



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR ROBOTIK
UND KOGNITIVE SYSTEME

Methods for Quasi-Static Tasks with Redundant Manipulators

Advances in Kinematics, Dexterity and Sensitivity

Ivo Kuhlemann

Dissertation

Universität zu Lübeck
Institut für Robotik und Kognitive Systeme



UNIVERSITÄT ZU LÜBECK

**From the Institute of Robotics and Cognitive Systems
of the University of Lübeck
Director: Prof. Dr.-Ing. Achim Schweikard**

**Methods for Quasi-Static Tasks
with Redundant Manipulators
Advances in Kinematics, Dexterity and Sensitivity**

Dissertation
for Fulfillment of
Requirements
for the Doctoral Degree
of the University of Lübeck

from the Department of Computer Sciences

Submitted by

Ivo Kuhlemann
from Lemgo (Germany)

Lübeck, 2019

First referee: Prof. Dr.-Ing. Achim Schweikard

Second referee: Prof. Dr.-Ing. Erik Maehle

Date of oral examination: 09.07.2019

Approved for printing. Lübeck, 11.07.2019

Zusammenfassung

Kooperative serielle Roboter haben in den letzten Jahren zunehmend an Bedeutung gewonnen. Anthropomorphe Robotermodelle wie der *KUKA LBR iiwa* bieten durch integrierte Kraft-Momenten-Sensorik die Möglichkeit zur Interaktion mit ihrer Umgebung. Mit Hilfe von Drehmomentsensoren in den Gelenken kann der Roboter seine Kontaktkräfte am Endeffektor präzise steuern und ermöglicht dadurch die Durchführung feinmechanischer Aufgaben. Darüber hinaus profitieren kinematisch redundante Roboter von einem zusätzlichen siebten Freiheitsgrad, mit dem eine Nullraumbewegung des Ellenbogens ermöglicht wird. Um das volle Potenzial dieser Roboter auszuschöpfen, müssen jedoch geeignete Steuerungsverfahren eingesetzt werden. Ziel dieser Arbeit ist die Entwicklung von Methoden zur Optimierung der Nutzbarkeit von kinematisch redundanten, kooperativen Manipulatoren für quasistatische Aufgaben, d.h. Roboterbewegungen mit vernachlässigbar geringen Dynamik-Effekten.

In dieser Arbeit wird eine optimierte, robuste inverse Kinematik für redundante Roboter in geschlossener Form entwickelt, mit dem Ziel mittels des neuen Algorithmus den siebten Freiheitsgrad über einen intuitiven Winkelparameter zu kontrollieren. Der entwickelte Algorithmus bietet einen vollständigen Lösungsraum und eine konsistente Steuerung sämtlicher Armkonfigurationen. Die inverse Kinematik wird anhand eines Modells des KUKA LBR iiwa evaluiert. Hierbei wird die Konfigurations-Zuverlässigkeit und die Kontinuität der Gelenk-Trajektorien für einen exemplarischen, anspruchsvollen Bewegungspfad untersucht. Die Ergebnisse zeigen, dass sich die vorgegebenen Armkonfigurationen entlang einer Trajektorie nicht spontan ändern. Selbst bei variierenden Ellenbogenwinkeln sind die berechneten Gelenktrajektorien kontinuierlich und der entwickelte Algorithmus liefert robuste Lösungen des inversen Problems.

Zusätzlich befasst sich die Arbeit mit einer quantitativen Analyse von Arbeitsräumen und der Beweglichkeit kinematisch redundanter Roboter wie dem LBR iiwa. Ziel ist es, zu un-

tersuchen, wie das zusätzliche siebte Gelenk die Beweglichkeit des Roboters beeinflusst. Von zentraler Bedeutung für diese Analyse ist die Entwicklung einer geeigneten Methode zur Isolierung und Quantisierung dieser Effekte. Hierfür wird der LBR iiwa als sechssachsige Kinematik simuliert, um einen direkten Vergleich von sieben- und sechssachsigen Robotern zu ermöglichen. Die Ergebnisse zeigen, dass das zusätzliche Gelenk die Beweglichkeit um etwa 17% erhöht.

Im weiteren Verlauf der Arbeit werden problematische Instabilitäten von gängigen Modellen zur Berechnung von Kontaktkräften am Endeffektor diskutiert. Die aufgezeigten Defizite motivieren die Entwicklung von optimierten Methoden zur genauen Abschätzung von quasistatischen Kontaktkräften am Endeffektor. Eine Methode des maschinellen Lernens wird verwendet, um die Abhängigkeiten zwischen Gelenkdrehmomenten und den daraus resultierenden Endeffektor-Kräften zu bestimmen. Für das Training wird ein umfangreicher Trainingsdatensatz mit 330.000 zufälligen Roboterpositionen und zugehörigen Kräften aufgenommen. Mit dem Datensatz wird ein tiefes Feed-Forward-Netz mit fünf verborgenen Schichten mit je 200 Neuronen trainiert. Die Ergebnisse zeigen, dass der neuentwickelte Lernansatz deutlich genauere Kraftmessungen ermöglicht als das proprietäre Kraftberechnungsmodell des LBR iiwa. Die neue Kalibrierungsmethode kann die Genauigkeit der Kontaktkräfte am Endeffektor um insgesamt mehr als 86% erhöhen. Darüber hinaus wird gezeigt, dass die Stabilität des Berechnungsmodells mit der entwickelten Methode gesteigert werden kann. Im Gegensatz zum proprietären Modell, welches hohe Fehler in der Nähe von Singularitäten aufweist, zeigt der neuronale Lernansatz robuste Ergebnisse.

Als Beispiel für eine quasistatische Anwendung wird die Integration des KUKA LBR iiwa in ein robotisiertes Ultraschallsystem beschrieben. Es wird gezeigt, wie die zuvor entwickelten Methoden zur Verbesserung des Systems eingesetzt werden können. Um die Handhabung und Effektivität des robotisierten Ultraschallsystems weiter zu verbessern, wird ein neuer Ansatz zur manuellen Handführung der Sonde vorgestellt. Ein großes Problem der Handführung ist die Schwierigkeit für den Anwender, die Gelenkgrenzen des Roboters einzuhalten. Um diesem Problem entgegenzuwirken, wird ein Sicherheitsmodul entwickelt, welches das Erreichen der Gelenkgrenzen verhindert. Das Modul erzeugt bei Annäherung an die Gelenkgrenzen entgegenwirkende Drehmomente.

Die Ergebnisse dieser Arbeit verbessern die Anwendung redundanter kooperativer Roboter durch gezielte Optimierungen der Robotersteuerung und Sensorik. Durch die entwickelten Methoden können die kinematischen Vorteile und die integrierte Kraftsensorik effektiv ausgenutzt werden.

Abstract

Compliant serial robots have increasingly gained relevance in recent years. Anthropomorphic robotic models like the *KUKA LBR iiwa* provide crucial advantages by an integrated force torque sensor technology to interact with their environment. Joint torque sensors enable the robot to precisely control its end-effector contact forces in order to perform fine-mechanical tasks. Furthermore, kinematically redundant manipulators profit from an additional seventh degree of freedom, creating the elbow self-motion manifold. To exploit the full potential of these robots, appropriate control methods have to be utilized. The purpose of this work is the development of methods to optimize the applicability of kinematically redundant compliant manipulators for quasi-static operations, i.e. robot motions with negligibly dynamics effects.

An optimized, robust inverse kinematics solution in closed form for kinematically redundant manipulators has been developed. The goal is to enable full control of the seventh degree of freedom by an intuitive angle parameter. The inverse kinematics algorithm provides a complete solution space and consistent control of arm configurations. The algorithm is evaluated with a model of the *KUKA LBR iiwa*, looking at configuration reliability and continuity of joint trajectories for an exemplary, challenging motion path. The results show that commanded arm configurations do not change spontaneously along a trajectory. Even for varying elbow angles, the joint trajectories remain consistent and the presented algorithm provides robust solutions.

Furthermore, the thesis addresses the quantitative analysis of workspaces and dexterities of kinematically redundant manipulators like the *LBR iiwa*. The goal is to analyze how dexterity is affected by an additional seventh joint. Central to this question is the development of an adequate method to isolate and investigate these effects. To isolate the influence of the additional joint, the *LBR iiwa* is simulated as a six-jointed robot to allow for direct comparison of seven- and six-jointed manipulators. The results show that the dexterity is enhanced by about 17%.

Also, problematic instabilities of common contact force estimation models are discussed. The deficits motivate for the development of methods to improve the accuracy of quasi-static end-effector contact force estimation. A learning approach is proposed to train the dependencies between joint torques and resulting end-effector forces. An extensive training data set is acquired with 330,000 randomized robot positions and corresponding forces and used for training a deep feed-forward net with five hidden layers, 200 neurons each. The results show advantages of the developed neural learning approach compared to the proprietary force estimation model of the LBR iiwa. The new calibration method can potentially increase the accuracy of end-effector forces by more than 86% overall. Furthermore, it is shown that the model stability can be increased with the proposed approach. In contrast to the proprietary model, which shows high outliers near singularities, the neural learning approach shows robust results.

As an example of a quasi-static application, the integration of the KUKA LBR iiwa in a robotized ultrasound system is presented. It is demonstrated how the developed methods can be used to improve the application performance. To further advance the usability and effectiveness of the robotized ultrasound system, a novel approach to hand guidance of the probe is proposed. Since hand guidance suffers greatly from the difficulty to adhere to the robot's joint limits, a safety module is developed which prevents the joint limits from being reached. The module generates counter-acting torques when the joint limits are being approached.

The findings of this work improve the applicability of redundant compliant manipulators through specific optimizations of robot control and sensory systems. The developed methods allow the kinematic advantages and the integrated force sensors to be effectively exploited.

Contents

Zusammenfassung	vii
Abstract	x
Mathematical Notations	xvii
1 Introduction	1
1.1 Purpose of this Work	2
1.2 Organization	3
2 Technical Background	7
2.1 Robot Kinematics	7
2.1.1 Forward Kinematics	8
2.1.2 Inverse Kinematics	10
2.1.3 Singular Configurations	16
2.2 Workspace and Dexterity	18
2.2.1 Configuration Space	21
2.3 Kinematically Redundant Manipulators	22
2.3.1 Collaborative Robots	26
2.3.2 The KUKA Light-Weight Robot	27
2.4 Contact Force Estimation	29
2.4.1 Mechatronic Joint Design	29
2.4.2 Dynamic Force Estimation	31
2.4.3 Quasi-Static Force Estimation	32
3 Robust Inverse Kinematics Solution	33
3.1 Introduction	33
3.2 State Of The Art	34

3.3	The Inverse Kinematics Algorithm	36
3.3.1	Algorithm Overview	37
3.3.2	Closed-Form Solution	38
3.4	Evaluation	44
3.4.1	Runtime	50
3.5	Conclusions	50
4	Workspace and Dexterity Analysis	53
4.1	Introduction	53
4.1.1	Related Work	54
4.2	Robot Kinematics and Approximation Methods	55
4.2.1	Workspace Approximation	57
4.2.2	End-Effector Orientations	58
4.2.3	Inverse Kinematics	59
4.2.4	Dexterity Calculation	59
4.3	Evaluation	60
4.4	Conclusions	68
5	Contact Force Optimization	69
5.1	Introduction	69
5.2	Hardware and Calibration Methods	72
5.2.1	Robotic Hardware	72
5.2.2	Training Data	73
5.2.3	Proprietary Force Estimation Model	76
5.2.4	Neural Network Model	77
5.3	Evaluation	80
5.3.1	Accuracy	81
5.3.2	Outlier Reduction	83
5.3.3	Generalization	83
5.3.4	Training Data Size	87
5.4	Conclusions	88
6	Robotized Ultrasound	91
6.1	Introduction	91

6.2 Hand Guidance	94
6.2.1 Active Joint Limit Avoidance	95
6.3 Conclusions	97
7 Conclusions	99
Bibliography	103
List of Figures	111
List of Tables	113
Abbreviations	114
Acknowledgment	117
Curriculum vitae	119

Mathematical Notations

Symbol	Description
\mathbf{M}	Homogeneous pose matrix
M	Rotational part of \mathbf{M}
\mathbf{m}	Translational part of \mathbf{M}
$R_{z,\alpha}$	Elemental rotation matrix about the z -axis by angle α
$\mathcal{R}(M)$	Homogeneous matrix rotation with matrix M as rotational part, i.e. $\begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix}$.
$\mathcal{T}(\mathbf{m})$	Homogeneous matrix translation with vector \mathbf{m} as translational part, i.e. $\begin{bmatrix} I_3 & \mathbf{m} \\ 0 & 1 \end{bmatrix}$.
I_3	3×3 identity matrix
$[\mathbf{m}]_x = [\mathbf{M}]_x$	x -coordinate of \mathbf{M} , i.e. $\mathbf{M}_{(1,4)}$ or \mathbf{m}_1
$[\mathbf{ab}]$	Axis from point a to point b
\bar{a}	Length of edge a
s_α, c_α	$\cos(\alpha)$ and $\sin(\alpha)$, respectively
s_1, c_1	$\cos(\theta_1)$ and $\sin(\theta_1)$, respectively
arm elbow wrist	Global configuration parameters
$\left. \begin{array}{l} \text{arm} \\ \text{elbow} \\ \text{wrist} \end{array} \right\} \in \{-1, +1\}$	
${}^{i-1}\mathbf{M}_i$	
${}^{i-1}M_i$	Rotational part of ${}^{i-1}\mathbf{M}_i$
${}^{i-1}\mathbf{m}_i$	Translational part of ${}^{i-1}\mathbf{M}_i$
\mathcal{T}_{z_i, d_i}	Homogeneous matrix translation along the z -axis of the

	coordinate frame i by scalar d_i
$\mathcal{R}_{z_i, \theta_i}$	Homogeneous matrix rotation about the z -axis of the coordinate frame i by the angle θ_i
q	Robot joint variables
Q	Joint space
C	Configuration space
W	Workspace of a manipulator
W_r	Reachable workspace
W_d	Dexterous workspace
$f(q)$	Forward kinematics function
\mathbf{J}	Jacobian Matrix
$SE(3)$	Three dimensional special Euclidean group

1 Introduction

Compliant serial robots have increasingly gained relevance in recent years. Especially anthropomorphic manipulators with seven degrees of freedom like the *LBR* series from *KUKA Roboter GmbH* are frequently used for applications in various challenging fields. New robotic models like the *LBR iiwa* provide crucial advantages by an integrated force torque sensor technology to interact with their environment. Joint torque sensors enable the robot to precisely control its end-effector contact forces in order to perform fine-mechanical tasks. In addition, these systems provide collision protection, which enables safe physical human-robot interactions. These interactions have gained importance over the last years and have advanced to be one of the biggest research fields in robotics nowadays [78]. A number of recent developments of new compliant robotic systems indicate a fundamental change in robotics and the way humans and machines work together. There is a trend to replacing the classical and potentially harmful position-controlled rigid robots for automation tasks with sensitive, compliant manipulators that can work safely in cooperation with humans [96].

Beside the ability to interact with their environment, anthropomorphic manipulators like the KUKA LBR iiwa profit from an additional degree of freedom, compared to common six-jointed industrial robots. In case a robot has more degrees of freedom than needed to fulfill a desired task, the system becomes kinematically redundant. Challenging applications occasionally require the use of kinematically redundant manipulators with more than six degrees of freedom, for example when high dexterity or obstacle avoidance are necessary. Typically, the introduction of redundancy to systems is driven by the intention of increasing reliability and robustness by duplicating critical components or functions, e.g. sensors. When integrating redundancy in robots, however, the motivation is different. It is the purpose to increase the dexterity, in order to approach the one of a human arm.

In fact, the additional degree of freedom allows for motions of the manipulator without changing the pose of the end-effector. These motions are also known as self-motion and are typical for kinematically redundant robots [14]. In other words, the self-motion can be used to accomplish the same tasks in different joint configurations. In case of the KUKA LBR iiwa, the additional degree of freedom creates a self-motion manifold of the elbow on a circular path. Increased dexterity can reduce the amount of configuration changes within single trajectories and thus allows following optimized paths which would be impossible without the additional joint. This is particularly important for applications in dynamic scenarios where spontaneous configuration changes, for example due to joint limitations, are strictly forbidden within a single motion trajectory. The additional dexterity is hence of great advantage [71]. A further benefit of the elbow's self-motion manifold is the specific positioning of the articulated arm to adapt its occupied space to other parallel applications in a shared domain.

However, all described advantages of these manipulators are only beneficial if appropriate methods are utilized to exploit their full potential. For example, the elbow's self-motion can only be efficiently used if specialized inverse kinematics algorithms control the robot. In other words, such algorithms must be able to fully control the additional seventh degree of freedom. Furthermore, the increased dexterity of seven-jointed manipulators can potentially help to use robotic systems more efficiently, by defining tasks in workspace areas with high dexterity. This would only be possible if the reachable workspace and the inherent dexterity were precisely known. Finally, reliable force-driven robot-interactions require precise contact force measurements. In turn, errors in the computational model can lead to dangerous and potentially harmful situations during force driven operations. Thus, higher accuracy of the determined forces would lead to an enhanced overall sensitivity of the robot and therefore allow more complex applications. All these requirements are leading to different research questions which will be discussed in the following section.

1.1 Purpose of this Work

The purpose of this work is the development of methods to optimize the applicability of kinematically redundant compliant manipulators for quasi-static operations, i.e. robot mo-

tions with negligible dynamics effects. Three research questions are defined to properly approach the objective as follows.

RQ1: How to exploit all degrees of freedom of kinematically redundant serial manipulators by an inverse kinematics algorithm under safety and algorithmic restrictions?

The first issue will deal with the development of an optimized, robust inverse kinematics solution in a closed form for kinematically redundant manipulators. The goal is to enable full control of the seventh degree of freedom, the elbow self-motion manifold, by an intuitive parameter. The inverse kinematics algorithm should additionally provide a complete solution space and consistent control of all possible configurations for one single target end-effector pose.

RQ2: What effect does the additional seventh degree of freedom have on the dexterous workspace of kinematically redundant manipulators?

The second issue addresses the quantitative analysis of workspaces and dexterities of kinematically redundant manipulators like the KUKA LBR iiwa. The goal is to analyze how a robot's dexterity is affected by an additional seventh joint. Central to this question is the development of an adequate method to isolate and investigate these effects.

RQ3: How can the accuracy and stability of contact force estimation models of compliant manipulators be increased?

The third research question addresses problematic instabilities of common force estimation models and motivates for the development of methods to improve the accuracy of quasi-static end-effector contact force estimation. A learning approach is proposed to train the dependencies between joint torques and resulting end-effector forces.

1.2 Organization

The work is structured as follows. This introductory chapter will be followed by six further chapters. At the beginning of each chapter the content will be summarized and the

structure will be outlined. In turn, the end of each chapter will conclude the main ideas discussed in more detail within the actual chapter. Conclusions will be given as answers to the aforementioned research questions.

Chapter 2: Technical Background

The subsequent **Chapter 2** covers theoretical and technical basics that are paramount for this thesis. At the beginning, the kinematic structure of redundant serial kinematics is explained, followed by an introduction of elementary tools for motion control of robots. Next, the definition and determination of workspaces and dexterities, which are crucial tools for manipulator-performance analysis, are described. In the following section kinematically redundant robot arms are described and analyzed in more detail, which play a central role in this thesis. The last section deals with the collaboration between humans and robots. Sensory technologies are discussed that allow manipulators to interact with their environment and, beyond that, fundamental strategies for force control of robotic arms are presented.

Chapter 3: Robust Inverse Kinematics Solution

Chapter 3 deals with the development of an inverse kinematics algorithm for optimized and robust control of kinematically redundant manipulators. The goal of this chapter is to answer **RQ1** by developing a new inverse kinematics algorithm to optimally exploit the additional seventh degree of freedom. To achieve this, the robot's elbow motion will be controlled to utilize the full potential of the kinematic. The chapter starts with our motivation for the development of a new algorithm and addresses the analysis of already existing solutions in the literature. In the next section, the developed algorithm is described in detail and subsequently evaluated.

Chapter 4: Workspace and Dexterity Analysis

In order to answer **RQ2**, **Chapter 4** focuses on the workspace and dexterity of kinematically redundant robots. The aim of this chapter is to analyze their reachability performance and to discover areas within their workspace where they provide high dexterity. Likewise, the effects of the additional seventh degree of freedom are studied and quantified. The chapter starts with an introduction and discusses related works. The following section presents a method to calculate highly detailed workspaces and inherent dexter-

ities of different robot kinematics. A method is presented to isolate the effects of the seventh degree of freedom and metrics are introduced to quantify corresponding impacts on dexterity.

Chapter 5: Contact Force Optimization

The improvement of the collaborative abilities of the LBR iiwa robot is the focus of [Chapter 5](#) in order to answer [RQ3](#). Therefore, a method is developed to increase the accuracy of the contact forces estimation at the end-effector. A learning approach is proposed to train the dependencies between joint torques and end-effector contact forces. After pointing out arising problems of inaccurate force estimation models, the chapter discusses related methods and approaches in the literature. The next section explains how the training data set was recorded using the robot. Applied learning methods and the network architecture are described. In a detailed evaluation, the performance of the developed calibration method is compared to KUKA's proprietary force estimation model.

Chapter 6: Robotized Ultrasound

As an example of a quasi-static application, [Chapter 6](#) addresses the application of robotized ultrasound with the KUKA LBR iiwa. While focusing on radiation therapy, the chapter shows how the manipulator can be used as an ultrasound robot and, furthermore, explains how the methods and findings from the previous chapters can be integrated to improve the application performance. After an introduction to ultrasound-based real-time tumor tracking in radiotherapy, the chapter proposes a novel approach to hand guidance of the ultrasound robot. The developed method is qualitatively evaluated by volunteers.

Chapter 7: Conclusions

The final chapter summarizes the main results of this thesis and concludes by linking back to the motivation of this first introductory chapter. [Chapter 7](#) concludes the results by reviewing risen research questions and evaluates the outcome of this work. Remaining directions for future work are discussed and examples for alternative applications are shown.

2 Technical Background

The purpose of this background chapter is to clarify relevant robotic terms and definitions which are used throughout the work. Furthermore, essential mathematical definitions are given to ensure consistent notations. Since this thesis focuses on serial manipulators, [Section 2.1](#) starts with describing their kinematic structure and explains important terms. For the inverse kinematics addressed in [Chapter 3](#), the most fundamental mathematical tools for motion control of robots are explained. Different approaches for inverse kinematics are discussed and differences are pointed out. [Chapter 4](#) focuses on the analysis of workspaces and dexterity of redundant manipulators. Therefore, [Section 2.2](#) gives a general mathematical definition of workspaces and discusses the difference between reachability and dexterity. Methods for calculating workspaces are described and metrics are introduced to derive the performance of robots based on their workspaces. Finally, the section shows how to use these metrics for application-driven construction of manipulators. [Section 2.3](#) explains the unique characteristics of kinematically redundant robots. Differences to conventional six-jointed manipulators are discussed. A special focus of this section is on the collaborative lightweight robot *KUKA LBR iiwa*. [Section 2.4](#) addresses the sensory technology of the *KUKA LBR iiwa* robot and shows how robots can interact with their environment via sensors. The robot's integrated proprietary controllers are explained with a focus on end-effector contact force estimation. Furthermore, differences between dynamic and quasi-static contact force estimations are described.

2.1 Robot Kinematics

A serial robotic manipulator consists of $n > 0$ rigid links, connected by $n - 1$ rotational or prismatic joints. The link attached to the ground is called base and the subsequent links are moving relative to it. The last link in the serial chain is called end-effector and normally provides a flange for mounting manipulation tools. In addition, commonly used robots with six or seven joints feature characteristic kinematic structures called shoulder,

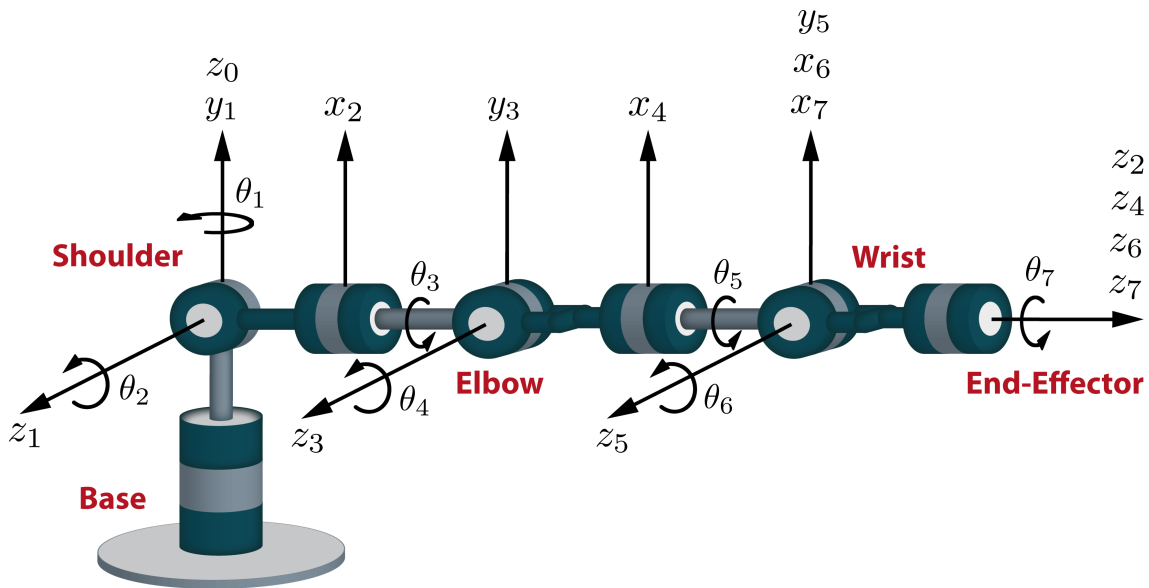


Figure 2.1: A common seven degree of freedom kinematic with relevant labels and joint axes.

elbow and wrist. The shoulder is normally located in the second joint, whereas joints three or four are called elbow. The last three joints in the kinematic chain are called wrist. Figure 2.1 shows a classic seven degree of freedom serial manipulator with relevant labels.

In order to control the spatial position of a robot end-effector, two paramount questions have to be answered: What is the current Cartesian pose of the end-effector and how does the robot get to the desired destination? The question of the current pose can be answered by the forward kinematics problem. In forward kinematics, the position and rotation of the coordinate frame attached to the last link is calculated based on given joint variables, relative to the base coordinate frame. The second question can be answered by the inverse kinematics problem by finding a set of joint variables to reach the desired target position with the end-effector.

2.1.1 Forward Kinematics

The forward kinematics problem treats the relation between given joint values and the resulting Cartesian position and rotation of the end-effector. In other words, it is a transformation from the joint variable space into the Cartesian space. Let our serial kinematic

chain consist of n joints and $n + 1$ links, then we index our joints with $i = 1, \dots, n$ and our links with $i = 0, \dots, n$. The motion of every joint is now expressed by a joint variable q_i . In case of rotational joints, q_i is an angle of rotation θ_i , and in case of a prismatic joint, q_i is a translational distance d_i . In a next step, we rigidly attach coordinate frames to each link. Each coordinate frame can be described by a 4×4 homogeneous transformation matrix \mathbf{M}_i or *pose* as

$$\mathbf{M}_i = \begin{bmatrix} M_i & \mathbf{m}_i \\ 0 & 1 \end{bmatrix}, \quad i \in \{0, \dots, n\} \quad (2.1)$$

where M_i is a 3×3 rotation matrix and \mathbf{m}_i denotes a 1×3 translational vector. Now let ${}^i\mathbf{M}_{i+1}$ be the homogeneous transformation matrix which transforms coordinate system i into $i + 1$. In other words, ${}^i\mathbf{M}_{i+1}$ expresses the position and orientation of the coordinate frame of joint $i + 1$ with respect to joint i . Since each joint is moving relative to a serial kinematic chain, we can now determine the position and rotation of the end-effector by a multiplicative combination, starting at the base coordinate frame to the last n -th link as

$${}^0\mathbf{M}_n = {}^0\mathbf{M}_1 {}^1\mathbf{M}_2 {}^2\mathbf{M}_3 \dots {}^{n-1}\mathbf{M}_n. \quad (2.2)$$

In a last step, we have to determine the joint-wise transformation matrices. In case of prismatic and rotational joints, these transformations are functions of single joint variables q_i , e.g.

$${}^{i-1}\mathbf{M}_i = {}^{i-1}\mathbf{M}_i(q_i). \quad (2.3)$$

Obviously, general kinematic information like link lengths must also be considered in the joint transformations. A commonly used method in robotics to include structural parameters in the forward kinematics is the Denavit-Hartenberg (DH) convention [17].

2.1.1.1 Denavit Hartenberg Representation

The DH-convention is a standard method to find suitable coordinate frames \mathbf{M}_i attached to each joint by relating kinematic information. The method uses four parameters d_i , θ_i , a_i , and α_i to determine each homogeneous transformation matrix by a combination of

elemental operations as

$$\begin{aligned}
{}^{i-1}\mathbf{M}_i &= \mathcal{T}_{z_{i-1},d_i} \mathcal{R}_{z_{i-1},\theta_i} \mathcal{T}_{x_i,a_i} \mathcal{R}_{x_i,\alpha_i} \\
&= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_ic_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_iss_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}.
\end{aligned} \tag{2.4}$$

Here, $\mathcal{T}_{z_{i-1},d_i}$ and \mathcal{T}_{x_i,a_i} denote homogeneous translations along the z -axis of the coordinate frame $i - 1$ by d_i and along the x -axis of coordinate frame i by a_i , respectively. $\mathcal{R}_{z_{i-1},\theta_i}$ and $\mathcal{R}_{x_i,\alpha_i}$ are homogeneous rotations around the z -axis of the coordinate frame $i - 1$ by the angle θ_i and around the x -axis of coordinate frame i by the angle α_i , respectively. The four DH-parameter d_i , θ_i , a_i , and α_i are commonly named *link offset*, *joint angle*, *link length*, and *link twist*, respectively. For any given link, three of these parameters are static, while only one parameter – the joint variable – is not. In case of a prismatic joint, d_i is variable whereas θ_i is static and vice versa in the case of a rotational joint. Figure 2.2 illustrates the concept of DH parameters.

2.1.2 Inverse Kinematics

In contrast to the forward kinematics problem, the goal of the inverse kinematics is to find a set of joint variables to reach a desired end-effector position and orientation. It is the reverse method to forward kinematics, or in other words, the inverse kinematics transforms from the Cartesian space into the joint variable space.

There are two major types of approaches to solve the inverse kinematics of a serial manipulator: analytical solutions and iterative solutions. Analytical or *closed form* approaches use geometric relations of kinematic structures and explicitly solve trigonometric functions. In contrast, iterative approaches typically rely on optimization methods and solve

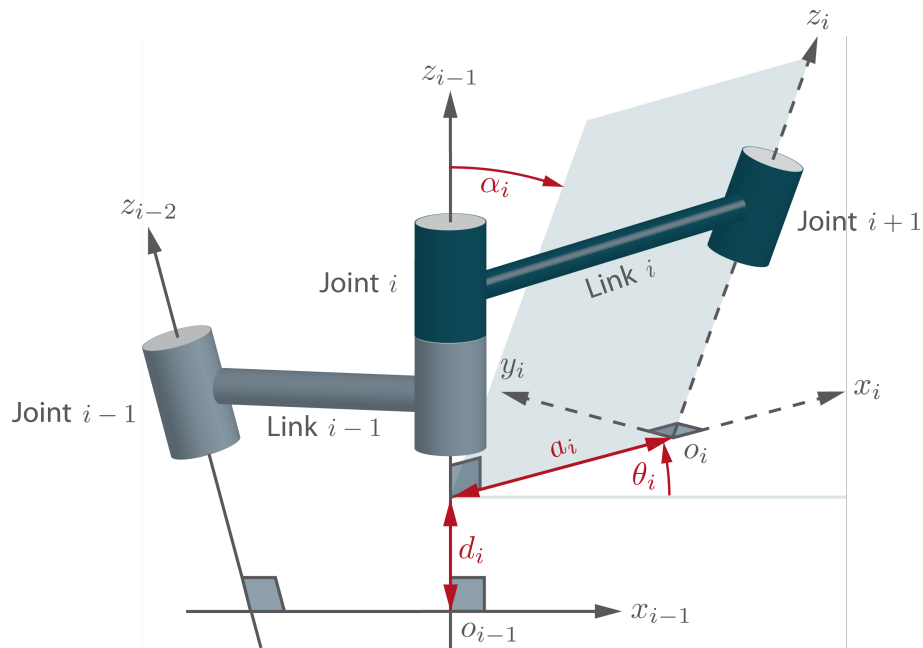


Figure 2.2: DH parameters a_i , α_i , d_i , θ_i defined for joint i and link i .

the inverse kinematics problem with an approximate solution. Looking at the computational cost, analytical inverse kinematics can be significantly faster than iterative solutions and provide more than one solution for a given end-effector pose.

