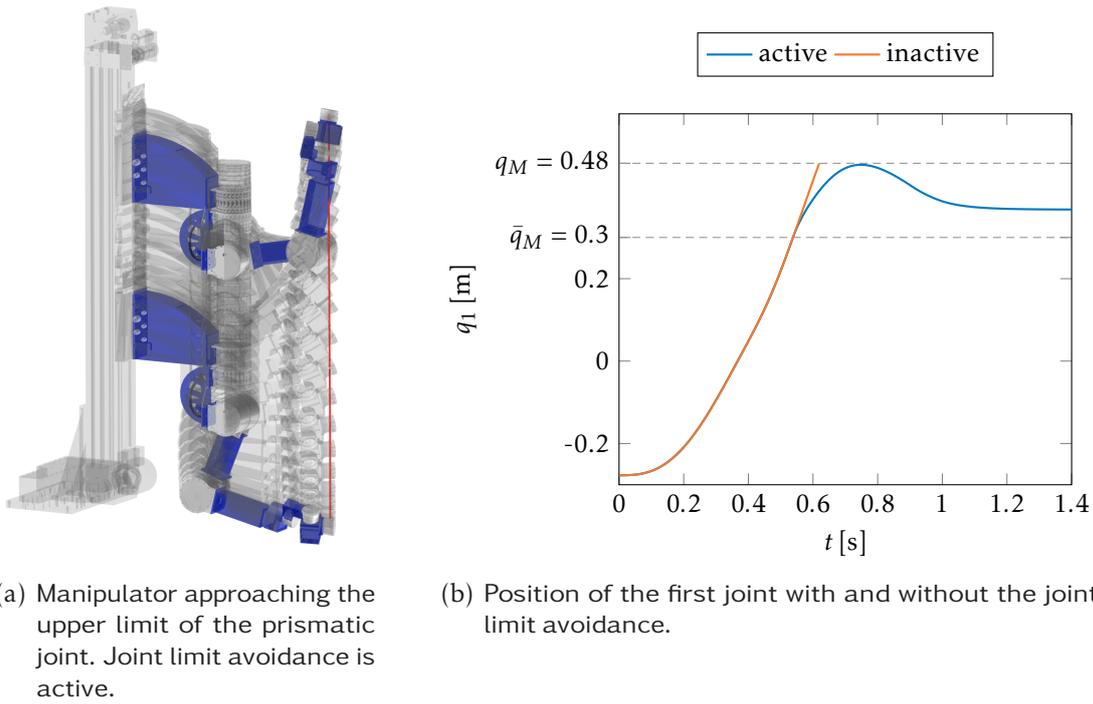


Figure 5.12: The joint limit avoidance algorithm for the 7-DoF agricultural manipulator.

Self-Collision Avoidance

In general, without precautionary measures and especially for manipulators with many DoFs, self-collisions can arise during normal operation. Therefore, the application of an efficient self-collision model (cf. Section 3.4.4) to check if a certain motion results in a collision and a collision avoidance algorithm is very important. A very accurate description of the framework and the distance calculation algorithms for real time applications can be found in the work of Schwienbacher [2013]. As a matter of fact, this framework can also be used for collision avoidance with objects in the environment. However, for arbitrary objects, the automatic generation of SSVs must be considered. Under the assumption that the minimum distance between two collision pairs is available, only the fundamental ideas of the collision avoidance algorithm are presented here for convenience.

The collision avoidance potential function H_c is defined as the sum of the single potentials $H_{c,i}$ for the N_c total amount of collision pairs:

$$H_c = \sum_{i=1}^{N_c} H_{c,i}(d_i) \quad (5.53)$$

The scalar function $H_{c,i}$ is dependent on the minimum distance $d_i(\mathbf{q})$ between two collision pairs i . The minimum distance is defined by the location vector $\mathbf{r}_{c,i}$ from point A to B on the surface of the SSVs corresponding to the collision pairs with

the shortest length:

$$d_i = \|\mathbf{r}_{c,i}\| = \sqrt{\mathbf{r}_{c,i}^T \mathbf{r}_{c,i}} \quad (5.54)$$

The points A and B are described by their location vectors \mathbf{r}_A and \mathbf{r}_B , respectively. The gradient of the cost functional $H_{c,i}$ with respect to \mathbf{q} is given by:

$$\frac{\partial H_{c,i}}{\partial \mathbf{q}} = \frac{\partial H_{c,i}}{\partial d_i} \frac{\partial d_i}{\partial \mathbf{q}} \quad (5.55)$$

And computation of the distance gradient $\frac{\partial d_i}{\partial \mathbf{q}}$ is obtained with the Jacobian matrices of translation $\mathbf{J}_{T,A}$ and $\mathbf{J}_{T,B}$ for the points A and B as follows (Toussaint et al. [2007]):

$$\frac{\partial d_i}{\partial \mathbf{q}} = \frac{1}{d_i} (\mathbf{r}_A - \mathbf{r}_B)^T (\mathbf{J}_{T,A} - \mathbf{J}_{T,B}) \quad (5.56)$$

Body fixed frames have been assigned to the robot links belonging to the points A and B. With the Jacobian matrices of translation of these body fixed frames, which are known by the direct kinematics computation, and the location vectors of A and B, $\mathbf{J}_{T,A}$ and $\mathbf{J}_{T,B}$ are easily obtained. Similar to the joint limit avoidance potential function, the intention is to design a potential function which increases when two collision pairs are approaching each other. Furthermore, the influence of the collision avoidance potential should be zero when the shortest distance d_i between two manipulator links is above an application specific safety distance $d_{a,i}$. According to Schwienbacher [2013], the combination of a quadratic H_{quad} and cubic H_{cubic} polynomial for the potential function of two collision pairs and a parameter $t \in [0,1]$ yields very promising results in practice:

$$H_{c,i} = \begin{cases} 0 & \text{for } d_i > d_{a,i} \\ H_{cubic} & \text{for } t d_{a,i} \leq d_i < d_{a,i} \\ H_{quad} & \text{for } 0 \leq d_{a,i} < t d_{a,i} \end{cases} \quad (5.57)$$

The coefficients of the two polynomials are computed with the boundary conditions ($H_{c,i}(0) = H_0$ and $H_{c,i}(d_{a,i}) = 0$) as well as the transient conditions of continuity at the borders of the piecewise defined functions and zero gradient and curvature at $d_{a,i}$. Please refer to Schwienbacher [2013] for further details.

The suitability of the algorithm to avoid self-collisions for redundant robots has been shown for the first agricultural manipulator prototype in Baur et al. [2013]. In Figure 5.13 the functioning of the algorithm is demonstrated in a sequence of pictures, representing two simulated motions of the final CROPS manipulator in the 7-DoFs configuration. The TCP of the robot should move along a straight line path, which is indicated in red. In the upper sequence of Figure 5.13 the collision avoidance algorithm is inactive during the motion and one can observe a collision in the pictures ② and ③ with a spherical obstacle which has been added as a collision object into the manipulator workspace. By using the here described

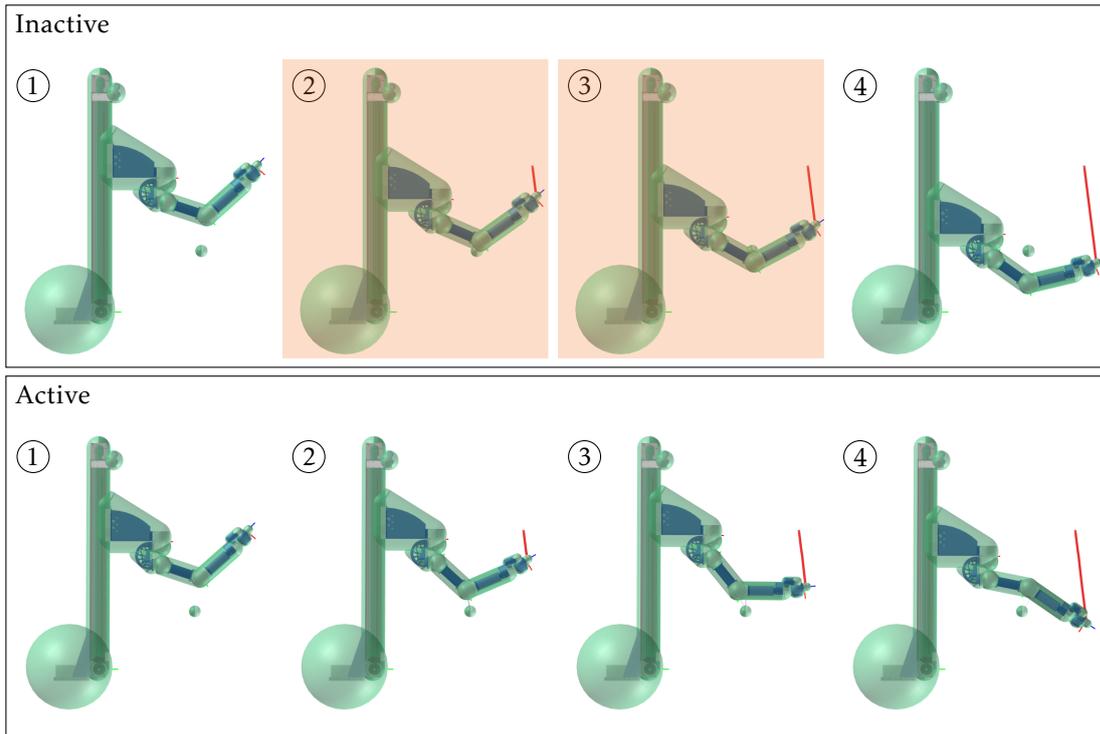


Figure 5.13: Two picture sequences, representing the same straight line motion of the TCP with a sphere obstacle in the manipulator workspace of the 7-DoFs agricultural manipulator. In the upper sequence the collision avoidance algorithm is inactive while in the lower sequence the collision avoidance is active. Frames with collisions are highlighted in red.

algorithm in the lower sequence of pictures, the collision with the obstacle can be avoided, while the TCP is still tracking the reference trajectory.

Maximization of Manipulability

Another common secondary objective suitable for null space optimization is related to the manipulability measure. As already presented in Baur et al. [2012], the author proposes the summation of the following two cost functions

$$H_m = H_{m,1} + H_{m,2} \quad (5.58)$$

and its meaning will be discussed in this Section. The manipulability measure was initially introduced by Yoshikawa [1985] and it can be used to keep the manipulator away from singular configurations. It is defined by³:

$$H_{m,1} = -\sqrt{\det(J_w J_w^T)} \quad (5.59)$$

³ Originally, the manipulability measure has been defined without the minus sign. It is added to the function $H_{m,1}$ since the secondary objectives in (5.46) are minimized.

At singular configurations $H_{m,1}$ vanishes and the solution to the inverse kinematics according to (5.46) degenerates. However, such a scenario can be handled with a singularity robust inverse kinematics algorithm as it will be presented in Section 5.4.4. The manipulability is a measure which depends on the joint configuration. It describes the ability of the TCP to change its pose in the robot workspace and can be geometrically visualized by so-called manipulability ellipsoids for task space dimensions $m \leq 3$. The volume of the manipulability ellipsoids is proportional to the manipulability measure (5.59). Exemplary and according to Siciliano et al. [2009], in Figure 5.14a the manipulability ellipsoids for a 2-DoFs planar robot are drawn in different configurations. Apparently, the ability for motions of the TCP along certain directions strongly depend on the joint configurations. For example, when the arm is fully stretched according to Figure 5.14a, a kinematic singularity occurs and motion in the x -direction is impossible. Utilization of the redundant

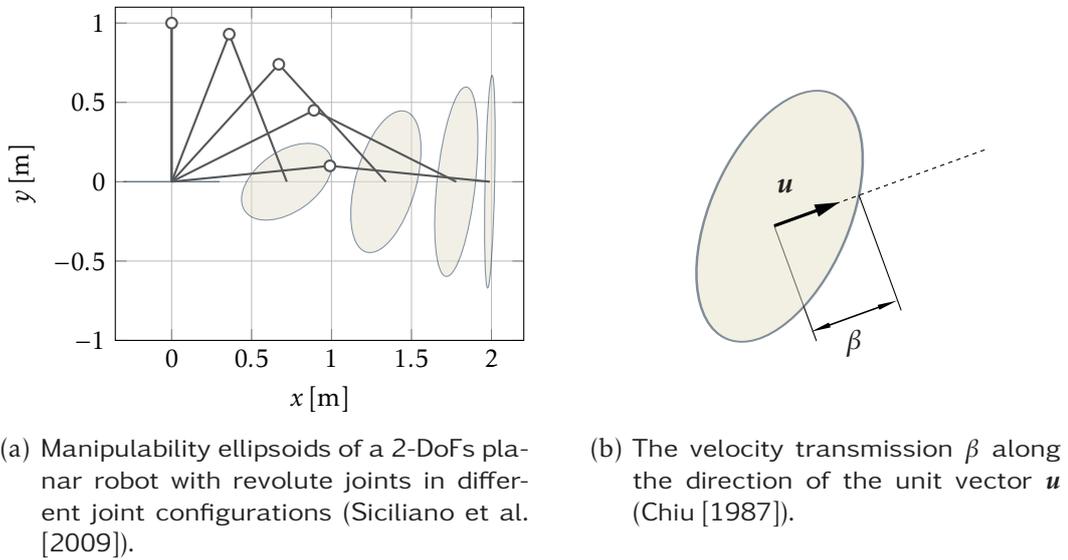


Figure 5.14: Manipulability ellipsoids.

