

Sensorbasierte Objektübergabe mit einem Manipulator für die Mensch-Roboter-Interaktion

Masterarbeit

am Fachbereich Informatik der Universität Hamburg,
Arbeitsbereich Technische Aspekte Multimodaler Systeme



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

vorgelegt von: Johannes Liebrecht
Studienbereich: Informatik
Matrikelnummer: 6052944
Erstgutachter: Dr. Norman Hendrich
Zweitgutachter: Prof. Dr. Jianwei Zhang

Abstract

One goal in human-robot interaction is the achievement of a safe, efficient and natural object handover. The main reason for the loss of natural object handover are the comparatively very limited sensory and motor abilities of the robot. The restoration of the natural object handover is the focus of this master thesis.

A part of this work deals with the development of a robot system, which can identify, pick, place and weigh objects. For this purpose, the integration of hardware and software components were made with the software framework MoveIt!. In addition, software components are designed and developed to capture the full functionality of existing hardware components.

The developed robot system was used in experiments for improving natural object handover. Among other things, the influence of the object weight, the transfer position and the compliance of the manipulator to the natural object handover is demonstrated. The findings could be used for the further development of controllers and motion planners.

Zusammenfassung

Ein Teilziel in der Mensch-Roboter-Interaktion ist die Erreichung einer sicheren, effizienten und natürlich wirkenden Objektübergabe. Die Hauptursache für den Verlust der natürlich wirkenden Objektübergabe sind die vergleichsweise stark eingeschränkten sensorischen und motorischen Fähigkeiten des Roboters. Die Wiederherstellung der natürlich wirkenden Objektübergabe ist der Fokus dieser Masterarbeit.

Ein Bestandteil der Arbeit beschäftigt sich mit der Entwicklung eines Robotersystems, welches unter anderem Objekte identifizieren, greifen, platzieren und wiegen kann. Hierzu erfolgt einerseits die Integration von Hardware- und Softwarekomponenten mit dem Softwareframework MoveIt!. Außerdem werden Softwarekomponenten entworfen und weiterentwickelt, um die Funktionalität der bestehenden Hardwarekomponenten vollständig auszuschöpfen.

Aufbauend auf dem entwickelten Robotersystem werden anhand von Experimenten neue Erkenntnisse zur Verbesserung der natürlich wirkende Objektübergabe gewonnen. Beispielsweise wird der Einfluss des Objektgewichtes, der Übergabestelle und der Nachgiebigkeit des Manipulators auf die natürlich wirkende Objektübergabe nachgewiesen. Die gewonnenen Erkenntnisse könnten unter anderem für die Weiterentwicklung von Steuerungsalgorithmen und Bewegungsplanern eingesetzt werden.

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VIII
Algorithmenverzeichnis	IX
1 Einleitung	1
1.1 Ziel der Masterarbeit	2
1.2 Aufbau der Arbeit	3
1.3 Vergleichbare Arbeiten	4
2 Theoretische Grundlagen	9
2.1 Industrieroboter	9
2.2 Kinematik	11
2.3 Dynamik	17
2.4 Bewegungsplanung	19
2.5 Greifer und Roboterhände	22
2.6 Tiefenmessung mit Tiefensensoren	26
2.7 Objektübergabe in der Mensch-Mensch-Interaktion	28
2.8 Sicherheit	29
3 Implementierung	31
3.1 Verwendete Hardware	31
3.1.1 Gesamtdarstellung des Robotersystems	31
3.1.2 KUKA Lightweight Robot	31
3.1.3 Weiss Robotics WSG 50 und WSG-DSA	33
3.1.4 Logitech HD Pro Webcam C910	35
3.1.5 Microsoft Kinect	35
3.2 Verwendete Software	37
3.2.1 Open Motion Planning Library	37
3.2.2 GStreamer	37
3.2.3 Fast Research Interface Library	38
3.2.4 Reflexxes Motion Libraries	38

3.2.5	Robot Operating System	39
3.2.6	MoveIt!	39
3.3	Roboterkonfiguration mit MoveIt!	39
3.4	Objekterkennung	44
3.5	Erweiterung des wsg50-ros-pkg um zusätzliche Funktionen	46
3.6	Entwicklung des ros_fri Controllers für den LWR	46
3.7	Greifen und Platzieren	54
3.8	Sicherheitsmechanismen	56
4	Experimentelle Ergebnisse	59
4.1	Objektübergabe bei Stillstand	59
4.2	Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslösekraft	63
4.3	Objektübergabe im Laufe des Bewegungsablaufs	92
5	Fazit und Ausblick	95
5.1	Fazit	95
5.2	Ausblick	96
	Literaturverzeichnis	98
	Danksagung	104
	Eidesstattliche Erklärung	105

Abkürzungsverzeichnis

C-collision	Kollisionsraum
C-free	Kollisionsfreier Raum
C-space	Konfigurationsraum
cm	Zentimeter
DH	Denavit-Hartenberg
DLR	Deutsches Zentrum für Luft- und Raumfahrt
Fraunhofer IFF	.	Fraunhofer-Institut für Fabrikbetrieb und -automatisierung
FRI	Fast Research Interface
g	Gramm
KCP	KUKA Control Panel
kg	Kilogramm
LWR	KUKA Lightweight Robot
m	Meter
mm	Millimeter
ms	Millisekunde
N	Newton
OMPL	Open Motion Planning Library
ROS	Robot Operating System
RRT	Rapidly-exploring Random Tree
RViz	ROS Visualizer
SIFT	Scale-Invariant-Feature-Transform
SRDF	Semantic Robot Description Format
TCP	Tool Center Point
UDP	User Datagram Protocol
URDF	Unified Robot Description Format

Abbildungsverzeichnis

1.1	In der oberen Zeile sind generierte Objektübergabestellungen und in der unteren Zeile vom Menschen gelernte Objektübergabestellungen dargestellt.	5
1.2	Oben links ist die Sichtbarkeit des Menschen illustriert. Das Sicherheitskriterium wird oben rechts durch einen schützenden Bereich um den Menschen beachtet. In der unteren Abbildung wird die Komfortzone des Menschen dargestellt.	6
1.3	Dargestellt sind die vier Trage- und Übergabestellungen zur Untersuchung des räumlichen und zeitlichen Kontrast.	7
2.1	Visualisiert ist auf der linken Seite die Position und Richtung der Gelenkkoordinatensysteme des KUKA Lightweight Robot (LWR). . .	10
2.2	Abgebildet sind die DH Parameter und die platzierten Referenzkoordinatensysteme der Gelenke durch Anwendung der DH Konvention.	14
2.3	Die EXTEND Operation.	21
2.4	Die obere Zeile zeigt den Wachstumsprozess eines RRT an und die untere Zeile das dazugehörige Voronoi Diagramm.	22
2.5	Dargestellt sind von oben links nach unten rechts die ELU-1 Hand, DLR-Hit Hand I, Sandia Hand und die Shadow Dexterous Hand C6M.	25
2.6	Bolzen-in-Loch-Montage.	25
2.7	Ein Lichtstreifen wird auf das zu untersuchende Objekt projiziert. Eine Kamera erfasst die Verformung des Lichtstreifens durch die Objektoberfläche.	27
2.8	Skizze und Formel des optischen Triangulationsprinzips.	27
2.9	Auf der linken Seite wird ein Punktemuster in die Szene projiziert. Auf der rechten Seite wird das Punktemuster wegen des Objektes verzerrt dargestellt.	27
3.1	Dargestellt ist der KUKA LWR, Weiss Robotics WSG 50 & WSG-DSA, Microsoft Kinect und die Logitech HD Pro Webcam C910. Letztere befindet sich unter der Microsoft Kinect.	32
3.2	Abgebildet ist der Arbeitsraum des LWR. Auf der linken Seite ist die Seitenansicht und auf der rechten Seite die Draufsicht illustriert.	34

3.3	Abgebildet ist der KUKA Lightweight Robot (LWR) mit dem KUKA Robot Controller und dem KUKA Control Panel (KCP).	34
3.4	Abgebildet ist der WSG 50 Greifer und die integrierte Sensormatrix vom Typ WSG-DSA auf den Greiferfingern.	35
3.5	Logitech HD Pro Webcam C910.	36
3.6	Microsoft Kinect.	36
3.7	Dargestellt ist eine stark vereinfachte Gesamtdarstellung der verwendeten ROS Softwarepakete des Robotersystems.	37
3.8	FRI Systemarchitektur	38
3.9	Auf der linken Seite ist das visualisierte Robotermodell und seine Umgebung dargestellt. In der Mitte sind beispielhaft die kartesischen Koordinatensysteme tcp und xtion_camera_link illustriert. Auf der rechten Seite sind alle definierten kartesischen Koordinatensysteme dargestellt.	40
3.10	Auf der linken Seite ist das volumetrische Model eines Octrees abgebildet und auf der rechten Seite der dazugehörige Baum.	42
3.11	Illustriert ist die Bewegungsplanung mit dem MoveIt! Plugin Motion-Plannung für RViz.	43
3.12	Illustriert ist in der oberen Zeile eine Umgebung ohne (linke Seite) und mit (rechte Seite) Hindernissen. Die Abbildungen in der unteren Zeile zeigen die dazugehörige OctoMap.	43
3.13	Auf der linken Seite ist eine Tischszene dargestellt. In der Mitte befindet sich die mit dem SIFT Algorithmus berechneten Merkmalspunkte der Tischszene. Auf der rechten Seite werden die übereinstimmenden Punkte mit der zuvor gelernten Nescafe Dose illustriert.	44
3.14	GStreamer Pipeline.	44
3.15	Auf der linken Seite ist das Kamerabild mit den detektierten Objekten Nescafe Dose und Kellogs Schachtel dargestellt. In der Mitte ist die seitliche Sicht des Gesamtaufbaus illustriert. Auf der rechten Seite ist das dazugehörige Robotermodell und die detektierten Gegenstände in RViz dargestellt.	45
3.16	Abgebildet ist der Aufbau der Datenpakete des FINGER_GET_DATA Befehls.	47
3.17	Abgebildet sind die Schnittstellen des Action Clients und Action Servers.	49
3.18	Abgebildet ist der Zustandsautomat eines ActionServers.	50
3.19	Abgebildet ist der Zustandsautomat des ros_fri.	52
3.20	Inhalt der Datei oml_planning_pipeline.launch.xml.	52
3.21	Dargestellt sind die Eingabe- und Ausgabeparameter des geschwindigkeitsbasierten Online-Trajektoriegenerator-Algorithmus der Reflexxes Type II Motion Library.	54

3.22	Abgebildet ist der Zustandsautomat der Pick & Place Demo.	55
3.23	Abgebildet ist im blauen Quader der Arbeitsraum des LWR.	57
4.1	Überblick der verwendeten Versuchsobjekte, Übergabestellungen und Übergabestrategien bei den Experimenten.	60
4.2	Zustandsautomat der TCP_FORCE_DEMO.	61
4.3	In der Abbildung wird in einer grasp und release Phase die Objektübergabe mit verschiedenen Objekten dargestellt.	62
4.4	Dargestellt ist auf der linken Seite die Übergabestellung P1 und in der mittleren Abbildung die Übergabestellung P2. Beide Übergabestellungen unterscheiden sich nur in der Orientierung. In der rechten Abbildung sind die beiden Versuchsobjekte, ein grüner Würfel und eine Nescafe Dose abgebildet.	65
4.5	Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P1 im Joint-Position Modus.	66
4.6	Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P2 im Joint-Position Modus.	67
4.7	Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P1 im Joint-Impedance Modus.	68
4.8	Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P2 im Joint-Impedance Modus.	69
4.9	Abgebildet sind die mit "gut" bewerteten Versuchsreihen für die Übergabestellung P1 im Joint-Position Modus.	70
4.10	Abgebildet sind die mit "akzeptabel" bewerteten Versuchsreihen für die Übergabestellung P1 im Joint-Position Modus.	71
4.11	Abgebildet sind die mit "schlecht" bewerteten Versuchsreihen für die Übergabestellung P1 im Joint-Position Modus.	72
4.12	Abgebildet sind die Versuchsergebnisse für die Übergabestellung P1 im Joint-Position Modus. Dargestellt sind in den Diagrammen die durchschnittlichen Bewertungen gut (grün), akzeptabel (gelb) und schlecht (rot) für alle Objektgewichte. Für jede Greifkraft ist ein Diagramm illustriert.	73
4.13	Abgebildet sind die Versuchsergebnisse für die Übergabestellung P1 im Joint-Impedance Modus.	74
4.14	Abgebildet sind die Versuchsergebnisse für die Übergabestellung P2 im Joint-Position Modus.	75
4.15	Abgebildet sind die Versuchsergebnisse für die Übergabestellung P2 im Joint-Impedance Modus.	76
4.16	Vergleich der Versuchsergebnisse für die Übergabestellung P1 & P2 im Joint-Position Modus.	77

4.17 Vergleich der Versuchsergebnisse für die Übergabestellung P1 & P2 im Joint-Impedance Modus.	78
4.18 Vergleich der Versuchsergebnisse zwischen dem Joint-Position Modus (P) und dem Joint-Impedance Modus (S) in der Übergabestellung P1.	79
4.19 Vergleich der Versuchsergebnisse zwischen dem Joint-Position Modus (P) und dem Joint-Impedance Modus (S) in der Übergabestellung P2.	80
4.20 Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P1 im Joint-Position Modus.	81
4.21 Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P2 im Joint-Position Modus.	82
4.22 Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P1 im Joint-Impedance Modus.	83
4.23 Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P2 im Joint-Impedance Modus.	84
4.24 Abgebildet sind die durchschnittlichen Ergebnisse des numerischen Scores und der Greifkräfte für beide Übergabestellungen und Übergabestrategien.	85
4.25 Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P1 im Joint-Position Modus.	86
4.26 Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P1 im Joint-Impedance Modus.	87
4.27 Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P2 im Joint-Position Modus.	88
4.28 Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestellung P2 im Joint-Impedance Modus.	89
4.29 Abgebildet sind die durchschnittlichen Ergebnisse des numerischen Scores, des Objektgewichtes und der Greifkräfte für beide Übergabestellungen und Übergabestrategien.	90
4.30 In der Abbildung sind die externen Kräfte, Gelenkwerte und Gelenkgeschwindigkeiten für eine Objektübergabe im Laufe der Bewegung dargestellt.	94

Tabellenverzeichnis

2.1	Dargestellt sind die DH Parameter für den KUKA LWR.	16
-----	---	----

Algorithmenverzeichnis

1	RRT Algorithmus	20
2	RRT-Connect Algorithmus	23

1 Einleitung

Heutzutage wird die Produktion in vielen Industriebereichen teilweise oder vollständig mit Industrierobotern vollzogen. Die fortschreitende Automatisierung entlastet den Menschen vor schwerer körperlicher und monotoner Arbeit, verbessert und vergleichmäßigt die Produktqualität, erhöht die Durchsatzleistung und reduziert die Personalkosten. Für einige Unternehmensbranchen, darunter beispielsweise die Automobilindustrie, ist die Automatisierung eine Notwendigkeit geworden, um die Wettbewerbsfähigkeit auf dem Weltmarkt zu erhalten [Prozessautomatisierung \[2014\]](#). Derzeit werden in der Industrie die Arbeitsräume der Industrieroboter mit Schutzzäunen abgegrenzt, um die Sicherheit der Arbeiter gewährleisten zu können. Diese verhindern allerdings effektive Arbeitsabläufe und benötigen viel Platz. Aus wirtschaftlicher Sicht ist langfristig die Aufhebung der Arbeitsraumtrennung von Menschen und Robotern sinnvoll [MaschinenMarkt \[2014\]](#). Der deutsche Roboterhersteller KUKA Roboter GmbH ist seit dem Jahre 1991 mit der Entwicklung eines leichtgewichtigen Roboterarmes beschäftigt, welcher einem Arbeiter bei der Produktion assistieren soll. Seit Dezember 2008 kann der KUKA LWR in der vierten Version erworben werden, welcher derzeit hauptsächlich für Forschungs- und Entwicklungszwecke eingesetzt wird [Bischoff \[2010\]](#). In naher Zukunft jedoch, sollen leichtgewichtige Roboterarme in der Industrie eine flexiblere Produktion und Montage ermöglichen.

Ein weiteres Anwendungsgebiet für Assistenzroboter ist der Pflegebereich. Die demografische Entwicklung in den meisten Industrienationen führt zu einer immer älter werdenden Gesellschaft. Die Zahl der Pflegebedürftigen in Deutschland lag im Jahre 2012 bei 2,4 Millionen Menschen und wird bis 2030 auf 3,4 Millionen ansteigen. Deutschland droht in den nächsten Jahren ein Personalmangel um die Versorgung der Pflegebedürftigen sicherzustellen [Pflegelandschaft \[2012\]](#). Mit dem zusätzlichen Einsatz von Pflegerobotern bzw. Servicerobotern könnte man dieser Entwicklung entgegenwirken. Das europäische Projekt Robot-Era befasst sich mit der Entwicklung eines Serviceroboters für den Pflegebereich, dessen Tauglichkeit zurzeit unter realen Bedingungen in Altersheimen untersucht wird [ROBOT-ERA \[2014\]](#).