By formulating the general inverse kinematics problem, we can mathematically describe the goal of inverse kinematics. Let \mathbf{A} be a desired end-effector pose given as a 4×4 homogeneous matrix and ${}^0\mathbf{M}_n$ be the resulting end-effector pose given by the forward kinematics as a function of all joints variables, then

$$\mathbf{A} = {}^0\mathbf{M}_n(q_1, q_2, \dots, q_n) = {}^0\mathbf{M}_1(q_1) {}^1\mathbf{M}_2(q_2) \dots {}^{n-1}\mathbf{M}_n(q_n). \quad (2.5)$$

Now, the goal of the inverse kinematics problem is finding suitable combinations of values for the joint variables q_1, q_2, \dots, q_n to solve the forward kinematics such that $\mathbf{A} = {}^0\mathbf{M}_n(q_1, q_2, \dots, q_n)$.

2.1.2.1 Analytical Inverse Kinematics

Analytical closed form inverse kinematics solutions are mostly preferable because they provide crucial advantages over iterative approaches. Since these algorithms solve a fixed number of trigonometric functions rather than using an iterative optimization, the computational cost is much lower and runtimes are deterministic. In real-time applications, fast, deterministic computation is a feature of paramount importance to realize high control loop rates. Due to the symmetry of trigonometric functions, multiple solutions are inherently provided. In other words, we can obtain multiple different arm configurations to reach the same desired end-effector pose. Solving the inverse kinematics problem in closed form allows for identifying all possible joint value combinations, resulting in a complete solution space. This allows for choosing a particular, best solution among several. Although defining the best solution can be very difficult as it depends on multiple factors.

Using (2.5) we define

$$\mathbf{A} = {}^0\mathbf{M}_n(q_1, q_2, \dots, q_n) = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

and the matrix elements $r_{11}, r_{12}, \dots, r_{33}$ and t_x, t_y, t_z represent non-trivial trigonometric terms, given by the forward kinematics formulation. Now, the general goal of a closed form approach is to find explicit solutions for

$$q_i = f_i(\mathbf{A}_{11}, \dots, \mathbf{A}_{34}), \quad i = 1, \dots, n, \quad (2.7)$$

as formulated in [102]. As mentioned before, multiple solutions are expected when the n equations are solved for the unknown joint variables. In practice, the range of joint motions is often mechanically restricted. For example, linear joints can only extend to a limited length or rotational joints cannot make a full 360 degrees turn. To this end, not all mathematical solutions will be physically realizable configurations of the manipulator and must be identified by adequate methods.

2.1.2.2 Decoupling Principle

Serial manipulators usually have six joints, because at least six mechanical degrees of freedom are needed to place the end-effector in an arbitrary position and orientation in the Cartesian workspace of the robot. Most of these kinematics are built with a *spherical wrist*. A spherical wrist means that the last three joints in the kinematic chain are revolute joints with orthogonal axes and an intersection point at the *wrist center*. Since inverse kinematics solutions are often difficult to derive, the complexity of the problem can be highly reduced by using the *decoupling principle*. For serial manipulators with a spherical wrist, it is possible to decompose the general inverse kinematics problem into two simpler sub-problems, called *inverse position* and *inverse orientation* kinematics [41]. As a result, we have to solve two independent problems, each with only three unknown parameters.

This is done by decomposing the given target transformation matrix into a translational part and a rotational part by using (2.1) as

$$\begin{aligned}
 {}^0\mathbf{M}_n(q_1, q_2, \dots, q_n) &= \begin{bmatrix} {}^0M_n & {}^0\mathbf{m}_n \\ 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} I_3 & {}^0\mathbf{m}_n \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^0M_n & 0 \\ 0 & 1 \end{bmatrix} \\
 &= \mathcal{T}({}^0\mathbf{m}_n)\mathcal{R}({}^0M_n).
 \end{aligned} \tag{2.8}$$

Here, I_3 is a 3×3 identity matrix. We can now express the general problem from (2.5) as two separate equations

$$\begin{aligned}
 \mathcal{T}(\mathbf{A}) &= \mathcal{T}({}^0\mathbf{m}_n) \\
 \mathcal{R}(\mathbf{A}) &= \mathcal{R}({}^0M_n).
 \end{aligned} \tag{2.9}$$

Since the target end-effector orientation $\mathcal{R}(\mathbf{A})$ and position $\mathcal{T}(\mathbf{A})$ are given, we have to solve the two equations for the joint variables q_1, q_2, \dots, q_n .

Three independent degrees of freedom are resulting from the kinematic structure of a spherical wrist, therefore allowing to realize every desired end-effector orientation. With this assumption we can decouple the kinematics of the first three joints up to the elbow and the last three joints from the wrist by using the general forward kinematics formulation

as

$$\begin{aligned}
 {}^0\mathbf{M}_n &= {}^0\mathbf{M}_1{}^1\mathbf{M}_2\dots{}^{n-1}\mathbf{M}_n \\
 &= {}^0\mathbf{M}_3{}^3\mathbf{M}_6 \\
 &= \begin{bmatrix} {}^0M_3 & {}^0\mathbf{m}_6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^3M_6 & 0 \\ 0 & 1 \end{bmatrix}
 \end{aligned} \tag{2.10}$$

where ${}^0\mathbf{m}_6$ denotes the wrist center position, henceforth called \mathbf{w} , and 3M_6 the wrist orientation. We see that the first three joints solve the inverse position problem up to the wrist center, while the end-effector orientation is realized by the last three joints. The essential part of the decoupling technique for the inverse kinematics is that the position of the wrist center will not change for any possible motion of the last three links. Therefore, the wrist center position is only a function of q_1, q_2 , and q_3 for a six degree of freedom manipulator. However, it is important to note that these assumptions only hold if the end-effector is also placed in the origin of the last joint. In other words, the last link must have zero length, which is highly unpractical and inefficient. Hence, we have to consider this modification in the inverse kinematics solution. Let $d_6 > 0$ be the length of the last link, pointing directly with the desired end-effector orientation $\mathcal{R}(\mathbf{A})$ on the target position $\mathcal{T}(\mathbf{A})$. We can define the transformation from the last link into the desired end-effector coordinate frame by

$$\mathbf{A} = {}^0\mathbf{M}_6 \begin{bmatrix} & 0 \\ I_3 & 0 \\ & d_6 \\ 0 & 1 \end{bmatrix}. \tag{2.11}$$

Now, the wrist coordinate frame can be determined by inversely transforming along the last link with

$$\mathbf{W} = {}^0\mathbf{M}_6 \begin{bmatrix} & 0 \\ I_3 & 0 \\ & d_6 \\ 0 & 1 \end{bmatrix}^{-1} = {}^0\mathbf{M}_6 \begin{bmatrix} & 0 \\ I_3 & 0 \\ & -d_6 \\ 0 & 1 \end{bmatrix}, \tag{2.12}$$

where the wrist center position vector is given by

$$\mathbf{w} = \begin{bmatrix} \mathbf{W}_{14} \\ \mathbf{W}_{24} \\ \mathbf{W}_{34} \end{bmatrix}. \tag{2.13}$$

Since we now have derived the wrist center position, the two inverse kinematics problems (2.9) can finally be solved for the joint variables in question.

2.1.2.3 Iterative Inverse Kinematics

The most variable yet simplest inverse kinematics methods typically rely on iterative optimization to search for an approximate solution, due to the difficulty of inverting the forward kinematics equations and finding closed-form solutions. In contrast to uniquely derived analytical inverse kinematics solutions, iterative approaches can be easily adapted to different types of manipulators. However, these approaches unfortunately provide a limited solution space with only one nearest set of resulting joint variables.

We can use equation (2.6) to define the general inverse kinematics problem as a set of nonlinear equations as

$$\mathbf{A} = {}^0\mathbf{M}_n = \mathbf{M}(q) \quad (2.14)$$

where $\mathbf{M}(q)$ is a set of trigonometric functions given by the forward kinematics analysis for a serial manipulator with n degrees of freedom. Using an iterative approach, the *Newton-Raphson* method, we now solve these functions for the zeros of

$$\mathbf{M}(q) - \mathbf{A} = 0. \quad (2.15)$$

We use initial heuristic values for the joint variables q as

$$q^* = q + \Delta q. \quad (2.16)$$

We can calculate the resulting end-effector pose for q^* using the forward kinematics equations (2.2, 2.3) as

$$\mathbf{M}^* = \mathbf{M}(q^*). \quad (2.17)$$

Now, the translational and rotational difference between \mathbf{M}^* and the desired end-effector pose \mathbf{A} is the target error $\Delta\mathbf{M}$, defined by

$$\Delta\mathbf{M} = \mathbf{A} - \mathbf{M}^*. \quad (2.18)$$

The goal is to iteratively minimize the error by adjusting q^* , to get as close as possible to \mathbf{A} with \mathbf{M}^* . Let \mathbf{J} be the *Jacobian* matrix of all first-order partial derivatives of the vector-valued functions in $\mathbf{M}(q)$, i.e.

$$\begin{aligned} \mathbf{J}(q) &= \begin{bmatrix} \frac{\partial M_1}{\partial q_1} & \dots & \frac{\partial M_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial M_m}{\partial q_1} & \dots & \frac{\partial M_m}{\partial q_n} \end{bmatrix} \\ &:= \left[\frac{\partial M_i}{\partial q_j} \right]_{\substack{i=1,\dots,m \\ j=1,\dots,n}} \end{aligned} \quad (2.19)$$

for $i = 1, \dots, m$ independent equations in $\mathbf{M}(q)$ and $j = 1, \dots, n$ number of joint variables. For a more detailed definition and derivation of the Jacobian matrix see [91]. Assuming very small increments of Δq , the Jacobian maps the effects of joint motions into the Cartesian space as end-effector motion approximately as

$$\Delta \mathbf{M} \approx \mathbf{J}(q) \Delta q. \quad (2.20)$$

We can see that

$$\Delta q \approx \mathbf{J}^{-1}(q) \Delta \mathbf{M} \quad (2.21)$$

and, furthermore, by using (2.16) that

$$q = q^* + \mathbf{J}^{-1}(q) \Delta \mathbf{M}. \quad (2.22)$$

With equation (2.22), we can finally determine the missing joint variables q . Due to very small Δq we may now have calculated a joint configuration resulting in a slightly closer end-effector pose to our desired target \mathbf{A} . The calculated joint values can now be used as a new set for q^* , repeatedly starting at (2.16). This iterative optimization method can be described as an algorithm as follows

2.1.3 Singular Configurations

Robotic manipulators can reach arm positions called singular configurations, in which the manipulator is not able to move in specific directions. Near singular configurations, the size of the joint velocities required to maintain a desired end-effector velocity in certain

Algorithm 1 Iterative inverse kinematics algorithm

```

1: initialize:
2:   set counter  $i = 0$ 
3:   find initial heuristic values for  $q^{(i)}$ 
4:   define termination criterion as  $\|\Delta\mathbf{M}\| < \varepsilon$  or  $q^{(i+1)} - q^{(i)} < \varepsilon$ 
5: begin:
6:   calculate error  $\Delta\mathbf{M}(q^{(i)})$ 
7:   if termination criterion satisfied then
8:     solution found as  $q^{(i)}$ 
9:     terminate
10:  else
11:    calculate  $q^{(i+1)} = q^{(i)} + \mathbf{J}^{-1}(q^{(i)})\Delta\mathbf{M}(q^{(i)})$ 
12:    set  $i = i + 1$ 
13:    go to begin

```

directions can be extremely large and will, theoretically, become infinite. In singular positions, the number of degrees of freedom of the effector is lower than the dimension of the desired motion task. Generally, singular configurations occur when at least one of the following two conditions is met [42]:

1. Two axes of linear joints become parallel
2. Two axes of rotational joints become identical

In a singular position, the effector loses at least one degree of freedom. Some kinematic equations become linearly dependent or certain solutions become undefined. As a result, the Jacobian loses rank and may become rank deficient. If possible, singularities should be avoided, since most manipulators are designed for tasks in which all of the degrees of freedom are needed and, therefore, a singularity would cause the system to malfunction. There are generally two different types of singular manipulator positions in which the Jacobian has no longer full rank [42]:

1. When the end-effector is near or at the inner or outer workspace boundaries, so called *workspace boundary singularities* occur. This happens when the manipulator is either at full radius or folded back on itself.
2. Singularities may also occur inside the workspace when two or more axes line up. These are called *workspace interior singularities*.

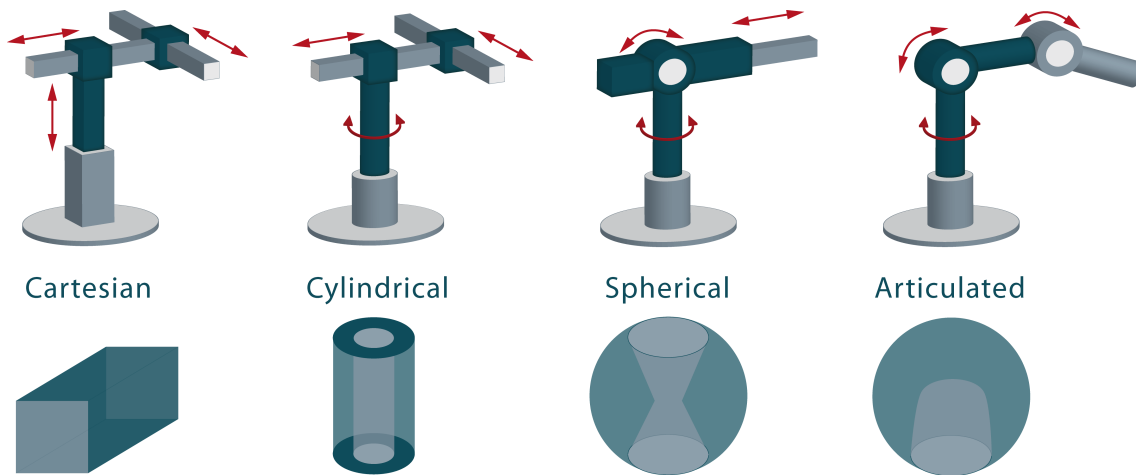


Figure 2.3: Four different basic kinematic structures with their corresponding workspaces.

However, to avoid singularities in advance, forbidden joint combinations can be determined from the Jacobian. Singularities can mathematically be found by calculating the conditions that make

$$|\mathbf{J}| = 0 \quad (2.23)$$

or

$$|\mathbf{J}\mathbf{J}^T| = 0. \quad (2.24)$$

2.2 Workspace and Dexterity

The workspace of a manipulator is the spatial volume in which the end-effector can move. It can be either a three dimensional volume or a two dimensional surface. When all joints move within their mechanical limits, the workspace is demarcated by the end-effector. The workspace is individually shaped by the kinematic structure of robots and constrained by mechanical limitations of joints. The shapes are composed of cuboids, cylinders, spheres and other complex structures. Some basic examples are shown in Figure 2.3.

Workspaces for manipulators are commonly described by technical drawings, showing the robotic arm in two different view planes. An example is given in Figure 2.4 for the *KUKA LBR iiwa 7 R800*. The figure depicts the dimensions of the robot and shows its

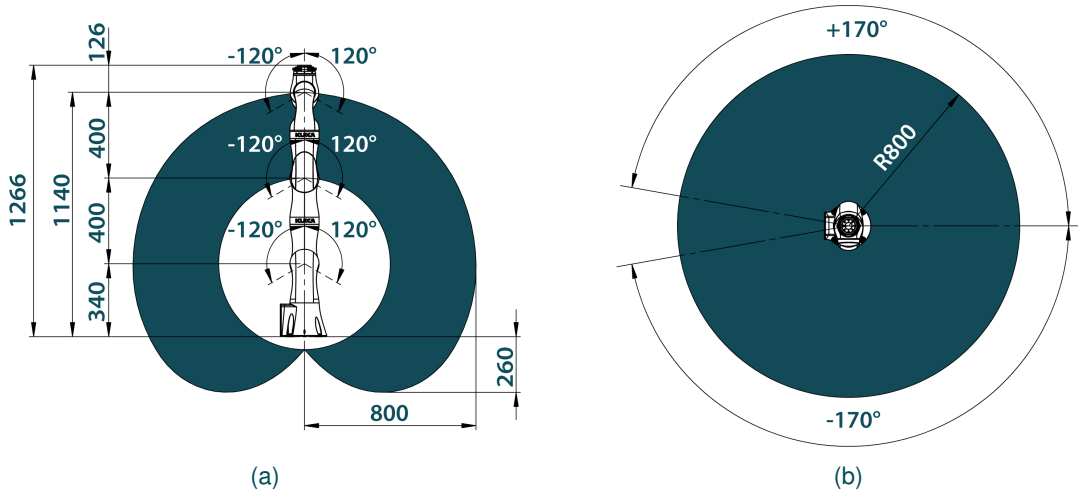


Figure 2.4: A standard mechanical drawing of the workspace specifications of the seven-axes S-R-R-S manipulator *KUKA LBR iiwa 7 R800*. The left figure (a) shows the side view in the x - z -plane and the right Figure (b) shows the top view in the x - y -plane. Lengths are given in mm.

principal ranges of motion. Note that the reference point of the working space is placed in the wrist center point.

However, the workspace is typically divided into a reachable workspace and a dexterous workspace. Points within the reachable workspace can be reached by the end-effector with at least one orientation. The dexterous workspace denotes points which can be reached in all possible orientations and is usually a subset of the reachable workspace.

Let Q be the joint space of a manipulator consisting of all possible values of the joint variables q of the robot. Then Q is also the configuration space of the manipulator, since specifying the joint angles also specifies the location of all of the links of the robot [73]. Now, the workspace of a manipulator is defined as the set of all end-effector configurations which can be reached by joint angle combinations. Let $f(q) : Q \rightarrow SE(3)$ be the forward kinematics function, then the workspace W can be described as the set

$$W = \{f(q) : q \in Q\} \subset SE(3), \quad (2.25)$$

where $SE(3)$ denotes the three dimensional special Euclidean group for rigid motions. This is the definition of a complete workspace of a manipulator. Instead of characterizing

the workspace as a subset of $SE(3)$, one can consider the set of positions in \mathbb{R}^3 which can be reached. This set is called the reachable workspace and can be defined as

$$W_r = \{p(q) : q \in Q\} \subset \mathbb{R}^3. \quad (2.26)$$

Here, $p(q) : Q \rightarrow \mathbb{R}^3$ denotes only the translational component of the forward kinematics map $f(q) : Q \rightarrow SE(3)$.

The reachable workspace is the volume of \mathbb{R}^3 which can be reached with at least one end-effector orientation. The dexterous workspace of a robot is the volume of space which can be reached by the manipulator with arbitrary end-effector orientation and can be defined as

$$W_d = \{p \in \mathbb{R}^3 : \forall R \in SO(3), \exists q \text{ with } f(q) = (p, R)\} \subset \mathbb{R}^3. \quad (2.27)$$

Due to highly varying geometries of workspaces and their dependencies on multiple parameters, it is not yet possible to find a unified mathematical representation of workspaces for all kinds of robots. Approximations, however, have been derived from analytical calculations for specific manipulator types [20, 107] and by numerical approaches [30, 112]. These approaches are not satisfying, since they provide only highly limited solutions. A reliable general approximation is provided by the discretization of the joint variables. A huge number of forward calculations is then performed using random, unique combinations of the discretized joint values. Finally, the resulting end-effector poses incrementally build the workspace. This method is obviously computation-intensive and, in addition, leads to discretization of the workspace and therefore yields a discontinuous representation.

In application-driven environments, robots are often selected to fulfill a single, specific task, e.g. pick-and-place of objects in an assembly line. When choosing a manipulator for a certain industrial purpose, it is important that the workspace is large enough to encompass all the points that the arm will need to reach, but it is wasteful to use a robot with a workspace much larger than necessary [49]. For a desired task, the operational space in which the robot has to move and, likewise, required end-effector orientations are known. This taskspace may be hardly restricted by prohibited zones due to obsta-

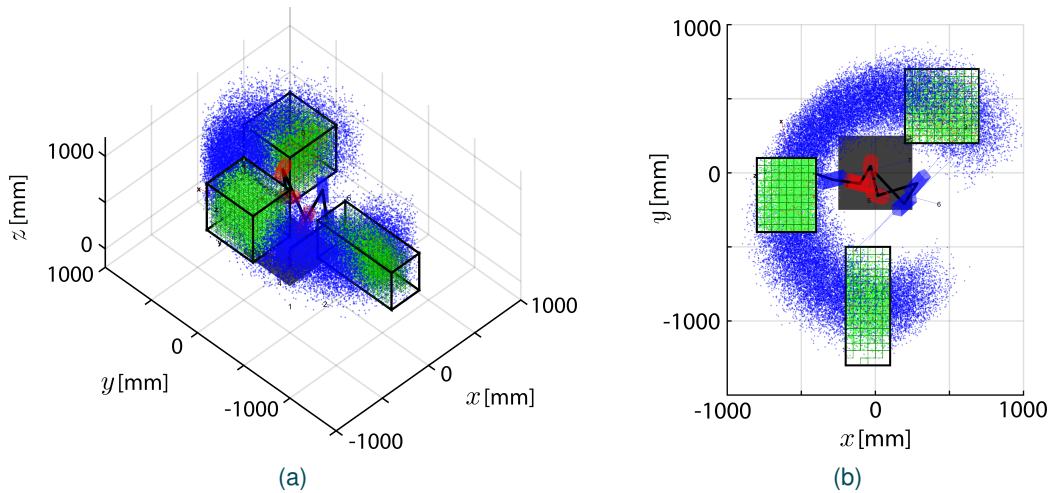


Figure 2.5: The volumes of three predefined tasks, shown as black cuboids, are partially covered (green points) by the workspace (blue points) of the optimized manipulator [89].

cles, e.g. other robots, objects or humans. However, by comparing the robots' calculated workspaces with the desired taskspace one can determine whether and to what degree the task could be performed. An example is shown in Figure 2.5. The graphic was generated using a toolbox for optimizing serial manipulators for predefined tasks [89]. The volumes of three predefined tasks, shown as black cuboids, are partially covered (green points) by the workspace (blue points) of the optimized manipulator. The kinematic structure has been highly specialized for the given task. However, the size and shape of workspaces are an important factor in the design of task-specific manipulators and can be used as a fitness metric of different robots for specific tasks [27, 111].

2.2.1 Configuration Space

The forward and backward kinematics of a manipulator relate the range of values available for the joint parameters, called the *configuration space* of the robot, to the geometric workspace of the effector. The configuration space, also called joint space, is an elementary tool for robotic motion planning [8, 12, 66]. The location of all parts of the robot geometry is called configuration q . A configuration can therefore be expressed as an n -tuple of the joint angles. The configuration space C , is the space of all possible configurations $q \in Q$. Thus, the configuration space represents the set of all transformations that can be applied to a robot based on its kinematics. The number of degrees of free-

dom, or the minimum number of parameters needed to specify a configuration q of a manipulator, define the dimension of the configuration space Q .

The advantage of observing arm motions in this representation is that complex geometric configurations can be mapped to a single point in the configuration space. In motion planning, the configuration space is a useful tool to abstract planning problems in a unified way. A major challenge in motion planning is to avoid obstacles in the workspace. The workspace is a static environment and may be populated with obstacles. The goal of robot motion planning is to find a collision-free path to move from an initial position and orientation to a desired goal pose [97]. Obstacles are forbidden zones in the workspace of a manipulator. Using the robot's kinematics, they can be mapped back to forbidden joint values in the robot's configuration space. Trajectories to a target position through the free space of Q can now be used to avoid the spatial obstacles.

2.3 Kinematically Redundant Manipulators

Challenging applications sometimes require the use of so called kinematically redundant manipulators with more than six degrees of freedom, for example when high dexterity or obstacle avoidance are necessary. Kinematically redundant manipulators, like the *KUKA LBR iiwa* with its seven degrees of freedom, have more joints than those required to execute a task. This provides the robot with additional dexterity [120]. An advantage of this additional dexterity is the ability to avoid joint limitations, singularities, and obstacles, but also to reduce joint torques, motion energy or, in general, to optimize suitable performance parameters [98]. Increased dexterity can also reduce the amount of configuration changes within single motion trajectories and thus allows following optimized paths which would be impossible without it. In dynamic scenarios where spontaneous configuration changes, for example due to joint limitations, are strictly forbidden, the additional dexterity is of great advantage [71]. A further benefit of kinematic redundancy is the specific positioning of the manipulator to adapt its occupied space to other parallel applications in a shared domain.

When a robot has more degrees of freedom than needed to fulfill a desired task, the system becomes kinematically redundant. To this end, manipulators may not be inherently

redundant, there are rather special tasks with respect to which they become redundant. For example, a planar manipulator with three degrees of freedom becomes redundant if the end-effector orientation angle is of no concern for a two-dimensional manipulation task. Assuming a complex spatial task including combined end-effector translations and rotations requiring six degrees of freedom, a robot with seven or more degrees of freedom is commonly considered as an inherently redundant manipulator. However, if the desired task is simple enough, even manipulators with less than seven degrees of freedom, like conventional six-joint industrial manipulators, may become kinematically redundant.

In traditional engineering, the introduction of redundancy to systems is motivated by the intention of increasing reliability and robustness by duplicating critical components or functions, e.g. redundant power supplies, sensors or cables. When integrating redundancy in robots, however, the motivation is different. It is the intention to increase the dexterity, like it is with a human arm.

A manipulator with a minimum number of degrees of freedom is highly specialized for specific types of task. Examples are SCARA (selective compliance assembly robot arm) robots and portal robots, which were designed for pick-and-place tasks only. This minimal design offers the advantages of low construction and maintenance costs. However, this high level of specialization also involves some drawbacks. The end-effector rotations are often limited to a single axis of operation, which leads to serious limitations. In addition, singularities and joint limits are major challenges in real-world applications. The remaining operational area in the joint space is therefore severely restricted by forbidden zones, making it difficult to avoid potential obstacles inside the workspace. All these factors require a very static and predictable workspace for these highly specialized robots.

In contrast, if the robot has more degrees of freedom than the planned task requires, this generally leads to more dexterity and, potentially, autonomy. The additional degrees of freedom allow for motions of the manipulator without changing the pose of the end-effector. These motions are also known as self-motion and are typical for kinematically redundant robots [14]. In other words, the self-motion can be used to accomplish the same tasks in different joint configurations. This massively increases the usable area in the joint space, making the robotic system more versatile. This also helps to avoid singularities, joint limits, and obstacles inside the workspace. These are key features for

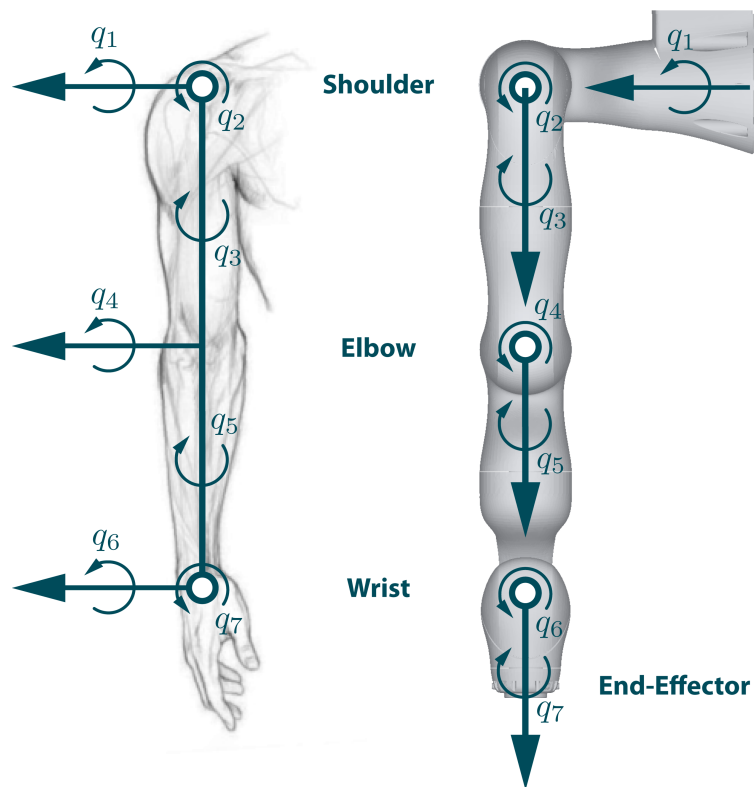
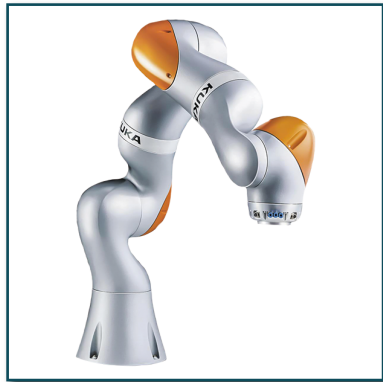


Figure 2.6: Comparison of kinematic models of the human arm and redundant manipulators.

applications in unstructured or dynamically varying environments as they occur in modern industrial applications and service robotics.

The human arm serves as a model for the structural design of kinematically redundant robots. The seven degrees of freedom of the human arm are comparable to those of commonly available seven-jointed manipulators. Figure 2.6 shows the comparison of the kinematic models of a human arm and a *KUKA LBR iiwa*. Although there are small differences in the layout of the joint axes, the resulting degrees of freedom and therefore potential motions are the same for both models. In fact, the human arm has three degrees of freedom at the shoulder, one at the elbow and three at the wrist. A number of today's robots with seven degrees of freedom have copied the functionality of the human arm. These manipulators are called anthropomorphic and share a common S-R-S (spherical-revolute-spherical) structure. They only differ in the link lengths and in the joint limits.



(a) KUKA LBR iiwa (Image by courtesy of KUKA AG, all rights reserved. [58])



(b) FRANKA EMIKA Panda (Image by courtesy of FRANKA EMIKA, all rights reserved. [19])



(c) Tokyo Robotics Torobo Arm (Image by courtesy of Tokyo Robotics Inc., all rights reserved. [105])



(d) Yaskawa MOTOMAN SDA10F (Image by courtesy of Yaskawa Europe GmbH, all rights reserved. [116])

Figure 2.7: Four examples of kinematically redundant robots. The manipulators (a)-(c) have seven joints and seven degrees of freedom. They share the same S-R-S (spherical-revolute-spherical) kinematic structure. The anthropomorphic robot (d) uses two arms of the same S-R-S design. With an additional rotational joint in its torso the robot consists of 15 actuated joints.

Figure 2.7 shows examples of current redundant robotic systems which are commercially available or in development.

Manipulators with a larger number of joints are often named hyper-redundant manipulators and include many snake-like robots. In general, adding degrees of freedom increases cycle time and reduces load capacity and accuracy for a manipulator.

2.3.1 Collaborative Robots

Per definition, collaborative robots, also known as cobots, are manipulators built for physical interaction with humans in a shared domain [78]. These interactions have gained importance over the last years and have advanced to be one of the biggest research fields in robotics nowadays. Robotics is currently undergoing a fundamental change in the way humans and machines work together. Today, the future lies in new concepts of sensitive, compliant manipulators that can work safely in cooperation with humans. In fact, they start replacing the classical and potentially harmful position-controlled rigid robots carrying out typical automation tasks, which dominated the market for the last decades.

The biological counterpart to stiff position-controlled manipulators are muscles with advanced functional performance in combination with a neuromechanical control system, allowing for fast and efficient parameter adaption. The most relevant difference is the compliance or springlike nature as we find it in biological systems [4]. However, such compliant actuators may not be suited for classical precise, position-controlled tasks, but might offer beneficial features for new applications, e.g. physical human-robot interaction (HRI), robotic prostheses, and legged robots [121]. A new generation of mechatronic concepts in general robot design within the compliance context are entering the field [26]. However, this trend leads towards the long-term goal of safe human-robot interaction for active and safe workspace sharing and collaboration with humans.

The human-centered design of robot mechanics and controls was a key technology for the development of HRI-compatible systems [29, 104]. Intrinsically elastic robots like series elastic actuators (SEA) [80] or variable impedance actuators (VIA) [110] have been introduced by this design paradigm. Whilst the physical contact between robot and operator is already considered during the design phase, unwanted and potentially harmful collisions can be prevented. In addition, understanding the intention and preferences of human motions and considering it in motion planning will enable the realization of safe cooperations and interactions in a shared workspace [70]. Some of the most advanced systems like the *KUKA LBR iiwa* robot that were developed are entering into industrial markets. These compliant systems are well accepted and show great potential for both industrial and service-oriented domains.

2.3.2 The KUKA Light-Weight Robot

A specific focus of this chapter is on the collaborative lightweight robot series from the manufacturer *KUKA Roboter GmbH*. Almost all experiments described in this thesis were performed with the type *LBR iiwa 7 R800* (Figure 2.7a). The KUKA LBR was the first commercially available seven-jointed robot equipped with integrated force sensors. The combination of an anthropomorphic, redundant kinematic structure with seven degrees of freedom and built-in torque sensors in each axis was unique on the market.

The robot was fundamentally developed by the Robotics and Mechatronics Center at the German Aerospace Center (DLR, Germany). There, research on redundant compliant manipulators started in 1991 with the *DLR Light-Weight Robot I*, known as *ROTEX*. It was the first robotic system which operated in space [33]. The development of lightweight robots was initially motivated by the lack of a light, flexible robot system for training astronauts. At that time, the robot arm already had seven axes in an anthropomorphic S-R-S structure and was equipped with torque sensors. Built for space, the robot was not able to move its own weight on earth. The development was finished in 1996.