DoFs to maintain configurations which provide high manipulability can therefore be considered as a desirable objective. On the other hand, in some cases it might be necessary to move the manipulator along a certain preliminary known direction in the robot workspace. For the given unit vector $\mathbf{u} \in \mathbb{R}^m$, Chiu [1987] derived an equation to obtain the scalar value β which he termed the velocity transmission ratio:

$$\beta = \left(\mathbf{u}^T (\mathbf{J}_w \mathbf{J}_w^T)^{-1} \mathbf{u} \right)^{-\frac{1}{2}} \quad (5.60)$$

Basically, by the vector $\beta \mathbf{u}$ a point from the origin to the surface of the manipulability ellipsoid is specified (cf. Figure 5.14b) and hence β provides a measure indicating the ability of the TCP to move in the direction \mathbf{u} . The velocity transmission ratio along that application specific direction \mathbf{u} is maximized by the

minimization of the objective function:

$$H_{m,2} = \mathbf{u}^T (\mathbf{J}_w \mathbf{J}_w^T)^{-1} \mathbf{u} \quad (5.61)$$

For the implementation in the inverse kinematics algorithm (5.46) not the objective functions (5.59) and (5.61) are required but their gradients. The derivative of (5.59) with respect to the i -th element of the vector \mathbf{q} is given by (Park et al. [1999]):

$$\frac{\partial H_{m,1}}{\partial q_i} = H_{m,1} \text{trace} \left[(\mathbf{J}_w \mathbf{J}_w^T)^{-1} \frac{\partial \mathbf{J}_w}{\partial q_i} \mathbf{J}_w^T \right] \quad (5.62)$$

The required derivative of the Jacobian $\frac{\partial \mathbf{J}_w}{\partial q_i}$ can be obtained during the recursive direct kinematics computation. With the matrix identity for an invertible matrix \mathbf{A}

$$\frac{\partial \mathbf{A}^{-1}}{\partial x} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{-1} \quad (5.63)$$

the expression for the gradient of (5.61) of the i -th vector element $\frac{\partial H_{m,2}}{\partial q_i}$ can be obtained:

$$\frac{\partial H_{m,2}}{\partial q_i} = -\mathbf{u}^T (\mathbf{J}_w \mathbf{J}_w^T)^{-1} \left[\frac{\partial \mathbf{J}_w}{\partial q_i} \mathbf{J}_w^T + \mathbf{J}_w \left(\frac{\partial \mathbf{J}_w}{\partial q_i} \right)^T \right] (\mathbf{J}_w \mathbf{J}_w^T)^{-1} \mathbf{u} \quad (5.64)$$

The application of the concept of manipulability was very useful to evaluate the suitability of the kinematic design of the CROPS manipulators at an early stage of the development and a great part of it has already been presented in Baur et al. [2012]. Exemplary, the manipulability in Figure 5.15 has been evaluated two times according to the task description in Section 3.2 with the first manipulator prototype. One time, the inverse kinematics has been solved without the local optimization of the manipulability (indicated by light gray ellipsoids) and another time, the manipulability and motion direction of the end-effector towards the stem center has been optimized (indicated with the ellipsoids in orange). In some cases, the ellipsoids without optimization of the manipulability indicate that a motion towards the stem center is not feasible, while with the local optimization the ellipsoids always indicate the ability of the robot to move the end-effector to the stem center and the goal.

However, for the application with the real manipulator one has to consider the effects on the velocity control of the system. If it is desired to apply large TCP velocities along a certain direction in the workspace, a high velocity transmission ratio is advantageous. But due to this high transmission, little variations in joint speeds have a big impact on end-effector velocities. Therefore, for an accurate control, a low transmission ratio is preferable since the sensitivity is higher. From the control point of view, the selection of \mathbf{u} must be considered carefully and the user must be aware that the accuracy of the velocity tracking is decreased by increasing the transmission ratio β .

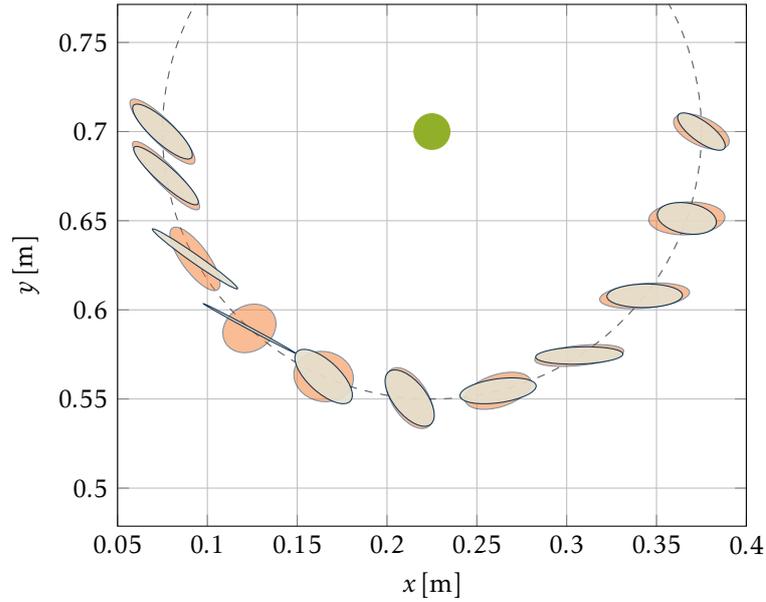


Figure 5.15: Manipulability ellipsoids for several TCP positions distributed in a half circle (dashed line) around a sweet pepper stem (green circle) for the task described in Section 3.2. The ellipsoids in light gray are without and the ellipsoids in orange are with the local optimization of the manipulability. For clarity the ellipsoids have been scaled to an appropriate size.

5.4.4 Singularity Robust Inverse Kinematics

The problem of singularities is an intrinsic problem to the inverse kinematics computation according to (5.46). For non-redundant manipulators, a mapping of the workspace velocities to joint speeds can be achieved by (5.44), only if the workspace Jacobian $J_w(\mathbf{q})$ is non-singular. Therefore, singular configurations appear if the Jacobian becomes singular, i.e. $\det J_w(\mathbf{q}) = 0$. For redundant manipulators, singular configurations are observed at configurations that reduce the rank of $J_w(\mathbf{q})$. It is worth mentioning that problems arise not only at the singular points but also in their neighborhood since in those regions small end-effector velocities $\dot{\mathbf{w}}_d$ require very high joint speeds $\dot{\mathbf{q}}_d$. Besides singularities at the boundary of the workspace, inner singularities in the reachable workspace can occur. Similar to (5.46) and (5.47), Nakamura and Hanafusa [1986] as well as Wampler [1986] derived independently a singularity robust inverse kinematics algorithm for redundant manipulators based on the damped least-squares inverse of the Jacobian J_w^* including the diagonal weighting matrix \mathbf{W} and a scalar damping parameter λ :

$$\dot{\mathbf{q}} = J_w^* \dot{\mathbf{w}}_d - \alpha (\mathbf{I} - J_w^* J_w) \left(\frac{\partial H}{\partial \mathbf{q}} \right)^T \quad (5.65a)$$

$$J_w^* = \mathbf{W}^{-1} J_w^T (J_w \mathbf{W}^{-1} J_w^T + \lambda^2 \mathbf{I})^{-1} \quad (5.65b)$$

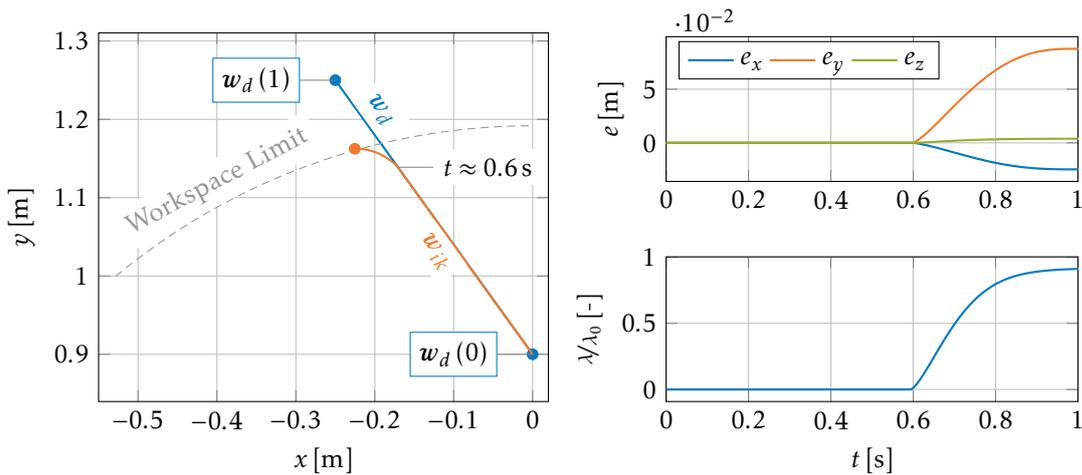
It is worth mentioning that the damped least-squares or singularity robust inverse is based on the solution to the following minimization of ϕ with respect to $\dot{\mathbf{q}}$:

$$\phi = \frac{1}{2} (\dot{\mathbf{w}}_d - \mathbf{J}_w \dot{\mathbf{q}})^T (\dot{\mathbf{w}}_d - \mathbf{J}_w \dot{\mathbf{q}}) + \frac{1}{2} \lambda^2 \dot{\mathbf{q}}^T \dot{\mathbf{q}} \rightarrow \min! \quad (5.66)$$

With the damping factor λ the robustness of the inverse kinematics algorithm can be tuned. While the robustness is increasing with higher values for λ the accuracy of the TCP tracking decreases. A detailed error analysis depending on λ was carried out by Nakamura [1991]. Furthermore, he suggests the usage of an adaptive damping factor, depending on the manipulability measure (5.59), which is non-zero at the proximity to singularities while it vanishes in regions far and away from singularities. By adapting the damping factor according to the condition of the Jacobian, an exact solution of the inverse kinematics far from singularities is provided, while close to kinematic singularities a stable, but less accurate solution can be maintained. With $\bar{m} := -H_{m,1}$, a common choice for the adaptive damping factor is provided by:

$$\lambda = \begin{cases} \lambda_0 \left(1 - \frac{\bar{m}}{\bar{m}_0}\right)^2 & \text{for } \bar{m} < \bar{m}_0 \\ 0 & \text{else} \end{cases} \quad (5.67)$$

with the application specific tuning parameter λ_0 and \bar{m}_0 . The resulting singularity robust inverse kinematics algorithm and its behavior applied to the CROPS manipulator are illustrated in an representative example scenario in Figure 5.16. According to Figure 5.16a the TCP of the manipulator should move along a straight line path \mathbf{w}_d in the task space. Worth mentioning, the goal coordinates of the desired trajectory are not within the reachable workspace of the robot. Although the task space definition of the TCP position contains all three dimensions, only



(a) The desired and actual path of the TCP. Only the projection into the (x, y) -plane of the workspace is shown. (b) Corresponding plots of the workspace tracking error (top) and the normalized adaptive damping factor λ (bottom).

Figure 5.16: An example for the singularity robust inverse kinematics algorithm.

the projection in the (x, y) -plane is plotted for clarity. At first, the damping factor λ is zero and the resulting inverse kinematics solution w_{ik} is therefore accurately tracking the desired TCP pose. However, at approximately $t \approx 0.6$ s the TCP approaches the workspace boundary which corresponds to a kinematically singular configuration. Due to the increase of the adaptive damping λ around that time (cf. lower plot in Figure 5.16*b*), an error in the computation is introduced but the stability of the algorithm is maintained. This is very well represented by the workspace error illustrated in the upper plot of Figure 5.16*b*.

The here introduced singularity robust inverse kinematics algorithm provided good results concerning the computational efficiency and tracking accuracy. Nevertheless, there are other methods described in the literature and a brief overview will be given in the following. With the damped-least squares method and the scalar parameter λ , a damping is uniformly added on all singular values and therefore an error is introduced in any direction. However, it is possible by a more detailed analysis of the Jacobian, using a Singular Value Decomposition (SVD), to replace the identity matrix I in (5.65*b*) by a matrix which allows for selective filtering along degenerated directions of the end-effector motions. According to Klein and Blaho [1987], the smallest singular value is the only accurate measure to detect the proximity to singularities. And based on an estimate of this smallest singular value, Maciejewski and Klein [1988] derived a method to selectively filter directions of TCP motion, which belong to the lost end-effector motion directions. Since the numerical calculation of a SVD is computational expensive and in general unsuitable for the real-time application, Maciejewski and Klein [1989] discuss an algorithm, exploiting previously calculated singular values of the Jacobian and assuming little changes in two consecutive time steps. A more recent review on inverse kinematics algorithms and their behavior close to singularities is given by Chiaverini et al. [1994]. He used an algorithm which is considering the degenerated end-effector motion directions while a user-defined accuracy can be specified. The selection of the damping parameter is based on an estimate of the two smallest singular values.

5.5 Chapter Summary

In this Chapter, the concept of the motion planning algorithm, implemented for the agricultural manipulator has been introduced. It is based on a decoupled planning approach in the manipulator workspace and an efficient inverse kinematics algorithm for kinematically redundant manipulators. Two different approaches for the workspace planning were considered and evaluated. The inverse kinematics algorithm can be applied to locally optimize secondary objectives in the null space of the manipulator. A detailed overview of the investigated objective functions has been provided in this Chapter. Finally, possible adaptations to handle kinematic singularities were addressed. The overall scheme is running in real-time, hence it is suitable for on-line motion planning.

6 Field Experiments and Results

According to the brief description of the European research project CROPS in Chapter 2, the developed agricultural manipulators should be integrated in three different harvesting applications and one precision spraying task. Many subsystems developed in the course of this interdisciplinary project as well as several partners were involved in the experiments. However, for each application, the manipulators with their motion planning algorithms are an integral part of the autonomous robot. The scope of this Section is to describe the most significant system components utilized for the different agricultural applications and provide the reader an idea on the challenges during the system integration and the complexity of the overall task.