Interessanterweise reagieren Menschen in verschiedenen Staaten unterschiedlich auf den bevorstehenden technologischen Wandel. In Japan herrscht eine allgemeine Begeisterung in der Bevölkerung den Robotern gegenüber, während die Menschen in

Europa dieser Technologie noch eher skeptisch gegenüberstehen. Um Ängste vor dem "Fremden" und für einige Menschen bedrohlich wirkenden Robotern abzubauen, ist die Entwicklung von humanoiden Robotern erforderlich. Ihre Integration in den Alltag des Menschen scheint am wenigsten problematisch zu sein. Hierbei stehen die Nachbildung des menschlichen Aussehens und besonders die Übernahme menschlicher Fähigkeiten im Mittelpunkt der Forschung. Eine wichtige Fähigkeit des Menschen ist die Kommunikation bzw. Interaktion mit anderen. Die Fähigkeit muss auch zwischen Robotern und Menschen bestehen, um Roboter in unser Alltagsleben zu integrieren. Ein Teilziel in der Mensch-Roboter-Interaktion ist die Erreichung einer sicheren, effizienten und natürlich wirkenden Objektübergabe. Menschen verwenden eine komplexe Sequenzen aus Sprache, Gesten, Blicken, Armbewegungen und taktilen Erkennungen, um den Wunsch zur Objektübergabe zu registrieren und in einem mehrphasigen Übergabeprozess zu realisieren. Nach derzeitigem technologischen Entwicklungsstand verfügen Roboter verglichen mit den menschlichen Fähigkeiten nur über sehr stark eingeschränkte sensorische und motorische Fähigkeiten. Dem Roboter fehlen unter anderem Informationen über den benötigten Kraftaufwand zur Freigabe des Objektes, sowie den genauen Zeitpunkt der Objektübergabe. Die Wiederherstellung der natürlich wirkenden Objektübergabe zwischen Mensch und Roboter ist eine wichtige Aufgabe im Bereich der Mensch-Roboter-Interaktion und Inhalt dieser Masterarbeit.

1.1 Ziel der Masterarbeit

Die vorliegende Masterarbeit verfolgt zwei Ziele. Das erste Ziel ist die Entwicklung eines Robotersystems bestehend aus dem KUKA LWR, Weiss Robotics WSG 50 Greifer mit WSG-DSA Fingern, Microsoft Kinect und der Logitech HD Pro Webcam C910. Das Robotersystem soll in der Lage sein Objekte zu identifizieren, Objekte zu greifen und platzieren und das Objektgewicht zu berechnen. Zudem sollen Softwarekomponenten erweitert bzw. entwickelt werden, um die vollständige Funktionalität der Hardwarekomponenten zu nutzen.

Aufbauend auf dem Robotersystem soll anhand von Experimenten die Objektübergabe zwischen Mensch und Roboter erforscht werden, um eine für den Menschen natürlich wirkende Objektübergabe herzustellen. Hierfür sollen die Einflussfaktoren Kraftaufwand, Kraftrichtung, Objektübergabestellungen und Nachgiebigkeit bei der Objektübergabe untersucht werden.

1.2 Aufbau der Arbeit

Die Gliederungsstruktur der vorliegenden Arbeit wird in diesem Abschnitt erläutert. Die Arbeit setzt sich aus 5 Kapiteln zusammen.

Im zweiten Kapitel werden die theoretischen Grundlagen für ein besseres Verständnis dieser Masterarbeit vermittelt. Als Einstieg wird der Aufbau und die Funktionsweise eines Industrieroboters beschrieben, sowie die Grundlagen der Kinematik und Dynamik behandelt. Daraufhin wird die Bewegungsplanung am Beispiel des Bewegungsplanungsalgorithmus RRT-Connect dargelegt. Anschließend werden Präferenzen für den Einsatz von Greifern und Roboterhänden für die Mensch-Roboter-Interaktion beschrieben. Danach wird das Tiefenmessverfahren der eingesetzten Microsoft Kinect vermittelt. Anschließend wird die Objektübergabe in der Mensch-Mensch-Interaktion aus einigen Studien betrachtet, um Entwurfskriterien für das Verhalten des Roboters festzustellen. Abschließend werden mögliche Sicherheitsmaßnahmen für eine sichere Mensch-Roboter-Interaktion beschrieben.

Das dritte Kapitel behandelt die einzelnen Schritte für die Entwicklung des Robotersystems. Die ersten beiden Unterkapitel befassen sich mit der verwendeten Hardware und Software des Robotersystems. Anschließend wird die Roboterkonfiguration mit MoveIt! und die Integration von Systemkomponenten erläutert. Weiterhin wird die Objekterkennung mittels der entwickelten GStreamer Pipeline von [Bistry \[2013\]](#) vorgestellt. Überdies wird die Erweiterung der WSG 50 Software um zusätzliche Funktionen und die Entwicklung des `ros_fri` Controllers zur Steuerung des KUKA LWR behandelt. Anschließend wird der Zustandsautomat der implementierten Pick & Place Demo vorgestellt. Abgeschlossen wird das Kapitel mit den verwendeten und implementierten Sicherheitsmechanismen während der Arbeit mit dem Robotersystem.

Im vierten Kapitel werden Experimente zur Erreichung einer natürlich wirkenden Objektübergabe zwischen Mensch und Roboter beschrieben. Im ersten Experiment liegt der Fokus auf den Einflussfaktoren Kraftaufwand und Kraftrichtung zur Objektübergabe während des Bewegungsstillstandes. Anschließend wird der Zusammenhang zwischen Objektgewicht und Greifkraft untersucht. Hierzu werden zudem unterschiedliche Übergabestellungen und der Einfluss der Nachgiebigkeit betrachtet. Abschließend wird die Objektübergabe im Laufe des Bewegungsablaufs kurz behandelt.

Im letzten Kapitel wird das Ergebnis dieser Arbeit und ein Ausblick beschrieben.

1.3 Vergleichbare Arbeiten

Die vorliegende Arbeit befasst sich mit zwei großen Themengebieten, wodurch es schwierig ist, vergleichbare Arbeiten die gleichzeitig beide Themengebiete behandeln zu finden. In diesem Abschnitt werden daher die einzelnen Themengebiete getrennt mit ähnlichen Arbeiten verglichen. Als erstes werden einige Arbeiten vorgestellt, welche sich mit der Systemintegration von Manipulatoren beschäftigen. Anschließend werden Arbeiten mit dem Schwerpunkt Objektübergabe zwischen Mensch und Roboter beschrieben.

In der Masterarbeit von [Pedersen \[2000\]](#) wird die Integration des KUKA Light Weight Robot auf den mobilen Manipulator Little Helper durchgeführt. Pedersen verwendet das Fast Research Interface um Echtzeitkontrolle über den LWR zu erlangen. Mit dem Position Controller und dem Cartesian Impedance Controller wird die Platziere-Bolzen-in-Loch Demo durchgeführt. Die Aufgabe ist schwer mit dem Position Controller zu lösen, weil die Position und Orientierung des Bolzens relativ zum Loch übereinstimmen muss. Der Cartesian Impedance Controller erlaubt vergleichsweise größere Positions- und Orientierungsabweichungen des Bolzens und ermöglicht mit seiner Nachgiebigkeit den Bolzen leichter einzufügen. Weiterhin hat Pedersen das Objektgewicht mit den Momentensensoren des LWR bestimmt und dabei die unzureichende Genauigkeit der Sensoren zur exakten Bestimmung bemängelt.

Die Arbeit von Pedersen wurde in den Masterarbeiten von [Schou et al. \[2012\]](#) mit der Weiterentwicklung des mobilen Manipulators Little Helper fortgesetzt. Dazu wurde beispielsweise auf den LWR der Schunk WSG 50 Greifer montiert. Der Fokus der Autoren liegt in der Integration von Manipulator und Greifer für industrielle Montageaufgaben. Hierfür wurden zuerst elementare Funktionen implementiert, wie beispielsweise das Öffnen des Greifers oder das Bewegen des Manipulators zu einem Punkt. Darauf aufbauend wurden Fähigkeiten wie z.B. das Greifen eines Objektes entwickelt. Für die Entwicklung neuer Montageaufgaben haben die Autoren ein Programmierinterface entwickelt, mit welchem neue Prozesse durch das sequentielle Auswählen der implementierten Fähigkeiten zusammengestellt werden können.

Der zweite Schwerpunkt der Arbeit ist die Objektübergabe zwischen Menschen und Robotern. In der Studie von [Chan et al. \[2012\]](#) wurde die Mensch zu Mensch Objektübergabe erforscht, um einen Controller zur Robotersteuerung zu entwickeln, welcher eine sichere, effiziente und intuitive Mensch zu Roboter Objektübergabe gewährleistet. Untersucht wurde die Greifkraft und Belastungskraft für Objekte mit unterschiedlichen Gewicht. Ein Ergebnis der Studie war, dass der Objektgeber für die Sicherheit der Objektübergabe und der Objektnehmer für den Objektübergabezeitpunkt verantwortlich ist.

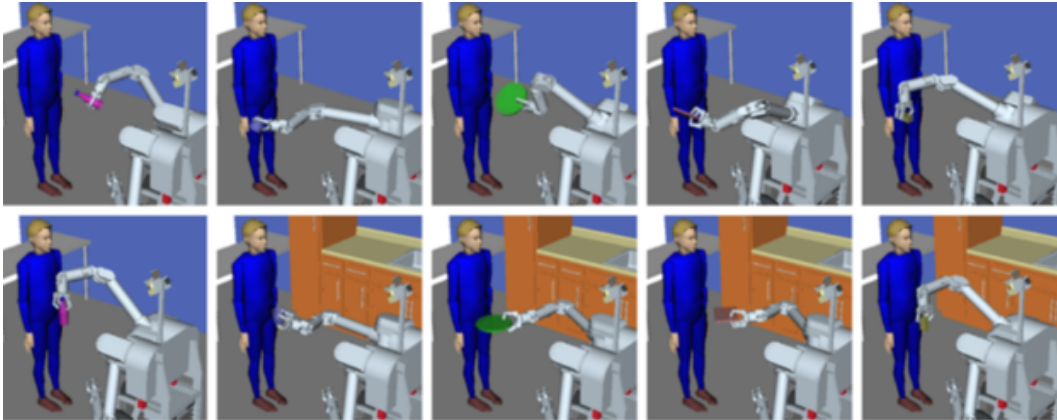


Abbildung 1.1: In der oberen Zeile sind generierte Objektübergabestellungen und in der unteren Zeile vom Menschen gelernte Objektübergabestellungen dargestellt. Abbildung entnommen aus [Cakmak et al., 2011 \[a\]](#)

In der Studie von [Cakmak et al., 2011 \[a\]](#) wurde nach dem menschlichen Vorbild gelernten Objektübergabestellungen mit generierte Objektübergabestellungen anhand eines kinematischen Modells miteinander verglichen. In [Abbildung 1.1](#) sind einige Beispiele für beide Übergabestellungsarten dargestellt. Die Autoren kamen zu dem Ergebnis, dass die generierten Objektübergabestellungen eine bessere Erreichbarkeit bieten. Während die gelernten Objektübergabestellungen für den Menschen natürlicher wirken. Weiterhin wird empfohlen die Sichtbarkeit des Objektes bei der Planung zu berücksichtigen, sowie das Objekt in der Orientierung zu übergeben, in welcher es am häufigsten im Alltag vorkommt. Bewegungsplaner können mit diesen Erkenntnissen modifiziert werden, um neben einer optimalen auch eine natürliche Objektübergabestellungen zu generieren.

Die Autoren [Dehais et al. \[2011\]](#) haben einen ergonomischen Bewegungsplaner mit dem Namen Human Aware Manipulation Planner entwickelt, welcher nach den Kriterien Sichtbarkeit, Sicherheit und Komfort Objektübergabestellungen generiert. In [Abbildung 1.2](#) ist der erlaubte (grüne) und nicht erlaubte (rote) Arbeitsraum für jedes der drei Kriterien dargestellt. Oben Links sind die sichtbaren Punkte in der Umgebung des Menschen illustriert. Oben Rechts befindet sich aus Sicherheitsgründen ein schützender Bereich um den Torso des Menschen. In der unteren Abbildung befindet sich die Komfortzone des Menschen. Die Komfortzone setzt sich aus der menschlichen Armstellung und der potentiellen Energie für die Erreichung eines Punktes in dieser zusammen. Die Trajektorie des Planers wurde in der Studie mit anderen verglichen und von den Testpersonen nach allen drei Kriterien am besten bewertet.

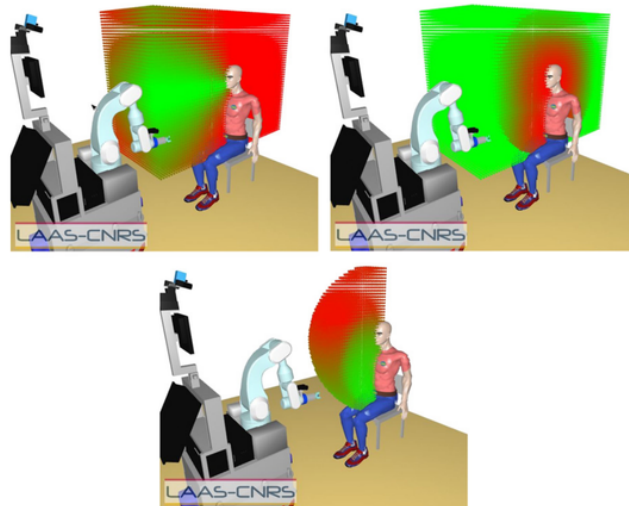


Abbildung 1.2: Oben Links ist die Sichtbarkeit des Menschen illustriert. Das Sicherheitskriterium wird oben rechts durch einen schützenden Bereich um den Menschen beachtet. In der unteren Abbildung wird die Komfortzone des Menschen dargestellt. Abbildung entnommen aus [Dehais et al. \[2011\]](#)

In der Studie von [Cakmak et al., 2011 \[b\]](#)) wurde der Zeitpunkt der Objektübergabe durch den Einsatz von Trajektorien mit räumlichen und zeitlichen Kontrast untersucht. Menschen halten Objekte in der Nähe des Torsos und signalisieren durch das Ausstrecken des Objektes zum Objektnehmer die Bereitschaft zur Objektübergabe. Der Objektnehmer hat dadurch ausreichend Zeit um den Übergabewunsch zu erkennen und den Verlauf der Trajektorie schätzen zu können. Die Autoren haben vier Arten von Trajektorien untersucht, welche in [Abbildung 1.3](#) für alle möglichen vier Fälle von räumlichen und zeitlichen Kontrast dargestellt sind. In der Studie konnten für Trajektorien mit und ohne räumlichen Kontrast keine Unterschiede beobachtet werden. Dagegen wurde bei Trajektorien mit zeitlichem Kontrast gegenüber Trajektorien ohne erheblich kürzere Wartezeiten vom Objektnehmer vor dem Greifvorgang beobachtet, weil dieser den Objektübernahmezeitpunkt planen konnte. Außerdem wurden von den Testpersonen keine Versuche unternommen, das Objekt vorzeitig zu greifen, wie es bei den Trajektorien ohne zeitlichen Kontrast auftrat. Insgesamt führen Trajektorien mit zeitlichen Kontrast zu einer flüssigeren Objektübergabe zwischen Mensch und Roboter. Die Autoren empfehlen das Objekt des Roboters solange in der Tragestellung zu behalten bis die Bereitschaft zur Objektübergabe vorliegt, um anschließend in die Objektübergabestellung zu fahren.

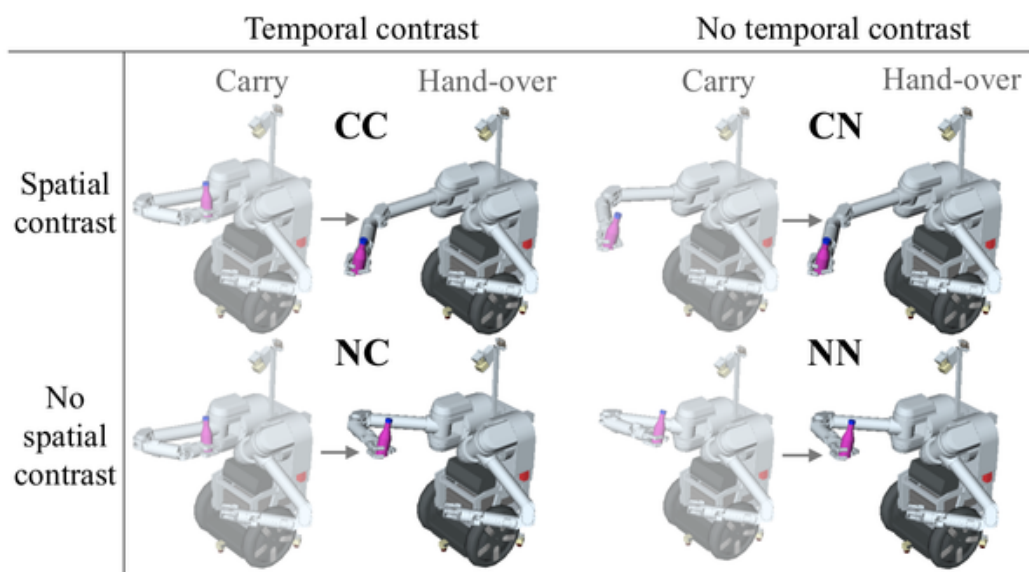


Abbildung 1.3: Dargestellt sind die vier Trage- und Übergabestellungen zur Untersuchung des räumlichen und zeitlichen Kontrast. Abbildung entnommen aus [Cakmak et al., 2011 \[b\]](#)

2 Theoretische Grundlagen

In diesem Kapitel werden die Grundlagen für das bessere Verständnis dieser Arbeit beschrieben. Zunächst wird der Aufbau des Industrieroboters betrachtet. Anschließend werden die Grundlagen der Kinematik, Dynamik und Bewegungsplanung für diesen behandelt. Weiterhin wird das Tiefenmessverfahren der eingesetzten Microsoft Kinect näher erläutert. Anschließend werden Greifer und Roboterhände auf ihre Eignung für die Mensch-Roboter-Interaktion beschrieben, sowie die Objektübergabe in der Mensch-Mensch-Interaktion untersucht, um daraus Entwurfskriterien für das Verhalten des Roboters Schlussfolgern zu können. Abschließend werden mögliche Sicherheitsmaßnahmen für eine sichere Mensch-Roboter-Interaktion behandelt.