The following version, the *DLR Light-Weight Robot II*, was developed from 1998 to 2001 [32]. The second generation was a technical improvement that could now also work on Earth under normal gravitational forces. The robot had a then unique load to weight ratio. With a self-weight of only 18kg, the manipulator could move a load with a maximum of 7kg. The kinematic structure was kept identical to the previous model.

In 2003, the third generation of sensitive lightweight robots, the *DLR Light Weight Robot III (LWR)*, was introduced [34]. With the improved version, a ratio of self-weight to load of 1:1 was achieved for the first time. The robot could carry 14kg load and weighed only 14kg. It was the first generation which was completely torque controlled and therefore particularly useful for human-machine-interaction [3].

In 2004, *KUKA Roboter GmbH* licensed the LWR III as a new trademark: the *Leichtbauroboter (LBR)* series. The company advanced the technology and, in 2008, released the *LBR 4*, the first commercial non-prototype version in small series production [10].

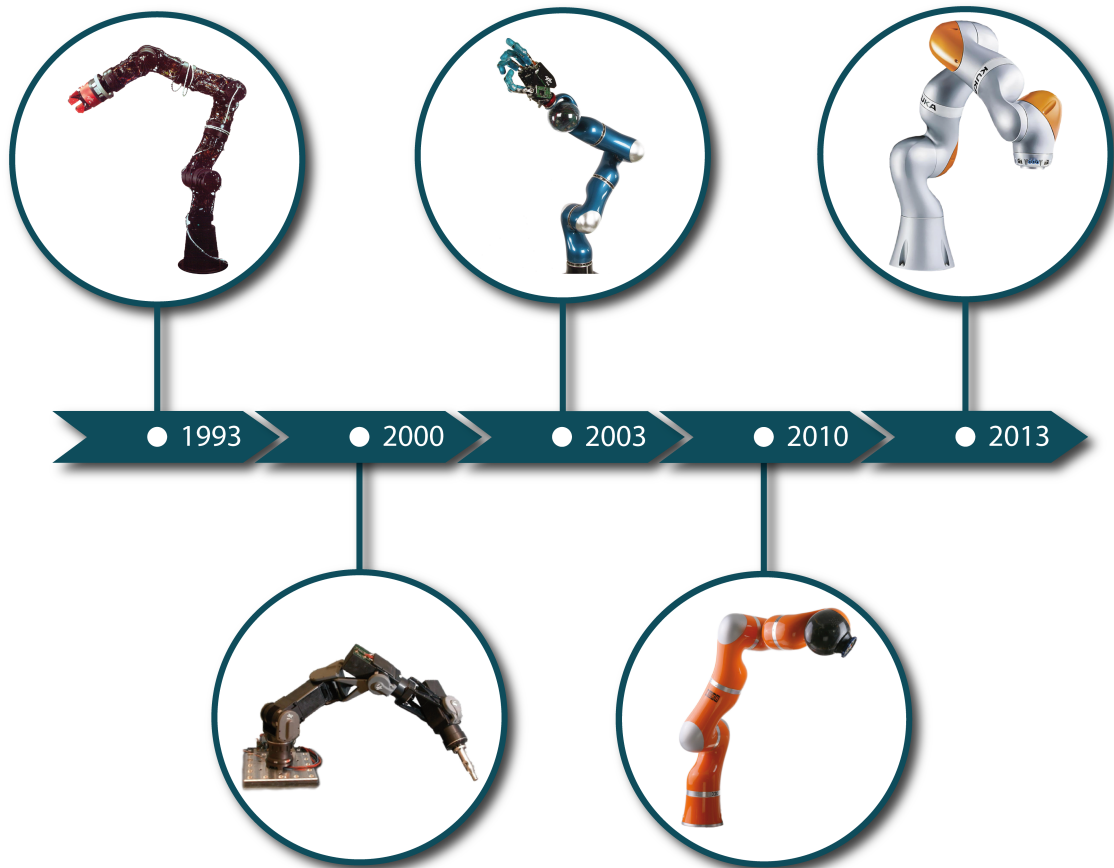


Figure 2.8: Five generations of the DLR lightweight robot. From left to right: LWR I (modified from [24]); LWR II (modified from [25]); LWR III [1]; LBR 4 (modified from [9]); LBR iiwa (Image by courtesy of KUKA AG, All rights reserved. [58]).

Two years later, in 2010, the refined *LBR 4+* was introduced, this time in larger quantities. Both, the *KUKA LBR 4*, and the *KUKA LBR 4+*, were a commercial success and showed large potential for industry and research.

To this end, KUKA released the *LBR iiwa* in 2013 with a total redesign of the control software and optimized hardware. All five generations of the lightweight robot series are shown in Figure 2.8 on a time line.

The *LBR iiwa* was designed for particularly user-friendly control with its new *KUKA Sunrise.OS* system software. With the expansion *KUKA Sunrise.HRC*, important functionalities were enabled for safe human-robot collaboration in Performance Level d with Cate-

gory 3 according to EN ISO 13849-1:2008 [40, 57]. The safety features include force and torque monitoring, collision detection, and velocity monitoring. The available expansion *KUKA Sunrise.FRI* with the *Fast Research Interface* enables real-time control and monitoring of robot parameters with a cycle time of up to 1kHz [90] and forms an important tool for research applications.

2.4 Contact Force Estimation

To enable safe physical human-machine-interaction, it is paramount for collaborative robots to measure and react accordingly to physical contacts with objects inside their workspaces. However, some manipulators like the *KUKA LBR iiwa* are not inherently compliant, but rather use active impedance control. These actuators mimic impedance behavior by using software controllers. Based on the measured system's state, a correction is calculated by the controller and applied by the stiff, position-controlled actuator. These manipulators require a precise determination of external forces to properly control their impedance. Therefore, mathematical models are used to simulate the physical and mechanical behaviour of the robots. In addition to the inherent dynamics induced by the manipulator, additional external forces from the environment can be included in the models.

However, depending on the nature of the robot's motion, it is necessary to differentiate between dynamic and quasi-static scenarios. Each integrated component of the links contributes to the total mass of the robot arm and behaves inertly, depending on speed and acceleration. In a dynamic scenario with high accelerations and velocities, mass induced inertial, centrifugal and Coriolis forces at the manipulator must be considered in the control algorithms [45]. These additional impacts increase significantly the complexity of the control laws. If the robot's motion is slow and steady enough, these mass-induced impacts become negligibly small and the manipulator operates in a quasi-static state. In these scenarios, the determination of contact forces, however, is greatly simplified.

2.4.1 Mechatronic Joint Design

The mechatronic design of the joints will be explained on the basis of the LWR III robot, whose electronics and mechanics are well documented in literature [1, 34]. The basic ar-

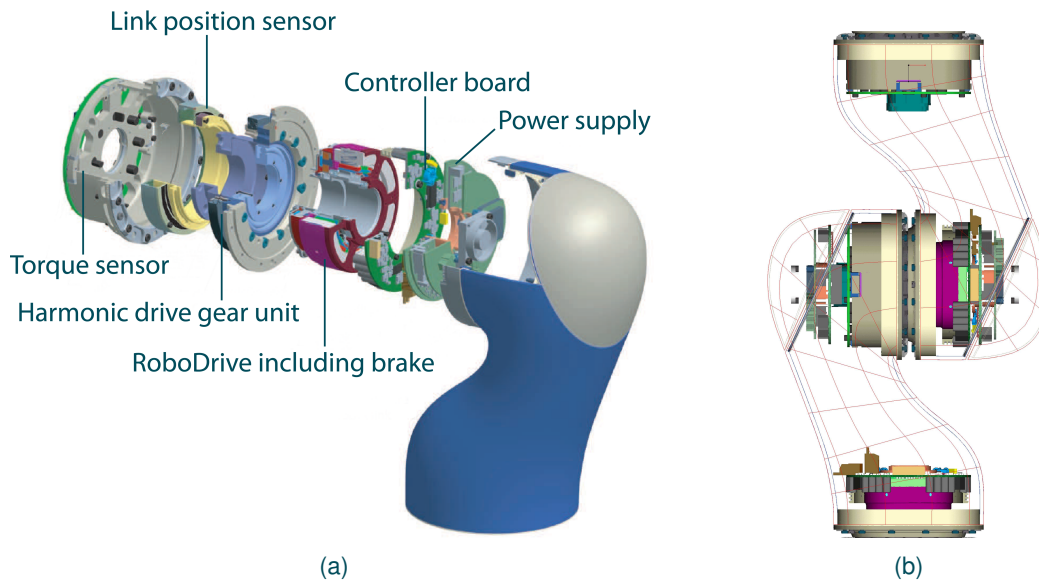


Figure 2.9: The mechatronic joint design of the DLR Light-Weight Robot III (modified from [1]). Essential components including actuation, electronics, and sensing are shown in an exploded view (a) and in an assembled construction (b).

chitecture was adopted in the later commercial version KUKA LBR iiwa. The mechatronic joint design with the essential components is shown in Figure 2.9.

The drive unit is composed of lightweight *Harmonic Drive* gears and *RoboDrive* motors possessing high energy density. RoboDrive motors have been developed at the DLR and are optimized servo motor drives, particularly suitable for service robotics applications where speed is relatively low, but minimal weight and high torques are essential. In addition, these motors provide high dynamics while frequently reversing operation (changing direction of rotation) around zero speed.

The integrated Harmonic Drive gears are strain wave gearings, known as harmonic gearings. These special types of mechanical gear systems are based on elastic dynamics and utilize the flexibility of metal. The mechanism is typically built of three basic components: a wave generator, a flexspline, and a circular spline. However, harmonic gears provide several advantages: negligible backlash, compact size and light weight, while still allowing for high gear ratios. They are typically used in actuated robotic manipulators.

Essential electronics including power supply, power converter units, and controller boards are integrated in the housing of every joint. This design reduces the number of wires inside the arm and also allows for fast joint level control. The state of every joint is measured at a frequency of 3kHz and includes the following information:

- Joint torques measured by strain-gauge torque sensors
- Motor position based on magneto-resistive encoders
- Link position based on potentiometers as redundant sensors for safety

The torque sensors are built as a four-way strain-gauge bridge in a classic load cell design. They are mounted directly to the *flexspline* component of the Harmonic Drive gear, and therefore connect the links with the drive units. This construction design provides the advantage of isolating the torques acting on the links, while an additional bearing decouples the disturbing forces and torques from the motor. When low-pass filtering the output signals at 600Hz, the torque sensor achieves a maximum relative error of 2% [57].

Joint level control is implemented as a state feedback controller at 3kHz rate on a controller board in each joint [2]. The robot dynamics and the Cartesian control are computed in a 1kHz cycle on a central controller unit, connected over an optical fiber bus.

2.4.2 Dynamic Force Estimation

In this section, the integrated proprietary force control method of the KUKA LBR iiwa will be briefly discussed. Instead of describing the highly complex dynamics models in detail, this section focuses on essential force control formulations and points out how the proprietary model calculates end-effector contact forces using internal sensors. However, a detailed description of the proprietary robot control architecture is presented in [3, 28].

In the presence of external joint torques, acting on the kinematic due to contact forces, the following dynamics model of robots with flexible joints is considered:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{g}(q) = \tau_J + \tau_{\text{ext}} \quad (2.28)$$

$\mathbf{M}(q) \in \mathbb{R}^{n \times n}$ is the symmetric and positive-definite inertia matrix, $\mathbf{C}(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is the centrifugal and Coriolis matrix, and $\mathbf{g}(q) \in \mathbb{R}^n$ is the gravity vector. We denote $\tau_{\text{ext}} \in \mathbb{R}^n$

as the external joint torque, $\tau_J \in \mathbb{R}^n$ is the elastic torque transmitted through the joints as

$$\tau_J = \mathbf{K}_J(\theta - q), \quad (2.29)$$

where $\mathbf{K}_J \in \mathbb{R}^{n \times n}$ is the diagonal and positive definite joint stiffness matrix and $\theta \in \mathbb{R}^n$ are the motor positions. Let $\mathbf{F}^{ecs} \in \mathbb{R}^6$ be the wrench at the end-effector as

$$\mathbf{F}^{ecs} = \begin{bmatrix} \mathbf{f}^{ecs} \\ \mathbf{m}^{ecs} \end{bmatrix}, \quad (2.30)$$

consisting of contact forces $\mathbf{f}^{ecs} \in \mathbb{R}^3$ and moments $\mathbf{m}^{ecs} \in \mathbb{R}^3$. By using the transposed geometric Jacobian matrix $\mathbf{J}(q)$ of the end-effector, we can calculate the external joint torques as

$$\tau_{\text{ext}} = \mathbf{J}^T(q) \mathbf{F}^{ecs}. \quad (2.31)$$

Finally, by including (5.13) in the general robot dynamics from (5.10), we can calculate the end-effector wrench as

$$\mathbf{F}^{ecs} = \mathbf{J}^T(q)^{-1} (\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{g}(q) - \tau_J). \quad (2.32)$$

2.4.3 Quasi-Static Force Estimation

Assuming the robot's motion is so slow and steady that mass induced impacts like inertial, centrifugal and Coriolis forces are becoming negligibly small, the manipulator operates in a quasi-static state. While the joint velocities \dot{q} and the joint accelerations \ddot{q} become zero, the terms $\mathbf{M}(q)\ddot{q}$ and $\mathbf{C}(q, \dot{q})\dot{q}$ are no longer affecting the resulting torques of the force estimation model. Now, the determination of contact forces is greatly simplified as

$$\mathbf{F}^{ecs} = \mathbf{J}^T(q)^{-1} (\mathbf{g}(q) - \tau_J), \quad (2.33)$$

given that the geometric Jacobian can be calculated for the configuration q and has full rank.

3 Robust Inverse Kinematics Solution¹

This chapter addresses the development of an optimized, robust inverse kinematics solution in a closed form for kinematically redundant manipulators with seven degrees of freedom and zero link offsets like the KUKA LBR iiwa lightweight robot. The goal is to enable full control of the additional seventh degree of freedom, the elbow self-motion manifold, by an intuitive angle parameter. The inverse kinematics algorithm should additionally provide a complete solution space and consistent control of all possible configurations for one single target end-effector pose. The developed algorithm will be evaluated looking at configuration reliability and continuity for exemplary, challenging motion paths where spontaneous configuration changes within a single trajectory must be avoided. The chapter begins with an introductory section describing the necessities and the resulting motivation for developing a new inverse kinematics solution by discussing prior related work. After an analysis of the underlying kinematic structure, an overview of the algorithm is given by pointing out major steps. Subsequently, the inverse kinematics solution is described in detail. It is followed by an evaluation of the developed solution with respect to initially defined requirements. The chapter concludes with an assessment of the most relevant findings.

3.1 Introduction

The effective and intuitive use of the self-motion manifold is a well known and much discussed problem since the introduction of kinematically redundant seven-axes articulated arms. Not only to exhaust the full potential of such manipulators, but also to achieve the establishment of such systems in demanding environments like medical technologies, the development of an efficient and robust inverse kinematics solution is of paramount importance.

¹Parts of this chapter have been published in [56]

During the analysis of the underlying kinematic characteristics, the seven-jointed S-R-S robot exhibits the same arm configuration manifolds as common six-axes robots (e.g. like the *PUMA*). In fact, there are eight possible configurations, commonly indicated by the three parameters `arm`, `elbow`, and `wrist` [62, 93]. The direct integration of these parameters in inverse kinematics algorithms is a common approach [41, 101] due to several advantages: Single configurations can be chosen consistently out of the set of all possible solutions, therefore the solution space becomes more transparent. Especially for robotic simulations, the opportunity to command specific arm configurations is an interesting feature. Furthermore, the consistent control of arm configurations increases the safety of robot motions, because spontaneous configuration changes within a single trajectory are avoided mathematically. This avoidance of joint flips is a common restriction due to special technical, safety, or efficiency requirements, e.g. to prevent collisions in medical applications with physical human-robot interaction [7, 51]. In this case, the possibility to reliably command arm configurations is essential.

However, iterating algorithms of inverse kinematics are often no alternative to closed-form solutions, because of higher computational costs, limited solution space with no configuration manifolds, or non-deterministic runtimes.

Unfortunately, no existing inverse kinematics solution in closed form for redundant robots provides the required combination of major features:

1. Intuitive elbow self-motion manifold control
2. Robust arm configuration control

Thus, the development of a closed-form inverse kinematics solution for kinematically redundant manipulators, achieving these two demands forms the goal of this chapter.

3.2 State Of The Art

A number of publications treat the solution of the inverse calculation problem for articulated arms with seven degrees of freedom. Most of these solutions are based on the

Jacobian matrix, either in closed or iterating form [37, 48, 63, 68, 75, 76, 117]. However, these solutions do not provide any active control of underlying arm configuration manifolds and are thus not satisfying and will not be considered further. There are also several publications pursuing approaches to closed-form algorithms. Although most of these solutions provide consistent results, none of them solves the inverse kinematics by combining an intuitive way to control the elbow motion, together with the possibility to select all arm configuration manifolds. To clarify the advantages and algorithmic differences of the developed solution from this chapter compared to those already existing, the differences are discussed in the following section.

Dahm and Joublin [16] were the first to formulate the inverse problem in closed-form, based on a geometrical approach. They showed that the elbow in its self-motion manifold describes a circular path and found a geometric way to calculate all desired joint angles for a desired elbow position. Unfortunately, the algorithm does not consider any arm configuration manifolds but the elbow and thus does not take the whole solution space into account. Consequently, it cannot provide active arm configuration control.

Shimizu *et. al.* [94] published an analytic solution for a similar redundant kinematic system, considering joint limitations. The algorithm leads to feasible results and also allows searching for possible elbow positions in the range of self-motion, which is in fact a very challenging task. Although they provide a smart solution, there are two drawbacks: First, there again is no way to use common configuration parameters e.g. `arm`, `elbow` or `wrist` to actively control the arm configuration manifolds. Second, the approach partially relies on a numeric optimization step, rendering it problematic in time-constrained settings due to its non-deterministic runtime.

Asfour and Dillmann [6] formulated the inverse problem for the anthropomorphic movement of redundant manipulators. They modeled a human arm using the DH-parameters and solved the inverse kinematics for this special robot kinematic. Unfortunately, they defined the elbow position in a less intuitive way using three solid angles and, furthermore, used joint limitations to imitate the characteristics of natural human arm movements. Detailed analysis of the algorithm reveals that the results are limited in the solution space and do not include all configuration manifolds.

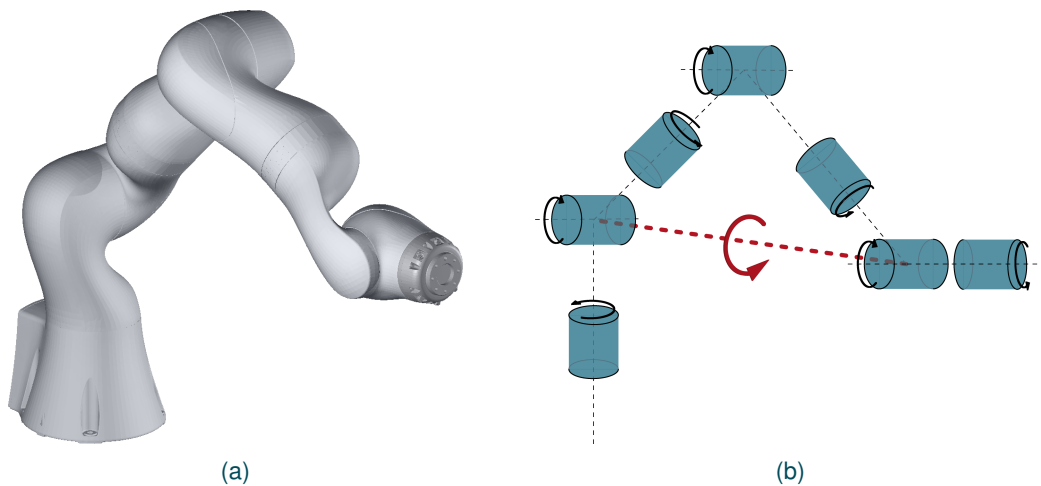


Figure 3.1: A seven-axis S-R-S redundant manipulator. Left: The KUKA LBR iiwa robot with the abstracted kinematic model on the right.

Other published closed-form solutions for anthropomorphic movement of redundant manipulators like these given by Wang and Artemiadis [113], Tondu [106], or Kim and Rosen [47] do have comparable anthropomorphic restrictions like Asfour and Dillmann [6] and do not solve the inverse problem in the full workspace of the robot, but rather in a very limited range of a human arm without considering configuration manifolds. Of course, anthropomorphic restrictions (e.g. $-90^\circ \leq \theta_1 \leq 90^\circ$) vastly simplify the inverse kinematics, but ultimately are not adequate to control a robot like the *LBR iiwa* in its full workspace. Unfortunately, further smart solutions lack of consistent, active arm configuration manifold control parameters [5, 72, 115].

This shows that an inverse kinematic solution combining active arm configuration control and parametrization of the elbow self-motion manifold in an intuitive way does not yet exist. The development of this solution is the purpose of this chapter.

3.3 The Inverse Kinematics Algorithm

The robot and an abstracted kinematic model is shown in Figure 3.1. To clarify the structure of this serial kinematic system, the underlying DH-parameters are shown in Table 3.1.

Table 3.1: DH-Parameter of the underlying seven-jointed redundant kinematic system

i	a_i	α_i	d_i	θ_i
1	0	$-\pi/2$	d_1	θ_1
2	0	$\pi/2$	0	θ_2
3	0	$-\pi/2$	d_3	θ_3
4	0	$\pi/2$	0	θ_4
5	0	$-\pi/2$	d_5	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	d_7	θ_7

3.3.1 Algorithm Overview

Based on the ability to realize all orientations with a spherical S-R-S wrist, the global inverse kinematics problem can be divided into two sub-problems. Following the decoupling principle, as classically used for common manipulators with six degrees of freedom, we define the two sub-problems as the inverse position and the inverse orientation problem as described in Section 2.1.2.2. To achieve this, the joints $\theta_{(1..4)}$ are used to solve the inverse position problem up to the wrist center point, while $\theta_{(5..7)}$ are used to realize the orientation of the end-effector. The deterministic elbow motion on a circular path allows the definition of an intuitive self-motion manifold angle δ . The angle describes the rotation of the elbow around the axis from shoulder to wrist (Figure 3.3a). In this figure, the position \mathbf{E}_0 shows the elbow position for an angle of $\delta = 0$.

However, to correctly determine the joint angles, it is necessary to view this circular path as a projection into the x - y -plane, there forming an ellipse. Based on this projection, we can choose θ_3 in the correct quadrant with respect to the underlying ambiguities. At this point, an elliptical tangent equation is used to resolve the ambiguities for θ_3 .

To simplify the calculation, the algorithm starts with an initial rotation of the scene. Assuming that the base coordinate system (BCS) is centered at the base of the robot, we rotate the desired target pose about the z -axis of the BCS, with the result that the center of the wrist now comes to lie exactly in the x - z -plane (Figure 3.3b). This is valid because the rotation can be reversed later on by a counter-rotation by θ_1 . Starting with this scenario, the position of the elbow in zero position ($\delta = 0^\circ$) is easy to determine and is further used to calculate the target elbow position for a desired target angle δ . Based on

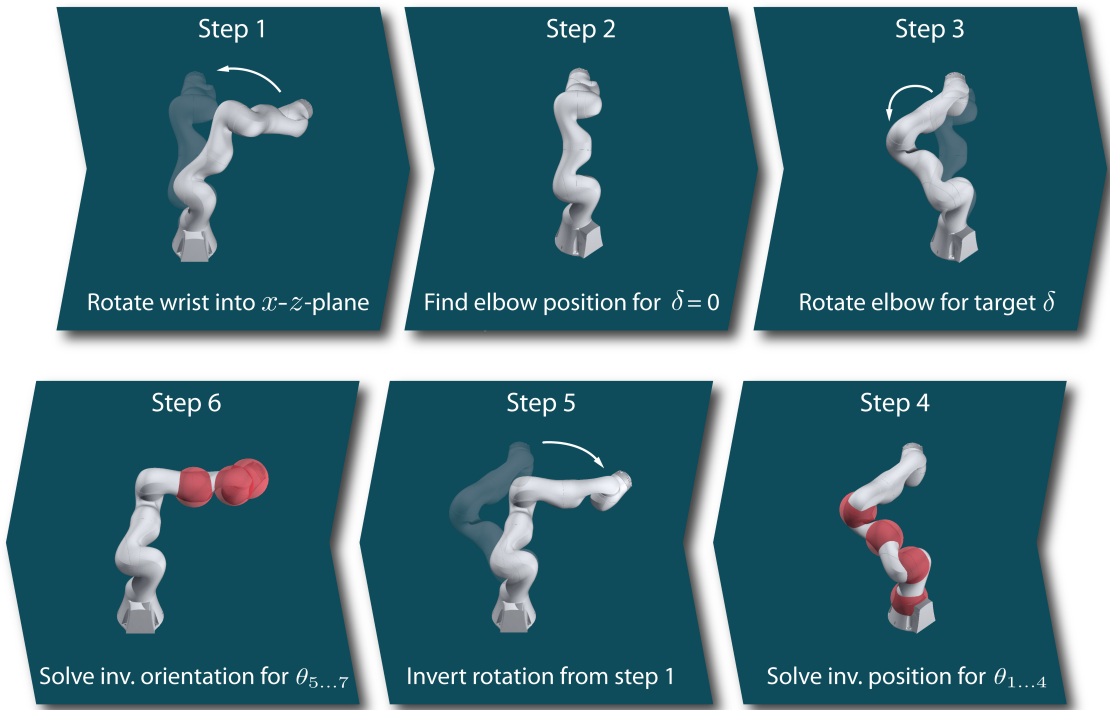


Figure 3.2: The six steps of the inverse kinematics solution: (1) Rotate the wrist center into the x - z -plane; (2) Determine the elbow zero position for $\delta = 0$; (3) Rotate the elbow for the desired δ angle; (4) Solve the inverse position up to the wrist for $\theta_{(1..4)}$; (5) Invert rotation from step 1 with θ_1 ; (6) Solve the inverse orientation for $\theta_{(5..7)}$.

this elbow position, all joint angles can be determined geometrically to solve the inverse position problem. After inverting the initial rotation by counter-rotating with θ_1 , we finally solve the last three joints analytically for the inverse orientation problem. In summary, we will take six essential steps as shown in Figure 3.2.

3.3.2 Closed-Form Solution

Considering the kinematics abstraction in Figure 3.3a, let \mathbf{M} be the target pose matrix with respect to the BCS. The inverse position is solved to the wrist center position \mathbf{w} , calculated by

$${}^0\mathbf{M}_6 = \mathbf{M} \cdot {}^6\mathbf{M}_7^{-1} =: \mathbf{W} = \begin{bmatrix} W & \mathbf{w} \\ 0 & 1 \end{bmatrix}. \quad (3.1)$$

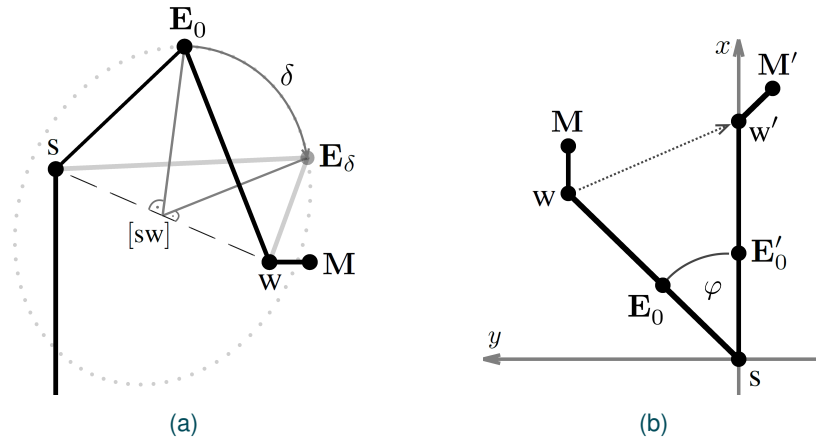


Figure 3.3: Left: The abstraction of the kinematic model. Right: Projection into the x - y -plane: The wrist center has to be placed in the x - z -plane of the BCS.

As mentioned before, it is necessary to place the wrist center in the x - z -plane of the BCS in an initial step. To achieve this, we can extract the rotation angle φ out of a projection into the x - y -plane and rotate w around the z -axis (BCS) with

$$\varphi = \text{atan2}([w]_y, [w]_x) \quad (3.2)$$

and

$$w' = R_{z, -\varphi} \cdot w, \quad (3.3)$$

where $R_{z, -\varphi}$ is an elemental rotation by the angle $-\varphi$ around the z -axis.

Let s be the shoulder position

$$s := \begin{bmatrix} 0 \\ 0 \\ d_1 \end{bmatrix} \quad (3.4)$$

and let \vec{e}_x be a unit vector pointing along the x -axis. Now let $R_{[sw]} = R_{z, \Psi} \cdot R_{x, \Theta} \cdot R_{z, \Phi}$ be

a rotation matrix defined by the three solid angles Ψ , Θ , Φ such that

$$R_{[sw']} \cdot \vec{e}_x = \frac{[sw']}{|[sw']|}. \quad (3.5)$$

Clearly, the angle Φ can be selected arbitrarily, we set $\Phi = 0$.

Now we can combine the known position of the shoulder with the rotation $R_{[sw']}$ to a homogeneous matrix as

$$\mathbf{S} := \begin{bmatrix} R_{[sw']} & \mathbf{s} \\ 0 & 1 \end{bmatrix}. \quad (3.6)$$

The elbow zero position \mathbf{E}'_0 for $\delta = 0$ can be calculated using the geometric angle α as shown in Figure 3.4b. Now let

$$d_{[sw']} = \|\mathbf{w}' - \mathbf{s}\|_2 \quad (3.7)$$

be the Euclidean distance between wrist and shoulder, then

$$\alpha = \text{acos} \left(\frac{d_5^2 - d_3^2 - d_{[sw']}^2}{-2d_3 d_{[sw']}} \right) \quad (3.8)$$

and finally

$$\mathbf{E}'_0 = \mathbf{S} \cdot \mathcal{T} \left(\begin{bmatrix} s_\alpha d_3 \\ 0 \\ c_\alpha d_3 \end{bmatrix} \right). \quad (3.9)$$

In the next step, we rotate the elbow from its zero position \mathbf{E}'_0 by the desired angle value δ to the final elbow position \mathbf{E}_δ . This is done by using elemental homogeneous transformations in the fashion of

$$\mathbf{E}_\delta = \mathbf{S} \cdot \mathcal{R}(R_{z,\delta}) \cdot \mathcal{R}(R_{[sw']})^{-1} \cdot \mathcal{T}(\mathbf{e}'_0 - \mathbf{s}). \quad (3.10)$$

To determine the first joint value θ_1 , we need to take the prior rotation of $-\varphi$ in (3.3) into account and thus temporarily counter-rotate \mathbf{E}_δ by $\mathcal{R}(R_{z,\varphi})$, resulting in

$$(\mathbf{E}_\delta)_\varphi := \mathcal{R}(R_{z,\varphi}) \cdot \mathbf{E}_\delta. \quad (3.11)$$

Now we can obtain the real elbow position and calculate the actual joint angle by

$$\theta_1 = \text{atan2} \left(\text{arm} \cdot [(\mathbf{E}_\delta)_\varphi]_y, \text{arm} \cdot [(\mathbf{E}_\delta)_\varphi]_x \right). \quad (3.12)$$

Please note that this temporary elbow position $(\mathbf{E}_\delta)_\varphi$ is now no longer needed.