6.1 Selective Harvesting

The selective harvesting applications are very challenging for the autonomous robot and many different modules must interact with each other for successful operation. Besides the agricultural manipulator, other important system components are the harvesting end-effector and the sensor system. In the following Subsections these hardware components with clear references to the responsible partners are introduced and the results of the field experiments are presented.

6.1.1 Sweet Pepper Harvesting

The experiments for sweet pepper harvesting were carried out by a team from the WAGENINGEN UNIVERSITY AND RESEARCH CENTER (Netherlands). The responsible person for the WP: *Sweet Pepper – Protected Cultivation* (see Section 2.2) was JOCHEN HEMMING. The following references provide detailed information on the experiments and results: Bac et al. [2015]; Hemming et al. [2014]. Figure 6.1 presents pictures of the first and final manipulator prototype during field tests in commercial sweet pepper greenhouses. The earliest experiments were carried out with the first manipulator prototype (cf. Figure 3.35a) in April 2012 to gain experience in the integration process, validate the interfaces and verify the suitability of the manipulator kinematics and hardware design for the sweet pepper harvesting application. The complete integrated and autonomously operating system has been tested in June 2014 (see Figure 6.1a and Figure 6.1c). The test set-up and major results of these experiments are in short summarized here.

The autonomous robot had the following most significant modules: Two different harvesting end-effectors, the 9-DoFs final manipulator prototype (cf. Figure 3.3b), and a sensor rig with lighting. All system components were mounted on a mobile platform designed by JENTJENS MACHINETCHNIEK BV (Netherlands).



(a) Integrated system for sweet pepper harvesting.



(b) First prototype.



(c) Final prototype (Picture: Hemming et al. [2014]).

Figure 6.1: Field experiments for sweet pepper harvesting in a greenhouse. The fin ray type end-effector is mounted in all pictures.

One end-effector, in the following referred to as fin ray type, was developed by FESTO (Germany) and is equipped with four adaptive gripper jaws, based on the Fin Ray Effect® (Gauchel and Saller [2012]). Above the jaws a scissor-like cutting mechanism has been mounted to cut through the peduncle. The other developed harvesting tool, so-called lip type end-effector, was developed by BART VAN TUIJL and EHUD WAIS. It fixates the fruit with a suction cup while two lips encircle the fruit. The lower lip will move until it reaches the backside of the peduncle and the upper lip, equipped with a knife, cuts through the peduncle from the front side. Both end-effectors are pneumatically actuated. Furthermore, a color camera¹ and a small Time-of-Flight (ToF) camera² are mounted on each end-effector for close-range fruit localization, respectively. The sensor rig (cf. right hand side in Figure 6.1a), located on the platform, was composed of two color cameras³ for stereo vision and a ToF camera⁴ to gain additional depth information of the environment. The detection algorithm were developed by the WAGENINGEN UNIVERSITY AND RESEARCH CENTER and provided the fruit coordinates as well as an approximation of the plant's main stem location (Bac et al. [2014a]).

For the experiments in a commercial sweet pepper greenhouse in the Netherlands, the platform was manually positioned in front of the canopy and the robot operated autonomously until a harvesting attempt was performed for each of the detected fruits. According to Bac et al. [2015], in an unmodified cultivation system, the robot only harvested 6 % with the fin ray type end-effector and 2 % with the lip type end-effector of the total amount of investigated fruits. In a simplified cultivation system with the removal of leaves and fruit clusters, the harvesting success improved to 26 % (fin ray type) and 33 % (lip type). Notably, only 176 fruits were investigated in all four experiments. Thus the results are not statistically significant, but they provide a tendency of the system's capability.

6.1.2 Apple and Grape Harvesting

Within the CROPS project the responsible partner for the WP: *Harvesting Systems in Orchards: Grapes and Apples* (cf. Section 2.2) was WOUTER SAEYS. He and his team from KU LEUVEN (Belgium) carried out the harvesting experiments in Belgian orchards. Similar to the sweet pepper harvesting application, initial tests with the first manipulator prototype were performed from July until November 2012 to validate interfaces and the manipulator design. The experiments with the fully integrated system were performed in the time period from July 2013 until September 2014. In the following the test set-ups of the harvesting robot for the apple and grape applications are summarized and significant results are highlighted.

For the apple harvesting application, the intention is to pick and remove single fruits, while for the grape harvesting the whole bunch of grapes should be grasped and cut at the stalk. Therefore, two end-effectors were developed by FESTO for

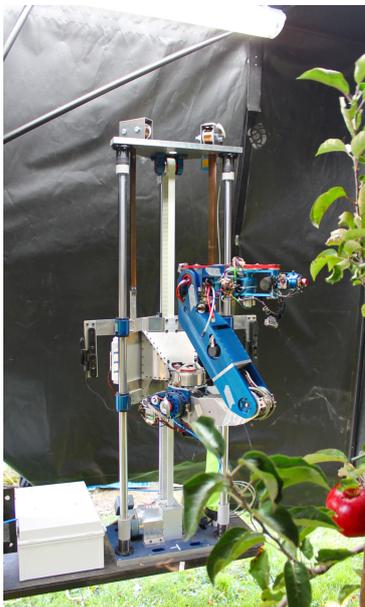
1 Type: VRmMS-12, Company: VRMAGIC HOLDING AG, Germany

2 Type: pmd[vision]@CamBoard nano, Company: PMDTECHNOLOGIES GMBH, Germany

3 Type: Prosilica GC2450, Company: ALLIED VISION TECHNOLOGIES GMBH, Germany

4 Type: SR4000, Company: MESA IMAGING AG, Switzerland

the different harvesting applications. The apple gripper design was reported by Gauchel and Saller [2012]. The gripping is achieved with two specialized jaws. On each jaw, an elastomer membrane, enclosing a granulate, is mounted. Upon gripping, the granulate is becoming distorted according to the shape of the apple and by evacuating the membrane, the outer membrane geometry is stabilized and a good grip can be achieved. The grape harvesting end-effector is a combined tool for cutting and grasping the stalks of grape bunches with a parallel stroke. Both end-effectors are pneumatically driven and they are equipped with the same sensor system as the sweet pepper harvesting end-effectors, i.e. a color and a small ToF camera. The experiments were carried out with the first manipulator prototype in the 9-DoFs configuration (cf. Figure 3.35a and Figure 6.2a). For the fruit and obstacle detection and localization, two Xtion PRO LIVE sensors from the company ASUS were utilized. In each device, a RGB camera and an infrared sensor are integrated. The integrated sensors were mounted on the first element of the manipulator moving vertically according to the motion of the prismatic joint. Detection and localization algorithms were developed by KU LEUVEN (Nguyen et al. [2014]). To avoid the exposure to direct sunlight and to maintain well-defined lighting conditions for the vision based sensors, the complete system is mounted on a trailer which is surrounded by a tarpaulin. The trailer is moved along the espalier canopy with a tractor from the company CNH INDUSTRIAL (cf. Figure 6.2b).



(a) First prototype.



(b) Harvesting platform.

Figure 6.2: Field experiments for apple harvesting in an orchard.

According to an internal project report by WOUTER SAEYS, the outdoor apple harvesting experiments showed that the sensor system is able to detect all apples in the harvesting area. The robot is capable of grasping 90 % of the detected apples while 72 % were successfully detached.

The grape harvesting experiments were carried out in September 2014. Unfortunately, the experiments were unsuccessful because a reliable stem localization was not possible.

6.2 Selective Spraying of Grapevine

The selective or precision spraying experiments of grapevine diseases were carried out at the UNIVERSITY OF MILAN (Italy) in the beginning of 2013. ROBERTO OBERTI, as the person responsible for the WP 7: *Precision Spraying* (cf. Section 2.2) within the CROPS project, and his team performed the tests in an experimental greenhouse. A detailed description of the experimental results is provided in Oberti et al. [2013, 2014] and major findings are highlighted in this Section.

In the test set-up, grapevine plants were grown in pots and arranged on a table in a very similar fashion compared to typical grapevine cultivation under outdoor conditions. Some plants were intentionally infected by a disease (powdery mildew) while the exact location has been marked by the visual observation of a plant pathologist for reference. The autonomous precision spraying system was composed of the agricultural manipulator (first prototype) in the 6-DoFs configuration (refer to Figure 3.2a), a precision spraying end-effector and a multispectral camera⁵. Figure 6.3a illustrates the manipulator with a water proof protection cover and the complete system integrated on a wheeled platform in the experimental greenhouse is show in Figure 6.3b. The manipulator task space w for

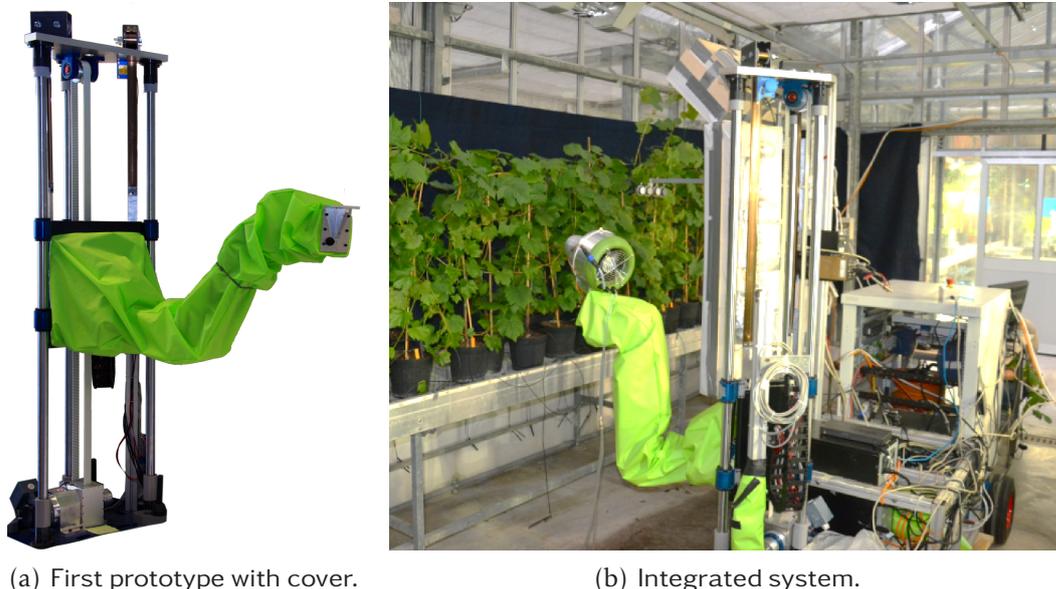


Figure 6.3: Field experiments for precision spraying of grapevine.

this application was defined according to (3.1) and the inverse kinematics were calculated as described in Section 5.4. The planning of the task space trajectory is straightforward since no collision objects are in the robot workspace. The spraying

⁵ Type: MS4100, Company: DUNCANTECH, USA

end-effector was designed by MARKO HOČEVAR and his group from the UNIVERSITY OF LJUBLJANA (Slovenia) and is described in Malnersic et al. [2012]. For the experiments, the nozzle provided a flow rate of 30 ml/min while the air flow velocity was controlled by a fan in the range of 5-30 m/s. The precision sprayer generates a jet of fluid with a circular spray pattern and constant diameter of 0.15-0.2 m along a distance ranging from 0.4-1.5 m. The valve of the nozzle and the fan speed are controlled by the digital and analog outputs of a microcontroller via the CAN board and the real-time control unit with the host computer (cf. Figure 3.26). The disease detection algorithm was developed by the UNIVERSITY OF MILAN. It is based on the chlorophyll breakdown of infected leaves and stems and the resulting changes in reflections. In the illustrative experiments, the platform was moved along the canopy wall of 5 m length and 1.8 m height. Based on the results from the sensor system, the spraying end-effector applied fluids on the detected areas from various directions to improve the spray coverage. Although the detection system provided some false positives, the amount of required pesticides was remarkably reduced by 84 % compared to conventional uniform treatment of the canopy.

7 Conclusion and Outlook

In the beginning of this Chapter, a brief summary of the main achievements of this thesis is provided. Finally, a discussion of the presented contents, and based on my personal experience gained during the design of the agricultural manipulators, some recommendations for future work are stated.

7.1 Summary

This thesis presented the simulation, design and motion planning algorithms of modular agricultural manipulators for selective harvesting and precision spraying applications. Starting from the year 2010 until the beginning of 2014, three manipulators were designed and built in total at the Institute of Applied Mechanics (AM). The first two manipulators are rather identical while the third (final) manipulator is a strongly enhanced prototype, in particular with respect to the modularity, performance, robustness, and the electronics architecture.

The initial challenge in the design of the manipulator was the derivation of suitable kinematics. To deal with the various tasks, a modular design approach has been chosen. However, due to the high variety of agricultural products, in particular for the harvesting applications, a very dexterous robot with 9-Degrees of Freedom (DoFs) was proposed. This unique manipulator kinematics combine a high amount of DoFs on a very narrow space for an extraordinary flexibility. For the task space, defining a pose in three dimensions, the manipulator is kinematically redundant. An evaluation of the kinematics has been performed with the manipulability measure.

At the beginning of the project, Pneumatic Artificial Muscles (PAMs) were theoretically and experimentally investigated on a test bed as actuator candidates for the agricultural manipulators. Although, this preliminary study revealed satisfactory results, the PAM drive concept has not been followed, mainly due to the inherent limited contraction capabilities of these linear actuators and the consequential limited joint ranges.

In accordance to the modular hardware, a modular software for the simulation of all developed prototypes and its different kinematic configurations has been implemented. Notably, the topology of all prototypes is described by open-chain kinematics. Besides the recursive kinematic computations, the implemented algorithms cover the multibody dynamics of the system. In addition to the multibody dynamics, the dynamics of the electrically driven actuators, the Harmonic Drive gear friction for revolute joints, and the experimentally determined friction of the linear bearing for the prismatic joints were taken into account.