2.1 Industrieroboter

Der Verein Deutscher Ingenieure definiert in der Richtlinie 2860 den Industrieroboter als "universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkel frei (d. h. ohne mechanischen Eingriff) programmierbar und ggf. sensorgeführt sind. Sie sind mit Greifern, Werkzeugen oder anderen Fertigungsmitteln ausrüstbar und können Handhabungs- und/oder Fertigungsaufgaben ausführen." [VDI2860 \[2014\]](#)

Die Bezeichnung Industrieroboter wird in dieser Arbeit synonym mit Manipulator und Roboterarm verwendet. Ein Manipulator kann als eine kinematische Kette betrachtet werden, weil alle seine Gelenke miteinander durch starre Glieder verbunden sind. In [Abbildung 2.1](#) ist der Industrieroboter KUKA Lightweight Robot (LWR) und seine kinematische Kette illustriert. Ein Ende der kinematischen Kette ist an einem fixen Ort montiert, welcher als die Basis des Manipulators bezeichnet wird. Am anderen Ende befindet sich der Endeffektor auf dem meistens Werkzeuge oder Greifer befestigt werden, auf diesen oder in ihrer Nähe wird wiederum ein spezieller Punkt mit der Bezeichnung Tool Center Point (TCP) definiert. Mit dem TCP wird der gewünschte Zielort des Manipulators relativ zum benutzerspezifischen Koordinatensystem definiert. Der TCP wird somit für die Beschreibung einer Manipulatorbewegung verwendet [[Craig 2005](#), S. 14].

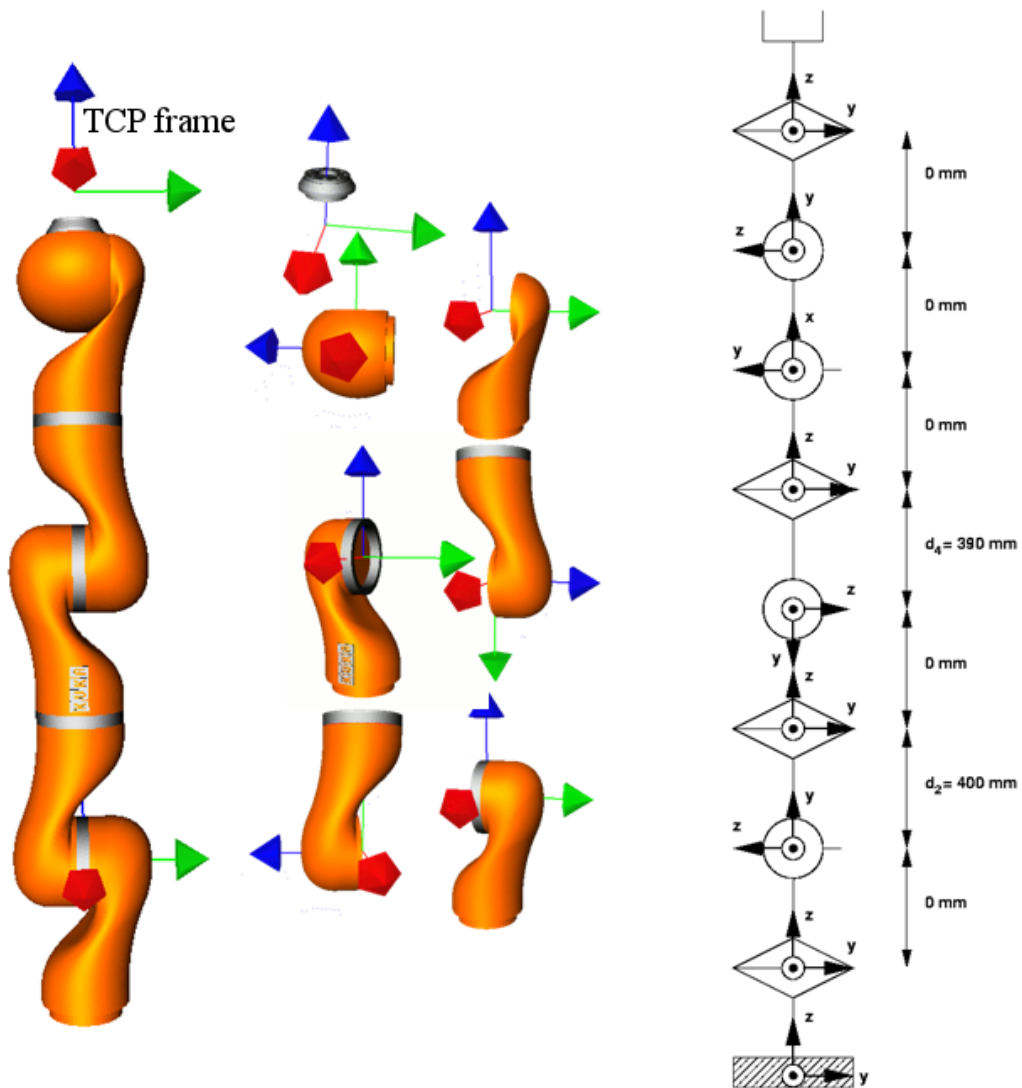


Abbildung 2.1: Visualisiert ist auf der linken Seite die Position und Richtung der Gelenkkoordinatensysteme des KUKA Lightweight Robot (LWR). Auf der rechten Seite befindet sich die zugehörige kinematische Kette. Abbildung entnommen aus [Zacharias 2012, S. 134].

Die am häufigsten vorkommenden Gelenkarten eines Roboterarmes sind die Schub- und Drehgelenke. Beide Gelenkarten verfügen jeweils nur über einen Freiheitsgrad, wodurch die Anzahl der Gelenke meistens mit der Anzahl der Freiheitsgrade eines Manipulators übereinstimmt. Die Anzahl der Freiheitsgrade wiederum ist die Anzahl der unabhängigen Parameter, welche spezifiziert werden müssen, um eine Manipulatorkonfiguration vollständig beschreiben zu können [Craig 2005, S. 5]. Ein Manipulator benötigt mindestens sechs Freiheitsgrade um einen beliebigen Punkt mit einer beliebigen Orientierung im Raum zu erreichen. Verfügt der Manipulator über mehr als sechs Freiheitsgrade, kann dieselbe Pose mit mehreren Gelenkstellungen erreicht werden. Einerseits wird dadurch der Manipulator robuster gegenüber Singularitäten eines sechs gelenkigen Manipulators. Hierbei ist mit einer Singularität eine spezielle Konfigurationen des Manipulators gemeint, in welcher dieser Freiheitsgrade verliert und somit nicht mehr jede Pose im Raum erreichen kann. Andererseits erhöht sich vergleichsweise die Komplexität eines Manipulators mit mehr als sechs Freiheitsgraden. Dadurch kann eine Teilmenge der bekannten Bewegungsplanungsalgorithmen durch die größeren Anforderungen nicht mehr gelöst werden, womit die Entwicklung neuer Bewegungsplanungsalgorithmen erforderlich wird.

2.2 Kinematik

John J. Craig definiert Kinematik als die Bewegungslehre, welche die Bewegung ohne die Kräfte die diese Verursachen betrachtet [Craig 2005, S. 4]. In diesem Abschnitt werden die Probleme und Lösungswege der Vorwärtskinematik und Rückwärtskinematik vorgestellt.

Die Pose (Position und Orientierung) des TCP kann im kartesischen Raum und im Gelenkwinkelraum beschrieben werden. Die Pose im kartesischen Raum wird durch ein kartesisches Koordinatensystem in Raum und im Gelenkwinkelraum durch die Gelenkwerte aller Gelenke beschrieben. In der Robotik gehört die Transformation zwischen dem kartesischen Raum und dem Gelenkwinkelraum zu zwei bekannten Problemen mit der Bezeichnung Vorwärtskinematik und Rückwärtskinematik. Die Vorwärtskinematik beschreibt das Problem, bei welchem die Gelenkwerte des Manipulators bekannt sind und die Pose des TCP ermitteln werden soll. Für die Lösung muss die Pose vom Gelenkwinkelraum zum kartesischen Raum transformiert werden. Die Rückwärtskinematik ist wiederum die Rückwärtstransformation vom kartesischen Raum zum Gelenkwinkelraum, bei welcher die Pose des TCP gegeben ist und die Gelenkstellungen berechnet werden sollen.

Für die Lösung der Vorwärtskinematik und Rückwärtskinematik müssen Transformationen zwischen den kartesischen Koordinatensystemen entlang der kinemati-

schen Kette stattfinden. Die Überführung eines kartesischen Koordinatensystems in ein anderes kann durch eine Translationsoperation und drei Rotationsoperationen erfolgen. Eine Translation ist die Verschiebung des Koordinatenursprungs im Raum, welche mathematisch gesehen als eine Punktverschiebung mit einem $(x, y, z)^T$ Vektor realisiert werden kann. Rotationen um einzelne Achsen eines Koordinatensystems lassen sich durch die folgenden Rotationsmatrizen realisieren.

$$R(x, \gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \quad (2.1)$$

$$R(y, \beta) = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.2)$$

$$R(z, \alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Hierbei sind der Roll Angle (γ), Pitch Angle (β) und Yaw Angle (α) die Rotationen um die x-Achse, y-Achse und z-Achse. Eine Verkettung der einzelnen Rotationsmatrizen erfolgt durch ihre Multiplikation, wobei beachtet werden muss, dass die Matrixmultiplikation nicht kommutativ ist. Im Bereich der Robotik hat sich die X-Y-Z Fixed-Angles (Roll, Pitch, Yaw Angles) Notation durchgesetzt, welche die Verkettung der Rotationsoperationen um ein festes Referenzkoordinatensystem in der Reihenfolge $R_{XYZ}(\gamma, \beta, \alpha) = R(z, \alpha) R(y, \beta) R(x, \gamma)$ vorsieht. Durch Linksmultiplikation der einzelnen Rotationsmatrizen ergibt sich schließlich eine einzige Rotationsmatrix, welche für die kartesische Koordinatensystemtransformation verwendet werden kann. Der Lesbarkeit halber steht in der Rotationsmatrix c für cos und s für sin.

$$R_{XYZ}(\gamma, \beta, \alpha) = \begin{bmatrix} c(\alpha)c(\beta) & c(\alpha)s(\beta)s(\gamma) - s(\alpha)c(\gamma) & c(\alpha)s(\beta)s(\gamma) + s(\alpha)c(\gamma) \\ s(\alpha)c(\beta) & s(\alpha)s(\beta)s(\gamma) + c(\alpha)c(\gamma) & s(\alpha)s(\beta)c(\gamma) - c(\alpha)s(\gamma) \\ -s(\beta) & c(\beta)s(\gamma) & c(\beta)c(\gamma) \end{bmatrix} \quad (2.4)$$

Der oben beschriebene 3x1 Translationsvektor und die 3x3 Rotationsmatrix

werden mit einem 1×3 Projektionsvektor und einem Skalierungsfaktor in einer 4×4 Matrix zusammengefasst, welche als die homogene Transformationsmatrix bezeichnet wird.

$$H = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

In der Robotik ist der Skalierungsfaktor immer eins und der Projektionsvektor ein Nullvektor. Mit der homogenen Transformationsmatrix wird für eine Koordinatensystemtransformation nur eine einzige Matrixmultiplikation benötigt. Mit der Inversen der homogenen Transformationsmatrix kann die umgekehrte Koordinatensystemtransformation bestimmt werden. Diese setzt sich mit den Spaltenvektoren r_1, r_2, r_3 und $p = (t_x, t_y, t_z)^T$ der homogenen Transformationsmatrix folgendermaßen zusammen.

$$H^{-1} = \begin{bmatrix} r_{11} & r_{21} & r_{31} & -p * r_1 \\ r_{12} & r_{22} & r_{32} & -p * r_2 \\ r_{13} & r_{23} & r_{33} & -p * r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Die Denavit-Hartenberg (DH) Konvention beschreibt die Zuordnung von Referenzkoordinatensystemen zu den Gelenken eines Manipulators. Die Konvention wird von Industrieroboterentwicklern berücksichtigt, um die Existenz einer Lösung für die Vorwärtskinematik garantieren zu können. Abbildung 2.2 illustriert die platzierten Referenzkoordinatensysteme durch die Anwendung der folgenden drei Regeln der DH Konvention [Craig 2005, S. 68].

- Die z -Achse des kartesischen Koordinatensystems i ist übereinstimmend mit der Gelenkachse i .
- Für den Fall $a_i = 0$ ist die x -Achse des kartesischen Koordinatensystems i parallel zu dem Kreuzprodukt: $x_i = z_i * z_{i-1}$. Für den Fall $a_i \neq 0$ liegt die x -Achse des kartesischen Koordinatensystems i auf a_i und zeigt vom Gelenk i in Richtung des Gelenkes $i+1$.
- Die y -Achse des kartesischen Koordinatensystems i ergibt sich aus dem rechtehändigen Koordinatensystem.

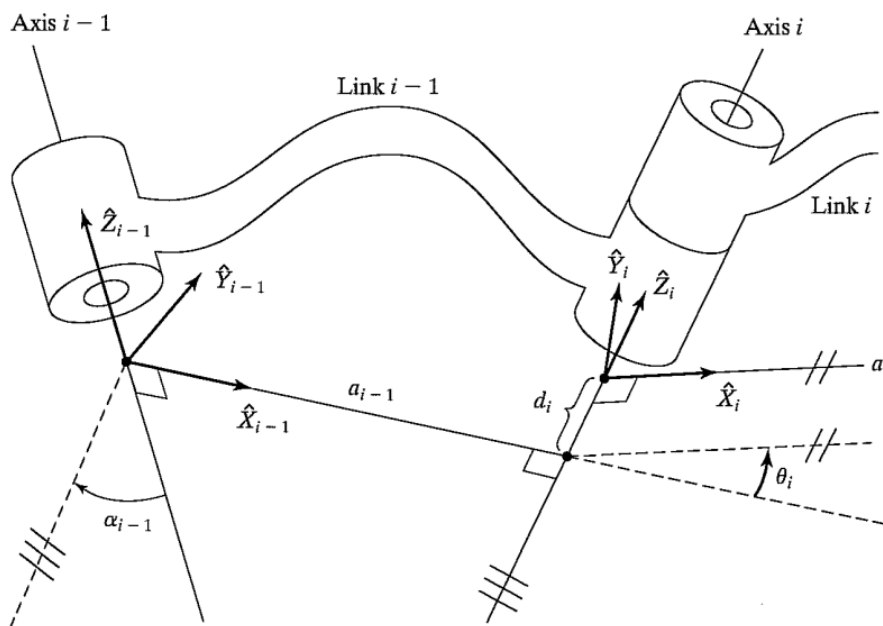


Abbildung 2.2: Abgebildet sind die DH Parameter und die platzierten Referenzkoordinatensysteme der Gelenke durch Anwendung der DH Konvention. Abbildung entnommen aus [Craig 2005, S. 66]

Durch Anwendung der DH Konvention ergibt sich allerdings keine eindeutige Zuordnung der Referenzkoordinatensysteme zu den Gelenken. Beispielsweise bestehen zwei Möglichkeiten die z-Achse bei der allerersten Zuordnung zu dem Gelenk i zuzuweisen [Craig 2005, S. 69]. Bei Einhaltung der DH Konvention werden nur die vier Parameter θ_i , d_i , a_i und α_i für eine Koordinatensystemtransformation benötigt, welche als die DH Parameter bezeichnet werden.

- Der Parameter a_i ist die Entfernung von Z_i bis Z_{i+1} entlang X_i gemessen.
- Der Parameter α_i ist der Winkel von Z_i bis Z_{i+1} um X_i gemessen.
- Der Parameter d_i ist die Entfernung von X_{i-1} bis X_i entlang Z_i gemessen.
- Der Parameter θ_i ist der Winkel von X_{i-1} bis X_i um Z_i gemessen.

Die DH Parameter werden durch die folgenden vier homogenen Matrizen beschrieben.

$$R(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

$$T(d_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

$$T(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

$$R(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

Durch die Multiplikation der vier Matrizen in der Reihenfolge $R(\theta_i)*T(d_i)*T(a_i)*R(\alpha_i)$ ergibt sich die allgemeine Transformationsmatrix ${}_{i-1}T^i$, welche für jede Transformation vom Koordinatensystem $i-1$ nach i in der Kinematischen Kette eingesetzt werden kann.

$${}_{i-1}T^i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

Für einen 6 gelenkigen Manipulator ergibt sich durch Multiplikation der Transformationsmatrix für jedes Gelenk eine Matrix ${}^0T^6 = {}^0T^1*{}_1T^2*{}_2T^3*{}_3T^4*{}_4T^5*{}_5T^6$. Diese hat 12 unbekannte Variablen und kann mit 12 linearen Gleichungen gelöst werden. Die benötigten DH Parameter des Manipulators werden vom Roboterhersteller bereitgestellt. In der Tabelle 2.2 sind beispielsweise die DH Parameter des KUKA LWR dargestellt. Auffällig ist die Häufigkeit der Werte 0° , -90° und 90° für die meisten Parameter, welche von den Entwicklern gewählt werden, um die Komplexität der zu lösenden Gleichungen erheblich zu vereinfachen.