Now, the joint angles θ_2 and θ_4 can be calculated using geometric approaches by

$$\theta_2 = \text{atan2} \left(\text{arm} \cdot \left\| \begin{bmatrix} [\mathbf{E}_\delta]_x \\ [\mathbf{E}_\delta]_y \end{bmatrix} \right\|_2, [\mathbf{E}_\delta]_z - [\mathbf{s}]_z \right) \quad (3.13)$$

and

$$\theta_4 = \text{arm} \cdot \text{elbow} \cdot \left(\pi - \text{acos} \left(\frac{d_{[\mathbf{sw}']}^2 - d_5^2 - d_3^2}{-2d_3d_5} \right) \right). \quad (3.14)$$

To determine θ_3 , we have to solve the forward kinematics to obtain the equation for the joint angle in an analytic way. We identify

$${}^1\mathbf{M}_6 = {}^0\mathbf{M}_1^{-1} \cdot {}^0\mathbf{M}_6 \quad (3.15)$$

using θ_1 and

$${}^1\mathbf{M}_4 = {}^1\mathbf{M}_2 \cdot {}^2\mathbf{M}_3 \cdot {}^3\mathbf{M}_4, \quad (3.16)$$

using θ_2 , θ_4 , and the unknown θ_3 . It is important to understand that based on the underlying decoupling principle, ${}^1\mathbf{M}_6$ and ${}^1\mathbf{M}_4$ have exactly the same translational part, the known wrist position w' . By using (3.16), we can extract the equation for θ_3 as

$${}^1\mathbf{M}_{4(3,4)} = [{}^1\mathbf{M}_4]_z = [w']_z = s_3s_4d_5 \quad (3.17)$$

and thus

$$s_3 = \frac{[\mathbf{W}']_z}{s_4 d_5} \quad (3.18)$$

and

$$c_3 = \sqrt{1 - s_3^2}. \quad (3.19)$$

Now, θ_3 also depends on the elbow position on the circular path between $0 \leq \delta \leq 2\pi$ and shows ambiguities which can only be resolved by a projection into the x - y -plane (Figure 3.4a and Figure 3.4b). To choose the correct position out of the two possible locations for \mathbf{E}_δ , we have to calculate the intersection point t of the tangent from the point s to the ellipse. Once we know this point, we can check if \mathbf{E}_δ is in front of or behind this intersection point. The ellipse has to be parametrized by the semi axes a and b . To this end, using (3.10) we can calculate the elbow position for $\delta = \frac{\pi}{2}$ which yields the maximum width of the ellipse as

$$\bar{a} = \left[\mathbf{E}_{\frac{\pi}{2}} \right]_y. \quad (3.20)$$

Furthermore, the length \bar{b} can be determined geometrically as

$$\bar{b} = [\mathbf{E}'_0]_x - \left[\mathbf{E}_{\frac{\pi}{2}} \right]_x. \quad (3.21)$$

For the case differentiation for θ_3 , only the y -coordinate of the intersection point is needed and can be calculated using the common tangent equation for ellipses as

$$[t]_y = \frac{-\bar{a} \sqrt{-\bar{b}^2 + \left[\mathbf{E}_{\frac{\pi}{2}} \right]_x^2}}{-\left[\mathbf{E}_{\frac{\pi}{2}} \right]_x}. \quad (3.22)$$

We can now calculate θ_3 as

$$\theta_3 = \begin{cases} \text{atan2}(s_3, \text{elbow} \cdot \text{sgn}(\bar{b}) \cdot c_3) & \text{if } |[\mathbf{E}_\delta]_y| > |[t]_y| \vee |[\mathbf{E}_\delta]_x| > \left| -\left[\mathbf{E}_{\frac{\pi}{2}} \right]_x \right| \\ \text{atan2}(s_3, -\text{elbow} \cdot \text{sgn}(\bar{b}) \cdot c_3) & \text{else.} \end{cases} \quad (3.23)$$

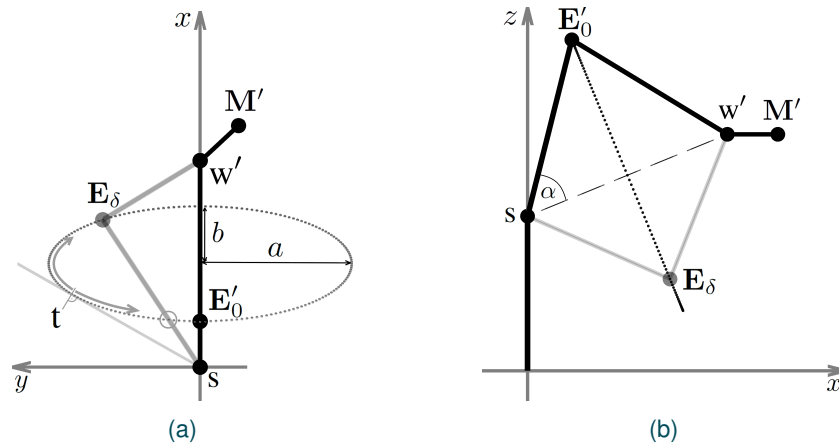


Figure 3.4: Left: Projecting into the x - y -plane: The parametrization of the ellipse and the calculation of tangent t are necessary to resolve the ambiguity for \mathbf{E}_δ . Right: Projecting into the x - z -plane: The ambiguity for \mathbf{E}_δ is not visible.

Besides these two cases, there are two special cases which have to be considered:

First case: The elbow and the wrist are at the same z -coordinate. In this case, the ellipse forms a line ($\bar{b} = 0$) and the angle θ_3 can be calculated directly as

$$\theta_3 = \text{atan2}(s_3, \text{elbow} \cdot \text{sgn}([\mathbf{E}_\delta]_z - [\mathbf{s}]_z) \cdot c_3). \quad (3.24)$$

Second case: The shoulder position is inside or on the elliptical path. In this case, the tangent is not valid and we have to use the equation of case one from (3.23) that holds even under these restrictions.

The last three joints $\theta_{5,6,7}$ are determined by solving the inverse orientation problem to align the end-effector with the target. The angles are calculated analytically by

$${}^4\mathbf{M}_7 = {}^0\mathbf{M}_4^{-1} \cdot \mathbf{M}, \quad (3.25)$$

resulting in

$$\theta_5 = \text{atan2}(\text{wrist} \cdot {}^4\mathbf{M}_{7(2,3)}, \text{wrist} \cdot {}^4\mathbf{M}_{7(1,3)}) \quad (3.26)$$

$$\theta_6 = \text{atan2}\left(\text{wrist} \cdot \sqrt{1 - {}^4\mathbf{M}_{7(3,3)}^2}, {}^4\mathbf{M}_{7(3,3)}\right) \quad (3.27)$$

$$\theta_7 = \text{atan2}(\text{wrist} \cdot {}^4\mathbf{M}_{7(3,2)}, -\text{wrist} \cdot {}^4\mathbf{M}_{7(3,1)}). \quad (3.28)$$

3.4 Evaluation

To evaluate the reliability of the inverse kinematics algorithm, the continuity of joint trajectories over a three dimensional motion path is analyzed. Aiming at direct, consistent control of arm configurations and therefore avoiding non-commanded configuration changes during motion, all joint trajectories have to be continuous. In turn, jump discontinuities during the joint trajectories would indicate spontaneous, potentially harmful joint changes. This is especially problematic in medical applications, where a collision between the patient and the robot must never occur.

However, the demanding part of geometric inverse kinematics approaches is to ensure continuity in the resulting joint values during octant crossings inside the Cartesian workspace. Thus, to evaluate the developed algorithm, a challenging motion-path is defined including eight octant crossings. The path follows an elliptical shape, centered at $(x = 0, y = 0, z = 100)[mm]$, described by

$$\mathbf{P}_i := \begin{bmatrix} & \cos(\alpha_i) \cdot r & & \\ I_3 & \sin(\alpha_i) \cdot r & & \\ & \cos(\alpha_i) \cdot r_z + z_{\text{offset}} & & \\ 0 & & & 1 \end{bmatrix}, i = 0 \dots n, \quad (3.29)$$

and

$$\alpha_i = -\pi + \sum_{j=0}^i \frac{2\pi}{n}, i = 0 \dots n \quad (3.30)$$

where \mathbf{P}_i are the single poses on the path for $n = 100$ steps, the identity matrix \mathbf{I}_3 as exemplary rotation, the two radii $r = 450 \text{ mm}$ and $r_z = 300 \text{ mm}$, and the length

$z_{\text{offset}} = 100 \text{ mm}$. The resulting motion-path is shown in blue in [Figure 3.5](#) and [Figure 3.6](#).

Furthermore, the following parameters are used to control the arm configurations during the motion:

$$\begin{aligned} \text{arm} &= \begin{cases} +1 \Rightarrow \text{lefty} \\ -1 \Rightarrow \text{righty} \end{cases} \\ \text{elbow} &= \begin{cases} +1 \Rightarrow \text{down} \\ -1 \Rightarrow \text{up} \end{cases} \\ \text{wrist} &= \begin{cases} +1 \Rightarrow \text{noflip} \\ -1 \Rightarrow \text{flip} \end{cases} \end{aligned}$$

In [Figure 3.5](#), a model of the *LBR iiwa* is moving along the predefined blue path with $n = 100$ steps. The path and related arm positions are plotted for three exemplary steps within the motion. Furthermore, the joint values are determined and plotted to realize this motion, calculated for three different, exemplary arm configurations and a fixed elbow angle of $\delta = 0^\circ$. To analyze the major evaluation parameter – the joint trajectory continuity – all joint values are plotted as unwrapped phases to eliminate uncritical discontinuities between $-\pi$ and π (and the other way around). The same simulation was repeated for an elbow angle of $\delta = 70^\circ$, as shown in [Figure 3.6](#).

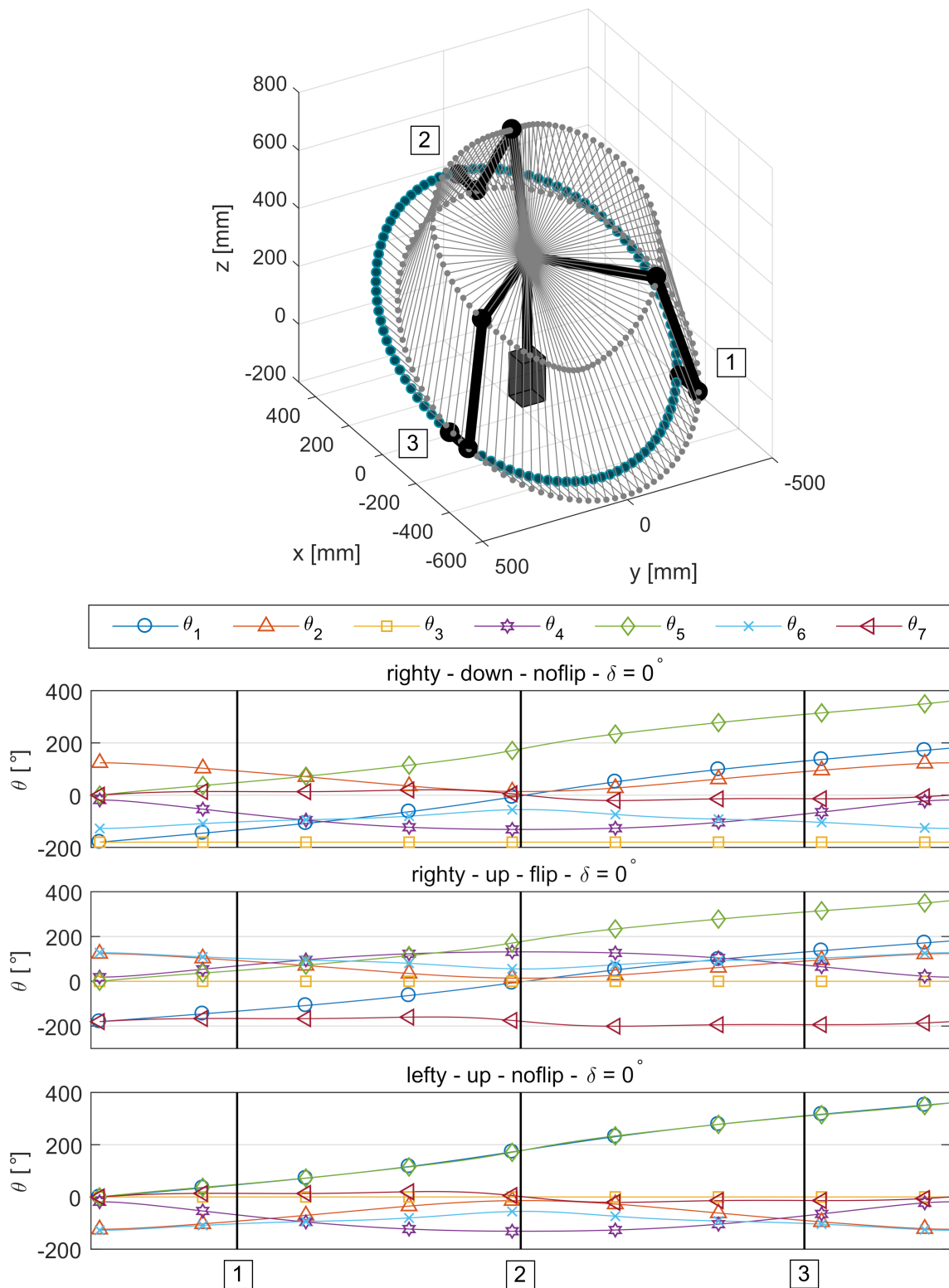


Figure 3.5: First motion simulation of the *LBR iiwa* robot using the proposed algorithm: The end-effector is moving along the generated blue path (top image) based on the calculated joint trajectories (bottom image). Three different arm configurations with an elbow angle of $\delta = 0^\circ$ are simulated. For three exemplary steps, the arm positions are drawn and the related joint values are marked in the bottom figures.

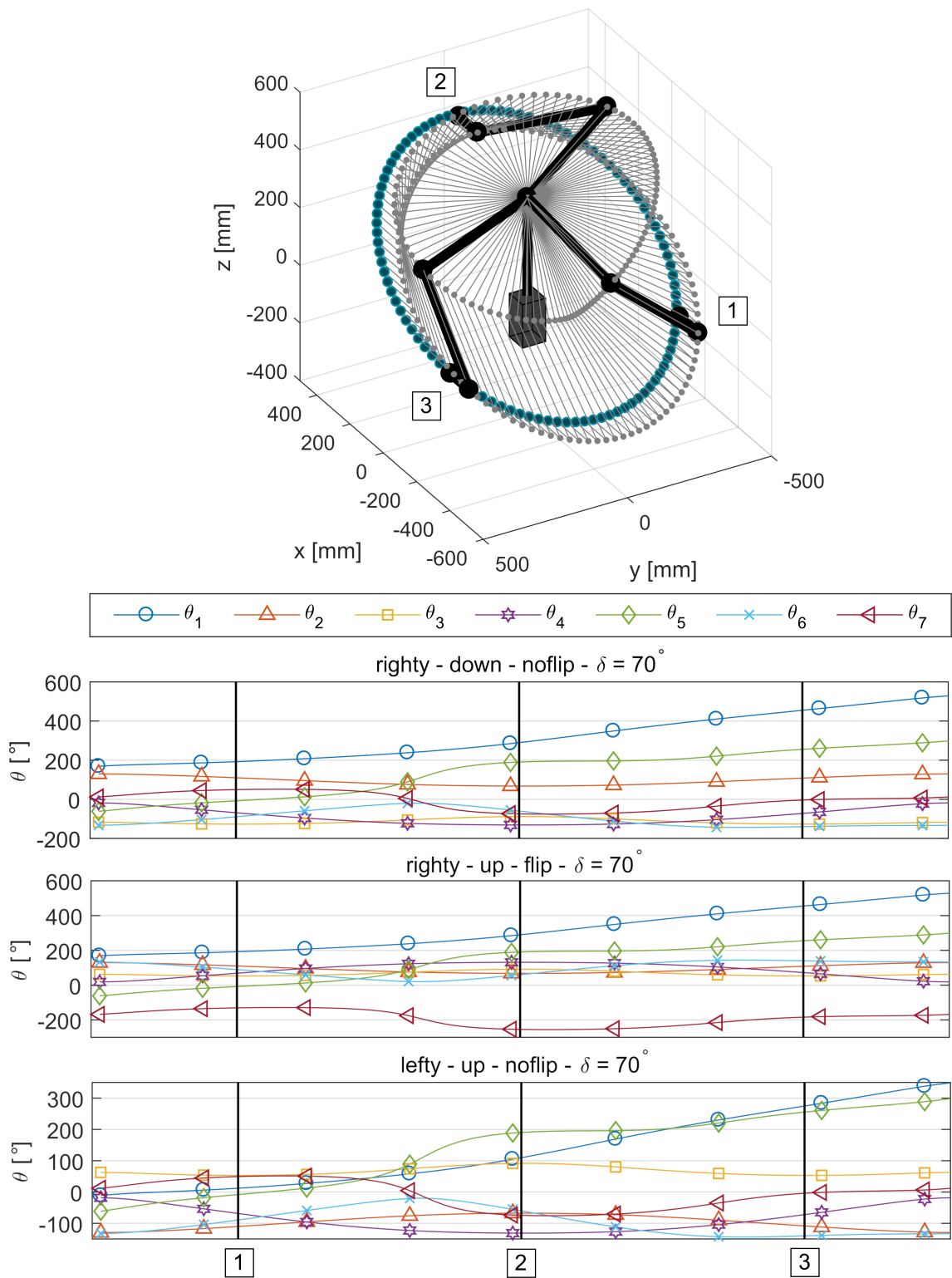


Figure 3.6: Second motion simulation of the *LBR iiwa* robot using the proposed algorithm: The end-effector is moving along the generated blue path (top image) based on the calculated joint trajectories (bottom image). Three different arm configurations with an elbow angle of $\delta = 70^\circ$ are simulated. For three exemplary steps, the arm positions are drawn and the related joint values are marked in the bottom figures.

Finally, to evaluate the joint trajectories when only an elbow motion is commanded, a third simulation was performed. This time, the end-effector of the *LBR iiwa* stays at a fixed pose of

$$\mathbf{P}_{fix} := \begin{bmatrix} & 600 \text{ mm} \\ I_3 & 0 \text{ mm} \\ & 600 \text{ mm} \\ 0 & 1 \end{bmatrix}, \quad (3.31)$$

while moving the elbow in a full circle from $\delta = 0^\circ$ to $\delta = 360^\circ$ with $n = 100$ steps. Again, the motion is simulated for three different arm configurations and the results are shown in [Figure 3.7](#).

The results clearly show an overall continuity and no jump discontinuities during the joint trajectories, and therefore validate consistent arm configurations within the whole motion. Even at challenging octant crossings, the algorithm provides reliable results and no spontaneous configuration changes.

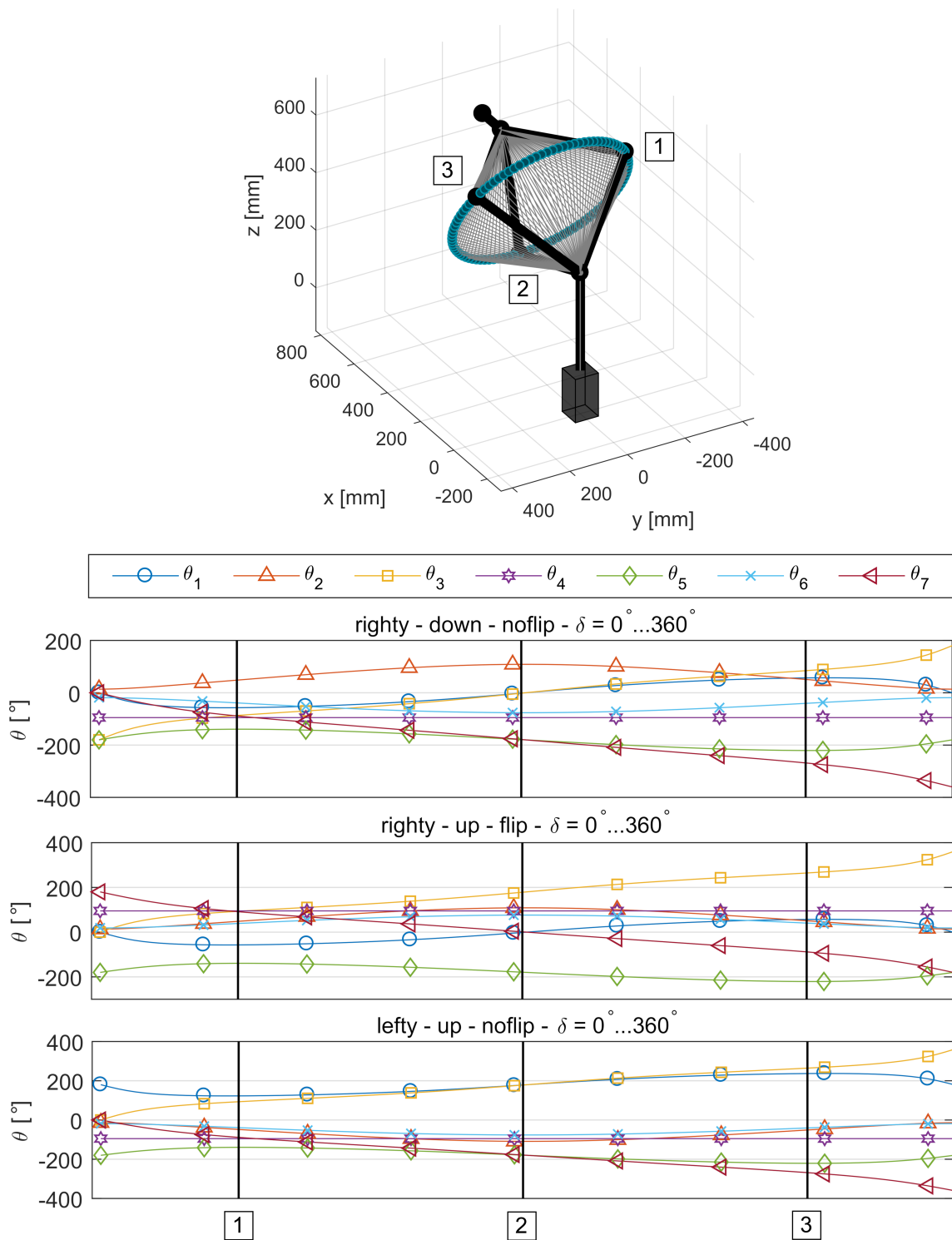


Figure 3.7: Third motion simulation of the *LBR iiwa* robot using the proposed algorithm: The end-effector is staying at a fixed pose, while moving the elbow on a full circle for $\delta = 0^\circ \dots 360^\circ$. For three exemplary steps, the arm positions are drawn and the related joint values are marked in the bottom figures.

3.4.1 Runtime

To analyze the runtime, the algorithm was implemented using three different programming languages and was tested on a desktop PC with an *Intel Core i7 970* (3.20 GHz) CPU, and an *NVIDIA GeForce GTX 980* GPU. The tests were performed on a non-real-time operating system (*Microsoft Windows 8.1*), where the runtimes may vary due to non-consistent scheduling. To overcome this issue, the final runtimes were calculated by means of several identical iterations. Table 3.2 shows the results of the runtime analysis. The results demonstrate short runtimes for all three languages, and therefore overall real-time capability.

Table 3.2: Runtime analysis for three different programming languages

Language	Runtime per iteration [μs]	Metric
MATLAB	750.0	mean of 1000 iterations
C++	1.754	mean of 1000 iterations
CUDA	0.010	mean of $8.6 \cdot 10^{10}$ iterations

3.5 Conclusions

The purpose of this chapter is to answer the first research question:

RQ1: How to exploit all degrees of freedom of kinematically redundant serial manipulators by an inverse kinematics under safety and algorithmic restrictions?

Therefore, a robust inverse kinematics solution in closed form for redundant manipulators with seven degrees of freedom, combining full arm configuration control and elbow self-motion manifold control by an intuitive parameter was developed. The proposed inverse kinematics solution provides two crucial advantages:

First, all theoretical solutions are consistently selectable, using common arm configuration parameters like `arm`, `elbow` and `wrist`. In other words, the algorithm takes all underlying arm configuration manifolds into account and thus provides a complete solution space.

Second, the elbow self-motion manifold parameter is defined in an intuitive way, making it easy to adapt the feature to practical applications. The elbow motion on a circular path is defined over the angle δ that describes the rotation of the elbow around the axis from shoulder to wrist.

The consistent control of arm configurations increases the safety of robot motions because spontaneous configuration changes within a single trajectory are mathematically avoided. The algorithm was evaluated with a model of the lightweight robot *KUKA LBR iiwa*, looking at configuration reliability and continuity for an exemplary, challenging motion path. The results show that commanded arm configurations do not change spontaneously along a trajectory. Even for varying elbow angles, the joint trajectories are continuous and the presented algorithm provides robust solutions. Thus, the inverse kinematics solution helps to improve the safety of robot motion in dynamic environments, where configuration changes are prohibited within a single motion, e.g. to avoid collisions in medical applications with direct human-robot interaction. The transparent strategy of the algorithm makes it easy to replicate and transfer it to practical applications. Furthermore, the lean structure of the algorithm is real-time capable at minimal computational effort and deterministic runtime.

4 Workspace and Dexterity Analysis¹

This chapter addresses the quantitative analysis of workspaces and dexterities of kinematically redundant manipulators. Methods are developed for approximating the reachable and dexterous workspace of serial robots at a high spatial resolution. The results will help to use robotic systems more efficiently, for example by defining tasks in workspace areas with high dexterity. A more efficient exploitation of the workspace would additionally help to avoid joint limits, therefore minimizing configuration changes and finally optimizing path planning. A further purpose of this chapter is the quantitative analysis of the effects of the additional seventh degree of freedom. It will be evaluated how dexterity is affected by an additional seventh joint. To this end, a method is being developed to isolate the effects of the degree of freedom in question. The chapter starts with an introductory section describing the purpose and the motivation for a detailed workspace and dexterity analysis. The section is followed by a discussion of prior related work and an overview of three different kinematic structures used in this work. Afterwards, essential methods for workspace approximation, inverse kinematics, and dexterity calculation are discussed. Major results from the experiments are presented in the subsequent evaluation section, including a detailed dexterity analysis while comparing three different kinematics with six and seven degrees of freedom. The chapter concludes with an assessment of the most relevant findings.

4.1 Introduction

An interesting research question is what quantitative benefits are provided by the additional seventh degree of freedom. Obviously, the dexterity of kinematically redundant manipulators should be higher compared to classical six-jointed kinematics, but does this hold in realistic scenarios? In order to answer these important questions, a detailed dex-

¹Parts of this chapter have been published in [53]

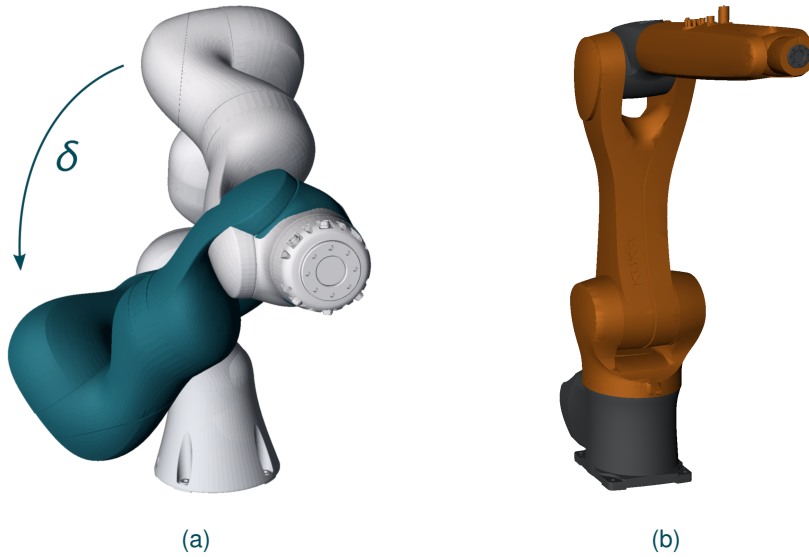


Figure 4.1: The seven-jointed *KUKA LBR iiwa 7* robot in Figure (a) and the six-jointed *KUKA KR 10 R900* robot in Figure (b). The *LBR iiwa* provides an additional seventh degree of freedom, creating a self-motion manifold of the elbow on a circular path, defined by the angle δ .

terity analysis was performed for three different robot structures. The *KUKA LBR iiwa 7 R800* robot represents the seven-jointed kinematic, while the *KUKA KR10 R900* robot was chosen for classic six-jointed manipulators involving comparable workspace dimensions. Both robots are shown in Figure 4.1. The figure also illustrates the additional degree of freedom of the *LBR iiwa*, creating a self-motion manifold of the elbow on a circular path, defined by the angle δ .

However, it is difficult to compare manipulators with different workspaces and different joint ranges. Therefore, to isolate the influence of the elbow motion, the *LBR iiwa* was simulated as a six-jointed robot to allow for a direct comparison to seven-jointed kinematics. Due to this simulation it was possible to directly analyze the effects of the additional degree of freedom.

4.1.1 Related Work

There are prior related works addressing the calculation of so-called dexterity maps and their use in dynamic path planning computations, validation of feasible manipulator poses

or optimal placement of mobile robot bases to realize optimal manipulator alignment for a specific task [108, 109, 119]. However, the work most closely related is the dexterity analysis presented by Porges in [79]. There, a novel way to accelerate on-line path planning computations for kinematically redundant serial robotic arms was presented. Furthermore, he proposed a new method to calculate the workspace of a robot and its inner dexterity. To compute the dexterity maps, multiple strategies based on forward kinematics, inverse kinematics, and hybrid methods have been compared to achieve low computational time. The highest resolution of the generated dexterity volumes was 100 mm in Cartesian space and 2,000 different rotations for each voxel. However, the work does not analyze the dexterous enhancements of the additional seventh joint and its spatial resolution is inadequate. Zacharias *et al.* presented in [118] a new method to analyze the workspace structure of a kinematically redundant manipulator and, furthermore, introduced the use of a reachability index. The work shows a low 300 mm spatial Cartesian resolution and, unfortunately, does not isolate the elbow motion. To this end, there is no existing work analyzing the direct effects of the additional degree of freedom of redundant manipulators like the KUKA LBR iiwa, which finally forms the goal of this chapter.

4.2 Robot Kinematics and Approximation Methods

Dexterity, also known as reachability, basically describes the capability of a robot to reach a point from different directions, whereby the more feasible directions exist, the higher the dexterity is. Consequently, a point with maximum dexterity is reachable from all existing orientations inside \mathbb{R}^3 . To obtain dexterity maps with a high spatial resolution, the workspace has been discretized with 1 mm resolution and the end-effector orientations have been discretized in 8,000 different rotations. In a next step, the number of feasible orientations were determined for every point inside the workspace by using inverse kinematics. Based on this method, a detailed dexterity analysis was performed and three different kinematic structures have been compared. The manipulators have been analyzed and compared using the underlying DH-parameters shown in Table 4.2.

While comparing the kinematic model of the LBR iiwa with the KR10 (and other classic serial six-jointed robots), the relevant difference is the additional third rotational joint of the LBR. By locking (fixing to 0°) this additional joint, it was possible to treat the LBR iiwa as a manipulator with six degrees of freedom. This simulation allows for directly analyzing the effects of the elbow motion.

Table 4.1: The DH parameters of three different kinematic models used for the analysis. The KUKA LBR iiwa represents the kinematically redundant manipulators with seven degrees of freedom. When locking the third joint, the elbow motion is prohibited and the robot is simulated as a six degree of freedom kinematic. In order to compare the modified LBR with other classic six-jointed robots, the KUKA KR10 with comparable workspace dimensions was selected.

i	LBR iiwa						LBR iiwa, six-jointed						KR 10 R900					
	a_i [mm]	α_i [rad]	d_i [mm]	θ_i [rad]	φ_{min} [deg]	φ_{max} [deg]	a_i [mm]	α_i [rad]	d_i [mm]	θ_i [rad]	φ_{min} [deg]	φ_{max} [deg]	a_i [mm]	α_i [rad]	d_i [mm]	θ_i [rad]	φ_{min} [deg]	φ_{max} [deg]
1	0	$-\pi/2$	340	θ_1	-170	+170	0	$-\pi/2$	340	θ_1	-170	+170	25	$-\pi/2$	400	θ_1	-170	+170
2	0	$\pi/2$	0	θ_2	-120	+120	0	$\pi/2$	0	θ_2	-120	+120	455	0	0	θ_2	-190	+45
3	0	$-\pi/2$	400	θ_3	-170	+170	0	$-\pi/2$	400	0	-	-	-35	$-\pi/2$	0	θ_3	-120	+156
4	0	$\pi/2$	0	θ_4	-120	+120	0	$\pi/2$	0	θ_4	-120	+120	0	$\pi/2$	-420	θ_4	-185	+185
5	0	$-\pi/2$	400	θ_5	-170	+170	0	$-\pi/2$	400	θ_5	-170	+170	0	$-\pi/2$	0	θ_5	-120	+120
6	0	$\pi/2$	0	θ_6	-120	+120	0	$\pi/2$	0	θ_6	-120	+120	0	0	-80	θ_6	-350	+350
7	0	0	126	θ_7	-175	+175	0	0	126	θ_7	-175	+175						

4.2.1 Workspace Approximation

The calculation of dexterity is necessary only for points inside the reachable workspace of a robot. Outlying areas must not be considered further and would unnecessarily increase the computational effort. To avoid this, the real workspace volume has to be determined for every manipulator in an initial step. Therefore, the valid inner points of the workspace are identified by using a voxel-grid method, as also described in [79] and [118]. This method fills an empty grid of voxels of the discretized workspace with valid workspace points, based on multiple repeated forward kinematics. Analyzing the underlying DH-parameters, all translational outer bounds of the workspace for the dimensions x , y , and z can be obtained. Please note that the outer bounds are overestimated for sake of algorithmic stability. The bounding box dimensions for workspace approximation are shown in Table 4.2.