To experimentally characterize the speed and load dependent friction of major robot drive modules of the final manipulator prototype, a modular test bed has

been designed. Apart from the measurements of the drive characteristics, the test bed was of particular importance during the start-up of the components and the validation of the mechanical design. The experimental results indicated substantially higher friction torques in the robot drive modules compared to the modeling solely based on the catalog data provided by the harmonic drive gear manufacturer. Partially, this can be explained by additional friction losses in the drive chain of the modules. A friction law, similar to a model from the literature, has been fitted very accurately to the measurements and therefore enhanced the dynamic simulation of the final manipulator prototype.

With the load on the structure, obtained by the simulation of the dynamics for several reference trajectories, the mechatronic design of the hardware components was iteratively derived, based on the Computer-Aided Design (CAD) modeling and actuator selection of JULIAN PFAFF. According to his mechanical design, the parameter for the simulation models were initially obtained from the CAD software tool and utilized for the simulation. In each iteration, the manipulator design was optimized with respect to a lightweight and stiff structure, as well as a high dynamic performance.

Another major part of this thesis is the work on the motion planning. To allow for real-time capabilities, a hierarchical design with a decoupled workspace planning on the highest level and an inverse kinematics on the following level was proposed. The unique workspace planner exploits known facts of the unstructured environment and provides a task space trajectory with high probability for a successful approach towards the target. Successful in this context means collision-free with relevant objects and suitable with respect to end-effector constraints. The inverse kinematics, according to Liégeois [1977] with local optimization of the redundant DoFs yields the corresponding joint trajectory. The selection of appropriate secondary objective functions to exploit the redundant DoFs to provide an optimized configuration was the particular focus on the implementation of the inverse kinematics algorithm. Remarkably, the presented algorithms were successfully applied by project partners in field and laboratory experiments.

Finally, the developed agricultural manipulators were evaluated in field experiments. These experiments were carried out by project partners in Italy, Belgium, and the Netherlands. Besides a reliable hardware design (described in Pfaff et al. [2012, 2014]), this required a well integrated, robust, easy-to-use and fail-safe software design. The central part of the manipulator application was implemented on a dedicated computer, the real-time control unit. Apart from running the algorithms and handling the low-level communication for proper operation of the manipulator, it was constantly monitoring the state of the hardware components as well as the risk of self-collisions or joint limit violations. Via an Ethernet connection, the real-time control unit has been interfaced to the Robot Operating System (ROS) framework, the common interface for any subsystem within the project. For the start-up, monitoring, testing of components, and debugging of the manipulator prototype, a Graphical User Interface (GUI) was programmed. Through the ROS framework each partner was able to operate the manipulator and perform an integration with the application specific highest level program, the supervisory control system. The first manipulator prototype has been integrated

for the sweet pepper, apples and grape harvesting applications in the 9-DoFs kinematic configuration, while the final manipulator prototype was tested in the 7- and 9-DoFs configurations for the sweet pepper harvesting application. The precision spraying experiments of grapevine were performed with the 6-DoFs kinematic configuration of the first manipulator prototype.

7.2 Discussion and Recommendations for Future Work

The contents discussed in this thesis covered diverse subjects in the field of robotics, in particular concerning the design of manipulators and motion planning algorithms. However, many additional topics, like the expertise in agricultural tasks, sensing or end-effectors, must be considered for the successful implementation of an autonomous agricultural robot in this challenging and interdisciplinary domain. To my knowledge, so far no custom designed manipulator exists for agricultural tasks that has been integrated and evaluated in four different applications. However, I want to point out that only demonstrators on the prototype level have been developed and the steps towards a commercial product which has an overall high reliability and an improved success rate are, in my opinion, still facing open questions in ongoing research and many improvements are required. Based on the gained experience during my work, I would like to present some ideas and directions for future research.

For the selective harvesting applications an individual trajectory has to be planned for each fruit. For time efficiency, only planners that have real-time planning capabilities were considered. However, **sample based motion planners**, like the Rapidly-Exploring Random Trees (RRT), are suitable to solve multiple-query path planning problems in a reasonable amount of time. Due to the suitability for parallelization and with the assumption of increasing computational power in the future, these planners might become promising.

The proposed decoupled and well established planning approach, based on the derivation of a task space trajectory in the manipulator workspace and the inverse kinematics computation is very time efficient. However, due to kinematic constraints of the manipulator, the workspace planner might provide a task space trajectory that is infeasible for the robot. One possible improvement of the currently implemented framework, might be to allow for **on-line modifications of the task space trajectory** during path execution, for example by a projection of a cost functional, similar to the secondary objective function and its projection in the null space, directly in the task space. Challenges must be overcome in the definition of the admissible error and how to compensate this task space tracking error at the proximity of the target. Another approach to increase the systems capability is to **relax the task space definition**. For example, it might still be possible to grasp a fruit with minor orientation errors. This could be accomplished by a redefinition of the task space description while a minimization of the orientation error is achieved by utilizing the redundant DoFs and a proper secondary objective function.

The inverse kinematics algorithm performs a local optimization of the joint configuration. However, it does not consider available information of the future course of the task space trajectory and disadvantageous configurations might be the consequence. Currently, global optimization of the whole trajectory can not be performed in real-time for the here discussed manipulators, therefore Schuetz et al. [2014] proposed the application of **model predictive control** algorithms with a moving time horizon.

For agricultural manipulators reliable sensing is still challenging. Within the CROPS project the object localization was performed with vision sensors and image processing algorithms. However, fruits might be occluded and not entirely visible in the direct line of sight. In particular for the selective harvesting applications, a lot of soft objects like leaves, branches, and fruits lead to a highly cluttered environment. If these soft objects are considered as obstacles, a collision-free path would often be impossible. Therefore, collision with these soft objects must be allowed, but to avoid plant damages the contact forces should be minimized. This requires force feedback obtained from **tactile or torque sensors** in the joints. The measurement of the resulting contact forces on each robot segment might be a promising approach. These contact forces can be integrated in the introduced inverse kinematics algorithm and minimized by exploitation of the redundant DoFs. Model based estimation of contact forces might be applied allowing for a prediction of contact forces and the implementation of predictive control algorithms.

The concept of **modularity** for agricultural manipulators is a promising idea, because there are many different tasks, besides harvesting or spraying, during one season. Even for one single task, the modularity can be beneficial, for example if the requirements on the manipulator workspace are changing, because of plant growth or seasonal variations in the cultivation. However, in general, automation for a single task of a single crop still remains a specialized and unique solution. In my opinion, a successful agricultural robot can only be designed when looking at the complete task in its entirety from a technical as well as a biological point of view.

Appendix A

Basic Technical Data

First Manipulator Prototype

Table A.1: Main joint components. All gears are from the company Harmonic Drive

Joint	Actuator		Gear	
	Type	Torque Constant [Nm/A]	Type	Ratio [-]
1	RD 115x50	$5.60 \cdot 10^{-1}$	n/a	
2	Maxon EC-i-40	$3.15 \cdot 10^{-2}$	CSD-17	100
3	Maxon EC-45	$7.04 \cdot 10^{-2}$	CSG-25	160
4	Maxon EC-45	$7.04 \cdot 10^{-2}$	CSG-25	160
5	Maxon EC-max-40	$5.00 \cdot 10^{-2}$	CSD-20	160
6	Maxon EC-i-40	$3.15 \cdot 10^{-2}$	CSD-14	100
7	Maxon EC-45-flat	$1.31 \cdot 10^{-1}$	CSD-14	100
8	Emoteq HT 01001	$3.46 \cdot 10^{-2}$	CSD-14	100
9	Emoteq HT 00801	$1.00 \cdot 10^{-2}$	HFUC-8	100

Final Manipulator Prototype

Table A.2: Main components of the manipulator joints. All actuators are from the company TQ-Systems (formerly RoboDrive) and all gears are from Harmonic Drive.

Joint	Actuator		Gear		Modul Index
	Type	Torque Constant [Nm/A]	Type	Ratio [-]	
1	115x50	$5.6 \cdot 10^{-1}$	n/a		n/a
2	85x13	$1.3 \cdot 10^{-1}$	CPL-32	50	A.4
3	85x13	$1.3 \cdot 10^{-1}$	CPL-32	100	A.2
4	85x13	$1.3 \cdot 10^{-1}$	CPL-32	100	A.3
5	50x08	$7.5 \cdot 10^{-2}$	CSD-20	100	B.1
6	50x08	$7.5 \cdot 10^{-2}$	CSD-20	100	B.2
7	38x06	$2.0 \cdot 10^{-2}$	CSD-14	100	C.1
8	38x06	$2.0 \cdot 10^{-2}$	CSD-14	100	C.2
9	38x06	$2.0 \cdot 10^{-2}$	CSD-14	100	C.3

Appendix B

Manipulator Kinematics

A few remarks on the nomenclature:

- Location, velocity, and acceleration vectors for the origin of the body fixed frame belonging to the segment i : $\mathbf{r}_i, \dot{\mathbf{r}}_i, \ddot{\mathbf{r}}_i$.
- Angular velocity and acceleration vectors of the body i : $\boldsymbol{\omega}_i, \dot{\boldsymbol{\omega}}_i$.
- Rotation matrix transforming an arbitrary vector ${}_p\mathbf{x}$, given in the frame p into the frame i : \mathbf{A}_{ip}
- The subindex on the left hand side of the kinematic quantities indicates the frame of reference.
- The tilde operator for a vector $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$ is defined by:

$$\tilde{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (\text{B.1})$$

Recursive Computation

Basic kinematic quantities of the body fixed frame i with respect to the kinematics of its parent link p (compare Figure 3.20) are:

$${}_i\mathbf{r}_i = \mathbf{A}_{ip} ({}_p\mathbf{r}_p + {}_p\mathbf{r}_{rel}) \quad (\text{B.2})$$

$${}_i\dot{\mathbf{r}}_i = \mathbf{A}_{ip} ({}_p\dot{\mathbf{r}}_p + {}_p\dot{\mathbf{r}}_{rel}) \quad (\text{B.3})$$

$${}_i\ddot{\mathbf{r}}_i = \mathbf{A}_{ip} ({}_p\ddot{\mathbf{r}}_p + {}_p\ddot{\mathbf{r}}_{rel}) \quad (\text{B.4})$$

$${}_i\boldsymbol{\omega}_i = \mathbf{A}_{ip} {}_p\boldsymbol{\omega}_p + {}_i\boldsymbol{\omega}_{rel} \quad (\text{B.5})$$

$${}_i\dot{\boldsymbol{\omega}}_i = \mathbf{A}_{ip} {}_p\dot{\boldsymbol{\omega}}_p + {}_i\dot{\boldsymbol{\omega}}_{rel} \quad (\text{B.6})$$

$${}_i\mathbf{J}_{R,i} = \mathbf{A}_{ip} {}_p\mathbf{J}_{R,p} + {}_i\mathbf{J}_{R,rel} \quad (\text{B.7})$$

$${}_i\mathbf{J}_{TO,i} = \mathbf{A}_{ip} ({}_p\mathbf{J}_{TO,p} + {}_p\mathbf{J}_{T,rel}) = \mathbf{A}_{ip} {}_p\mathbf{J}_{TO,p} + {}_i\mathbf{J}_{T,rel} \quad (\text{B.8})$$

The index *rel* indicates a relative measure between the parent link p and the reference link i .

Relative Kinematics

Depending on joint type, the relative kinematics are different and provided in the following¹:

1. Revolute Joints:

$${}^p\mathbf{r}_{rel} = \mathbf{r}_{const} \quad (\text{B.9})$$

$${}^p\dot{\mathbf{r}}_{rel} = {}_p\tilde{\omega}_p {}^p\mathbf{r}_{rel} \quad (\text{B.10})$$

$${}^p\ddot{\mathbf{r}}_{rel} = \frac{d}{dt}({}^p\dot{\mathbf{r}}_{rel}) = ({}_p\dot{\tilde{\omega}}_p + {}_p\tilde{\omega}_p {}_p\tilde{\omega}_p) {}^p\mathbf{r}_{rel} \quad (\text{B.11})$$

$${}^i\boldsymbol{\omega}_{rel} = \dot{q}_i \mathbf{e}_z \quad (\text{B.12})$$

$${}^i\dot{\boldsymbol{\omega}}_{rel} = \frac{d}{dt}({}^i\boldsymbol{\omega}_{rel}) = {}_i\tilde{\omega}_i \mathbf{e}_z \dot{q}_i + \ddot{q}_i \mathbf{e}_z \quad (\text{B.13})$$

$$({}^iJ_{R,rel})_{jk} = \begin{cases} 1 & \text{for } j = 3 \text{ and } k = i \\ 0 & \text{else} \end{cases} \quad (\text{B.14})$$

$${}^pJ_{T,rel} = {}_p\tilde{\mathbf{r}}_{rel}^T {}^pJ_{R,p} \quad (\text{B.15})$$

2. Prismatic Joints:

$${}^p\mathbf{r}_{rel} = \mathbf{r}_{const} + q_i \mathbf{A}_{ip}^T \mathbf{e}_z \quad (\text{B.16})$$

$${}^p\dot{\mathbf{r}}_{rel} = \mathbf{A}_{ip}^T \mathbf{e}_z \dot{q}_i + {}_p\tilde{\omega}_p {}^p\mathbf{r}_{rel} \quad (\text{B.17})$$

$${}^p\ddot{\mathbf{r}}_{rel} = \frac{d}{dt}({}^p\dot{\mathbf{r}}_{rel}) = \mathbf{A}_{ip}^T \mathbf{e}_z \ddot{q}_i + {}_p\dot{\tilde{\omega}}_p {}^p\mathbf{r}_{rel} + 2 {}_p\tilde{\omega}_p {}^p\dot{\mathbf{r}}_{rel} \quad (\text{B.18})$$

$${}^i\boldsymbol{\omega}_{rel} = {}^i\dot{\boldsymbol{\omega}}_{rel} = \mathbf{0} \quad (\text{B.19})$$

$${}^iJ_{R,rel} = \mathbf{0} \quad (\text{B.20})$$

$${}^iJ_{T,rel} = \mathbf{A}_{ip} ({}^pJ_{TO,p} + {}_p\tilde{\mathbf{r}}_{rel}^T {}^pJ_{R,p}) + J \quad (\text{B.21})$$

$$(J)_{jk} = \begin{cases} 1 & \text{for } j = 3 \text{ and } k = i \\ 0 & \text{else} \end{cases} \quad (\text{B.22})$$

¹ $(\cdot)_{jk}$ indicates the matrix element in the row j and column k .