Der beschriebene Lösungsweg für die Vorwärtskinematik wird momentan in der Robotik als das Standardlösungsverfahren betrachtet. Im Gegensatz dazu gibt es mehrere Lösungsmöglichkeiten für die Berechnung der Rückwärtskinematik. Im Weiteren

i	α_{i-1}	a_{i-1}	d_i	θ_i
0	0	0	0	0
1	0	90	0	0
2	0	-90	0.4	0
3	0	-90	0	0
4	0	90	0.39	0
5	0	90	0	0
6	0	-90	0	0

Tabelle 2.1: Dargestellt sind die DH Parameter für den KUKA LWR. Die Einheiten von α_{i-1} und θ_i sind in Grad und für a_{i-1} und d_i in Meter angegeben [Zacharias 2012, S. 132].

wird die algebraische Methode in geschlossener Form vorgestellt, wobei mit geschlossener Form eine Lösungsmöglichkeit gemeint ist, welche auf analytischen Ausdrücken basiert oder eine Lösung vom Polynom kleiner fünften Grades ist [Craig 2005, S. 106]. Das Lösungsverfahren berechnet die Rückwärtskinematik mit der inversen Transformationsmatrix ${}^i T^{i-1}$, welche durch die Multiplikation der oben beschriebenen vier DH Matrizen in umgekehrter Reihenfolge $R(\alpha_i) * T(a_i) * T(d_i) * R(\theta_i)$ berechnet werden kann.

$${}^{i-1} T_i^{-1} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) & 0 & -a_i \\ -\sin(\theta_i)\cos(\alpha_i) & \cos(\theta_i)\cos(\alpha_i) & \sin(\alpha_i) & -d_i\sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & -\cos(\theta_i)\sin(\alpha_i) & \cos(\alpha_i) & -d_i\cos(\alpha_i) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Mit der Transformationsmatrix ${}^i T^{i-1}$ kann eine einzige Matrix zur Lösung der Rückwärtskinematik für beispielsweise einen 6 gelenkigen Manipulator berechnet werden ${}^6 T^0 = {}^6 T^5 * {}^5 T^4 * {}^4 T^3 * {}^3 T^2 * {}^2 T^1 * {}^1 T^0$. Diese enthält 12 nicht lineare Gleichungen, von denen jedoch nur 3 Translationsgleichungen und 3 von 9 Rotationsgleichungen unabhängig sind. Dadurch müssen für die Berechnung der Rückwärtskinematik sechs nicht lineare Gleichungen mit sechs unbekanntem Variable (Gelenkwerte) gelöst werden [Craig 2005, S. 102]. Dies kann sehr schwer sein, weil das Lösen von nicht linearen Gleichungen zu keinen Lösungen oder zu multiplen Lösungen führen kann. Für die algebraischen Lösungsmethode in geschlossener Form kann jedoch in speziellen Fällen eine Lösung garantiert werden. Dafür muss der Manipulator drei benachbarte Gelenke haben die sich in einem Punkt schneiden oder drei Gelenke müssen parallel zueinander sein. Deshalb Erfüllen die meisten Manipulatoren mindestens eine der beiden Kriterien, um eine Lösung gewährleisten

zu können [Möller 2007, S. 74]. Falls mehrere Lösungen vorliegen, stellt sich die Frage, welches die beste Lösung ist? Die beste Lösung ist meistens die kürzeste, weil diese zu den geringsten Gelenkbewegungen führt. Allerdings kann die Lösung manchmal zu Manipulatorstellungen führen, welche sich im Kollisionszustand befinden. Multiple Lösungen bieten auf Kosten von Berechnungszeit den Vorteil, dass auf alternative kollisionsfreie Lösung zurückgegriffen werden können.

2.3 Dynamik

Die Kinetik ist ein Teilgebiet der Dynamik und befasst sich mit den Zusammenhängen zwischen Bewegungen und Kräften. In diesem Abschnitt wird die enorme Bedeutung der Jacobi-Matrix und der allgemeinen Bewegungsgleichung für die Bewegungsplanung gezeigt.

Die Jacobi-Matrix bzw. Ableitungsmatrix ist eine $m \times n$ Matrix mit Ableitungen erster Ordnung einer vektorbasierten Funktion $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$. Für $y = F(x)$ wird die Jacobi-Matrix folgendermaßen definiert, wobei m die Anzahl der kartesischen Freiheitsgrade und n die Gelenkzahl des Manipulators ist.

$$J = \frac{\delta F}{\delta x} = \begin{bmatrix} \frac{\delta y_1}{\delta x_1} & \cdots & \frac{\delta y_1}{\delta x_n} \\ \vdots & \ddots & \vdots \\ \frac{\delta y_m}{\delta x_1} & \cdots & \frac{\delta y_m}{\delta x_n} \end{bmatrix} \quad (2.13)$$

Die Gelenkwerte und Gelenkgeschwindigkeiten können in die kartesische Geschwindigkeit am Endeffektor umgerechnet werden, welche sich im Raum aus drei translatorischen und drei rotatorischen Komponenten zu dem Vektor $v = (v_x, v_y, v_z, w_x, w_y, w_z) \in \mathbb{R}^6$ zusammensetzt. Mit der Jacobi-Matrix lässt sich dieser berechnen, wobei q der Positionsvektor und \dot{q} der Geschwindigkeitsvektor ist.

$$v = J(q)\dot{q} \quad (2.14)$$

Die Berechnung der notwendigen Gelenkgeschwindigkeiten für die Erreichung einer bestimmten kartesischen Geschwindigkeit am Endeffektor wird für die Bewegungsplanung benötigt.

$$\dot{q} = J(q)^{-1}v \quad (2.15)$$

Weiterhin können Geschwindigkeitsänderungen nur durch Krafteinfluss bewirkt werden. Weshalb für die Bewegungsplanung der Zusammenhang zwischen den Gelenkkräften und den Kräften und Drehmomenten am Endeffektor von großer Bedeutung ist. Letztere sind mit dem Vektor $g = (f_x, f_y, f_z, m_x, m_y, m_z) \in \mathbb{R}^6$ definiert. Falls g sich im Weltkoordinatensystem befindet, kann mit der transponierten Jacobi-Matrix der generalisierte Gelenkkräftevektor Q berechnet werden.

$$Q = {}^0J(q)^{T0}g \quad (2.16)$$

Sobald ein Werkzeug oder Greifer bzw. Roboterhand am Manipulator montiert ist, wird ein TCP definiert. Daraufhin werden Messung der Kräfte und Drehmomente eher am TCP statt am Endeffektor benötigt. Mit der transponierten Jacobi-Matrix kann der Vektor g zwischen zwei kartesischen Koordinatensystemen der kinematischen Kette berechnet werden. Angenommen die kartesische Koordinatensystemtransformation AT_B erfolgt von A nach B.

$${}^AT_B = \begin{bmatrix} {}^AR_B & {}^At_B \\ 0 & 1 \end{bmatrix} \quad (2.17)$$

Dann lässt sich mit der transponierten Jacobi-Matrix der Vektor g vom Koordinatensystem B für das Koordinatensystem A berechnen.

$${}^Ag = ({}^BJ_A)^{TB}g \quad (2.18)$$

$${}^BJ_A = J_v({}^AT_B) = \begin{bmatrix} {}^BR_A & -{}^BR_AS({}^At_B) \\ 0_{3 \times 3} & {}^BR_A \end{bmatrix} \quad (2.19)$$

Jedes starre Glied in der kinematischen Kette erfährt eine Reaktionskraft und ein Reaktionsdrehmoment von dem vorherigen Glied, und ist abhängig vom

Eigengewicht, der Reaktionskraft und dem Reaktionsdrehmoment des nachfolgenden starren Gliedes. Mit der folgenden allgemeinen Bewegungsgleichung können die notwendigen Gelenkkräfte und Gelenkdrehmomente mit den bekannten Positions-, Geschwindigkeits- und Beschleunigungswerten für die Erreichung einer bestimmten Manipulatorstellung berechnet werden.

$$Q = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + J(q)^T g \quad (2.20)$$

q: Positionsvektor

\dot{q} : Geschwindigkeitsvektor

\ddot{q} : Beschleunigungsvektor

M: Trägheitsmatrix im Gelenkwinkelraum

C: Coriolis und zentripetale Kopplungsmatrix

F: Reibungskraft

G: Gravitationskräfte

Q: generalisierte Antriebskräfte mit den Gelenkkoordinaten q

J: Jacobi Matrix des Manipulators

g: $(f_x, f_y, f_z, m_x, m_y, m_z) \in \mathbb{R}^6$ Kräfte und Drehmomente am Endeffektor.

Mit der Gleichung kann unter anderem der Krafteinfluss durch die Bewegung eines Gelenkes auf die übrigen Gelenke beschrieben werden, sowie die Wirkung der Nutzlast auf den Manipulator. Die allgemeine Bewegungsgleichung kann mit dem rekursiven Newton-Euler Algorithmus sehr effizient gelöst werden, welcher jedoch nicht weiter in dieser Arbeit beschrieben wird [[Corke 2011](#), S. 191].

2.4 Bewegungsplanung

Mit der Lösung der Vorwärts- u. Rückwärtskinematik kann der Startzustand bzw. Endzustand des Manipulators im Gelenkwinkelraum und im kartesischen Raum berechnet werden. Die Berechnung der Konfigurationen zwischen der Startkonfiguration (q_{init}) und der Endkonfiguration (q_{goal}) wird als Bewegungsplanung bezeichnet. Hierbei ist mit Konfiguration die vollständige Spezifikation der Gelenkwerte eines Manipulators gemeint. Die Menge aller möglichen Konfigurationen wird als Konfigurationsraum (C-space) bezeichnet. Eine Konfiguration kann sich in einer Selbstkollision oder in einer Kollision mit einem Hindernis aus der Umgebung befinden, weshalb der C-space weiter unterteilt wird in kollisionsfreien Raum (C-free) und Kollisionsraum (C-collision). Eine Konfiguration wird in C-space als ein Punkt dar-

gestellt. Für die Erreichung des q_{goal} von q_{init} wird ein Bewegungsplaner benötigt, welcher in C-space nach einer kollisionsfreien kontinuierlichen Bahn sucht.

Die Bahnplanung kann mit der randomisierten Datenstruktur Rapidly-exploring Random Tree (RRT) berechnet werden. Hinter RRT verbirgt sich die einfache Idee einen Baum zu erstellen und mit zufälligen Konfigurationen aus C-free zu erweitern, bis eine Bahn zu q_{goal} entdeckt wird. Der RRT Algorithmus ist in Algorithmus 1 dargestellt. Anfangs erzeugt der RRT Algorithmus einen leeren Baum mit q_{init} als Wurzel. Anschließend wählt der Algorithmus in jedem Iterationsvorgang eine zufällige Konfiguration aus und versucht diese mit der Extend Funktion zu dem erzeugten Baum hinzuzufügen. Die Extend Funktion ist in Abbildung 2.3 illustriert. Im ersten Schritt wird mit einem Nächste-Nachbar-Algorithmus der am nächsten liegende Knoten im Baum (q_{near}) zum zufällig erzeugten Knoten (q) bestimmt. Danach wird die NEW_CONFIG Funktion aufgerufen, welche eine inkrementelle Bewegung zu q mit einer festen Distanz ϵ unternimmt und dabei auf Kollisionen prüft. Hierbei können die folgenden drei Situationen entstehen. Falls eine Kollision entdeckt wird, bricht die NEW_CONFIG Funktion ab und die nächste Schleifeniteration beginnt. Falls q sich innerhalb von ϵ befindet, wird q direkt zum RRT hinzugefügt. Ansonsten wird ein neuer Knoten q_{new} mit dem festen Abstand ϵ von q_{near} eingefügt [Kuffner & Lavelle 2010, Seite 2].

Algorithm 1 RRT Algorithmus

```

1: procedure BUILD_RRT( $q_{init}$ )
2:    $\tau$ .init( $q_{init}$ );
3:   for  $k = 1$  to  $K$  do
4:      $q_{rand} \leftarrow$  RANDOM_CONFIG();
5:   return  $\tau$ 

1: procedure EXTEND( $\tau, q$ )
2:    $q_{near} \leftarrow$  NEAREST_NEIGHBOR( $q, \tau$ );
3:   if NEW_CONFIG( $q, q_{near}, q_{new}$ ) then
4:      $\tau$ .add_vertex( $q_{new}$ );
5:      $\tau$ .add_edge( $q_{near}, q_{new}$ );
6:     if  $q_{new} = q$  then
7:       return Reached;
8:     else
9:       return Advanced;
10:  return Trapped;

```

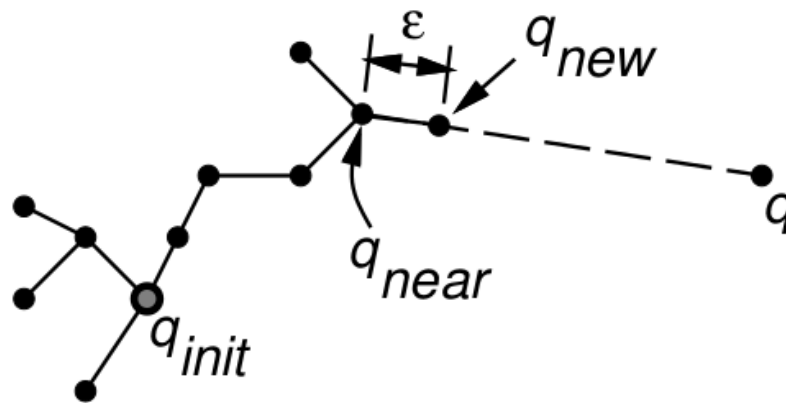


Abbildung 2.3: Die EXTEND Operation. Abbildung entnommen aus [Kuffner & Lavalle 2010, S. 2].

Der RRT Algorithmus verfügt über eine schnelle Explorations Eigenschaft, welche in Abbildung 2.4 illustriert ist. Die obere Zeile zeigt den Wachstumsprozess des RRT an und die untere Zeile das dazugehörige Voronoi Diagramm. Auffällig ist, dass der RRT in Richtung der freien Fläche des dazugehörigen Voronoi Diagramms wächst und dadurch ein schnelles Explorationsverhalten aufzeigt. Die Wahrscheinlichkeit das ein neuer Knoten an einer möglichst unentdeckten Stelle zum RRT hinzugefügt wird, ist proportional zu der Fläche der Voronoi Region [Kuffner & Lavalle 2010, Seite 3].

Eine Menge entwickelter Bahnplanungsalgorithmen basiert auf dem RRT Algorithmus, wie etwa RRT-Connect. RRT-Connect ist ein randomisierter Bahnplanungsalgorithmus der single-query planning problems löst, welche sehr schnell und ohne Vorverarbeitung berechnet werden müssen. Weiterhin eignet sich der Algorithmus für Manipulatoren mit einer hohen Anzahl an Freiheitsgraden. Nach Kuffner und Lavalle ist der RRT-Connect Algorithmus probabilistisch vollständig und die Knoten konvergieren zu einer gleichmäßigen Verteilung in C-free [Kuffner & Lavalle 2010, Seite 4]. Probabilistisch vollständige Algorithmen haben die Eigenschaft, dass bei Existenz einer Lösung, die Wahrscheinlichkeit eine Lösung zu finden gegen 1 konvergiert, falls die Anzahl der Stichproben gegen unendlich ansteigt. Folglich kann der RRT-Connect Algorithmus Probleme ohne Lösung nicht identifizieren [OMPL 2011, S. 4].

Der RRT-Connect Algorithmus ist in Algorithmus 2 dargestellt und verbessert den RRT Algorithmus in zwei Punkten. Als erste Verbesserung werden zwei unabhängige RRTs von den Konfigurationen q_{init} und q_{goal} aufgebaut und anschließend miteinander verbunden. Am Anfang initialisiert der RRT_CONNECT_PLANNER die Wurzel des Baumes T_a mit q_{init} und die Wurzel des Baumes T_b mit q_{goal} . Bei je-

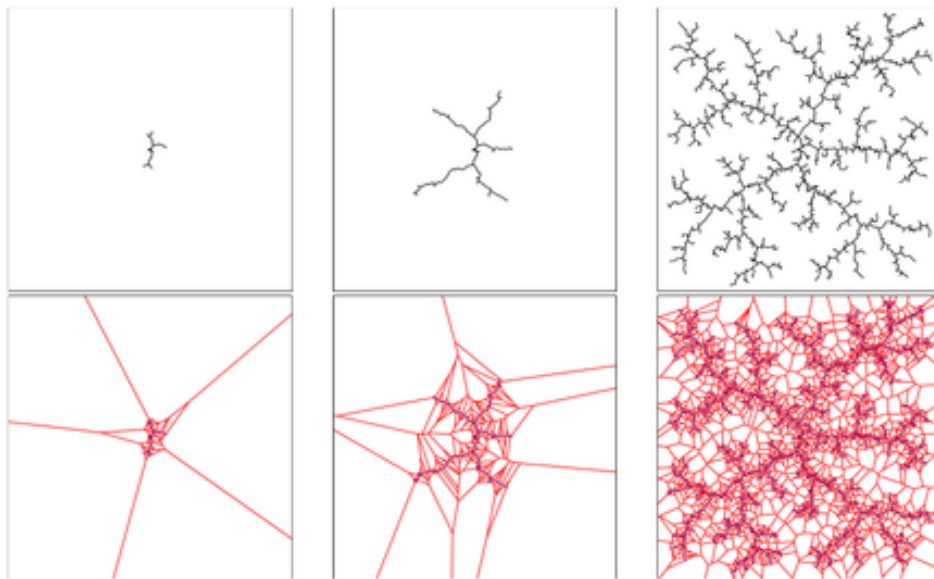


Abbildung 2.4: Die obere Zeile zeigt den Wachstumsprozess eines RRT an und die untere Zeile das dazugehörige Voronoi Diagramm. Abbildung entnommen aus [Kuffner & Lavelle 2010, S. 3].

dem Schleifendurchlauf versucht ein Baum mit der EXTEND Funktion zu wachsen. Bei Erfolg wird q_{new} bzw. q hinzugefügt und der andere Baum versucht mit der CONNECT Funktion seinen nächsten Knoten mit q_{new} bzw. q zu verbinden. Falls auch dieser Vorgang erfolgreich verläuft, wurde eine Bahn gefunden. Tritt jedoch ein Misserfolg während eines Schleifendurchlaufs auf, werden die Rollen von T_a und T_b getauscht und die nächste Schleifeniteration beginnt. Die zweite Verbesserung ist die Verwendung einer einfachen greedy Heuristik, welche in der CONNECT Funktion implementiert ist. Die CONNECT Funktion ruft die EXTEND Funktion solange auf bis q oder ein Hindernis erreicht werden. Dadurch wird der RRT schneller über längere Distanzen aufgebaut als mit der konstanten Distanz ϵ .

Nach Kuffner und LaValle hat der RRT-Connect Algorithmus unter anderem die Vorteile, dass keine Parameter abgestimmt werden müssen, keine Vorverarbeitung notwendig ist und eine Balance zwischen der greedy Suche und der uniformen Exploration besteht [Kuffner & Lavelle 2010, S. 4].