Table 4.2: Bounding box dimensions for workspace generation. The outer bounds are overestimated for sake of algorithmic stability.

robot	x [mm]		y [mm]		z [mm]		resolution [mm]	voxels in workspace
	min	max	min	max	min	max		
LBR iiwa	-950	+950	-950	+950	-600	+1300	1	$2.93 \cdot 10^9$
KR 10	-1000	+1000	-1000	+1000	-500	+1400	1	$3.75 \cdot 10^9$

Now, a cuboid can be constructed with the outer dimensions from Table 4.2, bounding the outer shell of the robots' workspaces. In a next step, this cuboid is discretized in $m \times n \times p$ voxels, forming the aforementioned voxel-grid. Figure 4.2 shows the results of this method based on the workspace of the LBR iiwa robot, using a Cartesian resolution of 50 mm. However, for the final results, a spatial resolution of 1 mm was used. The grey voxels show the dimensions of the pre-generated bounding cuboid. In a last step, the workspace can be determined by filling the grid with computed workspace points. To achieve this, a high number of forward kinematics calculations is performed, marking each time the reachable voxels as valid workspace points (blue voxels in Figure 4.2). The forward calculations are performed using random, equally distributed joint values inside the joint limits. Obviously, this method does not necessarily provide a complete solution, rather an approximation. However, to provide a solution with the utmost precision possible, $5 \cdot 10^{10}$ forward calculations have been performed for every robot, resulting in a sufficiently covered workspace.

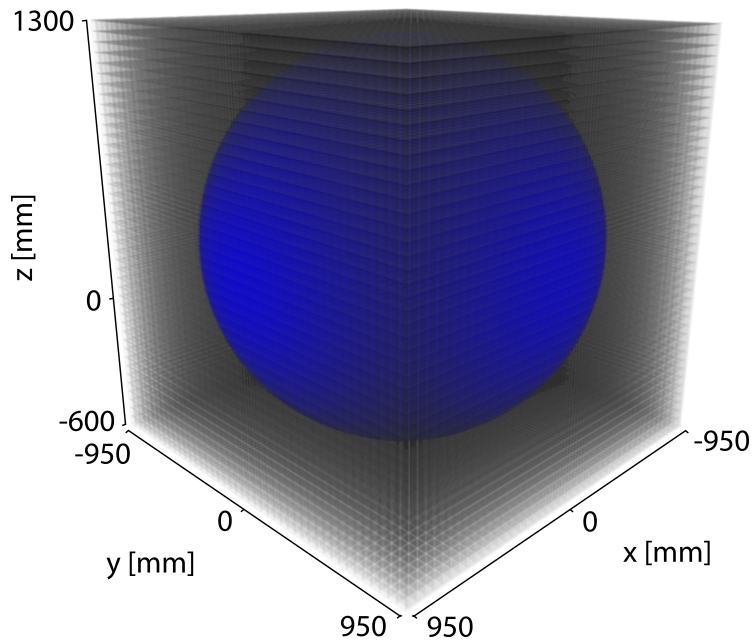


Figure 4.2: The three-dimensional workspace of the *KUKA LBR iiwa* robot in blue and an exemplary grid of voxels with 50 mm resolution to discretize the workspace. For the final results, a spatial resolution of 1 mm was used.

4.2.2 End-Effector Orientations

Dexterity describes the capability of a robot to reach a point from different directions. To this end, we have to analyze from how many directions a robot can reach all single points in its workspace to determine the robots' dexterity. However, we first have to define these orientations for the end-effector. To avoid unwanted weightings of specific directions, we have to ensure that the generated dexterity orientations are distributed equally over all existing rotations inside \mathbb{R}^3 . A set of 8,000 different orientations was generated, using the axis-angle representation (k, θ) . Here, k is a unit vector indicating the direction of an axis of rotation, and the angle θ describes the rotation about the axis. Therefore, we generated 400 equally distributed points on a three-dimensional unit sphere, which is known in the mathematical literature as the *hard-spheres problem* or *Tammes' problem* [18]. One solution is to find an equilibrium configuration of a system of n equally charged particles, by minimizing the electrostatic potential energy. In [86] Saff and Kuijlaars describe a method to find the equilibrium by modeling the problem as a system of n charged particles that repel each other according to Coulomb's law. This method was finally used for calculating the 400 equally distributed points on a unit sphere. The resulting coordinates of

every point on the sphere represents the vector k for the axis-angle representation. Subsequently, every vector k was combined with a set of 20 values for the angle θ , equally distributed between $[0, 2\pi]$, resulting in 8,000 different orientations.

4.2.3 Inverse Kinematics

For the dexterity calculations, two different inverse kinematics algorithms in closed form are used. For the six-axes KR 10 robot, a classical closed-form solution was used as described by Lee [62]. Using this algorithm, all eight solutions for a six-axes manipulator with shoulder offset can be obtained, resulting in a complete solution space. The KUKA LBR iiwa is a seven degree of freedom lightweight robot with zero link offsets. For this kinematically redundant manipulator, the previously developed inverse kinematics solution from Chapter 3 is used, utilizing two crucial advantages:

- First, the algorithm allows for full range elbow self-motion manifold control, defined by an intuitive angle parameter δ that describes the rotation of the elbow around the axis from shoulder to wrist.
- Second, by using common arm configuration parameters like `arm`, `elbow`, and `wrist`, all possible solutions for a single end-effector pose are taken into account and full arm configuration control is achieved.

The simulated 6-jointed KUKA LBR iiwa was controlled with the same inverse kinematics algorithm from Chapter 3, this time with a fixed elbow parameter $\delta = 0^\circ$.

4.2.4 Dexterity Calculation

In order to calculate the robots' dexterity we use the previously defined 8,000 orientations and check the number of orientations with which the manipulator can reach a given point in its workspace. In terms of robot dexterity maps, we have to extend this single observation to all points inside the discretized workspace, yielding a precise dexterity volume for the given manipulator. Furthermore, it is important to take all possible arm configurations into account. For a common six-jointed manipulator, we have to consider eight possible inverse kinematic solutions for a single end-effector pose. In case of the KUKA LBR iiwa, this solution becomes even bigger due to the elbow self-motion manifold angle δ . We expect an elbow motion in the range of $\delta \in [-\pi, \pi]$.

Table 4.3: Number of inverse kinematic calculations used for the dexterity maps.

robot	δ steps	inverse kinematic calculations
KR 10	–	$2.3979 \cdot 10^{14}$
six-jointed LBR iiwa	1	$1.8738 \cdot 10^{14}$
LBR iiwa	2	$3.7475 \cdot 10^{14}$
LBR iiwa	3	$5.6213 \cdot 10^{14}$
LBR iiwa	4	$7.4951 \cdot 10^{14}$
LBR iiwa	5	$9.3688 \cdot 10^{14}$
\vdots	\vdots	\vdots
LBR iiwa	150	$2.8106 \cdot 10^{16}$

One aforementioned benefit of the seventh degree of freedom is to avoid joint limitations by clever positioning of the elbow. Consequently, the more feasible elbow positions exist, the higher the possibility to find valid δ values to avoid joint limitations. We describe this effect as dexterous enhancements based on the additional seventh joint. To directly analyze this effect, the range of the elbow angle was discretized at different number of steps, starting from step one ($\delta = 0^\circ$ to simulate the six-jointed version of the KUKA LBR iiwa) up to 150 steps for a maximum search space. As described before, our developed inverse kinematics algorithm provides full control of the additional degree of freedom.

Due to a high number of combinations of voxels, end-effector orientations, arm configurations, and elbow angles, the dexterity analysis requires an immensely large number of inverse kinematic calculations. Table 4.3 shows the number of calculations done in this experiment, given for each robot. Handling this massive computational effort was only possible by computing simultaneously on to high-end GPUs (2x NVidia GTX 970), using the programming language CUDA. Due to this massive parallelization of the algorithm, the calculations were completed within three months.

4.3 Evaluation

High detailed dexterity maps are calculated for three different serial manipulator models. Based on this data, different analyses are performed, comparing the average dexterity of the robots and, furthermore, isolating the influence of the additional seventh degree of

freedom of the LBR iiwa. To begin with the dexterity map for the six-jointed KR 10 manipulator shown in [Figure 4.3](#), the attached colorbar indicates the dexterity value (number of valid orientations) inside the workspace. The volume consists of one symmetry plane, spanned between the z - and x -axis in blue. [Figure 4.4](#) shows the dexterity volume for the seven-jointed LBR iiwa robot for different cutting planes. The presented dexterity volume is based on 150 δ discretization steps, providing the largest search space for a suitable elbow position, and therefore the highest dexterity. However, the results show that the dexterous workspace of this robot consists of two symmetry planes: The z - x -plane in blue and the z - y -plane in green. Comparing the volume to the simulated six-jointed LBR iiwa ([Figure 4.5](#)), the same symmetries exist, but one can clearly see a significantly lower dexterity compared to the seven-jointed version. This effect becomes even more clear when directly comparing the two volumes voxel-by-voxel. [Figure 4.6](#) shows the differential dexterity volume, where the six-jointed dexterity map was subtracted voxelwise from the seven-jointed volume. Here, the isolated effect of the dexterous enhancements due to the additional degree of freedom becomes visible for the first time. Based on this map, we can analyze in which areas and by how much the additional joint enhances the dexterous workspace of the robot. Obviously, not all areas are affected equally.

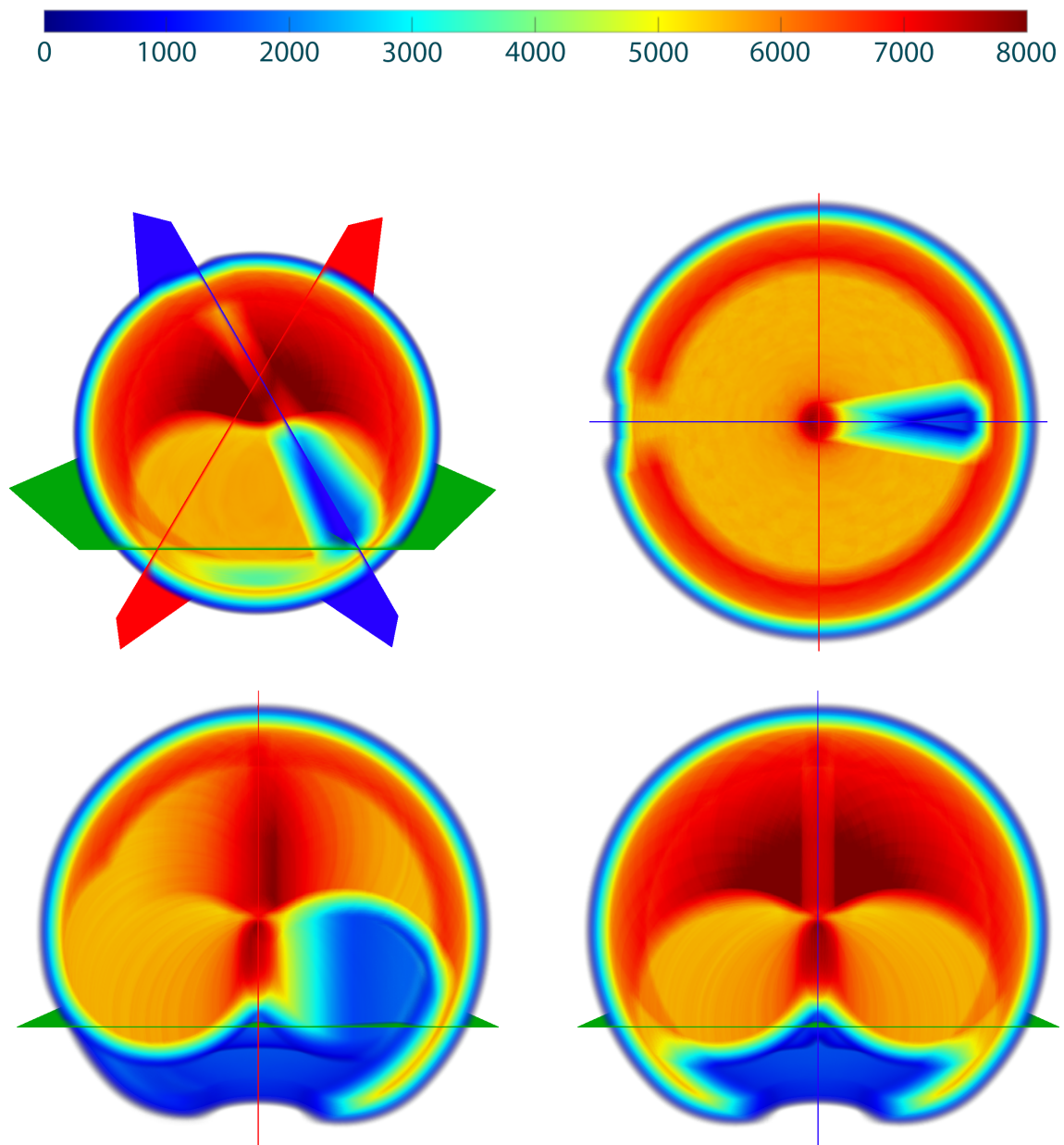


Figure 4.3: Dexterity volume of the KUKA KR 10, shown for different cutting planes. Green: x - y -plane; Blue: z - x -plane; Red: z - y -plane. While blue indicates low dexterity, red areas are at high dexterity.

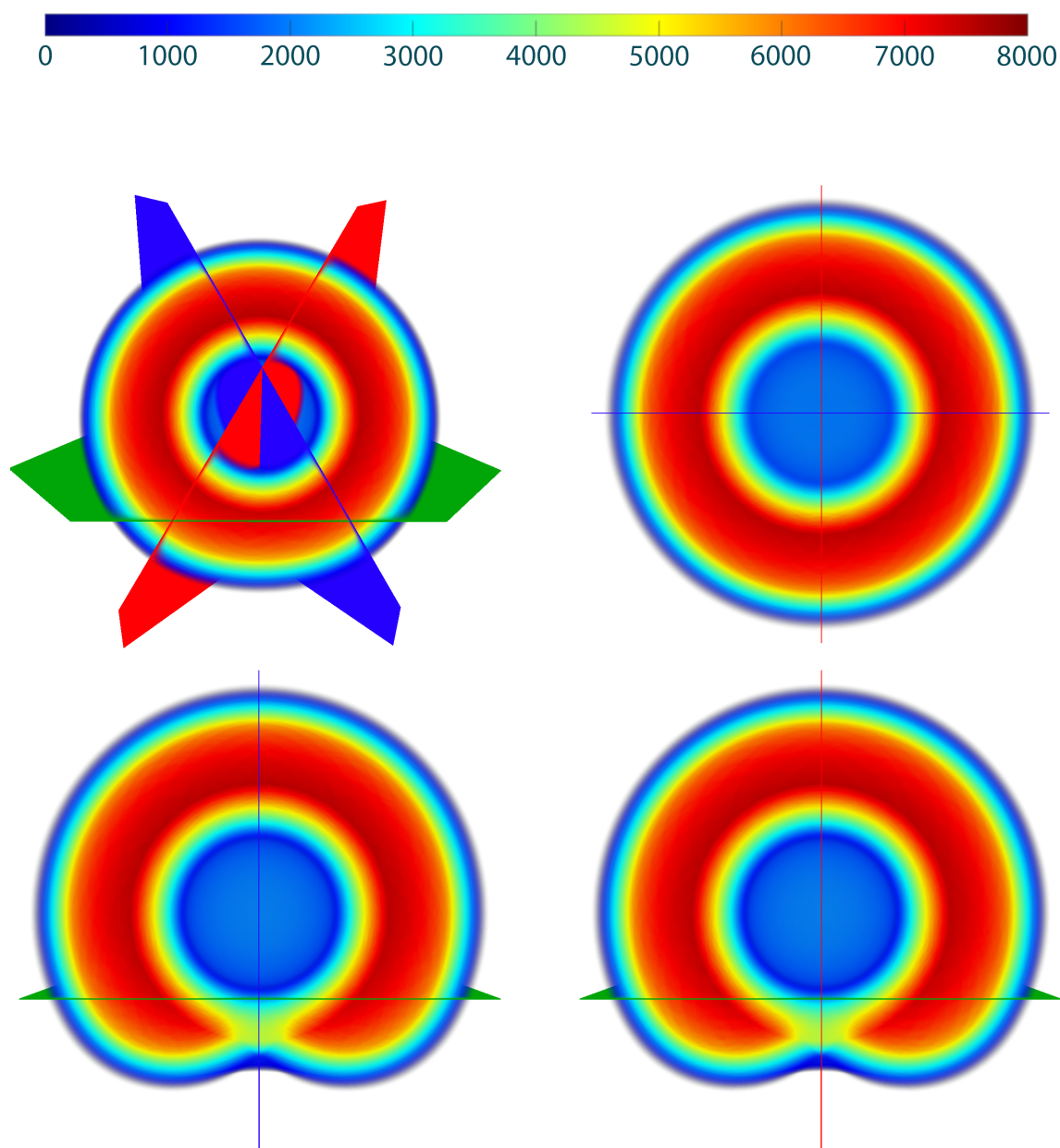


Figure 4.4: Dexterity volume of the KUKA LBR iiwa, shown for different cutting planes. Green: $x-y$ -plane; Blue: $z-x$ -plane; Red: $z-y$ -plane. While blue indicates low dexterity, red areas are at high dexterity.

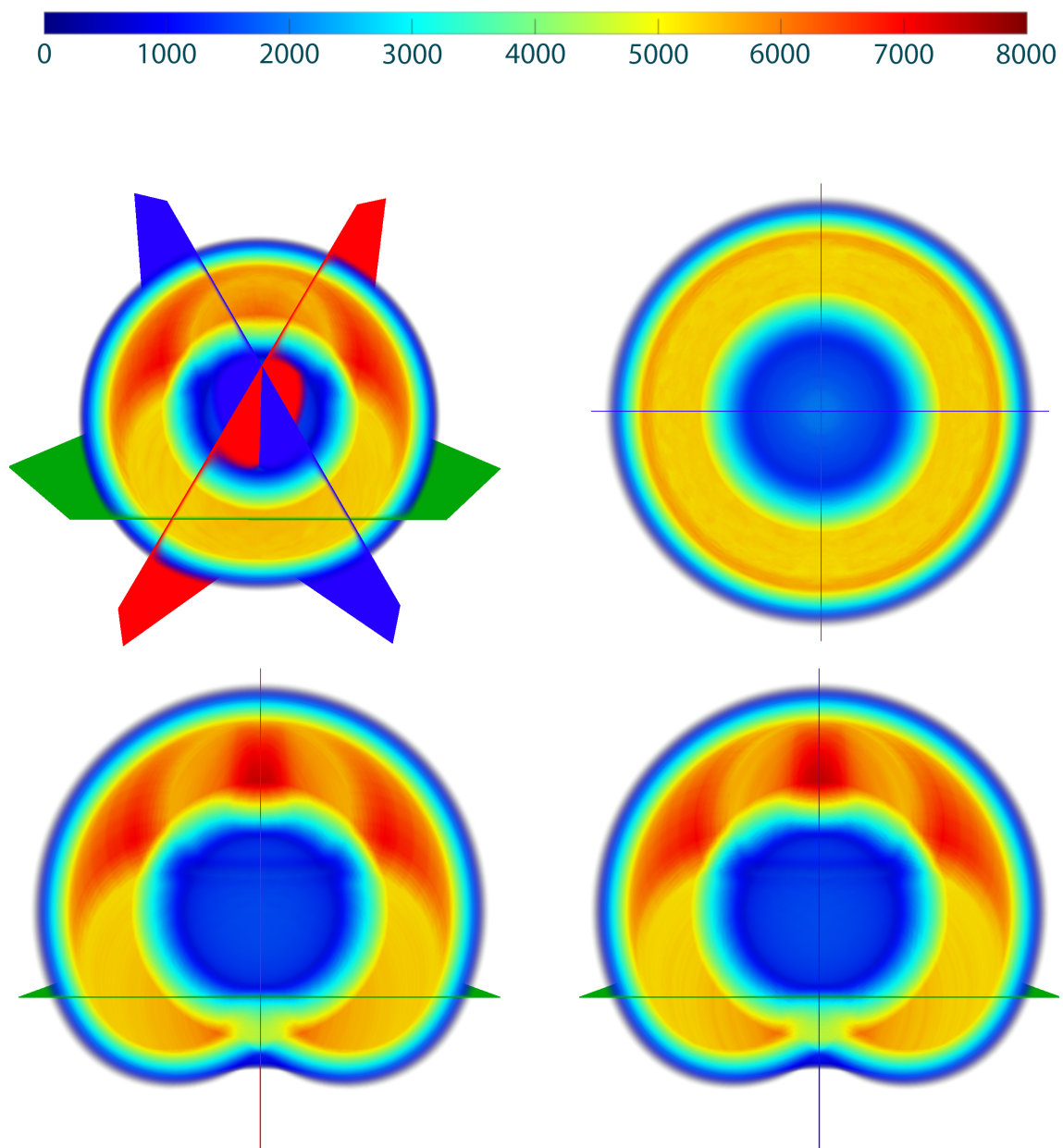


Figure 4.5: Dexterity volume of the simulated six-jointed KUKA LBR iiwa, shown for different cutting planes. Green: x - y -plane; Blue: z - x -plane; Red: z - y -plane. While blue indicates low dexterity, red areas are at high dexterity.

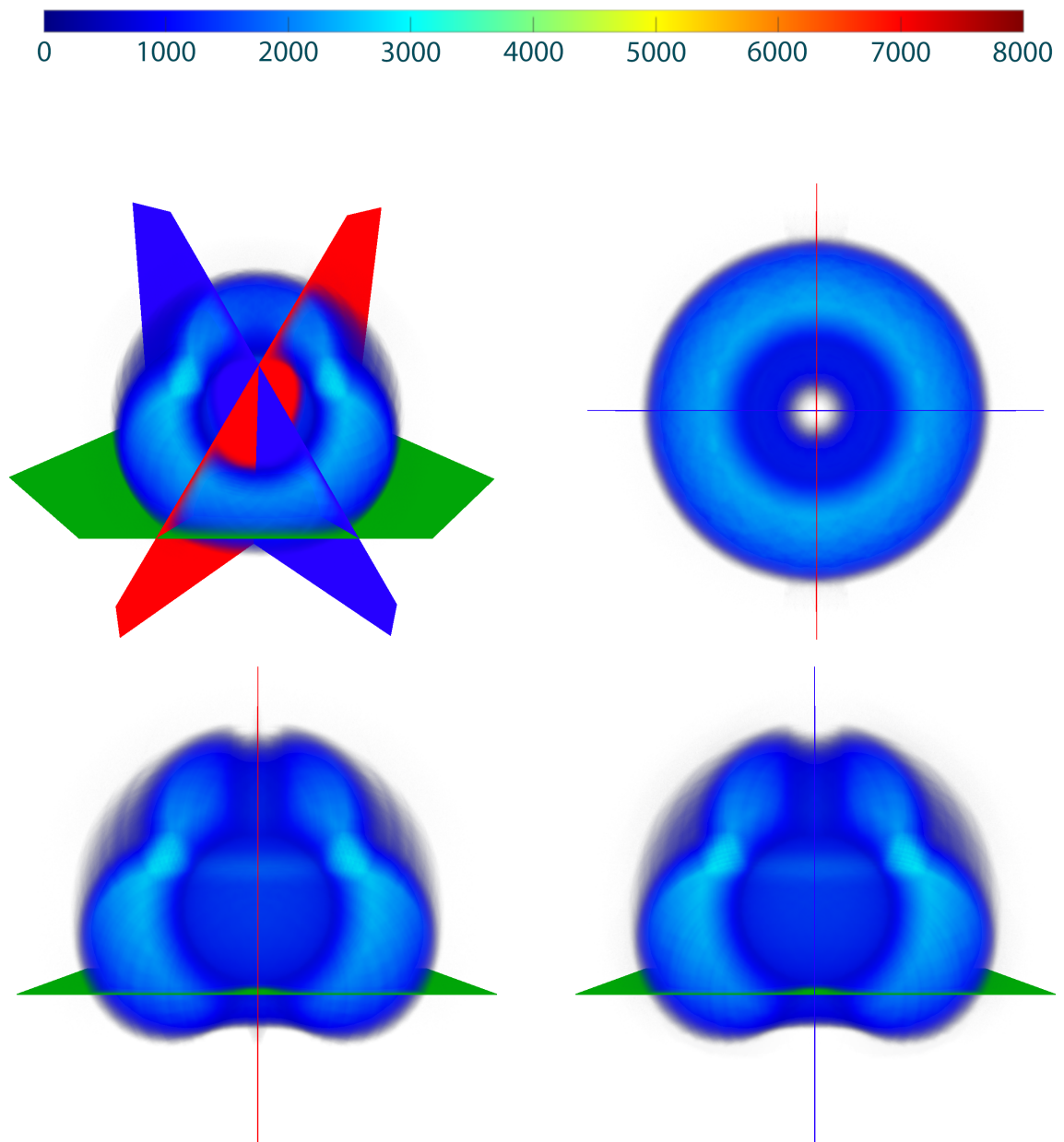


Figure 4.6: Differential dexterity volume of the KUKA LBR iiwa seven-jointed vs. six-jointed, shown for different cutting planes. Green: x - y -plane; Blue: z - x -plane; Red: z - y -plane. Blue indicates low dexterity enhancements.

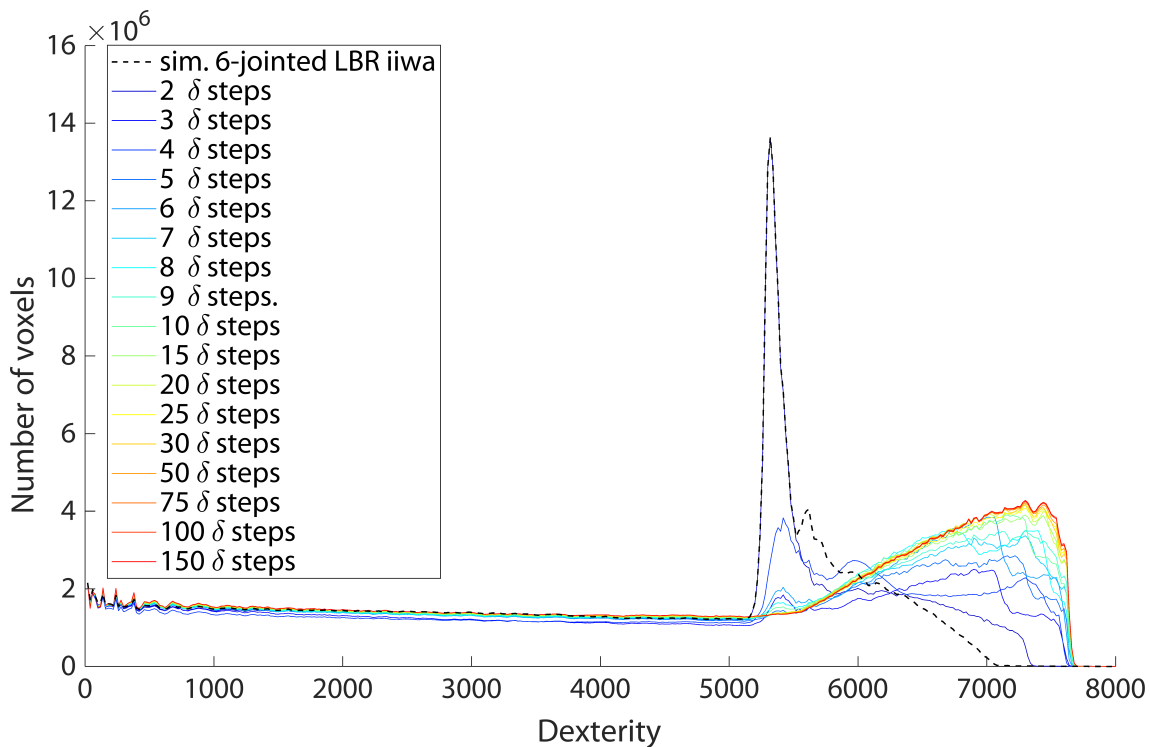


Figure 4.7: Analysis of the dexterous enhancements by the seventh degree of freedom. The dexterity histograms show the number of voxels attaining specific dexterity values. We see a shift to higher dexterity values when the search space for suitable elbow positions gets larger.

In a next step, these dexterity enhancements are quantified to better evaluate the effect. Therefore, a histogram is calculated of both the LBR iiwa as six- and seven-jointed robot. Figure 4.7 shows the histograms for the dexterity maps based on different discretization steps for the elbow angle δ . As stated before, the elbow motion itself corresponds to the additional seventh degree of freedom, which can now directly be validated by this data analysis method. The histograms visualize the number of voxels attaining specific dexterity values. Clearly, the larger the search space of suitable elbow positions is, the higher the dexterity is.

However, based on the histograms the relative dexterity enhancements of the elbow motion are not easy to point out. Thus, the overall percental increase of the average dexterity for different δ steps is shown in Figure 4.8. The results indicate a non-linear increasing dexterity as a function of the number of δ steps. Please note, the discrete distribution of

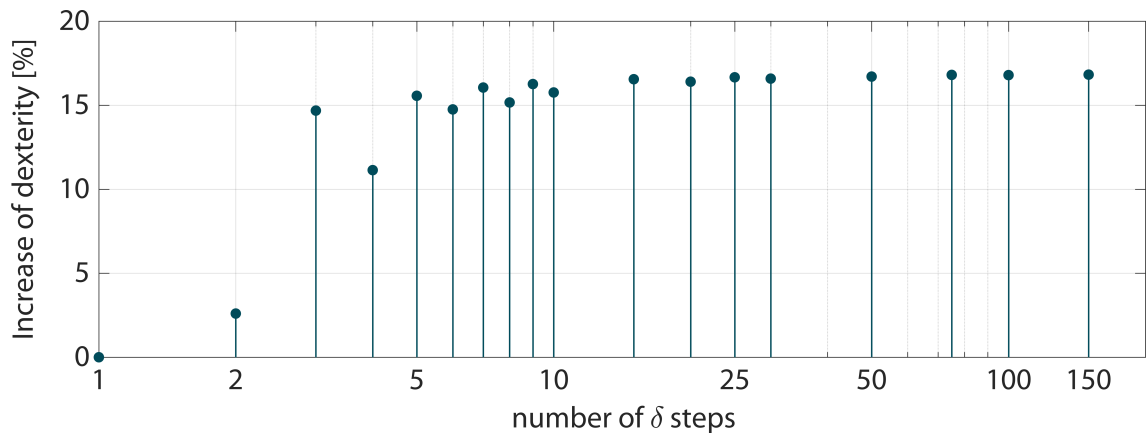


Figure 4.8: Percentual increase of average dexterity for different δ discretizations. The larger the search space of suitable elbow positions is, the higher the dexterity gets.

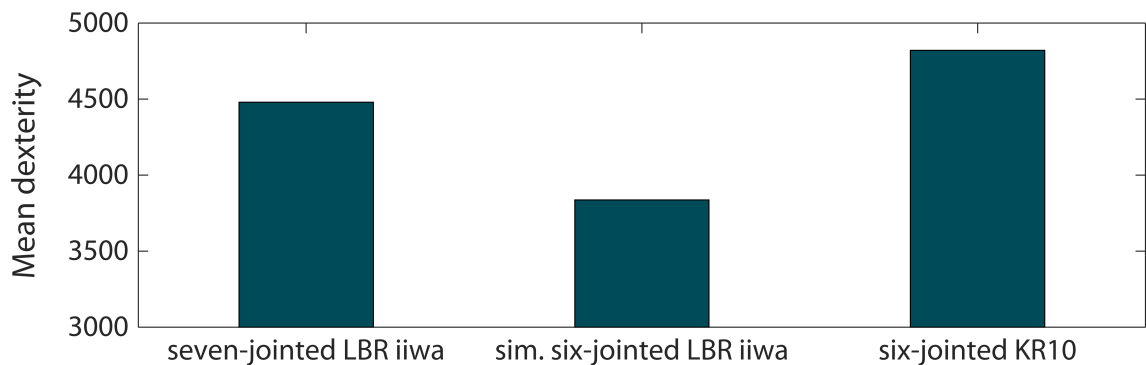


Figure 4.9: Comparison of three different robot models and their mean dexterity. The seven-jointed LBR iiwa shows a 16.81% higher dexterity than the simulated six-jointed model, but also a 7.6% lower dexterity than the six-jointed KR10 robot.

elbow positions between $\delta \in [0, 2\pi]$ in this experiment can lead to unsteady increases in dexterity. Especially in the low ranges of δ steps this effect becomes visible. In other words, a larger search space of suitable elbow positions does not necessarily lead to a higher dexterity, but the probability of finding a suitable elbow position increases.

Finally, the influence of the additional degree of freedom can be determined to be an improvement of 16.8%. In a last step, all three robots are compared based on their average dexterity, as shown in Figure 4.9.

The comparison shows the 16.81% lower dexterity of the simulated six-jointed LBR iiwa (3836.6) compared to the seven-jointed version (4479.2). However, the KR 10 model achieves the highest mean dexterity with 4820.9, which is 7.6% higher than the LBR iiwa, and 25.7% higher than the simulated six-jointed LBR iiwa.

The most plausible explanation for the higher dexterity of the KR 10 are the significantly larger joint ranges. For example, the KR 10 can realize any end-effector orientation in one axis (cf. the angle θ from Section 4.2.2) by using only the last joint ($\theta_6 \in [-350^\circ, +350^\circ]$). Similarly, other KR 10 joints provide a larger range than those of the LBR iiwa. The results indicate a high impact of the joint ranges on the robot's dexterity.