Appendix C

Gear Friction of Harmonic Drives

Table C.1 summarizes the parameter of the friction law (3.30) after a curve fit to the catalog data. The efficiency according to catalog data (for a temperature of 20 °C), plotted with the curve fit results, indicates an accurate match.

Table C.1: Results of the parameter estimation based on the friction model (3.30) with a least-squares fit to catalog data.

Gear	$T_{f,0}$ [Nm]	μ [-]	b [Nm s/rad]	γ [s/rad]	RMS
HFUC-8-100	$3.90 \cdot 10^{-3}$	$5.38 \cdot 10^{-5}$	$3.71 \cdot 10^{-6}$	$9.49 \cdot 10^{-6}$	$8.60 \cdot 10^{-3}$
HFUC-25-100	$1.10 \cdot 10^{-1}$	$6.11 \cdot 10^{-5}$	$1.06 \cdot 10^{-4}$	$9.44 \cdot 10^{-6}$	$8.60 \cdot 10^{-3}$
CSD-14-100	$1.88 \cdot 10^{-2}$	$1.04 \cdot 10^{-4}$	$2.84 \cdot 10^{-5}$	$1.50 \cdot 10^{-5}$	$4.60 \cdot 10^{-3}$
CSD-17-100	$4.58 \cdot 10^{-2}$	0	$8.64 \cdot 10^{-5}$	$7.23 \cdot 10^{-6}$	$7.50 \cdot 10^{-3}$
CSD-20-100	$8.02 \cdot 10^{-2}$	0	$1.51 \cdot 10^{-4}$	$7.23 \cdot 10^{-6}$	$7.50 \cdot 10^{-3}$
CSD-20-160	$5.51 \cdot 10^{-2}$	0	$1.06 \cdot 10^{-4}$	$8.03 \cdot 10^{-6}$	$5.20 \cdot 10^{-3}$
CSG-25-160	$9.23 \cdot 10^{-2}$	$3.05 \cdot 10^{-4}$	$1.03 \cdot 10^{-4}$	$7.04 \cdot 10^{-6}$	$7.50 \cdot 10^{-3}$
CPL-32-50	$2.51 \cdot 10^{-1}$	$3.90 \cdot 10^{-4}$	$1.53 \cdot 10^{-4}$	$1.21 \cdot 10^{-5}$	$1.20 \cdot 10^{-2}$
CPL-32-100	$2.29 \cdot 10^{-1}$	$3.36 \cdot 10^{-4}$	$1.73 \cdot 10^{-4}$	$7.52 \cdot 10^{-6}$	$9.90 \cdot 10^{-3}$

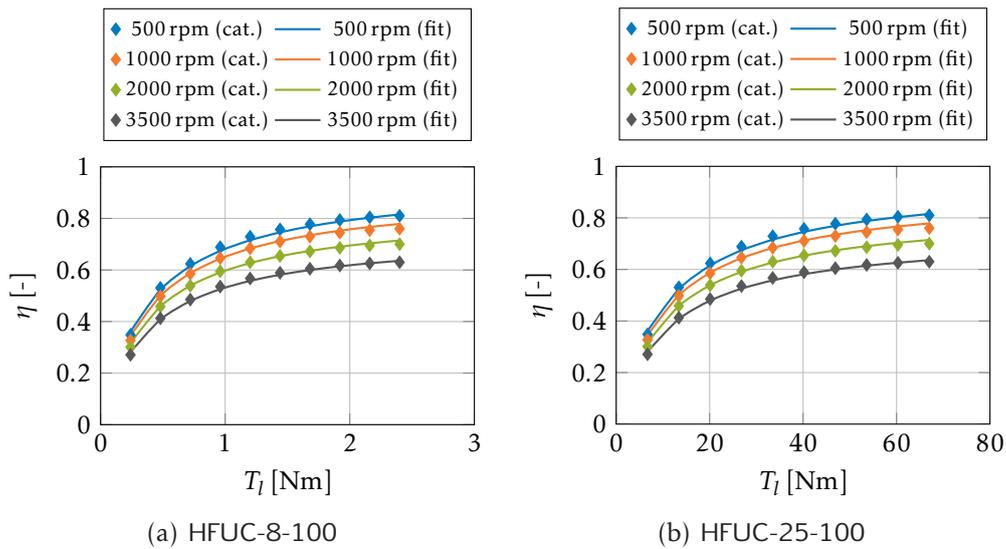


Figure C.1: Efficiency based on the catalog data provided by the Harmonic Drive gear manufacturer with the curve fit according to (3.30).

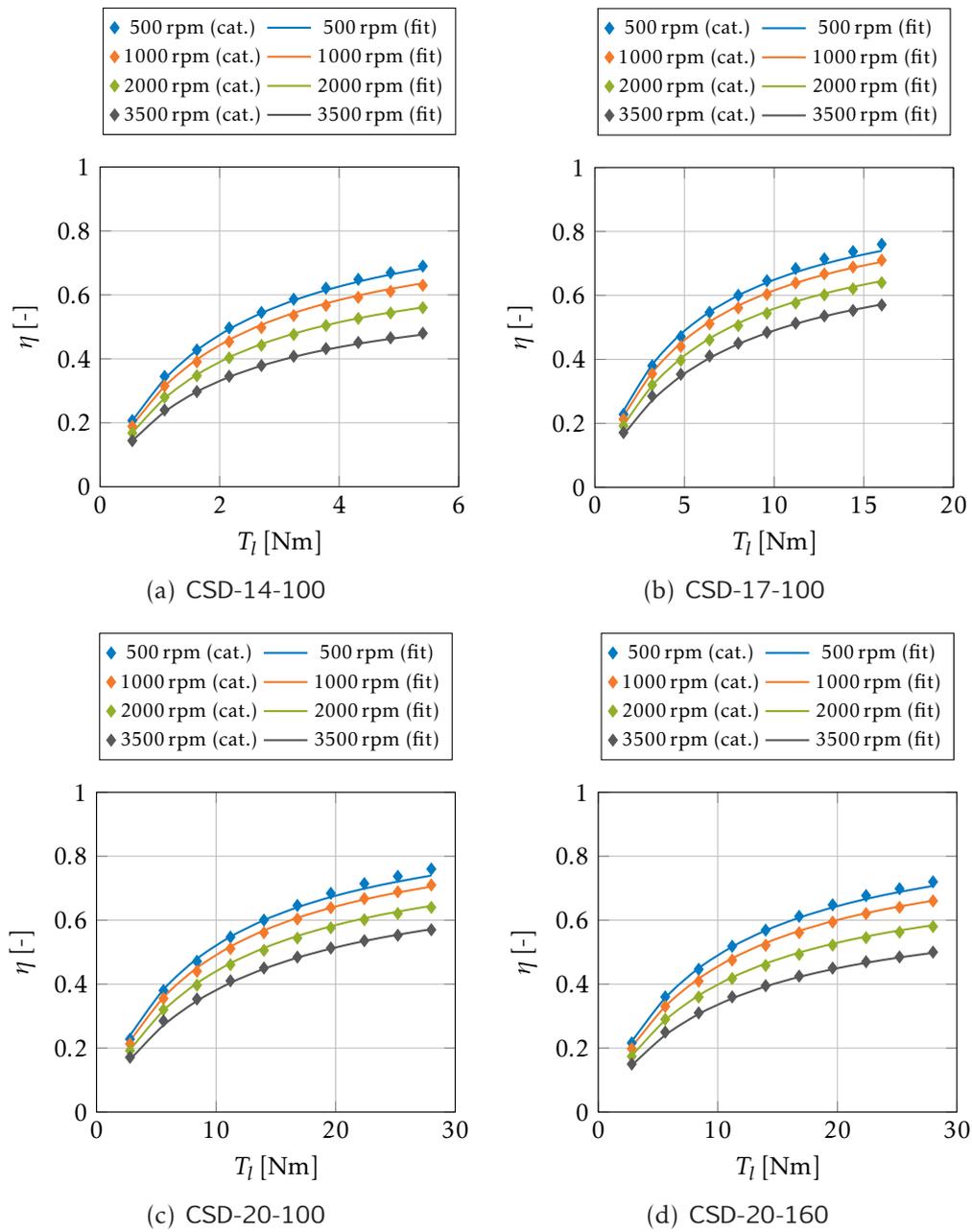


Figure C.2: Efficiency based on the catalog data provided by the Harmonic Drive gear manufacturer with the curve fit according to (3.30).

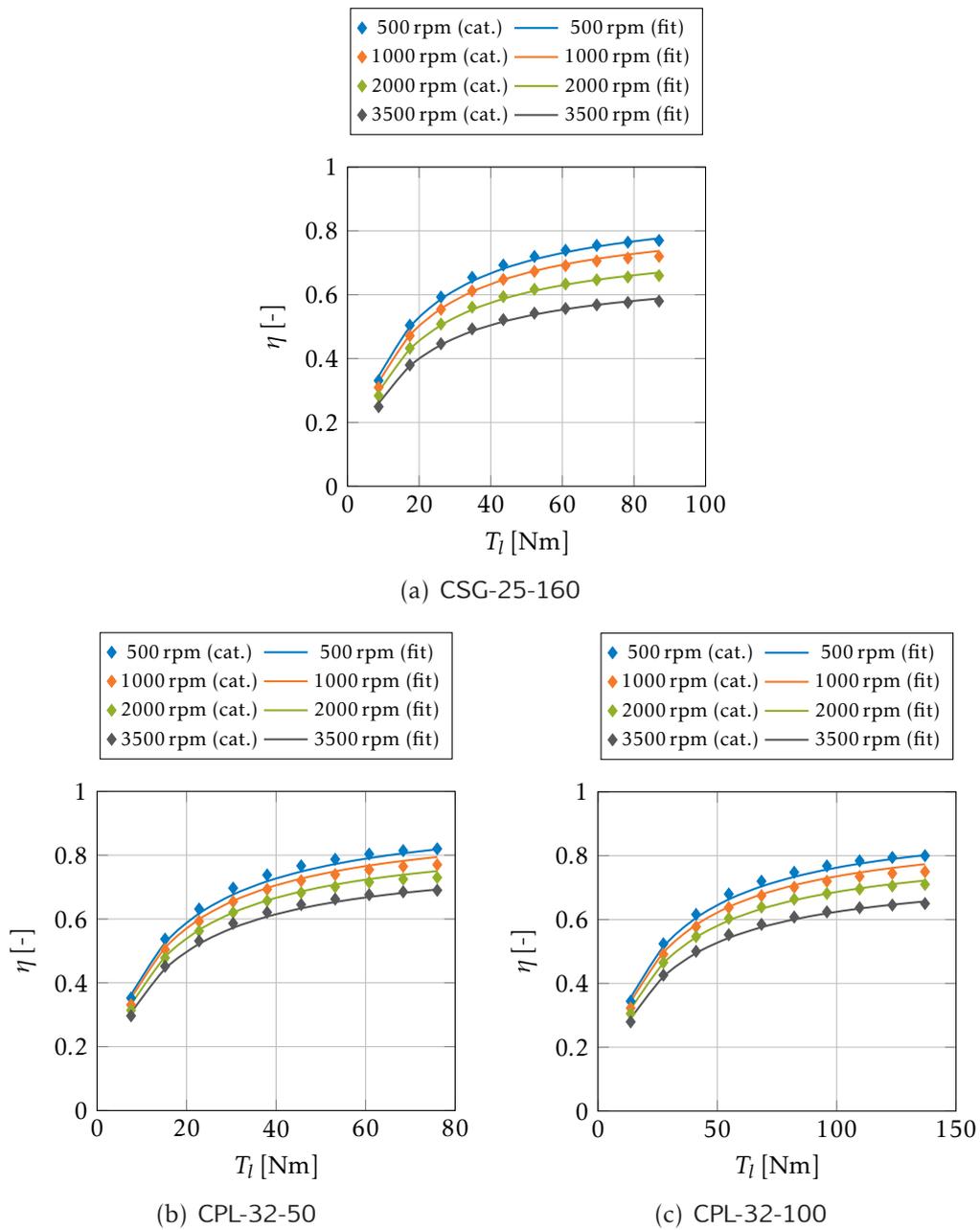


Figure C.3: Efficiency based on the catalog data provided by the Harmonic Drive gear manufacturer with the curve fit according to (3.30).

Appendix D

Experimental Friction Identification

Measurement from experiments on the robot modules B.0 (test module for the final manipulator), the module B.2 and the hip flexion and adduction of the humanoid robot LOLA are presented (test procedure according to Chapter 4).