2.5 Greifer und Roboterhände

In diesem Abschnitt werden die Eigenschaften von Greifern und Roboterhänden verglichen, um Präferenzen für die Mensch-Roboter-Interaktion zu bestimmen.

Algorithm 2 RRT-Connect Algorithmus

```

1: procedure CONNECT( $\tau, q$ )
2:   repeat
3:      $S \leftarrow$  EXTEND( $\tau, q$ );
4:   until not ( $S =$  Advanced)
5:   return S;

1: procedure RRT_CONNECT_PLANNER( $q_{init}, q_{goal}$ )
2:    $\tau_a.$ init( $q_{init}$ );  $\tau_b.$ init( $q_{goal}$ );
3:   for  $k = 1$  to  $K$  do
4:      $q_{rand} \leftarrow$  RANDOM_CONFIG();
5:     if not (EXTEND( $\tau_a, q_{rand}$ ) = Trapped) then
6:       if (CONNECT( $\tau_b, q_{new}$ ) = Reached) then
7:         return PATH( $\tau_a, \tau_b$ );
8:     SWAP( $\tau_a, \tau_b$ );
9:   return Failure;

```

Ein Greifer wird an einen Manipulator montiert, um Gegenstände greifen und manipulieren zu können. Der Aufbau des Greifers bestimmt dabei über die möglichen Interaktionen mit den Gegenständen. In industriellen Anwendungen werden meistens kostengünstige Greifer, die an das zu greifende Objekt genau angepasst, sind eingesetzt, weil die Größe, Form und ungefähre Position der Objekte meistens vor dem Greifvorgang bekannt sind. Der Einsatz von Greifern mit wenigen Freiheitsgraden und kaum bzw. keinen Sensoren reicht meistens aus. Sobald allerdings die industriellen Anwendungen verändert werden und das Greifen von neuen bzw. unbekannt Objekten erforderlich machen, werden in der Regel die alten Greifer durch neue speziell an die Aufgabe angepasste Greifer ersetzt. Die Anzahl der möglichen Griffe für das Greifen eines Objektes ist unter anderem von der Anzahl der Finger des Greifers abhängig. Untersuchungen haben gezeigt, dass ungefähr 90 % der vorkommenden Griffe in der Industrie mit drei Fingern ausführbar sind [Hesse 2011, S. 5]. Im Bereich der Serviceroboter versucht man hingegen, auch Objekte zu greifen, über die vorher keinerlei Informationen vorliegen. Aufgrund der Vielfalt an verschiedenen Objekten sind höhere Anforderungen an den Greifer und die Sensorik gestellt. Für die Konstruktion des Greifers orientiert man sich meistens an einem bereits bekannten natürlichem System, der menschlichen Hand. Diese besteht aus vier beweglichen Fingern und einem Daumen mit etwa dem halben Bewegungsvermögen. Der Daumen befindet sich bei jedem Griff in Opposition zu den anderen Fingern und ermöglicht damit viele Griffe. Die menschliche Hand ist ein komplexes Konstrukt mit 22 Freiheitsgraden, welches mit den heutigen technischen Möglichkeiten nicht exakt nachgebildet werden kann, sodass nur vereinfachte menschenähnliche

Hände eingesetzt werden können. In Abbildung 2.5 sind verschiedene Roboterhände illustriert. Von den Sandia National Laboratories wurde die modulare Sandia Hand mit 13 Freiheitsgraden entwickelt und ist für ungefähr 10.000 Dollar erhältlich [Sandia Hand \[2012\]](#). Weiterhin bietet das Unternehmen Elumotion ltd die ELU-1 Hand mit 9 Freiheitsgraden an [ELU-1 Hand \[2010\]](#). Das Harbin Institute of Technology und das Deutsche Zentrum für Luft- und Raumfahrt e. V. haben die vier Finger DLR-Hit Hand entwickelt, welche über jeweils drei Freiheitsgrade pro Finger und vier am Daumen verfügt. Die DLR-Hit Hand I befindet sich mit 50.000 € im mittleren Preissegment [DLR Hand \[2014\]](#), [DLR \[2006\]](#). Einer der führenden Hersteller im oberen Preissegment ist die Shadow Robot Company. Zu ihrem Produktsortiment gehört die Shadow Dexterous Hand C6M mit 20 Freiheitsgraden die 2009 für ungefähr 150.000 € zu erwerben war [Shadow \[2009\]](#).

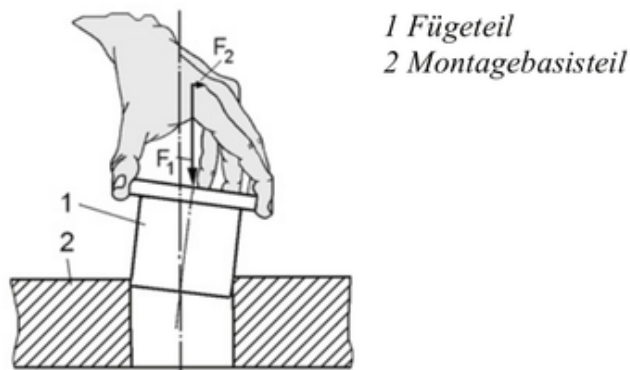
Neben der hohen Anzahl an Freiheitsgraden verfügt die menschliche Hand über eine sensorische Ausstattung mit der Vibration, Tastdruck, Feuchtigkeit und Wärme wahrgenommen werden können. Die sensorischen Fähigkeiten von Roboterhänden sind hingegen ebenfalls durch die technologischen Möglichkeiten stark eingeschränkt. Einige Roboterhände sind mit taktilen Sensoren an den Fingerspitzen ausgestattet und können damit z.B. den ausgeübten Druck auf die gegriffenen Objekte messen. Weiterhin spielt die Nachgiebigkeit bei Einfluss von externen Kräften auf die Roboterhand eine wichtige Rolle. Die Finger der Roboterhand können sehr schnell mit der Umwelt bzw. mit sich selbst in Kollision geraten, wodurch diese ohne Nachgiebigkeit unverzüglich beschädigt werden. Die Nachgiebigkeit bei der menschlichen Hand wird durch die Anspannung der Muskeln reguliert. Bei einer Roboterhand kann die Nachgiebigkeit durch die Reduzierung der Gelenksteifigkeit realisiert werden.

In der Abbildung 2.6 wird Anhand eines Beispiels die Notwendigkeit von sensorischen Fähigkeiten für das Einfügen eines Bolzens in ein Loch illustriert. Beim Versuch den Bolzen in das Loch zu stecken kann es zu einer Verkanntung des Bolzens kommen. In dem Fall lässt sich der Bolzen selbst mit erheblich hohem Kraftaufwand nicht weiter hineinschieben, weil eine Kraft F_2 entsteht, die den Klemmeffekt weiter verstärkt. Das Problem lässt sich ohne sensorische Fähigkeiten nicht lösen und zeigt die Notwendigkeit der Sensorintegration für Roboterhände bzw. Greifer [[Hesse 2011](#), S. 4]. Ein möglicher Lösungsansatz ist die Entwicklung einer künstlichen Haut, die mit einem taktilen Sensorsystem ausgestattet ist. Die künstliche Haut kann auf Roboterhände angebracht werden, um damit eine ortsauflösende Berührungswahrnehmung und Druckverteilung zu erhalten.

In Bezug auf die Objektübergabe zwischen Mensch und Roboter ist eine menschliche Roboterhand einem Greifer vorzuziehen, nicht nur wegen ihrem menschlichen Erscheinungsbild, sondern auch durch die deutlich besseren sensorischen und motor-



Abbildung 2.5: Dargestellt sind von oben links nach unten rechts die ELU-1 Hand, DLR-Hit Hand I, Sandia Hand und die Shadow Dexterous Hand C6M. Abbildung entnommen aus [ELU-1 Hand \[2010\]](#), [DLR Hand \[2014\]](#), [Sandia Hand \[2012\]](#), [Shadow \[2009\]](#).



*1 Fügeteil
2 Montagebasisteil*

Abbildung 2.6: Bolzen-in-Loch-Montage. Abbildung entnommen aus [Hesse \[2011\]](#).

ischen Fähigkeiten. Der Greifer ist hingegen als Schnittstelle zwischen Roboter und Werkzeug zu betrachten, weil dieser z.B. in Bezug auf die Traglast und Robustheit gegenüber einer Roboterhand vorzuziehen ist.

2.6 Tiefenmessung mit Tiefensensoren

Für die Objektmanipulation ist zunächst die Lokalisierung der Objekte notwendig. In diesem Abschnitt werden das Sheet of Light Triangulation Verfahren und das Structured Light Triangulation Verfahren zur Gewinnung von Tiefeninformationen beschrieben. Letzteres wird z.B. von der Microsoft Kinect eingesetzt [FM \[2014\]](#).

Eine Kamera und eine Lichtquelle werden in beiden Verfahren zur Gewinnung der Tiefeninformationen verwendet. In [Abbildung 2.7](#) ist die Idee des Sheet Of Light Triangulation Verfahren illustriert. Eine Lichtquelle projiziert einen Lichtstreifen in die Umgebung, welcher reflektiert von der Kamera empfangen wird. Der Lichtstreifen wird in Abhängigkeit vom Betrachtungswinkel der Kamera und der beleuchteten Oberfläche des Objektes verformt. Mit dem bekannten Abstand zwischen der Lichtquelle und der Kamera, dem Strahlungswinkel der Lichtquelle α und dem Eingangswinkel der Kamera β kann die Distanz zum Objektpunkt mittels Triangulation berechnet werden. Unter Triangulation versteht man die Anwendung des in der [Abbildung 2.8](#) dargestellten optische Triangulationsprinzips, mit welchem sich der Abstand e bestimmen lässt [[Wilke 2002](#), S. 13].

$$e = b * \frac{\tan(\alpha) * \tan(\beta)}{\tan(\alpha) + \tan(\beta)} \quad (2.21)$$

Für die Erstellung eines Tiefenbildes kann beispielsweise der Lichtstrahl einmal über die ganze Szene ablaufen. Die Auswertung von mehreren Kamerabildern ist mit diesem Verfahren erforderlich. Während mit dem Structured Light Triangulation Verfahren nur ein einziges Kamerabild benötigt wird. Das Structured Light Triangulation Verfahren projiziert statt einem einzigen Lichtstrahl ein Streifenmuster oder Punktemuster auf die Szene. Das projizierte Muster wird mit einem bei der Kalibrierung erstellten Referenzbild verglichen, um für jeden Punkt die Verschiebung zu bestimmen und daraus die Tiefeninformationen zu gewinnen. In [Abbildung 2.9](#) ist auf der linken Seite ein Punktemuster projiziert. Auf der rechten Seite wird das Punktemuster durch ein Objekt verzerrt dargestellt.

Das Tiefenbild kann mit beiden Verfahren in bestimmten Regionen Löcher aufweisen, sogenannte Tiefenschatten. Diese entstehen bei Regionen die vom der Kamera gesehen, jedoch nicht von der Lichtquelle getroffen werden, weil ohne das Lichtmuster keine Tiefe berechnet werden kann [[Tirpitz 2012](#), S. 16].

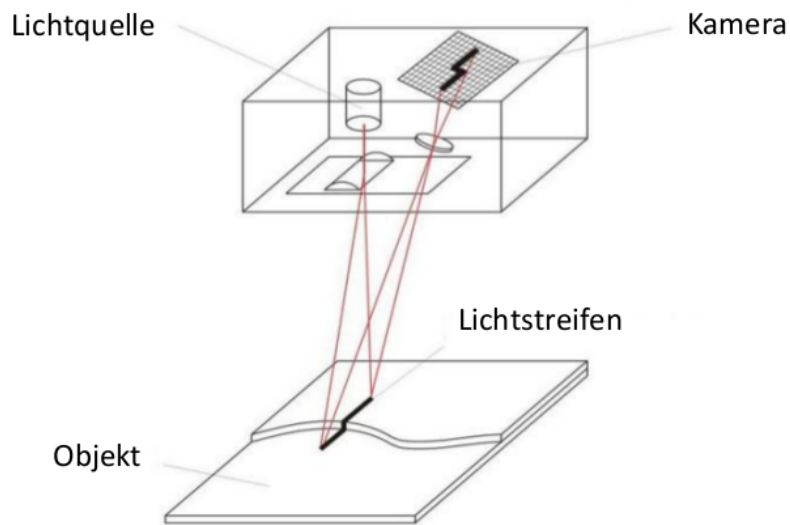


Abbildung 2.7: Ein Lichtstreifen wird auf das zu untersuchende Objekt projiziert. Eine Kamera erfasst die Verformung des Lichtstreifens durch die Objektoberfläche. Abbildung entnommen aus [Ploner \[2000\]](#).

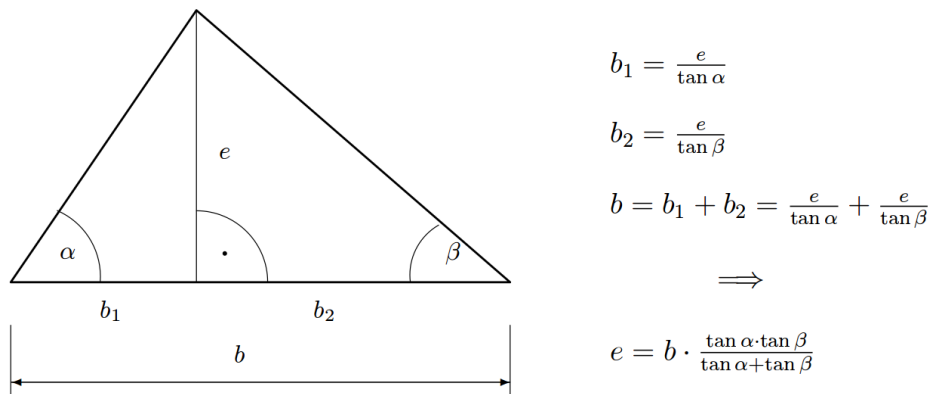


Abbildung 2.8: Skizze und Formel des optischen Triangulationsprinzips. Abbildung entnommen aus [Wilke \[2002\]](#).

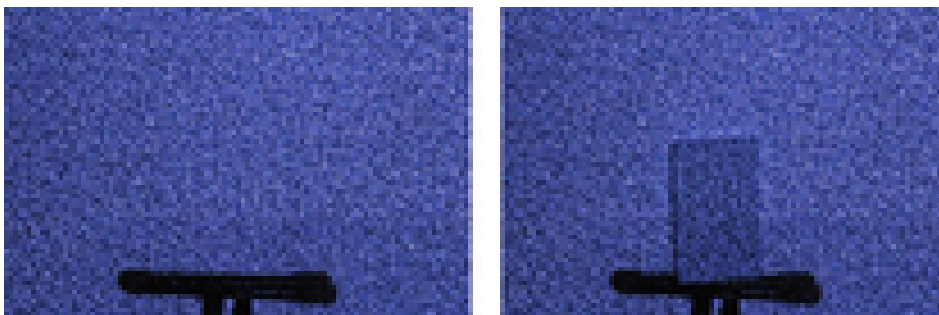


Abbildung 2.9: Auf der linken Seite wird ein Punktemuster in die Szene projiziert. Auf der rechten Seite wird das Punktemuster wegen des Objektes verzerrt dargestellt. Abbildung entnommen aus [Ploner \[2000\]](#).

2.7 Objektübergabe in der Mensch-Mensch-Interaktion

Für die Entwicklung einer natürlich wirkenden Objektübergabe zwischen Menschen und Robotern muss für die Imitation des menschlichen Verhaltens zunächst einmal die Objektübergabe zwischen Menschen betrachtet werden. Menschen verwenden komplexe Sequenzen aus Sprache, Gesten, Blicken, Armbewegungen und der taktilen Wahrnehmung, um den Wunsch bzw. die Bereitschaft für die Objektübergabe zu signalisieren und durchzuführen. Ein Teilproblem der Objektübergabe ist die Koordination zwischen den Akteuren. Bei jeder Objektübergabe gibt es einen Akteur in der Rolle des Objektgebers und einen in der Rolle des Objektnehmers, welche im weiteren als Geber und Nehmer bezeichnet werden. Der Freigabezeitpunkt des Gebers, die Dauer der Objektübergabe und die Greifkraft sind einige beeinflussende Faktoren einer koordinierten Objektübergabe. In der Studie von [Mason & MacKenzie \[2005\]](#) wurden die ausgeübten Greifkräfte während der menschlichen Objektübergabe, sowie ihre Koordination untersucht. Es wurde festgestellt, dass während der Objektübergabe, die Greifkraft des Gebers sinkt und die Greifkraft des Nehmers kontinuierlich steigt. Die durchschnittliche Zeit des Gebers und Nehmers zur Erreichung der Spitzenkraft beträgt 275 ± 9 ms nach dem Kontakt des Gebers mit dem Nehmer. Dies bedeutet wiederum, dass die Objektübergabe nach etwa einer halben Sekunde abgeschlossen ist. Weiterhin sind die ausgeübten Spitzenkräfte des Gebers und Nehmers bei einem stationären Geber in etwa identisch. Während bei einem mobilen Geber die Spitzenkraft des Nehmers wesentlich höher ist als die des Gebers. Überdies wurden die häufigsten beginnenden Freigabezeitpunkte des Gebers für die folgenden vier Fälle des stationären bzw. mobilen Gebers bzw. Nehmers festgehalten.

- stationärer Geber/stationärer Nehmer: Geber beginnt mit der Freigabe 60 bis 120 ms nach Kontakt.
- mobiler Geber/stationärer Nehmer: Geber beginnt mit der Freigabe mindestens 120 ms nach Kontakt.
- stationärer Geber/mobiler Nehmer: Geber beginnt mit der Freigabe 0 bis 60 ms vor Kontakt.
- mobiler Geber/mobiler Nehmer: Geber beginnt mit der Freigabe 0 bis 60 ms nach Kontakt.