4.4 Conclusions

The purpose of this chapter is to answer the second research question:

RQ2: What effect does the additional seventh degree of freedom have on the dexterous workspace of kinematically redundant manipulators?

To this end, the chapter addresses a detailed dexterity and workspace analysis for three different kinematic models. The KUKA LBR iiwa 7 R800 robot represents the seven-jointed manipulators, while the KUKA KR 10 R900 robot was chosen for common manipulators with six degrees of freedom with comparable workspace dimensions. To isolate the influence of the additional joint, the LBR iiwa was simulated as a six-jointed robot to allow for direct comparison of seven- and six-jointed manipulators. Due to this simulation it was possible to directly analyze the effects of the additional joint. First, a proper method was presented to determine the discretized workspace of a robot using a grid of voxels. Afterwards, 8,000 different end-effector orientations were defined to determine the robots' dexterity. Two different closed-form inverse kinematics algorithms were used for the calculations. The developed inverse kinematics algorithm for the seven-jointed LBR iiwa from Chapter 3 provides full control of the additional degree of freedom through the elbow self-motion manifold angle δ . The results clearly show that the additional joint of the LBR iiwa enhances the dexterity, but in contrast to our expectations by only 16.8%. Compared to the highest average dexterity of the seven-jointed manipulator, the KR 10 outperformed this value by 7.6%, due to significant larger joint ranges.

5 Contact Force Optimization

This chapter addresses methods to improve the accuracy of quasi-static end-effector contact force estimation using the integrated sensory technology of the KUKA LBR iiwa robot. Reliable force-driven robot-interaction requires precise contact force measurements. Often these measurements are severely incorrect and in most manipulation tasks expensive additional force sensors are installed. A more accurate force estimation would increase overall sensitivity of the robot and allow for more complex applications. The chapter proposes a learning approach to train the dependencies between joint torques and end-effector contact forces. Therefore, an extensive training data set is acquired for training a deep feed-forward net. The developed calibration method is evaluated looking at general accuracy improvements, stability near singularities and generalization performance. The chapter starts with an introductory section pointing out problems of inaccurate estimations of end-effector forces, and is thus motivating a new accurate calibration method to overcome these issues. After discussing prior related works, major contributions of this chapter are described. The following section discusses essential technical and theoretical methods used for the experiments and describes the data acquisition process. In a detailed evaluation the performance of the developed calibration method is compared to KUKA's proprietary force estimation model. The chapter concludes with a discussion of the most relevant findings.

5.1 Introduction

Integrated joint torque sensors of the KUKA LBR iiwa provide crucial functionalities for safe physical human-robot-interactions. Based on the joint torques τ , measured in each of the robot's axes, the corresponding contact forces f applied to the robot's end-effector can be calculated. Obviously, higher accuracy of the determined forces leads to an enhanced overall sensitivity of the robot and therefore allows for more complex applications.

On the other hand, errors in the computational model can lead to dangerous and potentially harmful situations during force driven operations. For industrial manipulators, no individual but rather general mechanical models are used for the control algorithms. Manufacturing inaccuracies and individual characteristics of bearings, sensors and actuators are not taken into account. As a result, this leads to further sources of error in contact force determination and can have a strong influence on the resulting accuracy.

In a dynamic scenario with high accelerations and velocities, mass induced inertial, centrifugal and Coriolis forces at the manipulator must be considered in the control algorithms [45]. In medical applications, motions of robotic arms are often restricted due to limited workspace and high safety requirements. In this case, robotic processes can be considered as quasi-static, because mass-induced impacts are small. For a static scenario, the determination of end-effector contact forces is greatly simplified. Finally, only the inverted geometric *Jacobian Matrix* is required to express the relationship between forces and joint torques [92]. There are multiple sources of error for inaccurate contact force determination. Numerical instabilities are not only a common problem in position control, but can also lead to high inaccuracies in force calculations. Whenever computational models use the *Jacobian Matrix* for motion and force control, the matrix loses full rank at singularities, leading to a limited model accuracy [69].

Furthermore, robots are usually not used in their full performance range, but operate in a very small range of forces due to the rather restricted type of specialized applications. Therefore, an optimized calibration within this specific range of forces could lead to very precise results, minimized errors and drastically increased overall system accuracy. One approach to increase the accuracy of the contact force calculation is machine learning. Artificial neural networks show impressive capabilities in solving direct [85] and inverse kinematics [82] as well as in dynamic control of redundant manipulators [59, 64, 65].

Earlier research has addressed the calibration of force/torque sensors for serial robotic manipulators, mostly focusing on optimizing linear relations between joint torques and dynamic parameters known as the Inverse Dynamic Identification Model (IDIM) [21, 36, 46]. Learning approaches to optimize the IDIM were presented in [43, 60, 77]. In [67], neural networks are used to map measured signals and resulting forces, to calibrate an exter-

nal force/torque sensor mounted to an end-effector. Although the authors show accurate results and provide time-saving routines, unfortunately, instabilities near singularities still exist or additional external sensors are required. Smith et al. showed interesting results by using a neural-network-based approach to determine contact forces for haptic devices [99]. Unfortunately, the authors have to use biased ground truth data and, therefore, it remains unclear how this affects the calibration accuracy. Furthermore, it has been shown that neural networks generally struggle with ambiguous data [38]. Although the previous works use simple planar kinematic structures not exceeding three degrees of freedom, there is no evidence yet that the proposed models still hold for kinematically redundant robots with seven or more degrees of freedom.

The use of neural networks for calibrating the robotic system provides several advantages. The system is capable of learning unique mechanical characteristics of the manipulator and the robot can therefore be calibrated in a highly specialized way. Furthermore, critical arm positions and singularities can be directly learned from the network as training points. Using a sufficient amount of diverse input data, these points can be uniquely identified and integrated into the model, resulting in a more robust calibration.

The aim of this chapter is to improve the accuracy of static end-effector contact force estimation using the robot's integrated sensory technology. Therefore, three major goals are addressed:

Contact Force Generation

Standard approaches for acquiring ground truth data for contact forces by using F/T (force/torque) sensors or additional collaborating manipulators require expensive hardware and introduce various sources of errors due to mechanical issues. This chapter presents an alternative method to generate contact forces by mounting calibration loads to the end-effector. By using gravity in combination with different robot base orientations, a homogeneous representation of contact forces in all directions is obtained. An extensive database consisting of 330,000 randomized data points was created.

KUKA LBR iiwa Accuracy Analysis

The database allows for a detailed analysis of the robot's integrated sensors and the accuracy of the **proprietary force estimation model (PFEM)**.

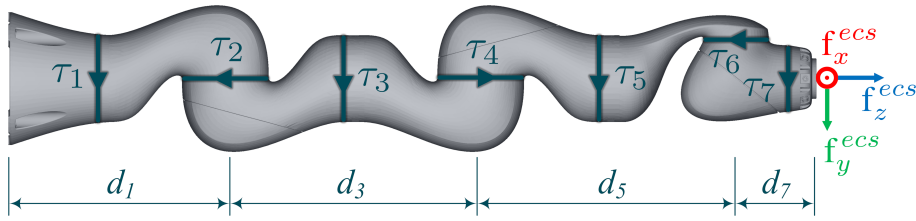


Figure 5.1: The end-effector contact forces f_x^{ecs} , f_y^{ecs} and f_z^{ecs} are related to the measured joint torques τ_1, \dots, τ_7 .

Estimation Error Reduction

A learning approach to train a deep feed-forward **artificial neural network (ANN)** to estimate the contact forces applied to the end-effector is proposed. It is based on the measured joint torques and information about the robot's current pose. Estimation error and robustness close to singular joint configurations will be improved.

In this first proof-of-concept study, the work focuses on improving contact forces. Generating training data for the end-effector torques would require much more complex calibration weights with a high demand for accuracy. However, there is currently no indication that the calibration method presented in this chapter could not be transferred to the contact torques.

5.2 Hardware and Calibration Methods

The *KUKA LBR iiwa 7 R800* robot was used for the experiments and data acquisition. However, the methods described in this chapter are applicable to general serial kinematics equipped with joint torque sensors.

5.2.1 Robotic Hardware

The kinematic structure, related joint torques and end-effector contact forces of the *KUKA LBR iiwa* are shown in Figure 5.1. Relevant specifications of the integrated joint torque sensors related to the DH-parameters of the robotic system used in this work are shown in Table 5.1. Each θ_i represents the i^{th} joint position variable, which is limited by its mechanical lower-limit $-q_i^{\text{max}}$ and upper-limit $+q_i^{\text{max}}$, with $i \in \{1, \dots, 7\}$. Two dedicated,

Table 5.1: Underlying DH-Parameters of the KUKA LBR iiwa 7 R800 robot and specifications of the integrated joint torque sensors.

i	a_i [mm]	α_i	d_i [mm]	θ_i	q_i^{max} [°]	τ_i^{max} [Nm]	$d\tau_i$ [10 ⁻³ Nm]
1	0	$-\pi/2$	340	θ_1	± 170	± 176	1.344
2	0	$+\pi/2$	0	θ_2	± 120	± 176	1.344
3	0	$-\pi/2$	400	θ_3	± 170	± 110	0.880
4	0	$+\pi/2$	0	θ_4	± 120	± 110	0.880
5	0	$-\pi/2$	400	θ_5	± 170	± 110	0.880
6	0	$+\pi/2$	0	θ_6	± 120	± 40	0.334
7	0	0	126	θ_7	± 175	± 40	0.334

redundant operating torque sensors are integrated in each axis with mechanical specifications shown in Table 5.1. Each sensor measures joint torques τ_i in the range between its mechanical limits $[-\tau_i^{max}, +\tau_i^{max}]$ with a resolution of $d\tau_i$ and an axis-specific relative error of 2%, $i \in \{1, \dots, 7\}$.

5.2.2 Training Data

The aim is to precisely determine corresponding contact forces at the end-effector from given joint positions and torque data. Contact forces at the end-effector can result from various impacts: by external forces such as pushing or pulling by hand, or while the robot actively pushes against an object with its end-effector. From a mechanical point of view, the resulting forces are the same for both cases and can be measured via the joint torques. In this study, the forces are simulated by mounting different masses on the end-effector. Gravity produces an equivalent force, which pulls the mass towards the ground. Obviously, this static force will always point in the same direction in the world coordinate system, but for varying robot positions it will create different contact forces in end-effector coordinates. With only one direction in world coordinates not all contact forces can be represented. Hence, six different base rotations were used to solve this problem. The robot was therefore mounted in different orientations, shown in Figure 5.2, resulting in a more homogeneous representation of the contact forces in all directions. To determine an exact ground truth for the end-effector forces, ten specially manufactured weights in the range of 0-2 kg (0-20 N) in 0.2 kg (2 N) steps were used for the measurements. The

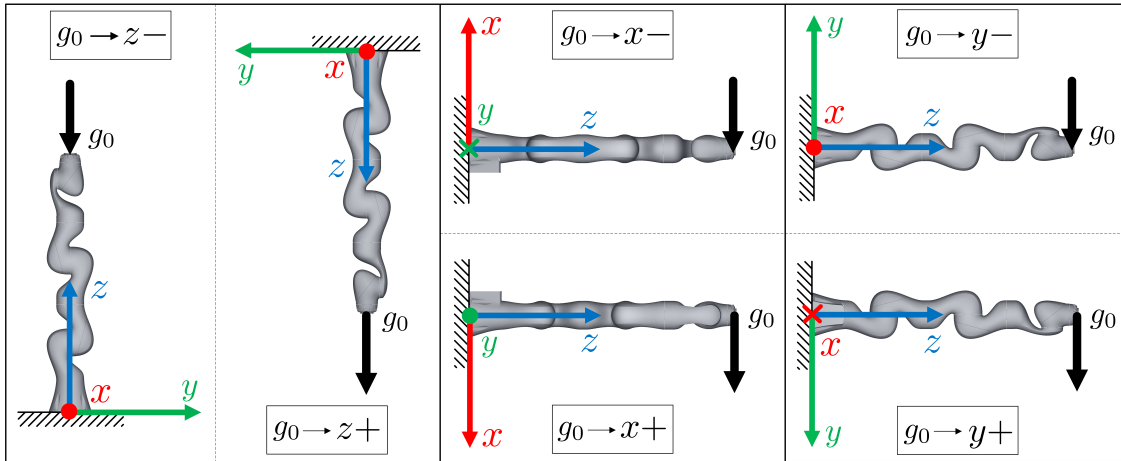


Figure 5.2: The robot was mounted in six different base orientations to record the training data with a homogeneous distribution of gravity vector directions.

precise weights are $m_j \in \{0, 202, 391, 616, 818, 995, 1199, 1401, 1603, 1808, 2002\} \text{ g} \pm 1 \text{ g}$, for $j \in \{1 \dots 11\}$. For each mass, 5,000 measurements were acquired, resulting in a total of randomized 330,000 data points with varying base rotations and calibration weights. The weights are built from symmetric metallic discs which can be stacked to result in ten different weight combinations. The exact contact forces for ground truth can be determined as follows:

For a serial manipulator we can compute the position and orientation of the end-effector by coordinate transformations from the base along the joints. We use the DH-convention to determine the transformation from joint $i - 1$ to i by

$${}^{i-1}\mathbf{T}_i = \mathcal{T}_{z_{i-1}, d_i} \mathcal{R}_{z_{i-1}, \theta_i} \mathcal{T}_{x_i, a_i} \mathcal{R}_{x_i, \alpha_i} \quad (5.1)$$

where $\mathcal{T}_{z_{i-1}, d_i}$ and \mathcal{T}_{x_i, a_i} denote homogeneous translations along the axis z_{i-1} by d_i and along x_i by a_i , respectively. $\mathcal{R}_{z_{i-1}, \theta_i}$ and $\mathcal{R}_{x_i, \alpha_i}$ are homogeneous rotations around the axis z_{i-1} by the angle θ_i and around x_i by the angle α_i , respectively. The variables a_i , α_i , d_i and θ_i are the robot-specific DH-parameter given in Table 5.1. We can now calculate the transformation from the robot's base up to the end-effector along the joints by

$${}^0\mathbf{M}_7 = {}^0\mathbf{T}_1 {}^1\mathbf{T}_2 {}^2\mathbf{T}_3 \dots {}^6\mathbf{T}_7. \quad (5.2)$$

Table 5.2: To calculate the six base transformations ${}^{-1}\mathbf{M}_{0,i}$ using (5.3) as shown in Figure 5.2, different combinations of Tait-Bryan angles are used.

i	Base rotation	Ψ [°]	Θ [°]	Φ [°]
1	$g_0 \rightarrow z-$	0	0	0
2	$g_0 \rightarrow z+$	0	0	180
3	$g_0 \rightarrow x-$	180	-90	0
4	$g_0 \rightarrow x+$	0	90	0
5	$g_0 \rightarrow y-$	90	0	90
6	$g_0 \rightarrow y+$	-90	0	-90

Using (5.2) we can also compute the transformation ${}^0\mathbf{M}_i$, $i \in \{1, \dots, 7\}$ to every joint i .

Let ${}^{-1}\mathbf{M}_{0,i}$ be the robot base transformation defined by

$${}^{-1}\mathbf{M}_{0,i} = \mathcal{R}_{z,\Psi_i} \mathcal{R}_{y,\Theta_i} \mathcal{R}_{x,\Phi_i} \quad (5.3)$$

using homogeneous rotations with six different combinations of the three Tait-Bryan angles Ψ_i , Θ_i and Φ_i , $i \in \{1..6\}$, denoted in Table 5.2.

Let $g_0 = 9.806 \frac{m}{s^2}$ be the acceleration due to gravity, then

$$\vec{g}_0 = \begin{pmatrix} 0 \\ 0 \\ -g_0 \end{pmatrix} \quad (5.4)$$

is the resulting gravity vector in the world coordinate frame pointing towards the ground. The forces induced by the calibration masses m_j acting in the robot base coordinate system (bcs) can be described by

$$\mathbf{t}_{i,j}^{bcs} = {}^{-1}\mathbf{M}_{0,i} \begin{pmatrix} \vec{g}_0 \cdot m_j \\ 1 \end{pmatrix} \quad (5.5)$$

for $i \in \{1..6\}$ and $j \in \{1..11\}$. Now let 0R_7 be the end-effector rotation as upper 3×3 rotational matrix based on the homogeneous transformation from (5.2), i.e.

$${}^0\mathbf{M}_7 = \begin{bmatrix} {}^0R_7 & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (5.6)$$

with vector t as translational part. We denote

$${}^0\mathcal{R}_7 = \begin{bmatrix} {}^0R_7 & 0 \\ 0 & 1 \end{bmatrix} \quad (5.7)$$

as the end-effector rotation as homogeneous transformation with zero translation in the robot base coordinate frame. Then the forces acting in the end-effector coordinate system (*ecs*) can be described by

$$\mathbf{f}_{i,j,n}^{ecs} = {}^0\mathcal{R}_{7n}^{-1} \cdot \mathbf{f}_{i,j}^{bes} \quad (5.8)$$

for $i \in \{1\dots6\}$ base rotations, $j \in \{1\dots11\}$ calibration masses and $n \in \{1\dots5000\}$ measurements.

For every force $\mathbf{f}_{i,j,n}^{ecs}$, the corresponding joint positions $q_{i,j,n}$ and joint torques $\tau_{i,j,n}$ are measured using the robot control software *KUKA Sunrise* with the *Fast Research Interface (FRI) v1.13* [90]. However, the measured torques include the self-weight of the robot links. The resulting torques depend on the joint positions and, more importantly, on the base orientation and the inherent direction of gravity. To decouple the data, only the torques generated by the specific calibration weights are used for training the neural network. To identify these isolated torques, two discrete measurements were performed at each of the 330,000 randomized positions. First, the joint torques without mounted calibration weight $\tau_0(m_0)$ were recorded. In a second measurement, the torques with mounted calibration weight $\tau_j(m_j)$ were recorded at the very same positions. The torques resulting only from the additional weight can therefore be determined by

$$\tau_j^\Delta = \tau_j - \tau_0 \quad (5.9)$$

for $j \in \{1\dots11\}$ calibration weights.

5.2.3 Proprietary Force Estimation Model

A detailed description of the proprietary robot control architecture is given in [3, 28]. For the sake of clarity, the following section discusses essential force control formulations and points out how the proprietary model calculates end-effector contact forces. In the presence of external joint torques due to contact forces, the following dynamics model of

robots with flexible joints is considered:

$$\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{g}(q) = \tau_J + \tau_{\text{ext}}. \quad (5.10)$$

$\mathbf{M}(q) \in \mathbb{R}^{n \times n}$ is the symmetric and positive-definite inertia matrix, $\mathbf{C}(q, \dot{q})\dot{q} \in \mathbb{R}^{n \times n}$ is the centrifugal and Coriolis matrix, and $\mathbf{g}(q) \in \mathbb{R}^n$ is the gravity vector. We denote $\tau_{\text{ext}} \in \mathbb{R}^n$ as the external joint torque and $\tau_J \in \mathbb{R}^n$ is the elastic torque transmitted through the joints as

$$\tau_J = \mathbf{K}_J(\theta - q) \quad (5.11)$$

where $\mathbf{K}_J \in \mathbb{R}^{n \times n}$ is the diagonal and positive definite joint stiffness matrix and $\theta \in \mathbb{R}^n$ are the motor positions. Let $\mathbf{F}^{ecs} \in \mathbb{R}^6$ be the wrench at the end-effector as

$$\mathbf{F}^{ecs} = \begin{bmatrix} \mathbf{f}^{ecs} \\ \mathbf{m}^{ecs} \end{bmatrix} \quad (5.12)$$

consisting of contact forces $\mathbf{f}^{ecs} \in \mathbb{R}^3$ and moments $\mathbf{m}^{ecs} \in \mathbb{R}^3$. By using the transposed geometric Jacobian matrix $\mathbf{J}(q)$ of the end-effector, we can calculate the external joint torques as

$$\tau_{\text{ext}} = \mathbf{J}^T(q)\mathbf{F}^{ecs}. \quad (5.13)$$

Finally, by including (5.13) in the general robot dynamics from (5.10), we can calculate the end-effector wrench as

$$\mathbf{F}^{ecs} = \mathbf{J}^T(q)^{-1}(\mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{g}(q) - \tau_J). \quad (5.14)$$

5.2.4 Neural Network Model

The aim is the estimation of the force \mathbf{f}^{ecs} applied to the end-effector based on the measured joint torques τ^Δ , the joint positions q_i and additional robot pose informations, combined in the input vector I . A detailed composition of the input vector I is given later in this section. This estimation is performed by training a regression model on the acquired training data. As the relationship between \mathbf{f}^{ecs} and τ^Δ is unknown close to singularities, our learning algorithm must be capable of approximating arbitrary functions. Additionally, a multidimensional prediction is needed to predict \mathbf{f}^{ecs} . Support Vector Machines are not inherently capable of multidimensional predictions [100] and Gaussian Processes show

an impractical increase in computing time for training on a set of our size [81, 83]. Both methods have the disadvantage that a suitable kernel must be chosen manually [35]. During the last years, deep neural networks showed good performance on approximating unknown functions, while scaling to big data is possible [61].

Therefore, a dense feed-forward neural network with five hidden layers is used. All of them were fully connected and consisted of 200 neurons each. The neurons are modeled with rectified linear unit (ReLU) activation functions [74]. The last layer consists of three neurons to predict the force vector \mathbf{f}^{ecs} with linear activation. The activation functions g_{ReLU} and g_{linear} are defined as

$$g_{ReLU}(x) = \max(0, x) \quad (5.15)$$

and

$$g_{linear}(x) = x \quad (5.16)$$

The force vector is predicted by propagating the input through the layers of the network [84]. The output o_i^k of the i -th neuron in the k -th hidden layer is given recursively by

$$o_i^k = g_{ReLU} \left(b_i^k + \sum_{j=1}^{r_{k-1}} w_{j,i}^k o_j^{k-1} \right), i = 1, \dots, r_k, \quad (5.17)$$

where $w_{j,i}^k$ is the weight of the i -th neuron in layer k for the incoming node j , b_i^k is this neuron's bias, r_k is the number of neurons in layer k and $o_i^0 = I_i$, the i -th element of the input vector I . Hence, the force vector estimation can be calculated as the output of the final sixth layer in dependency of the output of the last hidden layer:

$$\begin{aligned} \mathbf{f}_x^{ecs} &= g_{linear} \left(b_1^6 + \sum_{j=1}^{r_5} w_{j,1}^6 o_j^5 \right) \\ \mathbf{f}_y^{ecs} &= g_{linear} \left(b_2^6 + \sum_{j=1}^{r_5} w_{j,2}^6 o_j^5 \right) \\ \mathbf{f}_z^{ecs} &= g_{linear} \left(b_3^6 + \sum_{j=1}^{r_5} w_{j,3}^6 o_j^5 \right) \end{aligned} \quad (5.18)$$

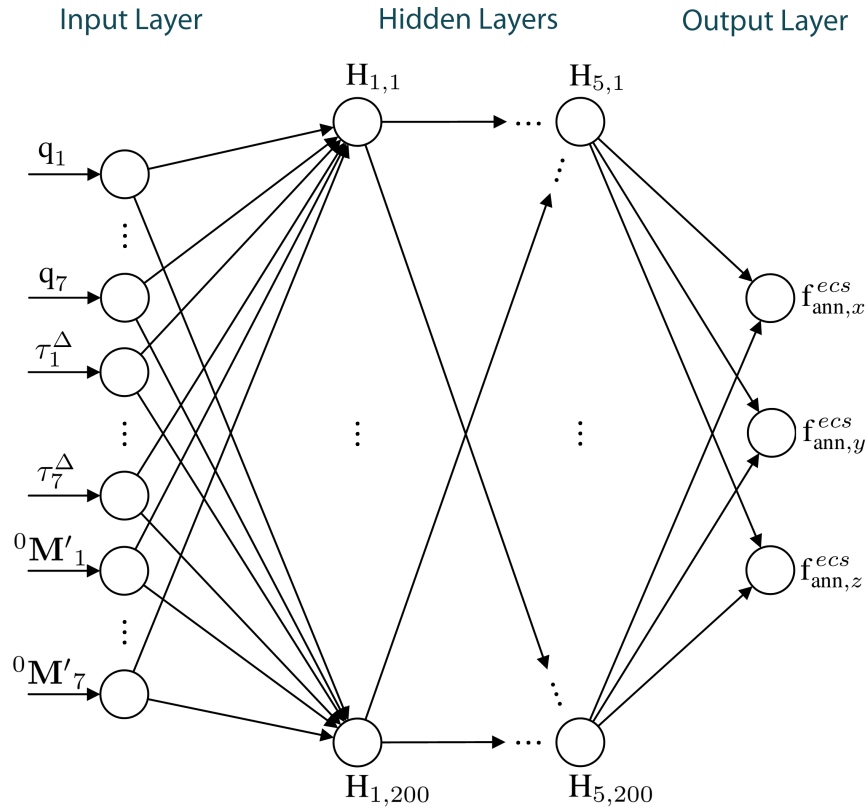


Figure 5.3: Architecture of the artificial neural network.

Figure 5.3 shows the neural network architecture. It has been also considered adding additional layers or neurons per layer. However, it has been observed that with less than five hidden layers, the network performance decreased, while no performance increase was observable as more hidden layers were taken into account. The same behavior could be seen for adding or removing neurons in each layer. Hence, all results shown in this work were calculated using the network architecture described above.

The relationship between f^{ecs} and the measured joint torques τ^Δ is highly dependent on the robot's current configuration. One challenge was the identification of a suitable, unique representation of the robot's pose. It has been observed that the joint angles $q_i, i \in \{1 \dots 7\}$, alone were not sufficient. Even by adding the end-effector rotation and position as inputs, the model was not able to learn the problem and did not converge. The high redundancy of the robotic arm, especially the additional seventh degree

of freedom, greatly increases the complexity of the function to be learned. Therefore, the translational and rotational part of the homogeneous pose matrices ${}^0\mathbf{M}_i, i \in \{1\dots7\}$, of the single joints were additionally fed to the network model. This results in 12 additional parameters per joint, i.e. the nine rotational parameters r_i^1, \dots, r_i^9 and three translational parameters t_i^1, \dots, t_i^3 for the i -th joint.

We define the vector ${}^0\mathbf{M}'_i$ as the element-wise representation of the pose matrix ${}^0\mathbf{M}_i$:

$${}^0\mathbf{M}'_i = \{r_i^1, \dots, r_i^9, t_i^1, \dots, t_i^3\} \in \mathbb{R}^{12}, i \in \{1\dots7\} \quad (5.19)$$

Hence, the input I and the output O of the neural network can be written as

$$I = \{q_1, \dots, q_7, \tau_1^\Delta, \dots, \tau_7^\Delta, {}^0\mathbf{M}'_1, \dots, {}^0\mathbf{M}'_7\} \in \mathbb{R}^{98} \quad (5.20)$$

and

$$O = \{f_x^{ecs}, f_y^{ecs}, f_z^{ecs}\} \in \mathbb{R}^3. \quad (5.21)$$

The input I was normalized to have zero mean and a unified variance of 1 (zero centering). The model was implemented in *Python* using *Keras* [15] with the *Theano*-Backend [103]. The Adam optimizer was used for training with mean squared error loss and a batch size of 6,000 samples.

5.3 Evaluation

An extensive database was built to develop and evaluate artificial neural network models. Therefore, ground truth contact forces are simulated by mounting ten specially manufactured weights in the range of 0-2 kg (0-20 N) on the end-effector. By using the constant gravity force and different robot base orientations, a homogeneous representation of contact forces in all directions is realized. Due to various combinations of base orientations, calibration weights and robot poses, the database consists of 330,000 randomized data points. See Section 5.2.2 for more details.

For evaluation purposes, a 20-fold cross-validation was performed allowing to validate the neural network on all 330,000 data points. The data set was divided into training (90%), validation (5%) and test data (5%).

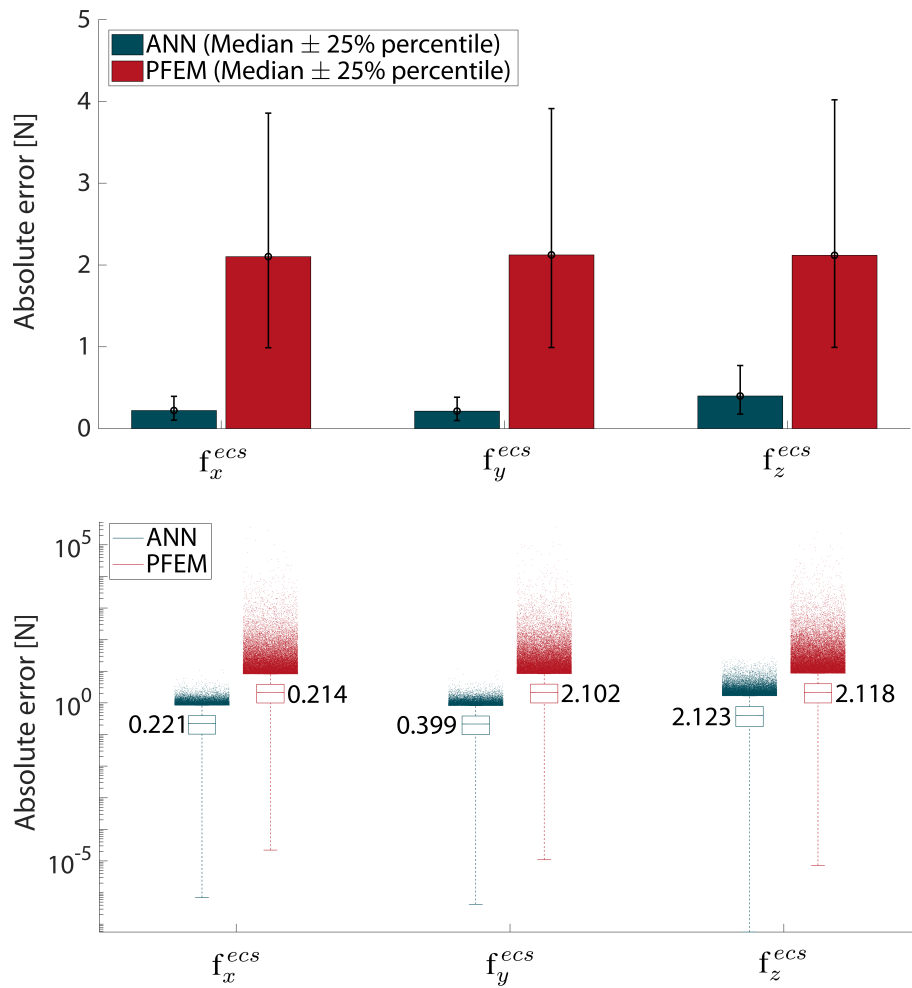


Figure 5.4: Absolute error of the ANN approach compared to the PFEM given for each axis in linear (top) and logarithmic (bottom) scale. The median error could be largely reduced by the ANN approach. Large outliers on the PFEM data with maximum errors in the range of 10^5 N could be avoided, resulting in much more precise contact force determination.

5.3.1 Accuracy

Figure 5.4 shows the overall accuracy of the developed calibration method compared to the PFEM, based on all measurements in logarithmic scale. The absolute error was calculated for the end-effector contact forces f^{ecs} for each axis separately. The calculation was based on the absolute difference between the ground truth (see Section 5.2.2) and the output of the trained neural network f_{ann}^{ecs} , as well as the measured contact forces given by the PFEM f_{pfem}^{ecs} .

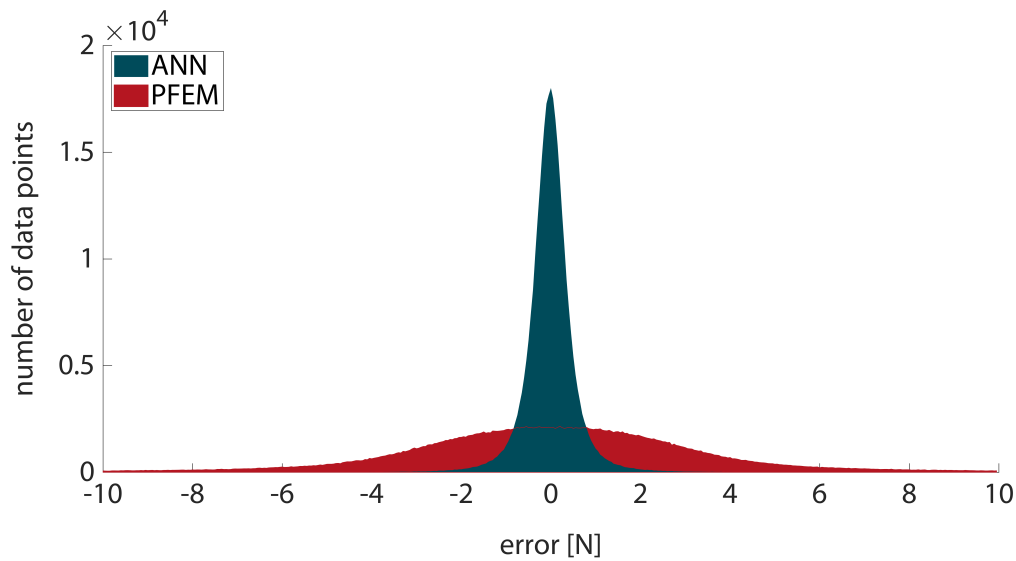


Figure 5.5: Error distribution of the ANN approach compared to the PFEM, evaluated on all data points within the full range of forces. The distribution of the ANN solution shows a significant peak around zero error. In contrast, data of the PFEM shows a flat shape, indicating less precise contact forces.