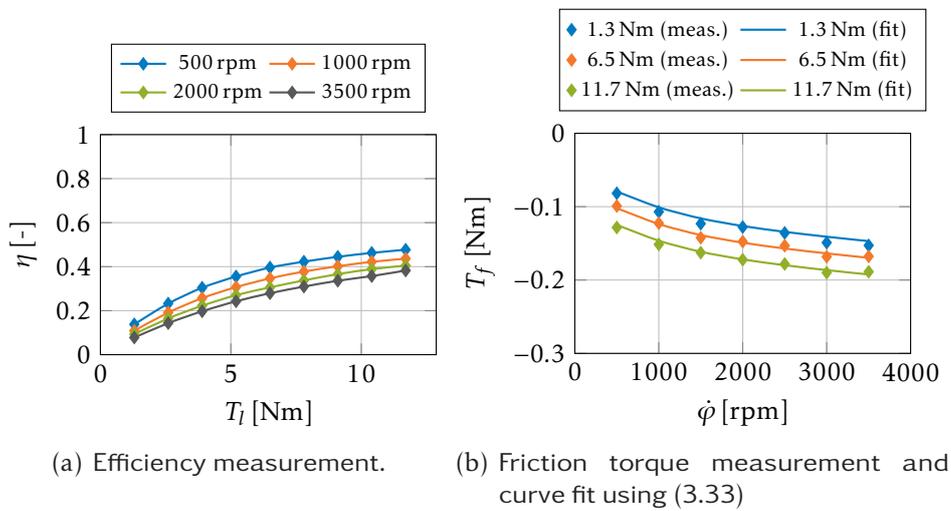


Figure D.1: Result of the experiments with the robot module B.0.

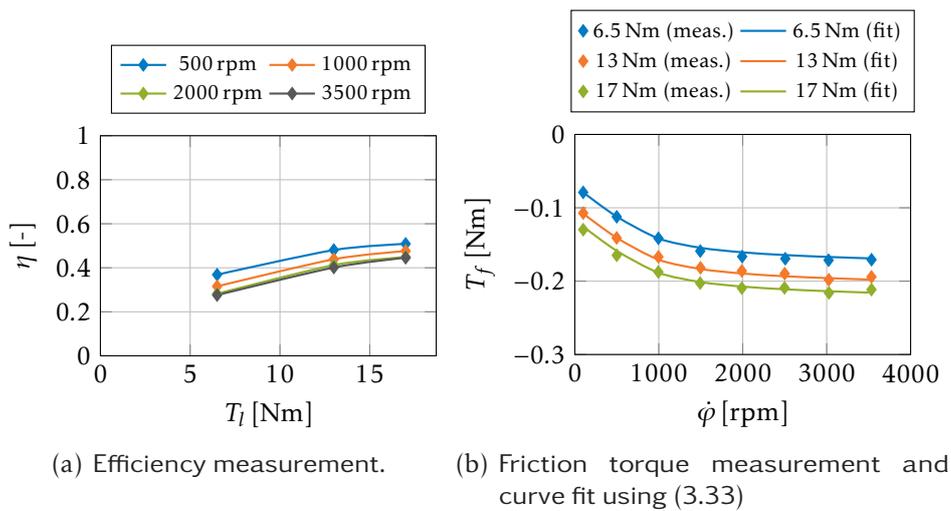


Figure D.2: Result of the experiments with the robot module B.2.

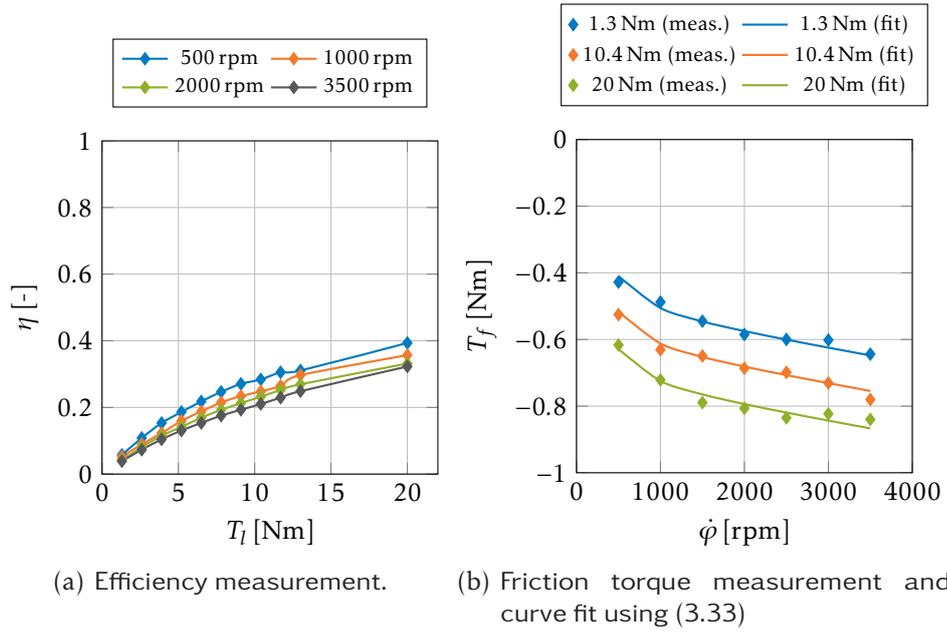


Figure D.3: Result of the experiments with the robot module of LOLA (flexion).

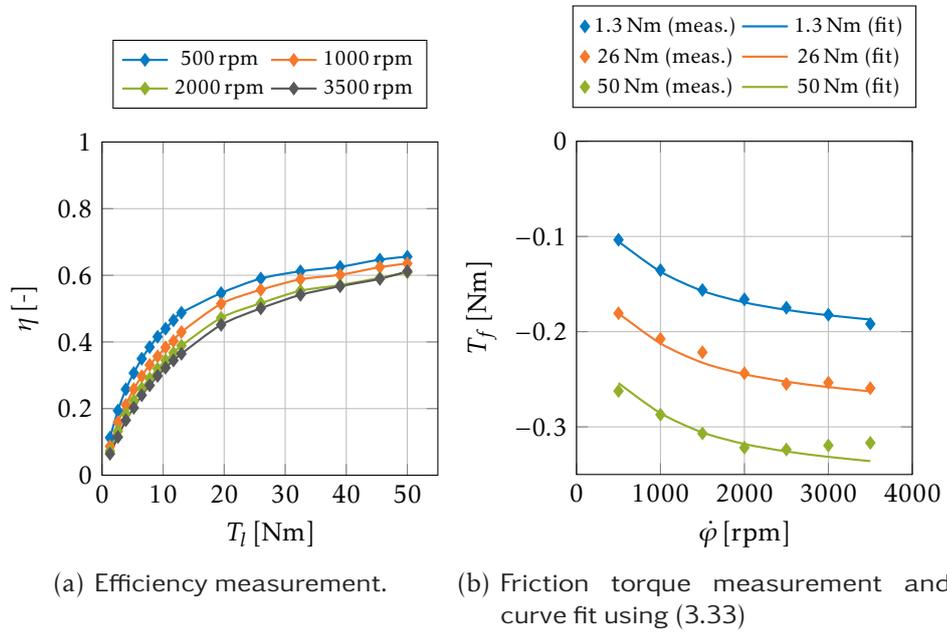


Figure D.4: Result of the experiments with the robot module of LOLA (adduction).

List of Abbreviations

AM Institute of Applied Mechanics.

BLDC Brushless Direct Current Electric Motor.

CAD Computer-Aided Design.

CAN Controller Area Network.

DLR German Aerospace Center.

DoF Degree of Freedom.

EU European Union.

GUI Graphical User Interface.

MRDS Microsoft Robotics Developer Studio.

PAM Pneumatic Artificial Muscle.

PCI Peripheral Component Interconnect.

PDO Process Data Object.

PMO Plant Maintenance Operation.

PRM Probabilistic Road-Map Planner.

PWM Pulse-Width Modulation.

RMSE Root Mean Square Error.

ROS Robot Operating System.

RRT Rapidly-Exploring Random Trees.

SCARA Selective Compliance Assembly Robot Arm.

SISO Single-Input and Single-Output.

SQP Sequential Quadratic Programming.

SSV Swept Sphere Volume.

SVD Singular Value Decomposition.

TCP Tool-Center-Point.

ToF Time-of-Flight.

UDP User Datagram Protocol.

UN United Nations.

URDF Unified Robot Description Format.

WP Work Package.

XML Extensible Markup Language.

Bibliography

- Aljanobi, A., Al-hamed, S., and Al-Suhaibani, S. A Setup of Mobile Robotic Unit for Fruit Harvesting. In *Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop on*, pages 105–108, June 2010. doi: 10.1109/RAAD.2010.5524602.
- Arima, S. and Kondo, N. Cucumber Harvesting Robot and Plant Training System. *Journal of Robotics and Mechatronics*, 11(3):208–212, 1999.
- Armstrong-Hélouvry, B., Dupont, P., and Canudas De Wit, C. A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction. *Automatica*, 30(7):1083–1138, 1994. ISSN 0005-1098. doi: [http://dx.doi.org/10.1016/0005-1098\(94\)90209-7](http://dx.doi.org/10.1016/0005-1098(94)90209-7).
- Bac, C. W., Hemming, J., and van Henten, E. J. Stem localization of sweet-pepper plants using the support wire as a visual cue. *Computers and Electronics in Agriculture*, 105(0):111–120, 2014a. ISSN 0168-1699. doi: <http://dx.doi.org/10.1016/j.compag.2014.04.011>.
- Bac, C. W., van Henten, E. J., Hemming, J., and Edan, Y. Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. *Journal of Field Robotics*, 31(6):888–911, 2014b. ISSN 1556-4967. doi: 10.1002/rob.21525.
- Bac, C. W., Hemming, J., van Tuijl, B., Barth, R., Wais, E., and van Henten, E. J. Performance evaluation of a harvesting robot for sweet-pepper. *Autonomous Robots*, 2015. (submitted).
- Bachche, S. G. *Automatic Harvesting for Sweet Peppers in Greenhouse Horticulture*. PhD thesis, Kochi University of Technology, Kochi, Japan, 2013. Available online at <http://hdl.handle.net/10173/979>, visted on September 9, 2014.
- Baeten, J., Donné, K., Boedrij, S., Beckers, W., and Claesen, E. Autonomous Fruit Picking Machine: A Robotic Apple Harvester. In *Field and Service Robotics*, volume 42 of *Springer Tracts in Advanced Robotics*, pages 531–539. Springer Berlin / Heidelberg, 2008. doi: 10.1007/978-3-540-75404-6_51.
- Barth, R., Baur, J., Buschmann, T., Edan, Y., Hellström, T., Nguyen, T., Ringdahl, O., Saeys, W., Salinas, C., and Vitzrabin, E. Using ROS for Agricultural Robotics - Design Considerations and Experiences. In *Proceedings of the Second RHEA International Conference on Robotics and associated High-technologies and Equipment for Agriculture*, pages 509 – 518, May 2014.
- Baur, J., Pfaff, J., Ulbrich, H., and Villgrattner, T. Design and development of a redundant modular multipurpose agricultural manipulator. In *Advanced*

- Intelligent Mechatronics (AIM)*, 2012 IEEE/ASME International Conference on, pages 823–830, July 2012. doi: 10.1109/AIM.2012.6265928.
- Baur, J., Pfaff, J., Schuetz, C., and Ulbrich, H. Dynamic modeling and realization of an agricultural manipulator. In *Proceedings of XV International Symposium on Dynamic Problems of Mechanics, DINAME*, February 2013.
- Baur, J., Dendorfer, S., Pfaff, J., Schuetz, C., Buschmann, T., and Ulbrich, H. Experimental Friction Identification in Robot Drives. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6006–6011, May 31–June 7 2014a. doi: 10.1109/ICRA.2014.6907744.
- Baur, J., Schuetz, C., Pfaff, J., Buschmann, T., and Ulbrich, H. Path Planning for a Fruit Picking Manipulator. In *International Conference of Agricultural Engineering - AgEng 2014 Zurich*. The European Society of Agricultural Engineers (EurAgEng), July 2014b. ISBN 978-0-9930236-0-6. ISBN 978-0-9930236-0-6.
- Baur, J., Schuetz, C., Pfaff, J., and Ulbrich, H. Modeling and Control of Pneumatic Artificial Muscles in an Antagonistic Set-up. In *Joint International Conference on Multibody System Dynamics*, June 30–July 3 2014c.
- Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999. ISBN 1-886529-00-0.
- Boblan, I. and Schulz, A. A Humanoid Muscle Robot Torso with Biologically Inspired Construction. In *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pages 1–6, June 2010.
- Buschmann, T., Lohmeier, S., Ulbrich, H., and Pfeiffer, F. Dynamics Simulation for a Biped Robot: Modeling and Experimental Verification. In *Robotics and Automation (ICRA), 2006 IEEE International Conference on*, pages 2673 –2678, May 2006. doi: 10.1109/ROBOT.2006.1642105.
- Buschmann, T. *Simulation and Control of Biped Walking Robots*. PhD thesis, Technische Universität München, 2010. ISBN 978-3-86853-804-5.
- Chiaverini, S., Siciliano, B., and Egeland, O. Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator. *Control Systems Technology, IEEE Transactions on*, 2(2):123 –134, June 1994. ISSN 1063-6536. doi: 10.1109/87.294335.
- Chiu, S. Control of Redundant Manipulators for Task Compatibility. In *Robotics and Automation (ICRA), 1987 IEEE International Conference on*, volume 4, pages 1718–1724, March 1987. doi: 10.1109/ROBOT.1987.1087795.
- Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Cambridge, MA, June 2005. ISBN 0-262-03327-5.