Aus den Beobachtungen schlossen die Autoren, dass die Objektübergabe von beiden Akteuren somatosensorisch koordiniert wird und der Geber den Beginn des Freigabezeitpunktes mit seiner visuellen Wahrnehmung ermittelt.

In der Publikation von [Chan et al. \[2012\]](#) wurde die Objektübergabe zwischen Menschen untersucht, um Entwurfsempfehlungen für den Roboter auszuarbeiten. Die Autoren haben festgestellt, dass der menschliche Geber für die Sicherheit bei der Objektübergabe verantwortlich ist und der Nehmer über den Objektübergabezeitpunkt bestimmt. Die Autoren haben eine durchschnittliche Objektübergabezeit von ungefähr 500 ms beobachtet, welche in der Regel linear abhängig von der Belastungskraft ist. Die Abhängigkeit besteht nicht mehr, sobald ein Akteur eine sehr kleine Belastungskraft wahrnimmt. Dieses Verhalten ist ein menschlicher Sicherheitsmechanismus der die Aufmerksamkeit des Gebers bei steigendem Objektgewicht erhöht, damit das Objekt nicht fallen gelassen wird. Weiterhin ist ein erhöhtes Verhältnis zwischen der Greifkraft und der Belastungskraft beim menschlichen Geben zum Ende der Objektübergabe erhöht festgestellt worden, welches ebenfalls auf die Rolle des Gebers zur Wahrung der Sicherheit schließen lässt. Die Autoren empfehlen den Roboter in der Rolle des Gebers auch dann noch eine Greifkraft ausüben zu lassen, selbst wenn die Belastungskraft nahe null liegt. Erst wenn die Belastungskraft negativ wird sollte die vollständige Objektfreigabe erfolgen.

2.8 Sicherheit

Der Arbeitsraum von klassischen Industrierobotern muss grundsätzlich abgezäunt werden, um Menschen nicht in Lebensgefahr zu bringen. Im Bereich der Mensch-Roboter-Interaktion fällt diese Sicherheitsmaßnahme weg, weil der Mensch und Roboter sich denselben Arbeitsbereich teilen sollen. Die Notwendigkeit von neuen Sicherheitsvorkehrungen für die sichere Interaktion zwischen Menschen und Robotern ist erforderlich. In diesem Abschnitt werden einige mögliche Sicherheitsmaßnahmen beschrieben.

Das europäische Projekt PHRIENDS untersuchte die Sicherheit der Mensch-Roboter-Interaktion mit Robotern von einer Traglast zwischen drei und fünfhundert Kilogramm. Das Ergebnis der Studie war, dass keiner der Roboterarme durch direkten Aufprall einen Menschen lebensgefährlich verletzen konnte, solange der Mensch während des Vorgangs nicht eingeschlossen wurde [Bischoff \[2010\]](#). Die wichtigste Sicherheitsvorkehrung ist somit die Platzierung des Manipulators in eine Umgebung, in der kein Einschluss des Menschen möglich ist. Weitere Vorkehrungen sollten zum Reduzieren des Verletzungspotenzials und des Verletzungsrisikos getroffen werden.

Viele Industrieroboterhersteller verwenden leistungsstarke Motoren um Manipulatoren mit einer möglichst hohen Traglast zu bauen, welche allerdings über ein hohes Eigengewicht verfügen. Die Reduzierung des Eigengewichts führt zu einer erheblichen Senkung des Verletzungspotenzials. Weshalb die Verwendung von leichtge-

wichtigen Manipulatoren für eine sichere Mensch-Roboter-Interaktion notwendig ist. Weiterhin sollte mindestens ein griffbereiter Notausschalter installiert sein und dem Menschen für seine Betätigung ausreichend Reaktionszeit gewährt werden. Hierfür muss die Geschwindigkeit und Leistungsfähigkeit besonders bei Manipulatoren aus dem industriellen Einsatz auf ein angemessenes Maß gedrosselt werden. Überdies sollte der Bewegungsbereich des Manipulators durch Markierungen des Arbeitsraumes sichtbar gemacht werden und auf das benötigte Volumen reduziert werden. Die Aufmerksamkeit des Menschen sinkt mit der Zeit selbst in möglicherweise gefährlichen Umgebungen. Die Unterscheidung zwischen gefährlichen und ungefährlichen Zonen erhöht die Sicherheit, weil der geforderte Aufmerksamkeitszeitraum sinkt und dem Menschen die Möglichkeit gewährt wird, von seiner Aufmerksamkeit nur in gefährlichen Zonen Gebrauch zu machen.

Weitere Sicherheitsvorkehrungen können mit dem Einsatz von Sensoren zur Kollisionserkennung getroffen werden. Beispielsweise verfügt der KUKA LWR über Momentensensoren in seinen Gelenken, welche es dem LWR ermöglichen mit einer Nachgiebigkeit betrieben zu werden. Falls der LWR von einer geplanten Bahn durch eine Kollision um mehr als den erlaubten Schwellenwert abweicht, wird dieser unverzüglich zum Stillstand gebracht. Eine weitere Möglichkeit ist die Ausstattung eines beliebigen Manipulators mit taktilen Sensoren an seiner Oberfläche. Das Fraunhofer-Institut für Fabrikbetrieb und -automatisierung (Fraunhofer IFF) hat im Projekt TAKSENS eine künstliche Haut bzw. ein taktilsensitives Sensorsystem auf Manipulatoren angebracht, um damit mögliche Kollisionen zu detektieren [FraunhoferIFF, 2014 \[a\]](#).

Noch wichtiger als die Detektion von eingetretenen Kollisionen für die sichere Mensch-Roboter-Interaktion ist die Detektion von bevorstehenden Kollisionen. Das Fraunhofer IFF hat den mittelschweren Industrieroboter ABB IRB4600 mit kapazitiven Sensoren ausgestattet, um zu demonstrieren, wie menschliche Annäherungen von z.B. einer Hand bereits aus 10 bis 15 cm Entfernung erkannt werden. Damit kann rechtzeitig vor dem Eintreten einer drohenden Kollision ein Abbremsungsvorgang eingeleitet werden [FraunhoferIFF, 2014 \[b\]](#). Überdies kann der Arbeitsraum des Roboters beobachtet werden, um auf bedrohliche Situationen zu reagieren. Das Fraunhofer IFF hat beispielsweise einen drucksensitiven Fußboden zur Arbeitsraumüberwachung entwickelt. Falls der Mensch in den Arbeitsraum des Roboters eindringt, wird die Geschwindigkeit auf ein nicht gefährliches Maß gedrosselt [FraunhoferIFF, 2014 \[c\]](#). Überdies kann der Arbeitsraum mit Kamerasensoren überwacht werden, um die Trajektorie des Manipulators auf mögliche Kollisionen mit dem Menschen zu überprüfen und bei Bedarf abubrechen. Anschließend könnte der Manipulator eine neue Trajektorie planen und den Menschen als Hindernis bei seiner Planung berücksichtigen.

3 Implementierung

In diesem Kapitel werden die einzelnen Schritte für die Entwicklung des Robotersystems beschrieben. Zunächst werden die verwendete Hardware und Software des Robotersystems behandelt. Daraufhin wird die Roboterkonfiguration und Integration von Systemkomponenten mit MoveIt! erläutert. Anschließend wird die Funktionsweise der Objekterkennung mit der GStreamer Software beschrieben. Danach wird die Erweiterung des `wsg50-ros-pkg` um zusätzliche Funktionen und die Entwicklung des `ros_fri` Controllers für die LWR Steuerung behandelt. Im Anschluss daran wird der Zustandsautomat der implementierten Pick & Place Demo vorgestellt. Abschließend werden die während der Arbeit verwendeten Sicherheitsmechanismen mit dem Robotersystem erläutert.

3.1 Verwendete Hardware

In diesem Unterkapitel wird zunächst die Hardware des Robotersystems näher erläutert und anschließend der Gesamtaufbau präsentiert.

3.1.1 Gesamtdarstellung des Robotersystems

Die beschriebenen Hardwarekomponenten des Systemaufbaus und deren Umgebung sind in [Abbildung 3.1](#) illustriert.

3.1.2 KUKA Lightweight Robot

Das Deutsche Zentrum für Luft- und Raumfahrt (DLR) hat ursprünglich den leichtgewichtigen Roboter LWR mit seinem hohen von Traglast zu Eigengewicht Verhältnis für den Einsatz in der Raumfahrt entwickelt. Dieser wurde im Laufe der Jahre weiterentwickelt und wird mittlerweile von der Firma KUKA Roboter GmbH in der vierten Version seit Dezember 2008 verkauft. Der LWR eignet sich laut KUKA derzeit noch nicht für den Industrieneinsatz und wird hauptsächlich für Forschungszwecke bzw. Entwicklungszwecke verwendet [Bischoff \[2010\]](#). Geliefert wird der LWR mit dem KUKA Robot Controller und einem KUKA Control Panel (KCP) (siehe



Abbildung 3.1: Dargestellt ist der KUKA LWR, Weiss Robotics WSG 50 & WSG-DSA, Microsoft Kinect und die Logitech HD Pro Webcam C910. Letztere befindet sich unter der Microsoft Kinect.

Abbildung 3.3). Das KCP ist ein Programmierhandgerät, mit dem die Bedienung und Programmierung des Robotersystems möglich sind.

Der LWR kann eine Traglast von 7 Kilogramm (kg) bei einem Eigengewicht von ungefähr 16 kg tragen. Das geringe Eigengewicht führt zu einem geringeren Energieverbrauch, wodurch der Betrieb mit einem Akku auf z.B. Servicerobotern ermöglicht wird. Weiterhin wird durch das geringe Eigengewicht das Verletzungspotenzial minimiert, welches als Voraussetzung für eine sichere Mensch-Roboter-Interaktion gilt.

Das Arbeitsraumvolumen des LWR beträgt 1.84 Kubikmeter und ist in der Seiten- und Draufsicht in Abbildung 3.2 illustriert. Erkennbar ist in der Seitensicht das hellgraue kugelförmige Volumen um das zweite Gelenk des Manipulator, welches für den LWR unerreichbar ist [Pedersen 2000, S. 38]. Weiterhin ist der Arbeitsraum des LWR durch die Montage auf dem Tisch, aufgrund der möglichen Kollisionen mit diesem nur eingeschränkt nutzbar. Um Beschädigungen zu vermeiden, müssen die Einschränkungen von der Bewegungsplanung berücksichtigt werden.

Ein Manipulator mit 6 Freiheitsgraden kann jede Pose im Raum mit maximal 8 verschiedenen Orientierungen erreichen. Der LWR verfügt durch einen zusätzlichen Freiheitsgrad über mehr Flexibilität. Die gleiche Pose im Raum kann theoretisch mit beliebig vielen Orientierungen angefahren werden, wodurch das Greifen von komplexen Objekten ermöglicht wird [Pedersen 2000, S. 7]. Beispielsweise ist das Greifen einer vollen Tasse von der Seite besser als von Oben, weil nur so der Inhalt der Tasse in ein anderes Behältnis geschüttet werden kann ohne den Greifer nass zu machen. Weiterhin können mit den sieben Freiheitsgraden typische Singularitäten eines Manipulators mit 6 Freiheitsgraden vermieden werden.

3.1.3 Weiss Robotics WSG 50 und WSG-DSA

Vom Hersteller Weiss Robotics wird der elektrische 2-Finger Parallelgreifer vom Typ WSG 50 verkauft. Der WSG 50 hat einen Hub von 55 mm pro Finger und kann mit einer nominalen Greifkraft von 5 bis 80 Newton (N) zugreifen. Auf dem WSG 50 befinden sich Greiferfinger mit einer integrierten Sensormatrix vom Typ WSG-DSA (siehe Abbildung 3.4). Der WSG-DSA verwendet zur Messung des Drucks die taktilen Aufnehmer vom Typs DSA9205i, welche aus einer 84 Zellen großen $6 * 14$ Sensormatrix besteht. Jede Sensorzelle kann die ausgeübte Kraft in einem Bereich von 0 bis 1023 messen. Die Sensoren sind zum Schutz mit einer Polymer-Gummibeschichtung bedeckt. Allerdings wirkt die Gummibeschichtung zugleich wie ein Tiefpassfilter auf die Sensoren und reduziert die Sensorempfindlichkeit [Carbone 2012, S. 83].

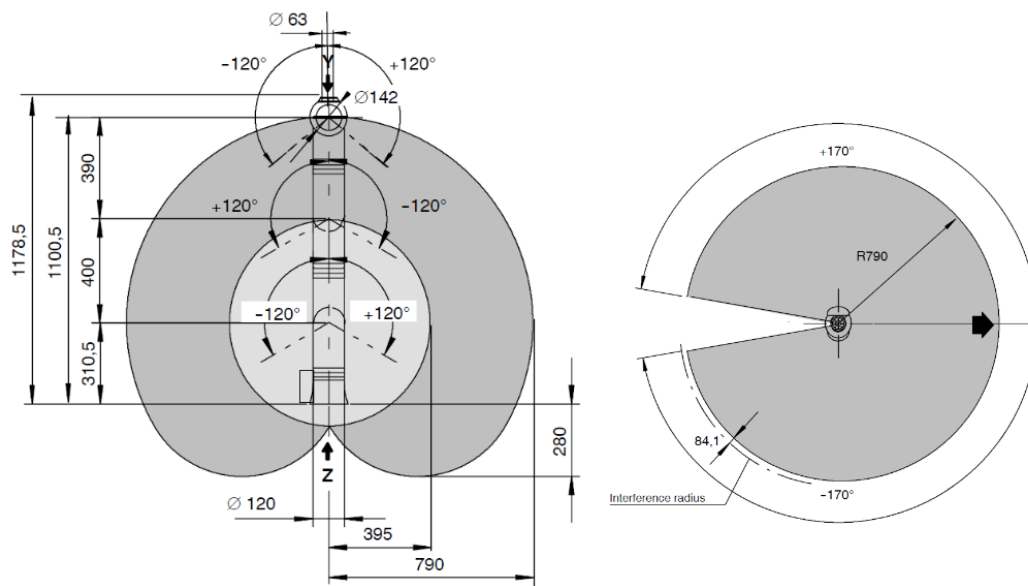


Abbildung 3.2: Abgebildet ist der Arbeitsraum des LWR. Auf der linken Seite ist die Seitenansicht und auf der rechten Seite die Draufsicht illustriert. Die Maße sind in mm angegeben. Abbildung entnommen aus [Pedersen \[2000\]](#).



Abbildung 3.3: Abgebildet ist der KUKA Lightweight Robot (LWR) mit dem KUKA Robot Controller und dem KUKA Control Panel (KCP). Abbildung entnommen aus [Bischoff \[2010\]](#).



Abbildung 3.4: Abgebildet ist der WSG 50 Greifer und die integrierte Sensormatrix vom Typ WSG-DSA auf den Greiferfingern. Abbildung entnommen aus [WSG-DSA, 2011](#) [b].

Weiterhin können mit der hohen Auflösung der taktilen Aufnehmer des WSG-DSA die Details der berührten Oberfläche erkannt werden, welche sich im beschränkten Umfang für eine Objekterkennung eignen [DSA \[2011\]](#). Der WSG 50 wird in dieser Arbeit durch das Transmission Control Protocol gesteuert. Die Kalibrierung des WSG 50 und des WSG-DSA kann über das verfügbare Webinterface erfolgen. Das Gesamtgewicht des WSG 50, der Greiferfinger und einer zusätzlich montierten Aluminiumplatte beträgt etwa 1.44 kg. Die Traglast des LWR reduziert sich dadurch auf 5.56 kg.

3.1.4 Logitech HD Pro Webcam C910

Der Hersteller Logitech bietet mit der Logitech HD Pro Webcam C910 (siehe [Abb. 3.5](#)) eine High Definition Webkamera mit 1080p-Auflösung an. Die Webkamera verfügt über eine maximale Digitalvideoauflösung von 1920 x 1080 Pixel und übertrifft damit die Farbkamera der Microsoft Kinect. Aufgrund der höheren Digitalvideoauflösung eignet sich die Webkamera besser zur Objekterkennung nach der skaleninvarianten Merkmalstransformation (SIFT-Verfahren), welches im [Abschnitt 3.4](#) näher beschrieben wird [Webcam \[2014\]](#).

3.1.5 Microsoft Kinect

Die Microsoft Kinect wird seit 2010 vom Hersteller Microsoft für die Xbox360 Spielekonsole verkauft. Die Kinect ist eine 3D-Kamera, die wegen ihres kostengünstigen Preises auch in anderen Bereichen eingesetzt wird. Die Kinect in [Abbildung 3.6](#) verfügt über eine VGA Kamera, einen Infrarot-Projektor und eine Infrarot-Kamera. Mit der VGA Kamera können Farbbilder mit einer maximalen Auflösung von 640x480



Abbildung 3.5: Logitech HD Pro Webcam C910. Abbildung entnommen aus [Webcam \[2014\]](#).



Abbildung 3.6: Microsoft Kinect. Abbildung entnommen aus [Kinect \[2014\]](#).

aufgezeichnet werden [FM \[2014\]](#). Das besondere an der Kinect ist die Tiefenkamera zur 3D Bilderzeugung, welche aus dem Infrarot-Projektor und der Infrarot-Kamera besteht. Der Infrarot-Projektor sendet im Infrarotbereich eine definierte Punkt-Matrix in den Raum und die Infrarot-Kamera nimmt das reflektierte Muster auf. Das Funktionsprinzip wurde in Abschnitt 2.6 beschrieben. Die Kinect sollte in einem geschlossenen Raum verwendet werden, weil dort der Infrarot-Anteil im Lichtspektrum gering ist und die Tiefenkamera nicht beeinflusst wird. Weiterhin sollte die Kinect an einen vibrationsarmen Ort platziert werden, weil Vibrationen zu Rauschen führen, welches fehlerhafte Tiefendaten liefert. Laut dem Hersteller hat die Kinect einen optimalen Erkennungsbereich zwischen 0.8 m und 3.5 m [[Tirpitz 2012](#), S. 16]. Mit den Openni Treibern lässt sich zwar ein Erkennungsbereich von 0.6 - 10 m realisieren, jedoch nimmt die Genauigkeit proportional zum Quadrat der Entfernung ab [FM \[2014\]](#). [Khoshelham \[2011\]](#) empfiehlt eine Objektentfernung zwischen 1 und 3 m, weil die gemessene Entfernung in der Nähe der Kinect nur um einige Millimeter variiert, jedoch bereits um vier Zentimeter bei einer Distanz von 5 Metern.