By using the neural network approach, the median absolute error could be reduced by 1.881 N (89.48%) for f_x^{ecs} , 1.909 N (89.92%) for f_y^{ecs} and 1.719 N (81.16%) for f_z^{ecs} . Furthermore, the PFEM data shows a high number of large outliers with maximum errors in the range of 10^5 N. Most likely, these high errors are caused by numerical instabilities of the computational model used by the PFEM near singular configurations (see [Section 5.3.2](#) for a detailed analysis). In contrast, the ANN solution provides much more stable results than the integrated model.

To further analyze the performance of both models, the error distributions of the two approaches were compared. [Figure 5.5](#) shows an error histogram of the test data. The number of data points, averaged over the three force axes, is shown for error values in a range between -10 N and 10 N. Comparing the shapes of the distributions, the PFEM data shows a flatter and broader shape than the ANN approach. This reflects the previously observed high number of large outliers of the PFEM data.

5.3.2 Outlier Reduction

In order to analyze the problems of numerical instabilities in more detail, the following evaluation focused on the errors as a function of the arm position. To acquire this, the manipulators' reach was used as a relevant parameter. The reach indicates how far the end-effector is displaced from the shoulder. At maximum reach, the arm is therefore fully extended and in a singular configuration. The PFEM model described in [Section 5.2.3](#) uses the geometric Jacobian matrix which would lose rank. [Figure 5.6](#) shows the contact force RMSE as a function of increasing reach. With an increasing reach over 700 mm, the error grows dramatically. In addition, the number and magnitude of outliers also rises massively. In contrast, the ANN data shows much more stable results. The outcome indicates that different arm positions have no impact on the model accuracy and, furthermore, that the model is capable of robustly handling singularities.

5.3.3 Generalization

Generalization is a crucial parameter when evaluating artificial neural networks. Therefore, two important parameters are analyzed as follows:

Force Generalization

To examine the network's capability to generalize between discrete loads, training was repeated with different modified data sets. Therefore, specific sub-sets from one up to three adjacent calibration weights were excluded and model accuracy was tested on this unknown data. [Figure 5.7](#) shows the resulting generalization performance. In the upper graph, one calibration mass was excluded. Each point shows the resulting root mean squared error whilst evaluating only on the excluded training data. In the second graph, two and three adjacent weights were missing for training, whereas evaluation was performed on each single excluded mass. The results show good generalization performance of the deep network model with a 56.01% higher RMSE on average, compared to the best network trained with the complete dataset. As one would expect, lower errors are observed for interpolated test data than for extrapolated output beyond the known training intervals.

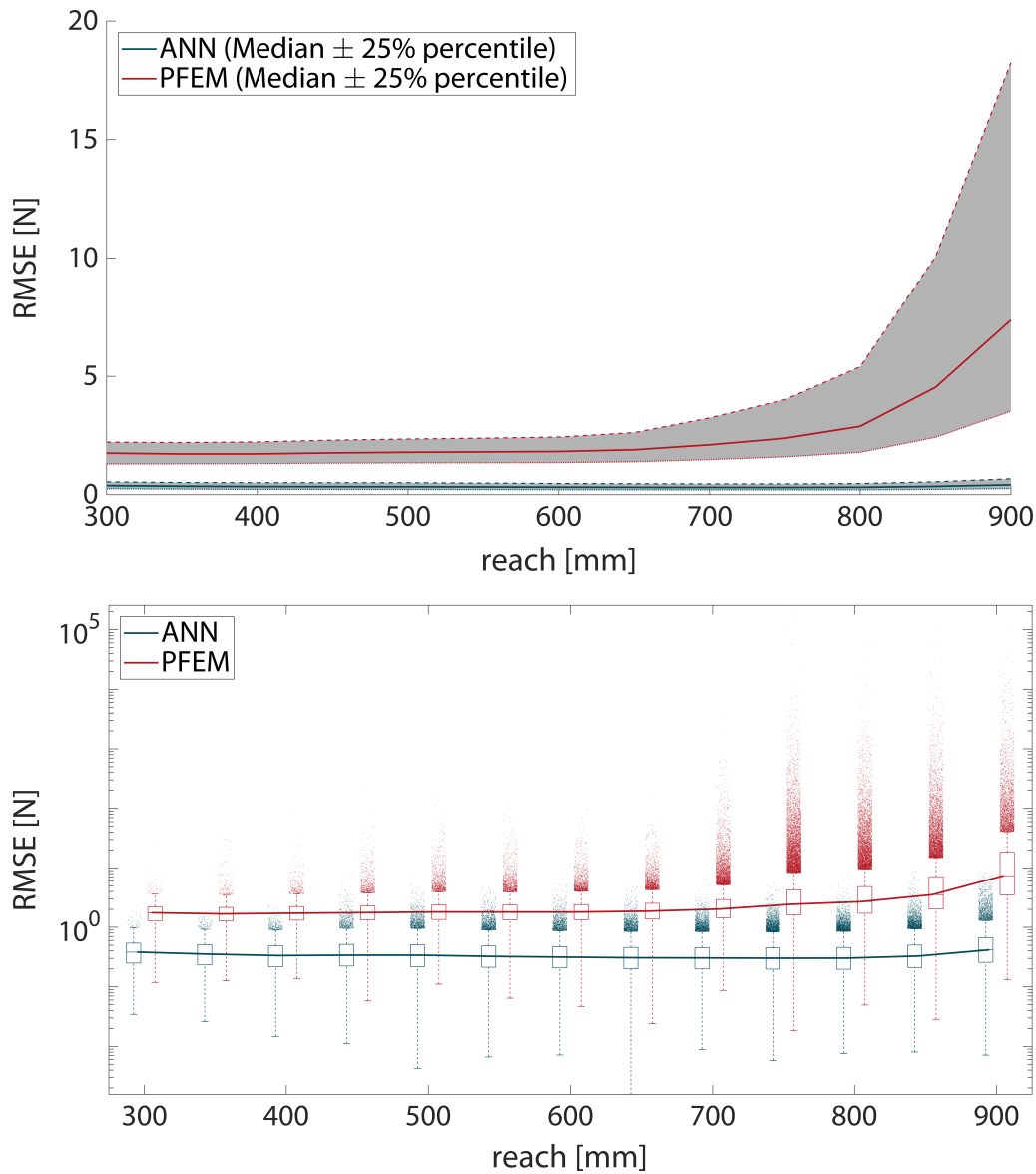


Figure 5.6: Contact force RMSE as a function of increasing manipulator reach in linear (top) and logarithmic (bottom) scale. A strongly rising error can be observed in the PFEM data from a reach above 700mm.

Directional Generalization

As described in Section 5.2.2, six different base orientations are used to generate combinations of contact forces. In a realistic scenario, however, these end-effector forces can occur from every direction. Thus, the network must be capable of generalizing intra-directionally. To analyze the directional generalization performance, additional test

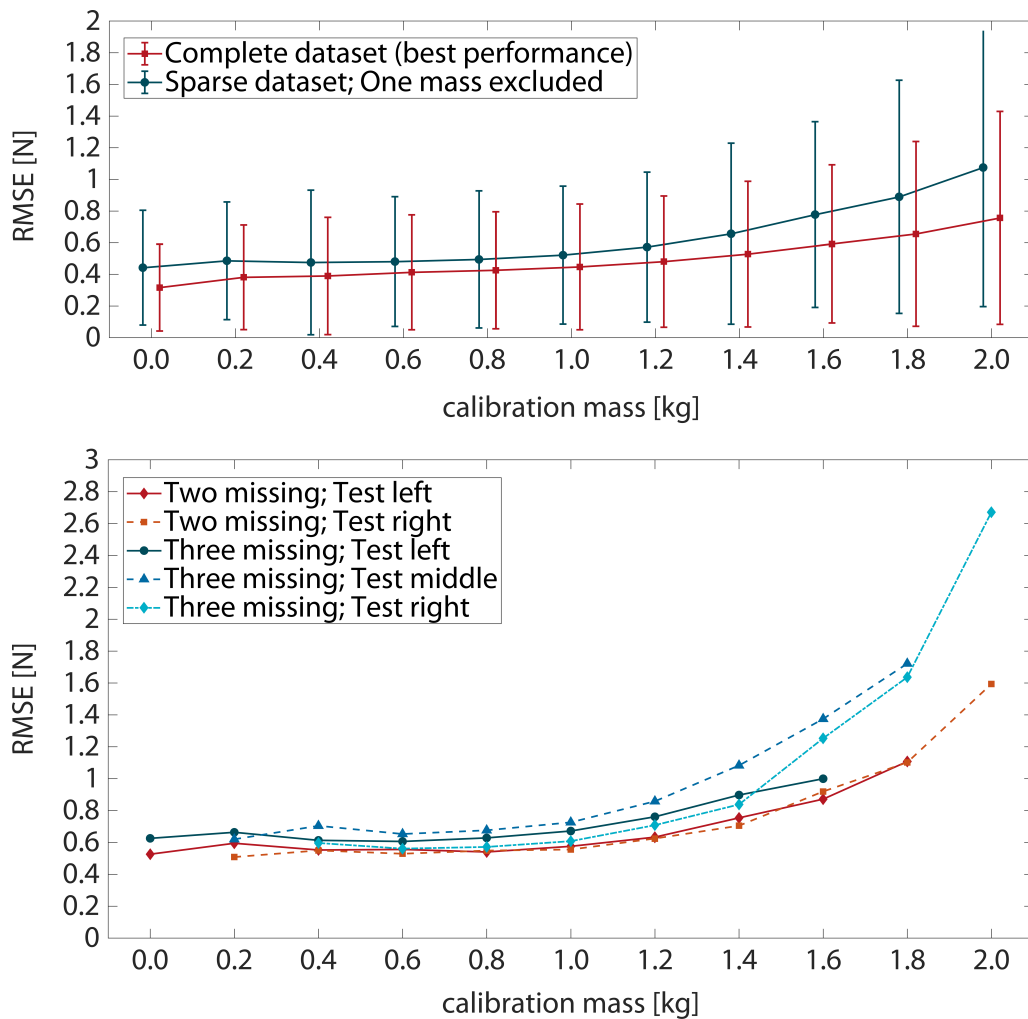


Figure 5.7: Analyzing the force generalization performance. In the upper graph one calibration mass was excluded for training, two and three adjacent weights in the lower figure. Each point shows the resulting RMSE whilst evaluating only on the excluded training data and on each single weight.

datasets are acquired using five varying, unlevelled base orientations. Therefore, the manipulator was mounted on a hexapod, as shown in Figure 5.8. Due to the lack of absolute precision of the hexapod, the actually reached Tait-Bryan angles, for calculating the base orientation (see Section 5.2.2), were measured using an inclinometer with an accuracy of 0.1° . A calibration mass of 1.0 kg (10N) and 1,500 measurements were used for each orientation. The data is evaluated using a pre-trained network with the highest perfor-



Figure 5.8: The LBR iiwa was mounted on a hexapod to acquire additional test datasets, to evaluate the intra-directional generalization performance.

Table 5.3: Directional generalization performance.

Ψ [°]	Θ [°]	Φ [°]	load [kg]	RMSE [N]	σ [N]
0.0	0.2	0.1	1.0	0.551	0.261
0.0	29.1	0.2	1.0	0.864	0.497
0.0	27.8	-14.6	1.0	0.856	0.467
0.0	-43.3	0.7	1.0	0.844	0.512
0.0	-31.7	39.4	1.0	0.901	0.563

mance. The results are shown in Table 5.3. The first dataset was acquired with a known base orientation (zero rotation) as reference. The following datasets including unknown contact forces, show 57.2% increased root mean squared errors on average, but staying under 1.0N. However, the resulting errors are still in an acceptable range. Hence, the

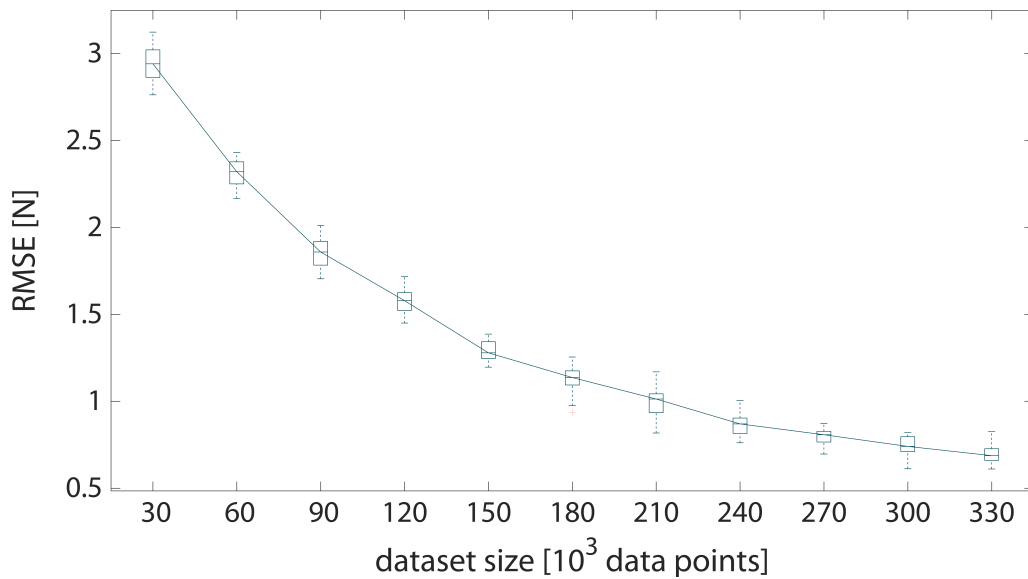


Figure 5.9: The mean squared error of the ANN approach is shown as a function of the dataset size. The precision of the calibration is exponentially related to the amount of data points used for training. The test error converges with more than 240,000 samples.

network shows robust intra-directional generalization performance and can thus be used to compute contact forces from any direction.

5.3.4 Training Data Size

To analyze how many data points are required for a precise calibration, the training was repeated with data sets of increasing sizes. The data points in the sub-datasets were distributed randomly and the division was kept comparable at 90%/5%/5% (Training/Validation/Test) for all experiments. Figure 5.9 shows the results of this evaluation, including the variance of all 20 cross-validations for each dataset. The mean squared error is shown as a function of the dataset size. The graph shows a strong inverse exponential relation between the amount of data points and calibration accuracy. First, the error is greatly reduced as the amount of data increases. The test error starts to converge once more than 240,000 samples are available. However, effects of overfitting could not be observed. The reason probably is the efficient training (discussed in Section 5.2.4) that employs robust batch updates using stochastic gradient descent.

Although the presented results improve the robot's sensitivity, minor limitations remain. The calibration was performed in a limited force range of 0-20 N, which is significantly below the maximum loads of the manipulator. However, compliant robots are often used in specialized practical applications within limited load ranges. Moreover, our results show good generalization performance, allowing for an increased force range to be calibrated with larger distances between the calibration weights.

Another possible issue is the time-consuming data acquisition process of the proposed method. Especially while calibrating multiple robots, this could be a major drawback. To overcome this, it is planned to transfer the trained model to similar robots. Due to the good performance of transfer learning methods in various fields of machine learning, it is plausible that pre-trained neural networks can be adapted to comparable kinematics by fine-tuning with only a small number of newly recorded data points.

5.4 Conclusions

The purpose of this chapter is to answer the third research question:

RQ3: How can the accuracy and stability of contact force estimation models of compliant manipulators be increased?

In other words, the aim of this chapter is to precisely determine corresponding contact forces at the end-effector from given joint position and torque data of a redundant serial lightweight manipulator (*KUKA LBR iiwa 7 R800*). The results show advantages of the developed neural learning approach compared to the PFEM. Firstly, the new calibration method can increase the accuracy of end-effector contact forces by 86.85% overall. Secondly, it has been shown that the calibration stability can be increased with the proposed approach. In contrast to the PFEM, which shows high outliers near singularities, the ANN approach shows robust results. The evaluation indicates that different arm positions do not affect the accuracy and, furthermore, the proposed model can robustly handle singularities.

After the promising findings of this chapter, it is planned to apply the presented meth-

ods to the end-effector torques. New optimized calibration weights are required, capable of generating specific torques at the end-effector. Furthermore, while switching input and output the approach could be further advanced. In this setup, contact forces could not only be determined, but actively controlled by learning the required joint torques.

6 Robotized Ultrasound¹

This chapter addresses the application of robotized ultrasound with the KUKA LBR iiwa, with a focus on radiation therapy, as an example of a quasi-static application. Tumor localization based on four dimensional ultrasound is a promising alternative to existing tracking systems for Image-guided radiation therapy, but requires a robotic system to navigate the ultrasound probe on the patient. The chapter shows how the redundant manipulator is used as an ultrasound robot and discusses how the methods and findings, developed in the previous chapters, are integrated into the application to improve the performance. A novel approach to hand guidance of the ultrasound probe is proposed. Since hand guidance is prone to put the robot into undesired configurations, like singularities or joint limits, a method is developed to increase usability and safety for the operator. The method is qualitatively evaluated by subjects in multiple realistic scenarios. The chapter concludes with a discussion of the most relevant findings.

6.1 Introduction

To compensate for tumor movements and thus to minimize the injury of healthy tissue is the aim of today's robotized radiation therapy systems like the CyberKnife. These systems would greatly benefit from developing a method for non-invasive live tracking of tumors. Given that the currently existing methods rely on live monitoring using fluoroscopy [95] or electromagnetic beacons [44, 114], the need to come up with an approach not using either ionizing radiation or invasive procedures is evident. In recent years, there have been multiple studies on using ultrasound for target localization during radiotherapy; see for example [11, 88]. To this end, real-time tumor tracking in radiotherapy is moving towards novel methods for image guidance [13, 39, 87]. Tumor localization based on four dimensional ultrasound is a promising alternative to existing tracking systems for

¹Parts of this chapter have been published in [52, 55]

Image-guided radiation therapy (IGRT) but requires continuous acquisition of high quality ultrasound images. In prior works, a robotized ultrasound system was presented, based on a common six-axes industrial robot (Adept Viper s850) carrying a three dimensional ultrasound transducer with a mounted force-torque sensor [51]. It has been shown that stable ultrasound image quality can be achieved over a whole treatment session (at least 30 minutes) by using this robotic system [50]. Furthermore, a method for fully automated patient localization has been presented and the feasibility of robotic ultrasound-guided Stereotactic body radiation therapy (SBRT) of the prostate has been shown [22, 23, 54].

Now, to further improve the robotic system, the Adept Viper s850 is replaced by the KUKA LBR iiwa manipulator, which is built for safe physical human-robot-interactions. Multiple integrated, certified safety features and collision detection modules ensure a safe environment for the patients during treatment. A more precise force sensor technology is capable of detecting the probe contact pressure with high accuracy. The robotized ultrasound system uses a three dimensional ultrasonic probe (GE 4V, connected to GE's Vivid E95 station) attached to the end-effector using a custom adapter plate as shown in [Figure 6.1](#). This setup allows for fully autonomous placement of the transducer anywhere on the patient.

However, in order to exploit the full potential of the kinematically redundant robot, the methods and findings developed in the previous chapters are integrated into the robotized ultrasound system. The inverse kinematics solution from [Chapter 3](#) is used to control the robot and safely navigate the probe over the body surface of the patient. It has been observed that the consistent control of the seventh degree of freedom allows the manipulator to adapt to different examination regions (e.g. prostate, liver, kidney). By a clever positioning of the elbow, locations on the body can be reached more effectively and the restriction by joint limits can be avoided. In addition, it could be shown that the impact of beam blocking in SBRT can largely be compensated by using a kinematically redundant robot controlled by the developed inverse kinematics method [22].

The findings from [Chapter 4](#) help to position the robot base in a diagnostic setup as shown in [Figure 6.1](#). The calculated workspace and dexterity maps are used to find an optimal position of the robot relative to the patient. In this process, the robot is placed to cover as much of the patient's body surface as possible with high dexterity.

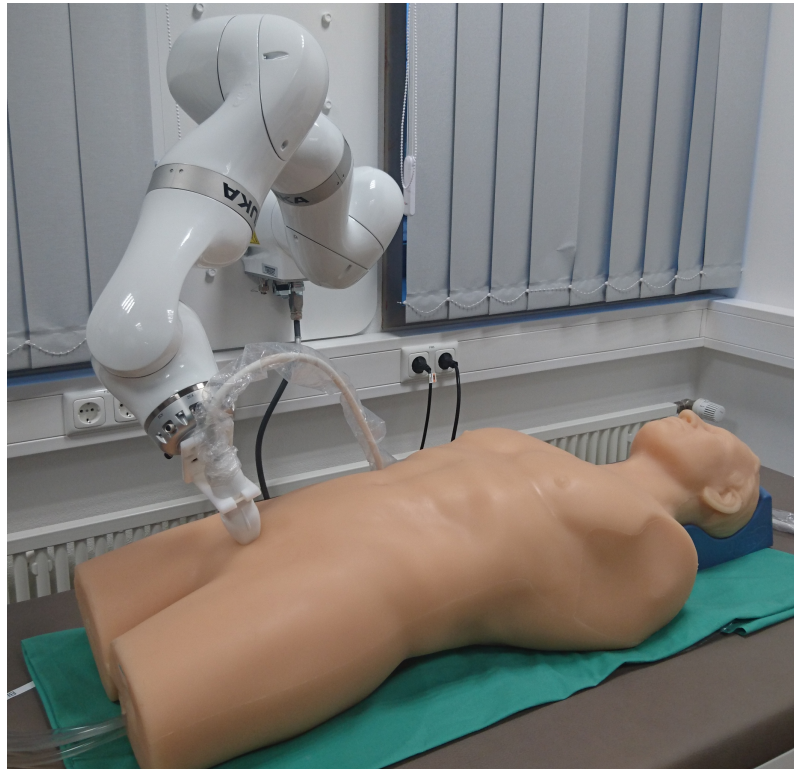


Figure 6.1: The robotized ultrasound system. The lightweight robot (KUKA LBR iiwa) is positioning the ultrasound probe (GE 4V) to monitor the prostate of the full-body ultrasound phantom (FAST, CAE Healthcare).

In order to acquire stable and high image quality for target tracking with robotized ultrasound systems, the contact pressure between ultrasonic probe and body surface must be measured and actively controlled. A continuous, minimum contact pressure is required to ensure good image quality. It can be significantly disturbed by respiratory movements that have to be compensated for. However, the accuracy strongly depends on a precise determination of contact forces at the end-effector using the embedded torque sensors of the KUKA LBR iiwa robot. The developed force estimation method from [Chapter 5](#) could potentially increase the performance of the contact pressure control, therefore improving the ultrasound image quality.

With the integration of the KUKA LBR iiwa into the robotic ultrasound system, further developments are made. As an alternative control mode for robotized ultrasound, a hand-

guidance mode has been developed, which enables intuitive handling for the operator. The ultrasound probe can be navigated by hand as in a standard ultrasound examination without a robot. Unfortunately, hand guidance is prone to put the robot into undesired configurations, like singularities or joint limits. To this end, a method is presented for safer and more stable hand guidance control, due to active joint limit avoidance.

6.2 Hand Guidance

To improve usability and effectiveness of the robotized ultrasound system, a novel approach to hand guidance of the probe is proposed. This manual control mode will allow the user to freely move the probe by simply grabbing it, just as it is done when working with a regular transducer. This is achieved by the robot's capability to directly and quickly measure external forces (caused by the user) which, in turn, are translated into robot motion. Making this approach as intuitive as possible has multiple advantages: skilled physicians can directly transfer their expertise and use the robotic system without having to think about it. Furthermore, it is possible to record motion trajectories for teaching, for Q&A or to later replay them. When compared to classical approaches (i.e. guiding the robot by means of a remote control), it is much easier to position the probe as desired when it can be simply pushed around. This position can then be used as a base line for target tracking. Hand guidance is realized by means of an integrated gravity compensation algorithm. This mode allows the robot to keep a position, compensating for its own weight and a previously calibrated tool with known weight, while still reacting to externally applied forces.

There is one severe drawback, however: hand guidance is prone to put the robot into undesired positions, like singularities or joint limits. If a joint is pushed into its limit during hand guidance, the robot controller will intervene by triggering an emergency stop, possibly causing abrupt motion of the effector. This is highly undesired and potentially dangerous. Consequently, it is imperative that these situations must be avoided. The current version of the robot's operating system (Sunrise 1.11) does not provide a straightforward method to achieve this.

The main contribution of this section is the development and testing of a safety layer

which continuously monitors the joints' positions and, if required, countermands external torques to avoid being pushed into joint limits. This new safety module allows the user to safely and intuitively guide the robot by hand, without the need of simultaneously monitoring the robot's state.

6.2.1 Active Joint Limit Avoidance

The robot control system can discriminate between innate (caused by the weight of the manipulator and the attached tool) and externally applied torques. When gravity compensation is activated, the robot is kept at its current position but can be moved by applying external forces, i.e. by pushing or pulling it. Internally, the system will adjust the joint positions such that the externally applied joint torques become zero – the robot moves and will stop once no external forces are acting.

The robot is controlled with a previously developed control and communication framework [55]. Based on a client/server concept, client-side control commands are received and processed by a central server unit and are executed by a client module running directly on the robot's controller. The framework integrates several special functionalities for robotized ultrasound applications, e.g. optimized inverse kinematics, probe contact pressure determination, and real time motion control.

The hand guidance module is implemented as a secondary process, running in parallel to the primary control application. The safety layer is built on top of KUKA's Fast Research Interface (FRI), which allows fast data exchange between the robot's control system and external programs. Since this exchange can occur in real-time (500-1000 Hz), it allows to manipulate the robot's trajectory and monitor its state with a very high frame rate. A schematic illustration of the control network architecture is shown in [Figure 6.2](#).

This network continuously monitors the joint values of all axes and, once a predefined angular distance to a joint's limit is underrun, creates a counter-acting torque in the joint in question. These torques are modeled according to a one-dimensional repulsive potential field: the closer a joint's value is to its limit, the higher the counter-acting torque will become. By appropriately selecting the parameters of the potential field, motion into the joint limits can be precluded.

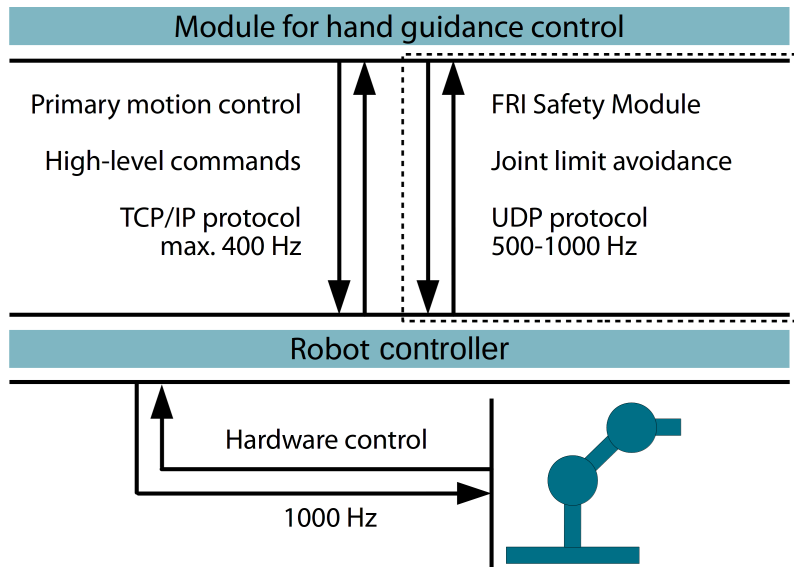


Figure 6.2: Schematic illustration of the control network architecture. The developed safety module (dashed box) works in parallel to the primary motion control process.

The torques τ_i are modeled as quadratic functions. Basically, there are several suitable types of functions. In this specific application, a quadratic function was chosen because it satisfies two essential criteria: a slow increase at a large distance to the limits and a strongly increasing counter-acting torque near the joint limits. We define ε_i , a joint-specific parameter, as the angular distance to a joint limit at which the counter-acting torques will be activated. The index i indicates the joints for $i = 1, \dots, n$, where n is the total number of joints. The torques τ_i are calculated as

$$\tau_i = \text{sgn}(q_i) \cdot \left(\frac{|q_i| - (|q_i^{\text{lim}}| - \varepsilon_i)}{\alpha_i} \right)^2, \quad \tau_i = 0 \forall |q_i| < |q_i^{\text{lim}}| - \varepsilon_i \quad (6.1)$$

where $\text{sgn}(q_i)$ is the sign function of the joint values q_i and q_i^{lim} are the joint-specific limits. Please note that the model assumes symmetric joint limits as with the KUKA LBR iiwa, e.g. $-170^\circ \leq q_1 \leq 170^\circ$. The parameter α_i controls the degree of curvature of the function. The parameters ε_i and α_i must be selected in such a way that reaching the joint limits is effectively avoided, but on the other hand the joint ranges are minimally restricted.

Because quantitative, detailed evaluations of the usability are difficult due to the high

subjectivity of the results, a qualitative analysis was performed [52]. The safety module was evaluated in multiple realistic scenarios. Novices in the field of robot control were asked to exemplarily locate several structures (heart, liver, prostate) in an anthropomorphic full-body ultrasound phantom (Bluephantom FAST Trauma). It has been observed that the use of the system became much more intuitive and easy for the users when the safety module was active. Additionally, the time needed for properly positioning the probe decreased markedly. Furthermore, the safety module completely prevented the emergency stop from being triggered.

6.3 Conclusions

The chapter presents the integration of the KUKA LBR iiwa in a robotized ultrasound system and discusses how the methods and findings developed in the previous chapters can be integrated to improve the performance. Furthermore, a novel and intuitive hand guidance mode for robotized ultrasound is presented. Since hand guidance suffers greatly from the difficulty to adhere to the robot's joint limits, a safety module is developed to increase usability and safety of the control mode. This module is based on KUKA's FRI and effectively prevents the joint limits from being reached. The joint angles of the seven axes are monitored in real time and, when the joint limits are being approached, counter-acting torques are generated. These repulsive torques are modeled by quadratic functions. The module was evaluated by volunteers and it was observed that even inexperienced users were able to use the system intuitively and safely.

7 Conclusions

The purpose of this work is the development of methods to optimize the applicability of kinematically redundant compliant manipulators for quasi-static tasks. Three research questions have been defined to properly approach the objective and each question has been addressed in a separate chapter. To conclude the results of this work, this chapter reviews the risen research questions and summarizes the proposed solutions.

The first research question [RQ1](#) dealt with the control of the additional seventh degree of freedom, the elbow self-motion manifold. The goal was to exploit the full potential of kinematically redundant serial manipulators by an inverse kinematics algorithm. In [Chapter 3](#), an inverse kinematics solution in closed form for redundant manipulators with seven degrees of freedom was developed. The algorithm combines robust arm configuration selection and elbow self-motion manifold control by an intuitive parameter. The proposed inverse kinematics solution provides two crucial advantages: First, all theoretical solutions are consistently selectable, using common arm configuration parameters and thus provide a complete solution space. Second, the elbow self-motion manifold parameter is defined in an intuitive way, making it easy to adapt the feature to practical applications. The elbow motion on a circular path is defined over the angle δ that describes the rotation of the elbow around the axis from shoulder to wrist. The algorithm was evaluated with a model of the lightweight robot KUKA LBR iiwa, looking at configuration reliability and continuity for an exemplary, challenging motion path. The results show that commanded arm configurations do not change spontaneously along a trajectory. Even for varying elbow angles, the joint trajectories are continuous and the presented algorithm provides robust solutions. The consistent control of arm configurations increases the safety of robot motions because spontaneous configuration changes within a single motion trajectory are avoided mathematically. Furthermore, the lean structure of the algorithm is real-time capable at minimal computational effort and deterministic runtime.

The goal of the second research question [RQ2](#) was to analyze how the robot's dexterity is affected by an additional seventh joint. Central to this question was the development of an adequate method to isolate these effects. Therefore, [Chapter 4](#) addressed a detailed dexterity and workspace analysis for three different kinematic models. The KUKA LBR iiwa 7 R800 robot represented the seven-jointed manipulators, while the KUKA KR 10 R900 robot was chosen for common manipulators with six degrees of freedom with comparable workspace dimensions. To isolate the influence of the additional joint, the LBR iiwa was simulated as a six-jointed robot to allow for direct comparison of seven- and six-jointed manipulators. A method was presented to determine the discretized workspace of a robot using a grid of voxels. Afterwards, 8,000 different effector orientations were defined to determine the robots' dexterity. The results show that the additional joint of the LBR iiwa enhances the dexterity, but contrary to initial expectations by only 16.8%. Compared to the average dexterity of the seven-jointed manipulator, the KR 10 outperformed this value by 7.6%, most likely due to significant larger joint ranges.