- Chou, C.-P. and Hannaford, B. Measurement and Modeling of McKibben Pneumatic Artificial Muscles. *Robotics and Automation, IEEE Transactions on*, 12(1): 90–102, February 1996. doi: 10.1109/70.481753.
- Cousins, S. Exponential Growth of ROS [ROS Topics]. *Robotics Automation Magazine, IEEE*, 18(1):19–20, March 2011. ISSN 1070-9932. doi: 10.1109/MRA.2010.940147.
- Coyette, C. and Schenk, H. *Agriculture, forestry and fishery statistics*. Number KS-FK-13-001-EN-C. European Commission, Eurostat, 2013. doi: 10.2785/45595. Available online at http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-FK-13-001/EN/KS-FK-13-001-EN.PDF, visited on July 10, 2014.
- Craig, J. J. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1986. ISBN 9780201103267.
- Creasy, G. L. and Creasy, L. L. *Grapes*. CABI Publishing, 2009. doi: 10.1079/9781845934019.0000.
- Daerden, F. *Conception and Realization of Pleated Pneumatic Artificial Muscles and their Use as Compliant Actuation Elements*. PhD thesis, Faculteit Toegepaste Wetenschappen Vakgroep Werktuigkunde, Vrije Universiteit Brussel, July 1999.
- Daerden, F. and Lefeber, D. Pneumatic Artificial Muscles: actuators for robotics and automation. *European Journal of Mechanical and Environmental Engineering*, 47(1):10–21, 2002.
- Davis, S., Tsagarakis, N., Canderle, J., and Caldwell, D. G. Enhanced Modelling and Performance in Braided Pneumatic Muscle Actuators. *The International Journal of Robotics Research*, 22(3-4):213–227, 2003. doi: 10.1177/0278364903022003006.
- De Haven, H. Tensioning Device for Producing a Linear Pull, September 27 1949. US Patent 2,483,088.
- De Hoog, D. Design evaluation of a sweet pepper harvesting manipulator using experiments and simulations. Master's thesis, Farm Technology Group, Wageningen UR, 2013. Available online at <http://edepot.wur.nl/257623>, visited on August 27, 2014.
- Denavit, J. and Hartenberg, R. S. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanics*, 22:215–221, 1955.
- Devol, J. G. C. Programmed Article Transfer, June 13 1961. US Patent 2,988,237.
- Edan, Y., Rogozin, D., Flash, T., and Miles, G. Robotic Melon Harvesting. *Robotics and Automation, IEEE Transactions on*, 16(6):831–835, 2000. ISSN 1042-296X. doi: 10.1109/70.897793.
- Ewald, A., Mayet, J., Buschmann, T., and Ulbrich, H. Generating Smooth Trajectories Free from Overshoot for Humanoid Robot Walking Pattern Replanning. In *Autonomous Mobile Systems (AMS)*, pages 89–97, 2012.

- Ferree, D. C. and Warrington, I. J. *Apples: Botany, Production, and Uses*. CABI Publishing, 2003. doi: 10.1079/9780851995922.0000.
- Foglia, M. M. and Reina, G. Agricultural Robot for Radicchio Harvesting. *Journal of Field Robotics*, 23(6-7):363–377, 2006. ISSN 1556-4967.
- Gauchel, W. and Saller, S. Adaptive gripper jaws for high-value crops harvesting. In *Fluid Power (8th IFK) in Dresden, 8th International Conference on*, 2012.
- Gaylord, R. H. Fluid Actuated Motor System and Stroking Device, July 22 1958. US Patent 2,844,126.
- Gienger, M., Löffler, K., and Pfeiffer, F. Towards the Design of a Biped Jogging Robot. In *Robotics and Automation (ICRA), 2001 IEEE International Conference on*, volume 4, pages 4140–4145, 2001. doi: 10.1109/ROBOT.2001.933265.
- Gillis, K. P., Giles, D. K., Slaughter, D. C., and Downey, D. Injection and fluid handling system for machine-vision controlled spraying. In *ASAE Annual Meeting*, 2001. Paper number: 011114.
- Gross, D., Hauger, W., Schrader, J., and Wall, W. *Technische Mechanik*. Number 3 in Springer-Lehrbuch. Springer, 2008. ISBN 9783540684244.
- Guo, J., Zhao, D.-A., Ji, W., and Xia, W. Design and Control of the Open Apple-Picking-Robot Manipulator. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 2, pages 5–8, July 2010. doi: 10.1109/ICCSIT.2010.5564770.
- Hamon, P., Gautier, M., and Garrec, P. Dynamic Identification of Robots with a Dry Friction Model Depending on Load and Velocity. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 6187–6193, 2010. doi: 10.1109/IROS.2010.5649189.
- Han, K.-S., Kim, S.-C., Lee, Y.-B., Kim, S.-C., Im, D.-H., Choi, H.-K., and Hwang, H. Strawberry Harvesting Robot for Bench-type Cultivation. *Journal of Biosystems Engineering*, 37(1):65–74, 2012. doi: 10.5307/JBE.2012.37.1.065.
- General Catalog*. Harmonic Drive AG, Limburg/Lahn, 2011.
- Harnoy, A., Friedland, B., and Cohn, S. Modeling and Measuring Friction Effects. *Control Systems Magazine, IEEE*, 28(6):82–91, 2008. ISSN 1066-033X. doi: 10.1109/MCS.2008.929546.
- Harnoy, A., Friedland, B., Semenock, R., Rachoor, H., and Aly, A. Apparatus for Empirical Determination of Dynamic Friction. In *Proceedings of the American Control Conference*, volume 1, pages 546–550, 1994.
- Hayashi, S., Ganno, K., Ishii, Y., and Tanaka, I. Robotic Harvesting System for Eggplants. *JARQ*, 36(3):163–168, 2002.

- Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., and Kurita, M. Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, 105(2):160 – 171, 2010. ISSN 1537-5110. doi: <http://dx.doi.org/10.1016/j.biosystemseng.2009.09.011>.
- Hemming, J., Bac, C. W., van Tuijl, B. A. J., Barth, R., Bontsema, J., and Pekkeriet, E. A robot for harvesting sweet-pepper in greenhouses. In *International Conference of Agricultural Engineering - AgEng 2014 Zurich*, Zurich, Switzerland, July 2014. The European Society of Agricultural Engineers (EurAgEng). ISBN 978-0-9930236-0-6.
- Hildebrandt, A. *Regelung und Auslegung servopneumatischer Aktorsysteme*. PhD thesis, Institut für Systemdynamik, Universität Stuttgart, July 2009. ISBN 978-3-8322-8660-6.
- Hildebrandt, A., Sawodny, O., Neumann, R., and Hartmann, A. A Flatness Based Design for Tracking Control of Pneumatic Muscle Actuators. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, volume 3, pages 1156–1161, December 2002. doi: 10.1109/ICARCV.2002.1234936.
- Hildebrandt, A., Sawodny, O., Neumann, R., and Hartmann, A. Cascaded control concept of a robot with two degrees of freedom driven by four artificial pneumatic muscle actuators. In *American Control Conference, 2005. Proceedings of the 2005*, pages 680 – 685 vol. 1, June 2005.
- Hirzinger, G., Sporer, N., Albu-Schäffer, A., Hähnle, M., Krenn, R., Pascucci, A., and Schedl, M. DLR's torque-controlled light weight robot III-are we reaching the technological limits now? In *Robotics and Automation (ICRA), 2002 IEEE International Conference on*, volume 2, pages 1710–1716 vol.2, 2002. doi: 10.1109/ROBOT.2002.1014788.
- Inoue, K. Rubbertuators and Applications for Robots. In *Proceedings of the 4th International Symposium on Robotics Research*, pages 57–63, Cambridge, MA, USA, 1988. MIT Press. ISBN 0-262-02272-9.
- Irie, N. and Taguchi, N. Asparagus Harvesting Robot. *Journal of Robotics and Mechatronics*, 26(2):267–268, 2014.
- Irie, N., Taguchi, N., Horie, T., and Ishimatsu, T. Asparagus Harvesting Robot Coordinated with 3-D Vision Sensor. In *Industrial Technology, 2009. ICIT 2009. IEEE International Conference on*, pages 1–6, February 2009. doi: 10.1109/ICIT.2009.4939556.
- Isermann, R. *Mechatronic Systems Fundamentals*. Springer London, 2005. ISBN 978-1-84628-259-1.
- Jackson, J. Microsoft Robotics Studio: A Technical Introduction. *Robotics Automation Magazine, IEEE*, 14(4):82–87, December 2007. ISSN 1070-9932. doi: 10.1109/M-RA.2007.905745.

- Kennedy, C. and Desai, J. Modeling and Control of the Mitsubishi PA-10 Robot Arm Harmonic Drive System. *Mechatronics, IEEE/ASME Transactions on*, 10(3): 263–274, 2005. ISSN 1083-4435. doi: 10.1109/TMECH.2005.848290.
- Kerscher, T., Albiez, J., Zollner, J., and Dillmann, R. Evaluation of the Dynamic Model of Fluidic Muscles using Quick-Release. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 637–642, February 2006.
- Khatib, O. Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1):90–98, April 1986. ISSN 0278-3649. doi: 10.1177/027836498600500106.
- Kitamura, S. and Oka, K. Recognition and Cutting System of Sweet Pepper for Picking Robot in Greenhouse Horticulture. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 4, pages 1807–1812, 2005. doi: 10.1109/ICMA.2005.1626834.
- Kitamura, S. and Oka, K. Improvement of the Ability to Recognize Sweet Peppers for Picking Robot in Greenhouse Horticulture. In *International Joint Conference SICE-ICASE*, pages 353–356, 2006. doi: 10.1109/SICE.2006.315789.
- Klein, C. A. and Blaho, B. E. Dexterity Measures for the Design and Control of Kinetically Redundant Manipulators. *International Journal of Robotics Research*, 6(2):72–83, July 1987. ISSN 0278-3649. doi: 10.1177/027836498700600206.
- Koditschek, D. E. and Rimon, E. Robot Navigation Functions on Manifolds with Boundary. *Advances in Applied Mathematics*, 11(4):412–442, 1990. ISSN 0196-8858. doi: 10.1016/0196-8858(90)90017-S.
- Kondo, N. and Ting, K. C. *Robotics for Bioproduction Systems*. American Society of Agricultural Engineers, 1998a. ISBN 0929355946.
- Kondo, N., Monta, M., and Fujiura, T. Fruit Harvesting Robots in Japan. *Advances in Space Research*, 18(1-2):181–184, 1996. ISSN 0273-1177. doi: DOI:10.1016/0273-1177(95)00806-P.
- Kondo, N. and Ting, K. Robotics for Plant Production. *Artificial Intelligence Review*, 12(1-3):227–243, 1998b. ISSN 0269-2821. doi: 10.1023/A:1006585732197.
- Kondo, N., Monta, M., and Fujiura, T. Basic constitution of a robot for agricultural use. *Advanced Robotics*, 10(4):339–353, 1995. doi: 10.1163/156855396X00020.
- Kuffner, J. J. and LaValle, S. M. RRT-connect: An Efficient Approach to Single-Query Path Planning. In *Robotics and Automation (ICRA), 2000 IEEE International Conference on*, volume 2, pages 995–1001, 2000. doi: 10.1109/ROBOT.2000.844730.
- Latombe, J.-C. *Robot Motion Planning*. Kluwer Academic Publishers, Norwell, MA, USA, 1991. ISBN 079239206X.

- Laumond, J.-P. *Robot Motion Planning and Control*. Springer-Verlag, Berlin, 1998. Available online at <http://homepages.laas.fr/jpl/book.html>, visited on May 15, 2014.
- LaValle, S. M. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available online at <http://planning.cs.uiuc.edu/>, visited on May 15, 2014.
- LaValle, S. M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*, 1998.
- Li, Y., Xia, C., and Lee, J. Vision-based Pest Detection and Automatic Spray of Greenhouse Plant. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 920–925, 2009. doi: 10.1109/ISIE.2009.5218251.
- Liégeois, A. Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms. *Systems, Man, and Cybernetics, IEEE Transactions on*, 7: 868–871, 1977. doi: 10.1109/TSMC.1977.4309644.
- Lohmeier, S., Buschmann, T., and Ulbrich, H. Humanoid robot LOLA. In *Robotics and Automation (ICRA), 2009 IEEE International Conference on*, pages 775–780, May 2009. doi: 10.1109/ROBOT.2009.5152578.
- Lohmeier, S. *Design and Realization of a Humanoid Robot for Fast and Autonomous Bipedal Locomotion*. PhD thesis, Technische Universität München, 2010. ISBN: 978-3-86853-734-5.
- Lozano-Perez, T. Spatial Planning: A Configuration Space Approach. *Computers, IEEE Transactions on*, C-32(2):108–120, February 1983. ISSN 0018-9340. doi: 10.1109/TC.1983.1676196.
- Maciejewski, A. A. and Klein, C. A. Numerical Filtering for the Operation of Robotic Manipulators through Kinematically Singular Configurations. *Journal of Robotic Systems*, 5(6):527–552, 1988. ISSN 1097-4563. doi: 10.1002/rob.4620050603.
- Maciejewski, A. A. and Klein, C. A. The Singular Value Decomposition: Computation and Applications to Robotics. *The International Journal of Robotics Research*, 8(6):63–79, 1989. doi: 10.1177/027836498900800605.
- Malnersic, A., Hočevár, M., Sirok, B., Marchi, M., Tirelli, P., and Oberti, R. Close range precision spraying airflow /plant interaction. In *Proceedings of the First International Conference on Robotics and associated High-technologies and Equipment for agriculture (RHEA)*, pages 107–112, 2012.
- Madow, A., Gomez-de Gabriel, J. M., Martinez, J. L., Munoz, V. F., Ollero, A., and Garcia-Cerezo, A. The Autonomous Mobile Robot AURORA for Greenhouse Operation. *Robotics Automation Magazine, IEEE*, 3(4):18–28, 1996. ISSN 1070-9932. doi: 10.1109/100.556479.