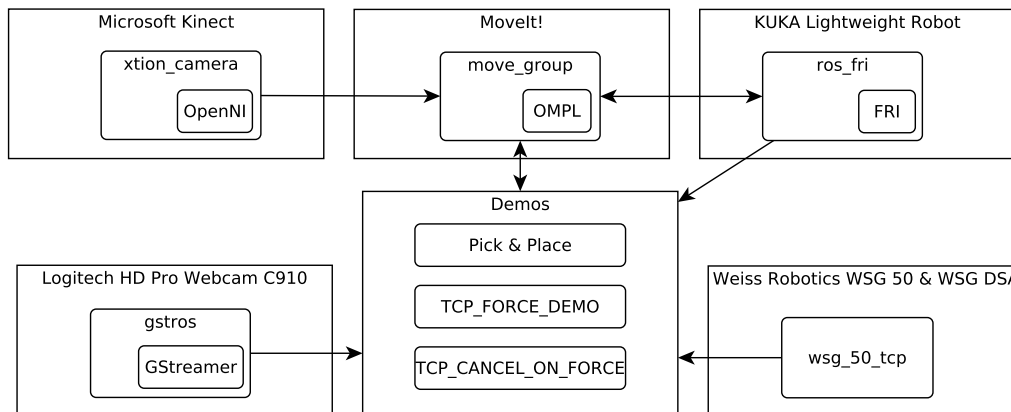


Abbildung 3.7: Dargestellt ist eine stark vereinfachte Gesamtdarstellung der verwendeten ROS Softwarepakete des Robotersystems.

3.2 Verwendete Software

In diesem Unterkapitel werden die verwendeten Softwarebibliotheken und Frameworks kurz vorgestellt. Eine stark vereinfachte Gesamtdarstellung der verwendeten Softwarekomponenten des Robotersystems und der Interaktion zwischen diesen ist in Abbildung 3.7 dargestellt.

3.2.1 Open Motion Planning Library

Die Open Motion Planning Library (OMPL) ist eine freie Softwarebibliothek mit vielen modernen stichprobenbasierenden Bewegungsplanungsalgorithmen (Sampling-based motion planning). Diese finden besonders schnell und effektiv eine Trajektorie auch in einem großen und komplexen Konfigurationsraum, welcher bei Systemen mit vielen Freiheitsgraden auftritt. In der OMPL ist unter anderem der in Abschnitt 2.4 beschriebene RRT-Connect Algorithmus enthalten [OMPL 2011, S. 3].

3.2.2 GStreamer

GStreamer ist ein quelloffenes Multimedia Framework, welches die freie Softwarebibliothek GObject verwendet. GStreamer ist für die Verarbeitung von Multimedia-Datenströmen gedacht, kann aber auch für andere Datenströme eingesetzt werden. Über die GStreamer-Pipeline werden Datenströme in mehrere hintereinander geschalteten Elemente (sog. Plugins) geleitet. Alle Plugins lassen sich in die drei Kategorien Eingabe, Transformation und Ausgabe unterteilen, welche bei GStreamer

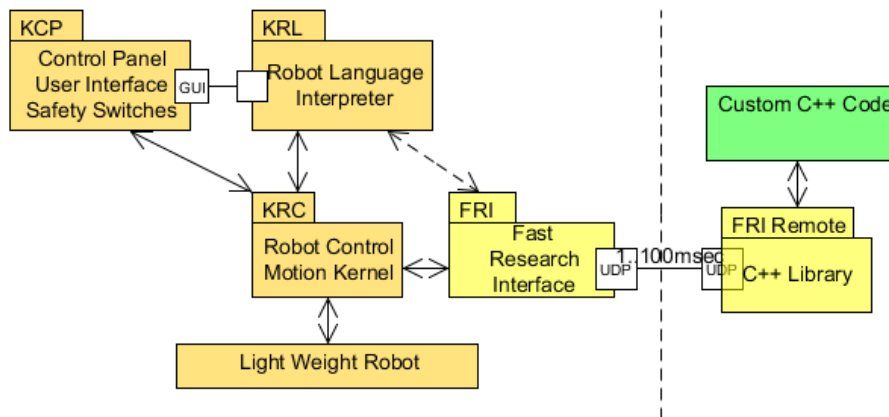


Abbildung 3.8: FRI Systemarchitektur. Abbildung entnommen aus [Bischoff \[2010\]](#).

die Bezeichnungen Source, Filter und Sink haben. Die zahlreichen GStreamer Plugins sind wiederverwendbare Bausteine, mit denen unter geringem Aufwand viele Anwendungen realisiert werden können [GStreamer \[2014\]](#), [Pipeline \[2014\]](#).

3.2.3 Fast Research Interface Library

Die Fast Research Interface Library wurde von Torsten Kröger an der Stanford Universität entwickelt und ermöglicht die Steuerung des KUKA Light-Weight Robot IV in der Programmiersprache C++. In [Abbildung 3.8](#) ist die Fast Research Interface (FRI) Systemarchitektur dargestellt. Das FRI ist eine Programmierschnittstelle, die den Zugriff auf alle verfügbaren Funktionen seitens des KUKA LWR ermöglicht. Die FRI Library wird auf einem entfernten Rechner ausgeführt, welcher zum KUKA Robot Controller (KRC) über das FRI eine Ethernet-Verbindung aufbaut. Beide Seiten tauschen User Datagram Protocol (UDP) Pakete in einem fest einstellbaren Intervall von 1-100 ms miteinander aus. Der KRC schickt periodisch Kontroll- und Statusinformationen an den entfernten Rechner und erwartet unmittelbar eine Antwortnachricht. In der Antwortnachricht können Eingangsdaten für die Steuerung der entsprechenden Controller enthalten sein [FRI \[2014\]](#).

3.2.4 Reflexxes Motion Libraries

Die von Torsten Kröger entwickelten Reflexxes Motion Libraries ermöglichen die Berechnung einer Trajektorie innerhalb eines low-level Kontrollzyklus von in der Regel weniger als einer Millisekunde (ms). Die Generierung einer Roboterbewegung

während der Bewegungsausführung ermöglicht es dem Roboter unmittelbar auf sensorbasierte Ereignisse zu reagieren. In dieser Arbeit wird die frei verfügbare Version Reflexxes Type II Motion Library verwendet [Kroeger \[2011\]](#).

3.2.5 Robot Operating System

Das Robot Operating System (ROS) ist ein quelloffenes Softwareframework für die Entwicklung von Roboteranwendungen. Die Robotik ist ein komplexes Themengebiet mit zahlreichen Problemen, die von keiner einzelnen Institution gelöst werden können. ROS ermöglicht es den unterschiedlich spezialisierten Organisationen bzw. Instituten auf der Welt, ihre Arbeiten anderen zur Verfügung zu stellen, um damit die technologische Entwicklung gemeinsam voranzubringen. Der Einsatz von ROS hat besonders im Bereich der Forschung zugenommen (siehe [3.2.6](#)) [ROS \[2014\]](#).

3.2.6 MoveIt!

MoveIt! ist das aktuelle Softwareframework für Bewegungsplanung, Kinematik, Kollisionskontrolle, Manipulation, Navigation, Perzeption und Kontrolle unter ROS [MoveIt!, 2014 \[a\]](#), [MoveIt!, 2014 \[b\]](#). Mit MoveIt! ist, unter anderem, die Objektmanipulation mit einer kollisionsfreien und selbstgefilterten Bewegungsplanung möglich. MoveIt! befindet sich im Alpha Entwicklungsstadium und wird schon auf über 65 Robotern verwendet [MoveIt!, 2014 \[c\]](#).

3.3 Roboterkonfiguration mit MoveIt!

Mit MoveIt! kann im Prinzip jeder beliebige Roboter gesteuert werden. Zunächst müssen notwendige Konfigurationsdateien mit dem MoveIt! Setup Assistent erstellt werden, welcher für den Vorgang ein Unified Robot Description Format (URDF) benötigt. Das URDF ist eine Datei im XML Format für die Repräsentation des Robotermodells. Der WSG 50 und der KUKA LWR sind Bestandteile des Roboterbaus und müssen deswegen in eine URDF integriert werden. Die URDF Dateien für den KUKA LWR und den WSG 50 wurden von den Herstellern entnommen und in `wsg50_on_lwr.urdf` zusammengefügt. Die Umgebung und das Robotermodell des Gesamtaufbaus aus [Abbildung 3.7](#) sind mit RViz in der [Abbildung 3.9](#) visualisiert. Hierbei wurde die Umgebung und die Platzierung des KUKA LWR von Dr. Norman Hendrich umgesetzt. In der Mitte sind beispielhaft die definierten kartesischen Koordinatensysteme `tcp` und `xtion_camera_link` dargestellt, welche erstellt und richtig platziert wurden. Auf der rechten Seite sind alle definierten kartesischen Koordinatensysteme der `wsg50_on_lwr.urdf` illustriert.

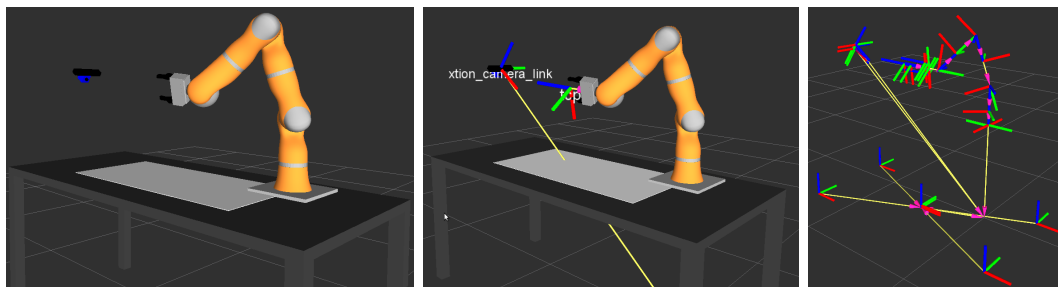


Abbildung 3.9: Auf der linken Seite ist das visualisierte Robotermodell und seine Umgebung dargestellt. In der Mitte sind beispielhaft die kartesischen Koordinatensysteme `tcp` und `xtion_camera_link` illustriert. Auf der rechten Seite sind alle definierten kartesischen Koordinatensysteme dargestellt. Die x,y,z -Achsen sind mit rot, grün und blau illustriert.

Aufbauend auf der URDF erstellt der MoveIt! Setup Assistent ein Modell im Semantic Robot Description Format (SRDF) [SRDF \[2014\]](#). Das SRDF verfügt über zusätzliche Systeminformationen gegenüber dem URDF, welche vom Anwender während der MoveIt! Setup Assistenten Routine angegeben werden. Zu den zusätzlichen Informationen zählen z.B. alle Gelenkpaare des Roboters, die nie miteinander in Kollision geraten können, oder alle Gelenke, die zur kinematischen Kette gehören. Der MoveIt! Setup Assistent erstellt Konfigurations- und Ausführungsdateien, welche für die Betriebsbereitschaft des Roboters benötigt werden. Die Controller- und Sensorintegration muss allerdings vom Anwender manuell durchgeführt werden. MoveIt! kann mit Roboterarmcontrollern vom Typ `FollowJointTrajectory` und Greifercontrollern vom Type `GripperCommand` kommunizieren. Damit kann MoveIt! die Funktionen der Controller auslösen und beispielsweise das Ausführen einer Trajektorie oder das Schließen des Greifers bewirken. Aufbauend auf den Befehlen können komplexe Befehle generiert werden z.B. das autonome Greifen oder Platzieren eines Objektes. Für die Kommunikation zwischen dem MoveIt! Framework und dem KUKA LWR wird ein Robotercontroller in ROS benötigt. Ein bekannter LWR Controller ist im `lwr_fri` package aus dem `lwr_hardware` Stack enthalten, welcher von Konrad Banachowicz von der Universität Warsaw entwickelt wurde. Dieser unterstützt bereits den Typ `FollowJointTrajectory`. In Abschnitt 3.6 werden die Nachteile des Packages beschrieben und durch das selbst entwickelte Package `ros_fri` ersetzt. Für den WSG 50 Greifer hat die Robotnik Automation SLL. das Package `wsg50-ros-pkg` entwickelt, welches allerdings den Typ `GripperCommand` nicht unterstützt. Für die MoveIt! Greiferintegration wurde mit der Entwicklung eines Action Server vom Typ `GripperCommand` mit dem Namen `gripper_adapter_server` begonnen, welcher die Funktionalitäten des `wsg50-ros-pkg` verwendet. In dieser Arbeit wird der `gripper_adapter_server` jedoch nicht weiter verwendet und steht lediglich für zukünftige Anwendungen bereit. Die MoveIt! Konfigurationsdatei `controller.yaml` muss für die erfolgreiche Controllerintegration mit dem folgenden Inhalt gefüllt werden.

```

controller_list:
- name: arm_controller
  type: FollowJointTrajectory
  default: true
  joints:
    - lwr_arm_0_joint
    - lwr_arm_1_joint
    - lwr_arm_2_joint
    - lwr_arm_3_joint
    - lwr_arm_4_joint
    - lwr_arm_5_joint
    - lwr_arm_6_joint
- name: gripper_adapter_server
  type: GripperCommand
  default: true
  joints:
    - wsg_50_gripper_base_joint_gripper_left
    - wsg_50_gripper_base_joint_gripper_right

```

Die Kollisionsvermeidung bei sich ändernden Umgebungen kann nur mit dem Einsatz von Sensoren erreicht werden. MoveIt! kann für die Erstellung seiner Umgebung auf die in Echtzeit generierten Sensordaten oder auf geladene Modelle z.B. CAD Modelle zugreifen [MoveIt!, 2014](#) [b]. Die Sensorintegration in MoveIt! lässt sich für die Microsoft Kinect durch die Erstellung der Datei `sensors_kinect.yaml` mit folgendem Inhalt durchführen.

```

sensors:
- sensor_plugin: occupancy_map_monitor/PointCloudOctomapUpdater
  point_cloud_topic: xtion_camera/depth_registered/points
  max_range: 1.1
  point_subsample: 1
  padding_offset: 0.1
  padding_scale: 1.0
  filtered_cloud_topic: filtered_cloud

```

Die Kinect liefert eine Punktwolke, mit welcher eine OctoMap erstellt wird. Die OctoMap ist eine Raumrepräsentation, die beliebige Umgebungen in belegte und freie Bereiche modelliert. Die Implementierung der OctoMap basiert auf einem Octree. Ein Octree ist eine baumartige Datenstruktur, in der jeder Knoten durch ein würfelförmiges Volumen repräsentiert wird. Das Volumen wird rekursiv in acht Teilmengen unterteilt, bis die gewünschte minimale Volumengröße erreicht ist [[OctoMap 2012](#), S. 4]. In [Abbildung 3.10](#) ist das volumetrische Modell auf der linken Seite und der dazugehörige Baum auf der rechten Seite dargestellt. Das Umgebungsmodell der

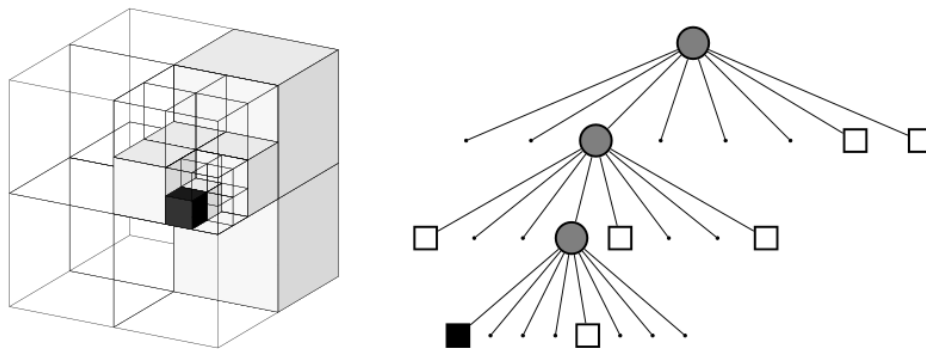


Abbildung 3.10: Auf der linken Seite ist das volumetrische Model eines Octrees abgebildet und auf der rechten Seite der dazugehörige Baum. Der Octree beinhaltet freie (weiße) und belegte (schwarze) Knoten. Die grauen Knoten müssen weiter unterteilt und mit Nachfolgeknoten repräsentiert werden. Abbildung entnommen aus [OctoMap 2012, S. 4].

OctoMap ermöglicht es dem Bahnplaner, die geplante Trajektorie auf Kollisionen zu überprüfen.

Der komplette Bewegungsplanungsprozess kann mit dem 3D Visualisierungswerkzeug ROS Visualizer (RViz) visualisiert werden. MoveIt! verfügt über das Motion-Planning Plugin für RViz, welches unter anderem das Robotermodell und die generierte Bewegungsplanung visualisiert. In Abbildung 3.11 ist der Planungsprozess mit RViz für den LWR dargestellt. Die Pose des Roboterarmes kann mit dem Interaktiven Marker verändern werden. Der grüne Manipulator repräsentiert den Startzustand und der gewünschte Endzustand wird durch den gelben bzw. orangen Manipulator dargestellt. Falls sich der Manipulator in Selbstkollision oder in Kollision mit einem Objekt aus der Umgebung befindet, nimmt der Manipulator die Farbe rot an. MoveIt! erlaubt es Bedingungen für die Bewegungsplanung zu definieren, wie etwa die Einschränkung des Arbeitsraumes, approximierte IK, Bahneinschränkungen und erneute Bahnplanung zu definieren. Für die Bewegungsplanung verwendet MoveIt! die OMPL. Die Ergebnisse der Bewegungsplanungsalgorithmen für eine spezifische Umgebungen können z.B. mit RViz verglichen werden. Mögliche Bewertungskriterien können die Planungszeit, Erfolgsrate und die geplante Trajektorie sein.