The third research question [RQ3](#) tackled problematic instabilities of KUKA's proprietary force estimation model and, therefore, motivated for methods to improve the accuracy of quasi-static end-effector contact force estimation. The aim of [Chapter 5](#) was to precisely determine corresponding contact forces from given joint position and torque data of a redundant serial lightweight manipulator (*KUKA LBR iiwa 7 R800*). An extensive training data set was acquired with 330,000 randomized robot positions and end-effector contact forces and used for training a deep feed-forward net with five hidden layers, 200 neurons each. The results show advantages of the developed neural learning approach compared to the proprietary force estimation model. The new calibration method can potentially increase the accuracy of end-effector forces by 86.85% overall. Furthermore, it has been shown that the model stability can be increased with the proposed approach. In contrast to the proprietary model, which shows high outliers near singularities, the neural learning approach shows robust results. The evaluation indicates that different arm positions do not affect the accuracy and, furthermore, the proposed model can robustly handle singularities.

[Chapter 6](#) presented the integration of the KUKA LBR iiwa in a robotized ultrasound system as an example of a quasi-static application. It has been demonstrated how the

methods developed in the previous chapters (RQ1-RQ3) can be used to improve the application performance. The inverse kinematics solution is used to control the robot and safely navigate the probe over the body surface of the patient. To cover as much of the patient's body surface as possible with high dexterity, the calculated workspace and dexterity maps are used to find an optimal position of the robot relative to the patient. Finally, the developed force estimation method could potentially increase the performance of the probe contact pressure control, therefore improving the ultrasound image quality. To further advance the usability and effectiveness of the robotized ultrasound system, a novel approach to hand guidance of the probe was presented. Since hand guidance suffers greatly from the difficulty to adhere to the robot's joint limits, a safety module was developed to increase usability and safety of the control mode.

The findings of this work improve the applicability of redundant compliant manipulators through specific optimizations of robot control and sensory systems. Especially for challenging applications it can be crucial to exploit the full potential of these robots. The presented methods allow for various advanced robotic tasks, as already demonstrated in robotized ultrasound. Other medical robotic applications could also potentially benefit from the findings of this thesis. The developed methods have already been integrated in a robotized TMS (transcranial magnetic stimulation) system [7, 31]. In this setup, the KUKA LBR iiwa navigates a magnetic coil, mounted to the end-effector, for precise placement on the patient's head. The developed inverse kinematics is used to ensure safe robot position control. Furthermore, in order to avoid collision between robot and patients, the robot's elbow position is continuously adapted to the position of the head, while aiming for maximum distance. During the stimulation process, the developed contact force estimation model helps to position the coil to the head with a constant contact pressure. However, due to the fundamental nature of the developed optimizations in this thesis, the range of possible alternative applications is extensive.

In future work, it would be interesting to investigate how the self-motion parameter δ could be used for secondary optimization objectives. While currently only the position of the elbow is controlled, it seems advantageous to add further target functions to the global problem of solving for the elbow self-motion manifold. For example, if the elbow position would be defined within a range and not as a single position, this margin could be used

for secondary optimization targets, e.g. avoiding joint limits. Furthermore, it would be interesting to apply the presented methods of force estimation to the end-effector torques. New optimized calibration weights are required, capable of generating specific known torques at the end-effector for training the neural network. Furthermore, while switching input and output the learning approach could be further advanced. In this case, contact forces could not only be determined, but actively controlled by learning the required joint torques.

Bibliography

- [1] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, and G. Hirzinger. The DLR lightweight robot: design and control concepts for robots in human environments. *Industrial Robot: An International Journal*, 34(5):376–385, 2007.
- [2] A. Albu-Schäffer, C. Ott, U. Frese, and G. Hirzinger. Cartesian impedance control of redundant robots: recent results with the DLR-light-weight-arms. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 3704–3709, 2003.
- [3] A. Albu-Schäffer, C. Ott, and G. Hirzinger. A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *The International Journal of Robotics Research*, 26(1):23–39, 2007.
- [4] R. M. Alexander. Three Uses for Springs in Legged Locomotion. *The International Journal of Robotics Research*, 9(2):53–61, 1990.
- [5] H. An, W. Clement, and B. Reed. Analytical inverse kinematic solution with self-motion constraint for the 7-DOF restore robot arm. In *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1325–1330, July 2014.
- [6] T. Asfour and R. Dillmann. Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1407–1412 vol.2, Oct 2003.
- [7] J. Bahlmann, I. Beckmann, I. Kuhlemann, A. Schweikard, and T. F. Münte. Transcranial magnetic stimulation reveals complex cognitive control representations in the rostral frontal cortex. *Neuroscience*, 300(0):425 – 431, 2015.
- [8] J. Barraquand and J.-C. Latombe. Robot Motion Planning: A Distributed Representation Approach. *The International Journal of Robotics Research*, 10(6):628–649, 1991.
- [9] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppel, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger. The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8, June 2010.
- [10] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppel, A. Stemmer, A. Albu-Schäffer, O. Eiberger, A. Beyer, G. Grunwald, and G. Hirzinger. From Research to an Industrial Product: The Development of the KUKA Lightweight Robot. *at - Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik*, 58(12):670–680, 2010.
- [11] O. Blanck, P. Jauer, F. Ernst, R. Bruder, and A. Schweikard. Pilot-Phantomtest zur ultraschallgeführten robotergestützten Radiochirurgie. In *44. Jahrestagung der DGMP*, volume 44, pages 122–123. DGMP, Sept. 2013.
- [12] M. Brady, J. M. Hollerbach, T. L. Johnson, M. T. Mason, and T. Lozano-Perez. *Robot Motion: Planning and Control*. MIT Press, Cambridge, MA, USA, 1983.

- [13] R. Bruder, F. Ernst, A. Schlaefer, and A. Schweikard. A Framework for Real-Time Target Tracking in Radiosurgery using Three-dimensional Ultrasound. In *Proceedings of the 25th International Congress and Exhibition on Computer Assisted Radiology and Surgery (CARS'11)*, volume 6 of *International Journal of Computer Assisted Radiology and Surgery*, pages S306–S307, June 2011. Motion Compensation in Radiosurgery.
- [14] J. W. Burdick. On the Inverse Kinematics of Redundant Manipulators: Characterization of the Self-Motion Manifolds. In K. J. Waldron, editor, *Advanced Robotics: 1989*, pages 25–34, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.
- [15] F. Chollet et al. Keras. <https://keras.io>, 2015.
- [16] P. Dahm and F. Joublin. Closed form solution for the inverse kinematics of a redundant robot arm. Technical Report 97-08, Institute für Neuroinformatik, Ruhr-Universität Bochum, Germany, 1997.
- [17] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Transactions of the ASME. Journal of Applied Mechanics*, 22:215–221, 1955.
- [18] T. Erber and G. M. Hockney. Equilibrium configurations of N equal charges on a sphere. *Journal of Physics A: Mathematical and General*, 24(23):1–12, 1991.
- [19] FRANKA EMIKA GmbH. <https://voith.com/robotics-en/index.html>. Accessed: 2018-09-29.
- [20] F. Freudenstein and E. J. F. Primrose. On the Analysis and Synthesis of the Workspace of a Three-Link, Turning-Pair Connected Robot Arm. *Journal of Mechanisms, Transmissions, and Automation in Design*, 106(3):365–370, 1984.
- [21] M. Gautier and A. Jubien. Force calibration of KUKA LWR-like robots including embedded joint torque sensors and robot structure. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 416–421, Sept 2014.
- [22] S. Gerlach, I. Kuhlemann, F. Ernst, C. Fürweger, and A. Schlaefer. Impact of robotic ultrasound image guidance on plan quality in SBRT of the prostate. *The British Journal of Radiology*, 90(1078):20160926, 2017. PMID: 28749165.
- [23] S. Gerlach, I. Kuhlemann, P. Jauer, R. Bruder, F. Ernst, C. Fürweger, and A. Schlaefer. Robotic ultrasound-guided SBRT of the prostate: feasibility with respect to plan quality. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–11, 2016.
- [24] German Aerospace Center, Institute of Robotics and Mechatronics, Mechatronic Systems. https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-9467/16255_read-8930. Accessed: 2018-11-07.
- [25] German Aerospace Center, Institute of Robotics and Mechatronics, Mechatronic Systems. https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3978/6178_read-8941. Accessed: 2018-11-07.
- [26] M. A. Goodrich and A. C. Schultz. Human–Robot Interaction: A Survey. *Foundations and Trends in Human–Computer Interaction*, 1(3):203–275, 2008.
- [27] C. M. Gosselin and M. Guillot. The Synthesis of Manipulators with Prescribed Workspace. *Journal of Mechanical Design*, 113(4):451–455, Dec. 1991.
- [28] S. Haddadin, A. D. Luca, and A. Albu-Schäffer. Robot Collisions: A Survey on Detection, Isolation, and Identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.
- [29] R. V. Ham, T. G. Sugar, B. Vanderborght, K. W. Hollander, and D. Lefeber. Compliant actuator designs. *IEEE Robotics & Automation Magazine*, 16(3):81–94, 2009.

- [30] E. J. Haug, C.-M. Luh, F. A. Adkins, and J.-Y. Wang. Numerical Algorithms for Mapping Boundaries of Manipulator Workspaces. *Journal of Mechanical Design*, 118(2):228–234, June 1996.
- [31] B. Hippmann, I. Kuhlemann, T. Bäumer, J. Bahlmann, T. F. Münte, and S. Jessen. Boosting the effect of reward on cognitive control using TMS over the left IFJ. *Neuropsychologia*, 125:109 – 115, 2019.
- [32] G. Hirzinger, A. Albu-Schaffer, M. Hahnle, I. Schaefer, and N. Sporer. On a new generation of torque controlled light-weight robots. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3356–3363, 2001.
- [33] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. ROTEX-the first remotely controlled robot in space. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2604–2611, 1994.
- [34] G. Hirzinger, N. Sporer, A. Albu-Schaffer, M. Hahnle, R. Krenn, A. Pascucci, and M. Schedl. DLR's torque-controlled light weight robot III-are we reaching the technological limits now? In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1710–1716, 2002.
- [35] T. Hofmann, B. Schölkopf, and A. Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.
- [36] J. Hollerbach, W. Khalil, and M. Gautier. *Model Identification*, pages 321–344. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [37] J. Hollerbach and K. Suh. Redundancy resolution of manipulators through torque optimization. *IEEE Journal of Robotics and Automation*, 3(4):308–316, August 1987.
- [38] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.
- [39] S. Ipsen, R. Bruder, R. O'Brian, P. J. Keall, A. Schweikard, and P. R. Poulsen. Online 4D ultrasound guidance for real-time motion compensation by MLC tracking. *Medical Physics*, pages 5695–5704, Oct. 2016.
- [40] ISO Central Secretary. Safety of Machinery – Safety-Related Parts of Control Systems – Part 1: General Principles for Design. Technical Report EN ISO 13849-1:2008, International Organization for Standardization, Geneva, CH, Dec. 2008.
- [41] R. N. Jazar. *Theory of Applied Robotics: Kinematics, Dynamics, and Control*, chapter 6.1 Decoupling Technique, pages 325–341. Springer, 2007.
- [42] R. N. Jazar. *Theory of Applied Robotics: Kinematics, Dynamics, and Control*, chapter 6.5 Singular Configuration, pages 363–367. Springer, 2007.
- [43] S. Jung, S. B. Yim, and T. C. Hsia. Experimental studies of neural network impedance force control for robot manipulators. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3453–3458, 2001.
- [44] P. J. Keall, E. Colvill, R. O'Brien, J. A. Ng, P. R. Poulsen, T. Eade, A. Kneebone, and J. T. Booth. The first clinical implementation of electromagnetic transponder-guided MLC tracking. *Medical Physics Letter*, 41(2):020702, Dec. 2018.
- [45] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.
- [46] P. K. Khosla and T. Kanade. Parameter identification of robot dynamics. In *24th IEEE Conference on Decision and Control*, pages 1754–1760, Dec 1985.

- [47] H. Kim and J. Rosen. Predicting Redundancy of a 7 DOF Upper Limb Exoskeleton Toward Improved Transparency between Human and Robot. *Journal of Intelligent and Robotic Systems*, 80(1):99–119, February 2015.
- [48] K. Kreutz-Delgado, M. Long, and H. Seraji. Kinematic Analysis of 7-DOF Manipulators. *The International Journal of Robotics Research*, 11(5):469–481, 1992.
- [49] S. Kucuk and Z. Bingul. Robot Workspace Optimization Based on a Novel Local and Global Performance Indices. In *IEEE International Symposium on Industrial Electronics*, volume 4, pages 1593–1598, June 2005.
- [50] I. Kuhlemann. Force and Image Adaptive Strategies for Robotised Placement of 4D Ultrasound Probes. Master's thesis, University of Luebeck, 2013. RobLab, Motion Compensation in Radiosurgery.
- [51] I. Kuhlemann, R. Bruder, F. Ernst, and A. Schweikard. WE-G-BRF-09: Force- and Image-Adaptive Strategies for Robotised Placement of 4D Ultrasound Probes. *Medical Physics*, 41(6):523, Aug. 2014.
- [52] I. Kuhlemann and F. Ernst. A safety module for active joint limit avoidance and intuitive hand guidance of a robotic ultrasound system. In *International Journal of Computer Assisted Radiology and Surgery*, volume 12, pages 38–39. CARS 2017, Springer, June 2017.
- [53] I. Kuhlemann, P. Jauer, F. Ernst, and A. Schweikard. Robots with seven degrees of freedom: Is the additional DoF worth it? In *IEEE 2nd International Conference on Control, Automation and Robotics*, pages 80–84, April 2016.
- [54] I. Kuhlemann, P. Jauer, A. Schweikard, and F. Ernst. Patient localization for robotized ultrasound-guided radiation therapy. In *Imaging and Computer Assistance in Radiation Therapy, ICART, 18th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 105–112, Oct. 2015.
- [55] I. Kuhlemann, P. Jauer, A. Schweikard, and F. Ernst. SU-G-JeP3-08: Robotic System for Ultrasound Tracking in Radiation Therapy. *Medical Physics*, 43(6):3672–3672, 2016.
- [56] I. Kuhlemann, A. Schweikard, P. Jauer, and F. Ernst. Robust inverse kinematics by configuration control for redundant manipulators with seven DoF. In *2nd International Conference on Control, Automation and Robotics*, pages 49–55, April 2016.
- [57] KUKA Laboratories GmbH. *LBR iiwa Betriebsanleitung*. KUKA Roboter GmbH, Augsburg, GER, second edition, Feb. 2014.
- [58] KUKA Roboter AG. <https://www.robots.com/robots/lbr-iiwa-7-r800>. Accessed: 2018-09-29.
- [59] N. Kumar, V. Panwar, N. Sukavanam, S. Sharma, and J. Borm. Neural network-based nonlinear tracking control of kinematically redundant robot manipulators. *Mathematical and Computer Modelling*, 53(9-10):1889–1901, 2011.
- [60] N. Kumar, V. Panwar, N. Sukavanam, S. P. Sharma, and J.-H. Borm. Neural network based hybrid force/position control for robot manipulators. *International Journal of Precision Engineering and Manufacturing*, 12(3):419–426, Jun 2011.
- [61] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [62] C. Lee and M. Ziegler. Geometric Approach in Solving Inverse Kinematics of PUMA Robots. *IEEE Transactions on Aerospace and Electronic Systems*, 20(6):695–706, Nov 1984.
- [63] S. Lee and A. Bejczy. Redundant arm kinematic control based on parameterization. In *IEEE International Conference on Robotics and Automation*, pages 458–465 vol.1, Apr 1991.

- [64] S. Li, Y. Zhang, and L. Jin. Kinematic control of redundant manipulators using neural networks. *IEEE transactions on neural networks and learning systems*, 28(10):2243–2254, 2017.
- [65] R.-J. Lian. Adaptive self-organizing fuzzy sliding-mode radial basis-function neural-network controller for robotic systems. *IEEE Transactions on Industrial Electronics*, 61(3):1493–1503, 2014.
- [66] Lozano-Perez. Spatial Planning: A Configuration Space Approach. *IEEE Transactions on Computers*, C-32(2):108–120, 1983.
- [67] T. Lu, G. Lin, and J. He. Neural-network-based 3D force/torque sensor calibration for robot applications. *Engineering Applications of Artificial Intelligence*, 10(1):87–97, 1997.
- [68] C. Lück and S. Lee. Self-motion topology for redundant manipulators with joint limits. In *IEEE International Conference on Robotics and Automation*, pages 626–631 vol.3, May 1993.
- [69] A. Maciejewski. Kinetic limitations on the use of redundancy in robotic manipulators. *IEEE Transactions on Robotics and Automation*, 7(2):205–210, 1991.
- [70] J. Mainprice and D. Berenson. Human-robot collaborative manipulation planning using early prediction of human motion. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 299–306, 2013.
- [71] L. Matthäus. *A robotic assistance system for transcranial magnetic stimulation and its application to motor cortex mapping*. PhD thesis, Institute for Robotics and Cognitive Systems, Lübeck, Germany, 2008.
- [72] H. Moradi and S. Lee. *Advances in Intelligent Computing: International Conference on Intelligent Computing*, chapter Joint Limit Analysis and Elbow Movement Minimization for Redundant Manipulators Using Closed Form Method, pages 423–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [73] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*, chapter 2.4 Manipulator workspace, pages 95–96. CRC Press, Inc., Boca Raton, FL, USA, first edition, 1994.
- [74] V. Nair and G. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning*, pages 807–814, 2010.
- [75] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task-Priority Based Redundancy Control of Robot Manipulators. *The International Journal of Robotic Research*, 6(2):3–15, 1987.
- [76] D. Nenchev, Y. Tsumaki, and M. Takahashi. Singularity-consistent kinematic redundancy resolution for the S-R-S manipulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 3607–3612 vol.4, Sept 2004.
- [77] H.-L. Pei, Q.-J. Zhou, and T. P. Leung. A Neural Network Robot Force Controller. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1974–1979, Jul 1992.
- [78] M. A. Peshkin, J. E. Colgate, W. Wannasuphprasit, C. A. Moore, R. B. Gillespie, and P. Akella. Cobot architecture. *IEEE Transactions on Robotics and Automation*, 17(4):377–390, 2001.
- [79] O. Porges. Analysis and applications of reachability and capability maps for robotic manipulators. Master's thesis, Julius Maximilian University Würzburg, Germany, 2014.
- [80] G. A. Pratt and M. M. Williamson. Series elastic actuators. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 399–406 vol.1, 1995.

- [81] J. Quinonero-Candela, C. Rasmussen, and C. Williams. Approximation methods for Gaussian process regression. *Large-scale kernel machines*, pages 203–224, 2007.
- [82] D. Raj, I. Raglend, and M. Anand. Inverse kinematics solution of a five joint robot using Feed forward and Elman network. In *International Conference on Circuit, Power and Computing Technologies*, pages 1–5. IEEE, 2015.
- [83] E. Rueckert, M. Nakatenus, S. Tosatto, and J. Peters. Learning Inverse Dynamics Models in $O(n)$ time with LSTM networks. In *Proceedings of the International Conference on Humanoid Robots*, 2017.
- [84] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323:533 EP –, 10 1986.
- [85] H. Sadjadian, H. Taghirad, and A. Fatehi. Neural networks approaches for computing the forward kinematics of a redundant parallel manipulator. *International Journal of Computational Intelligence*, 2(1):40–47, 2005.
- [86] E. Saff and A. Kuijlaars. Distributing many points on a sphere. *The Mathematical Intelligencer*, 19(1):5–11, December 1997.
- [87] J. Schlosser, R. Gong, R. Bruder, A. Schweikard, S. Jang, J. Henrie, A. Kamaya, A. Koong, D. Chang, and D. Hristov. Robotic intra-fractional US guidance for liver SABR: system design, beam avoidance, and clinical imaging. *Medical Physics*, pages 5951–5963, Oct. 2016.
- [88] J. Schlosser, K. Salisbury, and D. Hristov. Telerobotic system concept for real-time soft-tissue imaging during radiotherapy beam delivery. *Medical Physics*, 37(12):6357–6367, 2010.
- [89] H. Schoenert. An Evolutionary Algorithm for the Kinematic Synthesis of Serial Manipulators. Master's thesis, University of Luebeck, Institute for Robotics and Cognitive Systems, 2016.
- [90] G. Schreiber, A. Stemmer, and R. Bischoff. The Fast Research Interface for the KUKA Lightweight Robot. In *Workshop on IEEE ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications - How to Modify and Enhance Commercial Controllers*, pages 15–21, 05 2010.
- [91] A. Schweikard and F. Ernst. *Medical Robotics*, chapter 4.2 Jacobi-Matrices, pages 137–142. Springer, 2015.
- [92] A. Schweikard and F. Ernst. *Medical Robotics*, chapter 9.3 Joint Torques and Jacobi-Matrices, pages 339–346. Springer, 2015.
- [93] H. Seraji. Configuration control of redundant manipulators: theory and implementation. *IEEE Transactions on Robotics and Automation*, 5(4):472–490, Aug 1989.
- [94] M. Shimizu, H. Kakuya, W. Yoon, K. Kitagaki, and K. Kosuge. Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution. *IEEE Transactions on Robotics*, 24(5):1131–1142, Oct 2008.
- [95] H. Shirato, S. Shimizu, K. Kitamura, T. Nishioka, K. Kagei, S. Hashimoto, H. Aoyama, T. Kunieda, N. Shinohara, H. Dosaka-Akita, and K. Miyasaka. Four-dimensional treatment planning and fluoroscopic real-time tumor tracking radiotherapy for moving tumor. *International Journal of Radiation Oncology, Biology, Physics*, 48(2):435–442, 2000.
- [96] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*, chapter 69. Physical Human-Robot Interaction, pages 1835–1869. Springer Publishing Company, Incorporated, first edition, 2016.

- [97] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*, chapter 7. Motion Planning, pages 139–161. Springer Publishing Company, Incorporated, first edition, 2016.
- [98] B. Siciliano and O. Khatib. *Springer Handbook of Robotics*, chapter 10. Redundant Robots, pages 221–240. Springer Publishing Company, Incorporated, first edition, 2016.
- [99] A. C. Smith, F. Mobasser, and K. Hashtrudi-Zaad. Neural-Network-Based Contact Force Observers for Haptic Applications. *IEEE Transactions on Robotics*, 22(6):1163–1175, 2006.
- [100] A. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [101] M. Spong and S. Hutchinson. *Robot Modeling and Control*, volume 1. Wiley, Hoboken, NJ, USA, December 2005.
- [102] M. Spong and S. Hutchinson. *Robot Modeling and Control*, volume 2, chapter The General Inverse Kinematics Problem. Wiley, Hoboken, NJ, USA, December 2006.
- [103] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.
- [104] A. Thobbi, Y. Gu, and W. Sheng. Using human motion estimation for human-robot cooperative manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2873–2878, 2011.
- [105] Tokyo Robotics Inc. http://robotics.tokyo/products/torobo_arm. Accessed: 2018-09-29.
- [106] B. Tondu. A closed-form inverse kinematic modelling of a 7R anthropomorphic upper limb based on a joint parametrization. In *IEEE-RAS International Conference on Humanoid Robots*, pages 390–397, Dec 2006.
- [107] Y. C. Tsai and A. H. Soni. Accessible Region and Synthesis of Robot Arms. *Journal of Mechanical Design*, 103(4):803–811, 1981.
- [108] N. Vahrenkamp, T. Asfour, and R. Dillmann. Robot placement based on reachability inversion. In *IEEE International Conference on Robotics and Automation*, pages 1970–1975, May 2013.
- [109] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann. Manipulability analysis. In *IEEE-RAS International Conference on Humanoid Robots*, pages 568–573, Nov 2012.
- [110] B. Vanderborght, A. Albu-Schaeffer, A. Bicchi, E. Burdet, D. Caldwell, R. Carloni, M. Catalano, O. Eiberger, W. Friedl, G. Ganesh, M. Garabini, M. Grebenstein, G. Grioli, S. Haddadin, H. Hoppner, A. Jafari, M. Laffranchi, D. Lefeber, F. Petit, S. Stramigioli, N. Tsagarakis, M. V. Damme, R. V. Ham, L. Visser, and S. Wolf. Variable impedance actuators: A review. *Robotics and Autonomous Systems*, 61(12):1601–1614, 2013.
- [111] R. Vijaykumar, M. Tsai, and K. Waldron. Geometric optimization of manipulator structures for working volume and dexterity. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 228–236, 1985.
- [112] J. Y. Wang and J. K. Wu. Dextrous Workspaces of Manipulators, Part 2: Computational Methods. *Mechanics of Structures and Machines*, 21(4):471–506, 1993.
- [113] Y. Wang and P. Artemiadis. Closed-Form Inverse Kinematic Solution for Anthropomorphic Motion in Redundant Robot Arms. *Adv. Robot Autom.*, 2(3):–, 2013.

- [114] T. R. Willoughby, P. A. Kupelian, J. Pouliot, K. Shinohara, M. Aubin, M. Roach, L. L. Skrumeda, J. M. Balter, D. W. Litzenberg, S. W. Hadley, J. T. Wei, and H. M. Sandler. Target localization and real-time tracking using the Calypso 4D localization system in patients with localized prostate cancer. *International Journal of Radiation Oncology*Biophysics*, 65(2):528–534, 2006.
- [115] L. Yan, Z. Mu, and W. Xu. Analytical inverse kinematics of a class of redundant manipulator based on dual arm-angle parameterization. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3744–3749, October 2014.
- [116] Yaskawa Europe GmbH. <https://www.yaskawa.co.uk/uk/products/robotics/motoman-robots/productdetail/product/sda10f>. Accessed: 2018-09-29.
- [117] T. Yoshikawa. Manipulability and redundancy control of robotic mechanisms. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1004–1009, Mar 1985.
- [118] F. Zacharias, C. Borst, and G. Hirzinger. Capturing robot workspace structure: representing robot capabilities. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3229–3236, 2007.
- [119] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger. Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories. In *IEEE-RAS International Conference on Humanoid Robots*, pages 55–61, Dec 2009.
- [120] Z.-L. Zhou and C. Nguyen. Joint configuration conservation and joint limit avoidance of redundant manipulators. In *IEEE International Conference on Robotics and Automation*, volume 3, pages 2421–2426 vol.3, Apr 1997.
- [121] M. Zinn, O. Khatib, B. Roth, and J. K. Salisbury. Playing it safe [human-friendly robots]. *IEEE Robotics and Automation Magazine*, 11(2):12–21, 2004.

List of Figures

2.1	A common seven degree of freedom kinematic with relevant labels and joint axes.	8
2.2	DH parameters $a_i, \alpha_i, d_i, \theta_i$ defined for joint i and link i	11
2.3	Four different basic kinematic structures with their corresponding workspaces.	18
2.4	A standard mechanical drawing of the workspace specifications of the seven-axes S-R-S manipulator <i>KUKA LBR iiwa 7 R800</i>	19
2.5	An example of calculated workspaces with the desired taskspaces.	21
2.6	Comparison of kinematic models of the human arm and redundant manipulators.	24
2.7	Four examples of kinematically redundant robots.	25
2.8	Five generations of the DLR lightweight robot.	28
2.9	The mechatronic joint design of the DLR Light-Weight Robot III.	30
3.1	A seven-axes S-R-S redundant manipulator.	36
3.2	The six steps of the inverse kinematics solution.	38
3.3	The abstraction of the kinematic model with a projection into the x - y -plane.	39
3.4	Two projections to parametrize the ellipse for the inverse kinematics algorithm.	43
3.5	First motion simulation of the <i>LBR iiwa</i> robot.	46
3.6	Second motion simulation of the <i>LBR iiwa</i> robot.	47
3.7	Third motion simulation of the <i>LBR iiwa</i> robot.	49
4.1	The seven-jointed <i>KUKA LBR iiwa 7</i> robot and the six-jointed <i>KUKA KR 10 R900</i> robot.	54
4.2	The three-dimensional workspace of the <i>KUKA LBR iiwa</i> robot.	58
4.3	Dexterity volume of the KUKA KR 10, shown for different cutting planes.	62
4.4	Dexterity volume of the KUKA LBR iiwa, shown for different cutting planes.	63

4.5	Dexterity volume of the simulated six-jointed KUKA LBR iiwa, shown for different cutting planes.	64
4.6	Differential dexterity volume of the KUKA LBR iiwa seven-jointed vs. six-jointed, shown for different cutting planes.	65
4.7	Analysis of the dexterous enhancements by the seventh degree of freedom.	66
4.8	Percental increase of average dexterity for different δ discretizations.	67
4.9	Comparison of three different robot models and their mean dexterity.	67
5.1	The relation between end-effector contact forces and measured joint torques.	72
5.2	Robot mounting positions and distribution of gravity vector directions.	74
5.3	Architecture of the artificial neural network.	79
5.4	Absolute error of the ANN approach compared to the PFEM given for each axis.	81
5.5	Error distribution of the ANN approach compared to the PFEM.	82
5.6	Contact force RMSE as a function of increasing manipulator reach.	84
5.7	Analysis of the force generalization performance.	85
5.8	The LBR iiwa mounted on a hexapod to acquire additional test datasets.	86
5.9	The mean squared error of the ANN approach shown as a function of the dataset size.	87
6.1	The robotized ultrasound system.	93
6.2	Schematic illustration of the control network architecture.	96

List of Tables

3.1	DH-Parameter of the underlying seven-jointed redundant kinematic system	37
3.2	Runtime analysis for three different programming languages	50
4.1	The DH parameters of three different kinematic models used for the dexterity analysis.	56
4.2	Bounding box dimensions for workspace generation.	57
4.3	Number of inverse kinematic calculations used for the dexterity maps. . . .	60
5.1	Underlying DH-Parameters of the KUKA LBR iiwa 7 R800 robot and specifications of the integrated joint torque sensors.	73
5.2	The six base transformations and combinations of Tait-Bryan angles.	75
5.3	Directional generalization performance.	86

Abbreviations

1D	One dimensional
2D	Two dimensional
3D	Three dimensional
ANN	Artificial Neural Network
BCS	Base Coordinate System
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture
DH	Denavit-Hartenberg
DOF	Degree of Freedom
ECS	End-effector Coordinate System
FRI	Fast Research Interface
GPU	Graphics Processing Unit
HRI	Human-Robot-Interaction
IDIM	Inverse Dynamic Identification Model
IGRT	Image-guided Radiation Therapy
LBR	Leichtbauroboter
LWR	Light-Weight Robot
Med	RMSE Median

MSE	Mean Square Error
PFEM	Proprietary Force Estimation Model
RMSE	Root Mean Square Error
SBRT	Stereotactic Body Radiation Therapy
SEA	Series Alastic Actuators
S-R-S	Spherical-Revolute-Spherical kinematic structure
TMS	Transcranial Magnetic Stimulation
VIA	Variable Impedance Actuators

Acknowledgment

An erster Stelle möchte ich mich bei meiner Freundin Alisa bedanken. Durch ihre uneingeschränkte Unterstützung, besonders in stressigen und schwierigen Zeiten, gab sie mir stets unverzichtbaren Rückhalt und Motivation. Auch meinen Eltern Jutta und Norbert gilt ein besonderer Dank für all ihre Unterstützung auf meinem Lebensweg. Diese Arbeit wäre ohne sie nicht entstanden!

Zudem möchte ich meinem Doktorvater Herrn Prof. Dr.-Ing. Achim Schweikard danken, dass er mir die Möglichkeit gab, an diesem interessanten Thema zu arbeiten und mich dabei unterstützt hat, meine Ideen frei zu verfolgen. Bei Prof. Dr. Floris Ernst möchte ich mich für die vielen fachlichen Diskussionen und konstruktiven Kritiken, sowie das hilfreiche Korrigieren bedanken, was erheblich zum Gelingen der Doktorarbeit beigetragen hat. Gleichmaßen möchte ich mich bei meinen Kollegen bedanken, mit denen ich in den letzten sechs Jahren am Institut für Robotik und Kognitive Systeme arbeiten durfte. Es war mir eine besondere Freude mit euch zu kickern und zu arbeiten. Ich werde immer gern an diese Zeit zurückdenken. An dieser Stelle möchte ich auch allen Freiwilligen danken, die sich bereit erklärt haben, an Studien für diese Arbeit teilzunehmen. Abschließend möchte ich mich bei allen meinen Freunden und Kollegen bedanken, die mich während des Schreibens dieser Arbeit unterstützt und abgelenkt haben.

Curriculum vitae



Personal information

Name Ivo Kuhlemann
Date of birth December 7th 1986
Place of birth Lemgo

Education

2011–2013 **Master of Science**, University of Lübeck, Medical Engineering Science, *Thesis: Force- and image-adaptive strategies for robotized placement of 4D ultrasound probes*, Lübeck, Germany
2006–2011 **Bachelor of Engineering**, University of Applied Science OWL, Production Engineering, *Thesis: Optimization of WPC production parameters regarding impact strength*, Lemgo, Germany
2003–2006 **Higher education**, Higher Technical College Lüttfeld, Lemgo, Germany