- Mizuuchi, I., Kawamura, M., Asaoka, T., and Kumakura, S. Design and Development of a Compressor-Embedded Pneumatic-Driven Musculoskeletal Humanoid. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, pages 811–816, November 2012. doi: 10.1109/HUMANOIDS.2012.6651613.
- Monta, M., Kondo, N., and Shibano, Y. Agricultural Robot in Grape Production System. In *Robotics and Automation (ICRA), 1995 IEEE International Conference on*, volume 3, pages 2504–2509, 1995. doi: 10.1109/ROBOT.1995.525635.
- Muscato, G., Prestifilippo, M., Abbate, N., and Rizzuto, I. A prototype of an orange picking robot: past history, the new robot and experimental results. *Industrial Robot: An International Journal*, 32(2):128–138, 2005. doi: 10.1108/01439910510582255.
- Nakamura, Y. and Hanafusa, H. Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control. *Journal of Dynamic Systems, Measurement and Control*, 108(3):163–171, 1986. doi: 10.1115/1.3143764.
- Nakamura, Y. *Advanced Robotics: Redundancy and Optimization*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1991. ISBN 0201151987.
- Nguyen, T. T., Kayacan, E., De Baerdemaeker, J., and Saeys, W. Task and Motion Planning for Apple-Harvesting Robot. In *Modelling and Control in Agriculture, Horticulture and Post Harvest Industry, 2013, 4th IFAC Conference on*, volume 4, pages 247–252, 2013.
- Nguyen, T. T., Vandevoorde, K., Kayacan, E., De Baerdemaeker, J., and Saeys, W. Apple detection algorithm for robotic harvesting using a RGB-D camera. In *International Conference of Agricultural Engineering - AgEng 2014 Zurich*, July 2014. ISBN 978-0-9930236-0-6.
- Niiyama, R., Nishikawa, S., and Kuniyoshi, Y. Athlete Robot with Applied Human Muscle Activation Patterns for Bipedal Running. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 498–503, December 2010. doi: 10.1109/ICHR.2010.5686316.
- Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Hočevár, M., Baur, J., Pfaff, J., Schuetz, C., and Ulbrich, H. Selective spraying of grapevine's diseases by a modular agricultural robot. In *Journal of Agricultural Engineering*, volume 44, pages 149–153, 2013.
- Oberti, R., Marchi, M., Tirelli, P., Calcante, A., Iriti, M., Hočevár, M., Baur, J., Pfaff, J., Schuetz, C., and Ulbrich, H. CROPS Agricultural Robot: Application to Selective Spraying of Grapevine's Diseases. In *Proceedings of the Second RHEA International Conference on Robotics and associated High-technologies and Equipment for Agriculture*, pages 49–58, 2014.

- Park, J., Chung, W., and Youm, Y. Computation of Gradient of Manipulability for Kinematically Redundant Manipulators Including Dual Manipulators Systems. *Transaction on Control Automation, and Systems Engineering*, 1:8–15, 1999.
- Park, R. Two-Reaction Theory of Synchronous Machines Generalized Method of Analysis-Part I. *American Institute of Electrical Engineers, Transactions of the*, 48 (3):716–727, July 1929. ISSN 0096-3860. doi: 10.1109/T-AIEE.1929.5055275.
- Peterson, D. L., Bennedsen, B. S., Anger, W. C., and Wolford, S. D. A Systems Approach to Robotic Bulk Harvesting of Apples. *Transactions of the ASAE*, 42(4): 871–876, 1999.
- Pfaff, J., Baur, J., Ulbrich, H., and Villgrattner, T. Development of a multipurpose agricultural manipulator. In *Proceedings of the First RHEA International Conference on Robotics and associated High-technologies and Equipment for Agriculture*, 2012.
- Pfaff, J., Baur, J., Schuetz, C., Buschmann, T., and Ulbrich, H. Design of Drive Units for Agricultural Robots. In *International Conference of Agricultural Engineering - AgEng 2014 Zurich*. The European Society of Agricultural Engineers (EurAgEng), July 2014. ISBN 978-0-9930236-0-6.
- Pfeiffer, F. and Schindler, T. *Einfuehrung in die Dynamik*. Springer, Berlin, 2014. ISBN 978-3-642-41046-8.
- Pfeiffer, F. The TUM walking machines. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1850):109–131, 2007. doi: 10.1098/rsta.2006.1922.
- Pieper, D. *The Kinematics of Manipulators Under Computer Control*. Memo (Stanford artificial intelligence Laboratory). Department of Mechanical Engineering, Stanford University, 1968.
- Pierce, R. C. Expansible Cover, May 26 1936. US Patent 2,041,950.
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- Rajendra, P., Mitsutaka, K., Kazunori, N., Gosei, O., and Shigehiko, H. Shading Compensation Methods for Robots to Harvest Strawberries in Tabletop Culture. In *System Integration (SII), 2011 IEEE/SICE International Symposium on*, pages 172–177, December 2011. doi: 10.1109/SII.2011.6147440.
- Reed, J., Miles, S., Butler, J., Baldwin, M., and Noble, R. Automatic Mushroom Harvester Development. *Journal of Agricultural Engineering Research*, 78(1): 15–23, January 2001. ISSN 0021-8634. doi: <http://dx.doi.org/10.1006/jaer.2000.0629>.

- Rimon, E. and Koditschek, D. E. The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds. In *Robotics and Automation (ICRA), 1989 IEEE International Conference on*, volume 1, pages 21–26, May 1989. doi: 10.1109/ROBOT.1989.99962.
- Sakai, S., Iida, M., and Umeda, M. Heavy Material Handling for Agricultural Robot. In *Robotics and Automation (ICRA), 2002 IEEE International Conference on*, volume 1, pages 1062–1068, 2002. doi: 10.1109/ROBOT.2002.1013496.
- Sakai, S., Iida, M., Osuka, K., and Umeda, M. Design and control of a heavy material handling manipulator for agricultural robots. *Autonomous Robots*, 25: 189–204, 2008. ISSN 0929-5593. doi: 10.1007/s10514-008-9090-y.
- Sammons, P., Furukawa, T., and Bulgin, A. Autonomous Pesticide Spraying Robot for Use in a Greenhouse. In *Australasian Conference on Robotics and Automation*, Sydney, Australia, December 2005. ISBN 0-9587583-7-9.
- Sarig, Y. Robotics of Fruit Harvesting: A State-of-the-art Review. *Journal of Agricultural Engineering Research*, 54(4):265–280, 1993. ISSN 0021-8634. doi: <http://dx.doi.org/10.1006/jaer.1993.1020>.
- Scarfe, A. J., Flemmer, R. C., Bakker, H. H., and Flemmer, C. L. Development of An Autonomous Kiwifruit Picking Robot. In *Autonomous Robots and Agents (ICARA), 2009 4th International Conference on*, pages 380–384, February 2009. doi: 10.1109/ICARA.2009.4804023.
- Schertz, C. E. and Brown, G. K. Basic Considerations in Mechanizing Citrus Harvest. *Transactions of the ASAE*, 11(3):343–346, 1968. doi: 10.13031/2013.39405.
- Schröder, J., Erol, D., Kawamura, K., and Dillman, R. Dynamic Pneumatic Actuator Model for a Model-Based Torque Controller. In *Computational Intelligence in Robotics and Automation. Proceedings. 2003 IEEE International Symposium on*, volume 1, pages 342–347, July 2003. doi: doi:10.1109/CIRA.2003.1222113.
- Schuetz, C., Buschmann, T., Baur, J., Pfaff, J., and Ulbrich, H. Predictive Online Inverse Kinematics for Redundant Manipulators. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5056–5061, May 2014. doi: 10.1109/ICRA.2014.6907600.
- Schwienbacher, M. *Efficient Algorithms for Biped Robots - Simulation, Collision Avoidance and Angular Momentum Tracking*. PhD thesis, Technische Universität München, 2013.
- Schwienbacher, M., Buschmann, T., Lohmeier, S., Favot, V., and Ulbrich, H. Self-Collision Avoidance and Angular Momentum Compensation for a Biped Humanoid Robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 581–586, May 2011. doi: 10.1109/ICRA.2011.5980350.

- Seyfferth, W., Maghzal, A. J., and Angeles, J. Nonlinear Modeling and Parameter Identification of Harmonic Drive Robotic Transmissions. In *Robotics and Automation (ICRA), 1995 IEEE International Conference on*, volume 3, pages 3027–3032, 1995. doi: 10.1109/ROBOT.1995.525714.
- Shapiro, A., Korkidi, E., Demri, A., Ben-Shahar, O., Riemer, R., and Edan, Y. Toward Elevated Agrobotics: Development of a Scaled-Down Prototype for Visually Guided Date Palm Tree Sprayer. *Journal of Field Robotics*, 26:572–590, June 2009. ISSN 1556-4959. doi: 10.1002/rob.v26:6/7.
- Siciliano, B., Sciavicco, L., and Villani, L. *Robotics: Modelling, Planning and Control*. Springer, 2009. doi: 10.1007/978-1-84628-642-1.
- Siciliano, B. and Khatib, O. *Springer Handbook of Robotics*. Springer, 2008. doi: 10.1007/978-3-540-30301-5.
- Sistler, F. Robotics and Intelligent Machines in Agriculture. *Robotics and Automation, IEEE Journal of*, 3(1):3–6, February 1987. ISSN 0882-4967. doi: 10.1109/JRA.1987.1087074.
- Sivaraman, B. *Design and Development of a Robot Manipulator for Citrus Harvesting*. PhD thesis, University of Florida, Department of Agricultural and Biological Engineering, Gainesville, FL, USA, 2006. Available online at http://etd.fcla.edu/UF/UFE0017536/sivaraman_b.pdf, visited on September 10, 2014.
- Sivaraman, B. and Burks, T. F. Robot Manipulator for Citrus Harvesting: Configuration Selection. In *ASABE Annual International Meeting*, Minneapolis, MN, USA, June 17-20 2007. Paper Number: 071144.
- Slotine, J.-J. and Li, W. *Applied Nonlinear Control*. Prentice Hall, October 1991. ISBN 978-0130408907.
- Sucan, I. A. and Chitta, S. „MoveIt!“. Available online at <http://moveit.ros.org>, visited on July 29, 2014.
- Taghirad, H. D. and Belanger, P. R. An Experimental Study on Modelling and Identification of Harmonic Drive Systems. In *Decision and Control, 1996. Proceedings of the 35th IEEE Conference on*, volume 4, pages 4725–4730, December 1996. doi: 10.1109/CDC.1996.577625.
- Taghirad, H. D. and Belanger, P. R. Modeling and Parameter Identification of Harmonic Drive Systems. *Journal of Dynamic Systems, Measurement, and Control*, 120(4):439–444, December 1998. ISSN 0022-0434.
- Tanigaki, K., Fujiura, T., Akase, A., and Imagawa, J. Cherry-harvesting robot. *Computers and Electronics in Agriculture*, 63(1):65–72, August 2008. ISSN 0168-1699. doi: 10.1016/j.compag.2008.01.018.
- Tillett, N. D. Robotic Manipulators in Horticulture: A Review. *Journal of Agricultural Engineering Research*, 55(2):89–105, 1993. ISSN 0021-8634. doi: <http://dx.doi.org/10.1006/jaer.1993.1035>.

- Tondu, B. and Lopez, P. Modeling and Control of McKibben Artificial Muscle Robot Actuators. *Control Systems, IEEE*, 20(2):15–38, April 2000. ISSN 1066-033X. doi: 10.1109/37.833638.
- Tondu, B., Ippolito, S., Guiochet, J., and Daidie, A. A Seven-degrees-of-freedom Robot-arm Driven by Pneumatic Artificial Muscles for Humanoid Robots. *International Journal of Robotics Research*, 24(4):257–274, April 2005. doi: 10.1177/0278364905052437.
- Toussaint, M., Gienger, M., and Goerick, C. Optimization of sequential attractor-based movement for compact behaviour generation. In *Humanoid Robots (Humanoids), 2007 11th IEEE-RAS International Conference on*, pages 122–129, November 29–December 1 2007. doi: 10.1109/ICHR.2007.4813858.
- Tuttle, T. D. Understanding and Modeling the Behavior of a Harmonic Drive Gear Transmission. Technical Report No. 1365, MIT, 1992.
- Tuttle, T. D. and Seering, W. P. A Nonlinear Model of a Harmonic Drive Gear Transmission. *Robotics and Automation, IEEE Transactions on*, 12(3):368–374, 1996. ISSN 1042-296X. doi: 10.1109/70.499819.
- Ulbrich, H. *Maschinendynamik*. Teubner Studienbücher. Teubner, 1996. ISBN 9783519032335.
- van Henten, E. J., Hemming, J., van Tuijl, B. A. J., Kornet, J. G., Meuleman, J., Bontsema, J., and van Os, E. A. An Autonomous Robot for Harvesting Cucumbers in Greenhouses. *Autonomous Robots*, 13:241–258, November 2002. ISSN 0929-5593. doi: 10.1023/A:1020568125418.
- Vanderborght, B., Verrelst, B., Van Ham, R., Vermeulen, J., and Lefeber, D. Dynamic Control of a Bipedal Walking Robot actuated with Pneumatic Artificial Muscles. In *Robotics and Automation (ICRA), 2005 IEEE International Conference on*, pages 1–6, April 2005. doi: 10.1109/ROBOT.2005.1570087.
- Wampler, C. W. Manipulator Inverse Kinematic Solutions Based on Vector Formulations and Damped Least-Squares Methods. *Systems, Man and Cybernetics, IEEE Transactions on*, 16(1):93–101, January 1986. ISSN 0018-9472. doi: 10.1109/TSMC.1986.289285.
- Whitney, D. E. Resolved Motion Rate Control of Manipulators and Human Prostheses. *Man-Machine Systems, IEEE Transactions on*, 10(2):47–53, June 1969. ISSN 0536-1540. doi: 10.1109/TMMS.1969.299896.
- Wittwer, S. H. and Robb, W. M. Carbon Dioxide Enrichment of Greenhouse Atmospheres for Food Crop Production. *Economic Botany*, 18(1):34–56, 1964. ISSN 0013-0001. doi: 10.1007/BF02904000.
- Yoshikawa, T. Manipulability of Robotic Mechanisms. *The International Journal of Robotics Research*, 4(2):3–9, June 1985. doi: 10.1177/027836498500400201.