Für die schnelle Wahrnehmung von Veränderungen in der Umgebung wird die OctoMap periodisch aktualisiert. Beispielhaft ist in Abbildung 3.12 die OctoMap in RViz für Umgebungen ohne und mit Hindernis (Arm) dargestellt. Damit kann die Detektion des Menschen im Arbeitsraum des Robotersystems erfolgen und Kollisionen können vermieden werden.

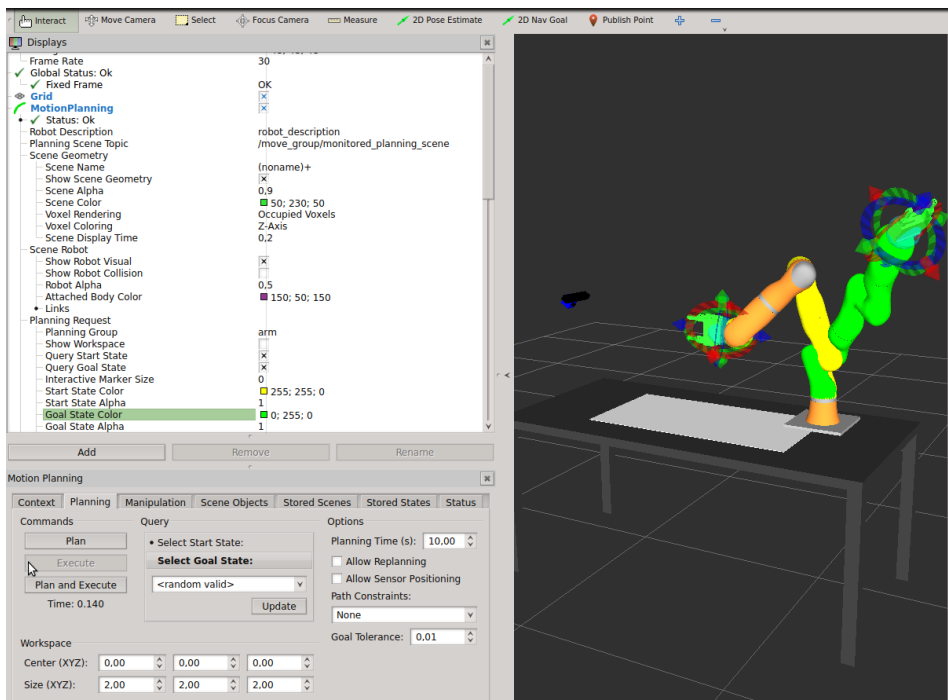


Abbildung 3.11: Illustriert ist die Bewegungsplanung mit dem MoveIt! Plugin MotionPlanning für RViz. Der grüne Manipulator ist der Startzustand und der gelbe bzw. orange Manipulator der Endzustand.

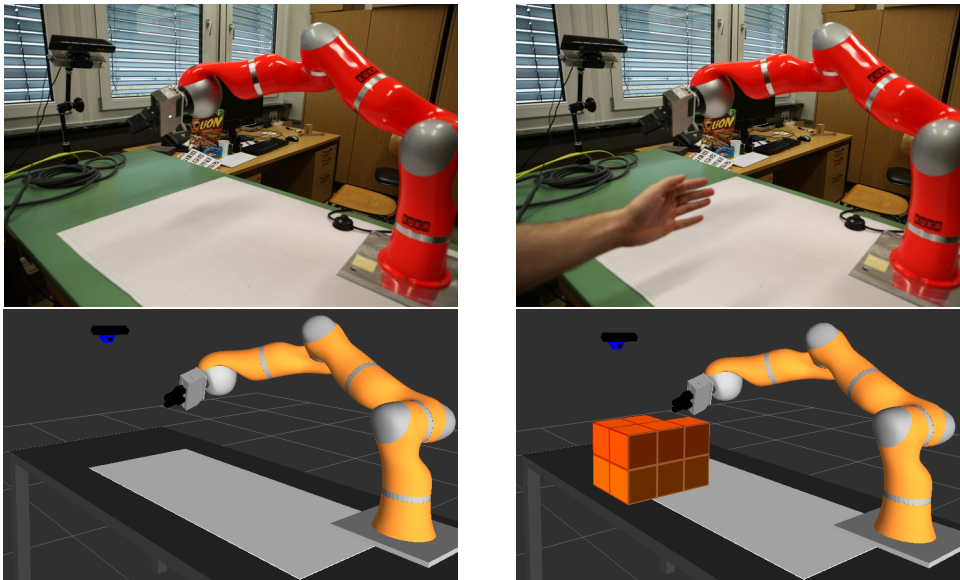


Abbildung 3.12: Illustriert ist in der oberen Zeile eine Umgebung ohne (linke Seite) und mit (rechte Seite) Hindernissen. Die Abbildungen in der unteren Zeile zeigen die dazugehörige OctoMap.

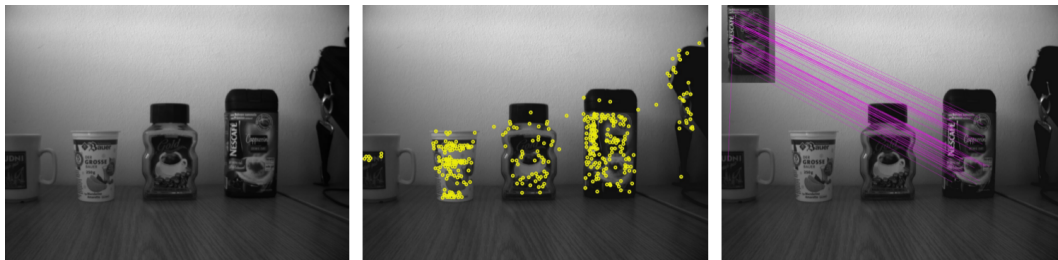


Abbildung 3.13: Auf der linken Seite ist eine Tischszene dargestellt. In der Mitte befindet sich die mit dem SIFT Algorithmus berechneten Merkmalspunkte der Tischszene. Auf der rechten Seite werden die übereinstimmenden Punkte mit der zuvor gelernten Nescafe Dose illustriert. Abbildung entnommen aus [Bistry \[2013\]](#).

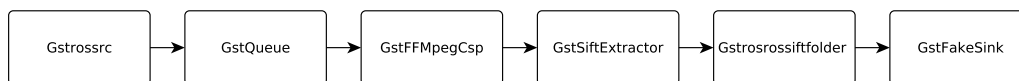


Abbildung 3.14: GStreamer Pipeline.

3.4 Objekterkennung

Für die Objekterkennung mit der Farbkamera Logitech HD Pro Webcam C910 wird der von D. Lowe im Jahre 1999 vorgestellte Scale-Invariant-Feature-Transform (SIFT) Algorithmus verwendet [Lowe \[1999\]](#). Der SIFT Algorithmus extrahiert Bildmerkmale, die invariant gegenüber Translation, Rotation und Skalierung sind. Die Bildmerkmale werden in einem SIFT-Vektor gespeichert und für den Vergleich mit anderen Bildern genutzt. In [Abbildung 3.13](#) sind die einzelnen Schritte der Objekterkennung anhand einer Tischszene mit einem gelernten Objekt Nescafe illustriert. Die gefundenen übereinstimmenden Merkmalspunkte der Tischszene mit dem bekannten Objekt werden in der rechten Abbildung transformiert illustriert. Die Position und Orientierung eines Objektes kann durch mindestens vier übereinstimmende Punkte berechnet werden, wenn Kalibrierungsdaten vorliegen und die Position der Punkte bezüglich eines Objektkoordinatensystems bekannt ist [[Bistry 2013](#), S. 11].

Der SIFT Algorithmus wird in der von Dr. Hannes Bistry entwickelten GStreamer Pipeline eingesetzt, welche in [Abbildung 3.14](#) vereinfacht visualisiert ist. Die aufgebaute GStreamer Pipeline besteht aus einer Kette von Elementen. Das erste Element `Gstrossrc` greift den Bilddatenstrom der Kamera aus dem ROS-Framework auf. Die Datenpakete werden anschließend mit dem `GstQueue` Element nach dem

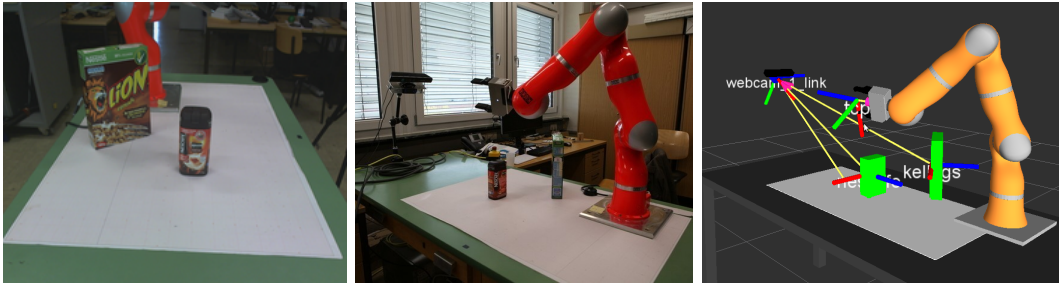


Abbildung 3.15: Auf der linken Seite ist das Kamerabild mit den detektierten Objekten Nescafe Dose und Kellogs Schachtel dargestellt. In der Mitte ist die seitliche Sicht des Gesamtaufbaus illustriert. Auf der rechten Seite ist das dazugehörige Robotermodell und die detektierten Gegenstände in RViz dargestellt.

FIFO-Prinzip zwischengespeichert und an das nächste Element weitergeleitet. Das Zwischenspeichern ist notwendig um Datenverlust zu vermeiden, weil Situationen auftreten können, in denen der Scheduler eines Betriebssystems das einzelne Datenpaket der Kamera nicht in der vorgesehenen Zeit verarbeitet kann. Danach konvertiert GstFFMpegCsp den Videoframe von video/x-raw-yuv nach video/x-raw-rgb. Das Element GstSiftExtractor berechnet daraufhin den SIFT-Vektor aus den Eingangsdaten. Anschließend wird im Element Gstrosssiftfolder der eingehenden SIFT-Vektor mit allen Bilddateien aus einem Ordner verglichen und die Ergebnisse werden im ROS-Framework publiziert. Der Ordner beinhaltet mit der Nescafe Dose und einer Kellogs Schachtel zwei bekannte Objekte. Das letzte Element GstFakeSink publiziert die Bilddaten aus GStreamer zum ROS-Framework [Bistry \[2013\]](#). Auf der ROS Ebene werden Daten von dem gstros Package geliefert.

In [Abbildung 3.15](#) wird die Detektion der Objekte Nescafe Dose und Kellogs Schachtel in RViz mit dem Interaktiven Marker illustriert. Die detektierten kartesischen Koordinaten stimmen weitgehend mit den tatsächlichen überein. Für die Nescafe Dose wurden die Werte $x = 2.2$ cm, $y = -9.0$ cm, $z = 82.7$ cm und für die Kellogs Schachtel $x = -19.3$ Zentimeter (cm), $y = -23.0$ cm, $z = 95.0$ cm ermittelt. Die x und y Werte sind in der Regel nur minimalen Schwankungen von ungefähr ± 0.1 cm ausgesetzt. Während bei der Tiefeninformation Schwankungen von ± 0.7 cm beobachtet wurden. Die Genauigkeit der Objektdetektion ist allerdings ausreichend, um beispielsweise Pick & Place Operationen durchzuführen.

3.5 Erweiterung des wsg50-ros-pkg um zusätzliche Funktionen

Für die Steuerung des Weiss Robotics WSG 50 unter ROS wurde das von Robotnik Automation SLL. entwickelte Package wsg50-ros-pkg eingesetzt. Dieses musste im Laufe der Arbeit um zusätzliche Funktionen erweitert werden, um die Integration des WSG 50 und des WSG-DSA an das Robotersystem zu erreichen. In diesem Abschnitt werden die implementierten Erweiterungen vorgestellt.

Um das Greifen und Loslassen von Objekten in der Simulation darstellen zu können und Kollisionen mit der Umgebung zu vermeiden, werden die Gelenkstellungen der Finger des WSG 50 benötigt. Mit diesen kann MoveIt! zudem Kollisionen zwischen den Greiferfingern und dem zu greifenden Objekt feststellen, welche auftreten, wenn die Finger mit einer zu starken Kraft geschlossen werden und dadurch in ein leicht verformbares Objekt hineinfahren. Das wsg50-ros-pkg bietet nicht die Möglichkeit die aktuellen Gelenkstellungen MoveIt! mitzuteilen, weshalb dieses um einen Joint-State-Publisher erweitert wurde, welcher die Gelenkwerte dem Topic `joint_states` mitteilt. Die Gelenkwerte werden aus der aktuellen Finger Position `op` berechnet. Der WSG 50 ist ein 2 Finger Parallelgreifer, welcher in der Nullstellung die Finger geschlossen hat. Für die Berechnung der Gelenkwerte muss `op` nur durch zwei geteilt werden und bei einem Gelenkwert muss ein negatives Vorzeichen stehen, weil die Greiferfinger beim Öffnen in entgegengesetzte Richtung fahren. Abschließend wird nochmal um 1000 geteilt, um die Einheit von Millimeter auf Meter umzuwandeln.

```
joint_states.position[0] = -op/2000.0;  
joint_states.position[1] = op/2000.0;
```

Weiterhin sind für die Experimente unbedingt die Funktionalitäten der taktilen Sensoren erforderlich, für welche keine offiziellen ROS Treiber existieren. Für ihre Implementierung wurden die dargestellten Informationen in [Abbildung 3.16](#) verwendet. Gezeigt ist der Aufbau der Datenpakete des Befehls `FINGER_GET_DATA` für das Senden und Empfangen der Sensordaten [WSG-DSA, 2011 \[a\]](#). Mit den Informationen wurde das wsg50-ros-pkg erweitert, welches die taktilen Sensorinformationen periodisch ausliest und anschließend in einer selbstdefinierten Message auf das ROS Topic `wsg_50/sensor` sendet. Die Sensordaten wurden für die Experimente verwendet, um beispielsweise die externe Kraftausübung eines gegriffenen Objektes zu erkennen.

3.6 Entwicklung des ros_fri Controllers für den LWR

Jedes der sieben Gelenke des LWR ist neben einem Positionssensor auch mit einem Momentensensor ausgestattet, wodurch ein positions-, geschwindigkeits- und kraft-

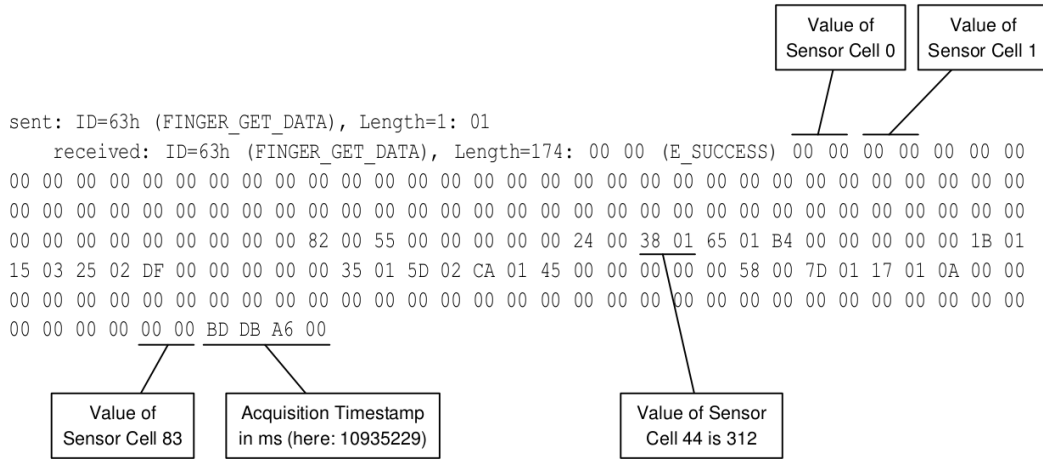


Abbildung 3.16: Abgebildet ist der Aufbau der Datenpakete des FINGER_GET_DATA Befehls. Abbildung entnommen aus WSG-DSA, 2011 [a].

geregelter Betrieb ermöglicht wird. Mit dem Momentensensoren kann eine nachgebende Bewegung ausgeführt werden, wobei nachgebend bedeutet, dass der Roboterarm durch externe Krafteinwirkung nachgibt [DLR [2014]]. Der LWR verfügt über die folgenden vier Kontrollstrategien.

Der Position-Controller bzw. Joint-Position-Controller nutzt die Positionssensoren des LWR und steuert diesen zu der gewünschten Position ohne ein nachgiebiges Verhalten zu zeigen.

Der Cartesian-Stiffness-Controller bzw. Cartesian-Impedance-Controller verwendet neben den Positionssensoren zusätzlich noch die Momentensensoren, mit denen ein nachgiebiges Verhalten am Referenzpunkt des Werkzeugs im kartesischen Raum erzeugt werden kann. Die Nachgiebigkeit kann in jede Richtung und Drehung mittels den Steifigkeits- und Dämpfungsparametern geregelt werden. Die maximale erlaubte kartesische Kraft und die maximale kartesische Abweichung vom TCP werden in dem Modus ebenfalls angegeben. Weiterhin besteht noch die Möglichkeit eine Kraft in eine beliebige Richtung am TCP auszuüben.

Der Axis-specific-Stiffness-Controller bzw. Joint-Impedance-Controller nutzt ebenfalls neben den Positionssensoren die Momentensensoren um ein nachgiebiges Verhalten an den Gelenken auszulösen. Die regelbaren Parameter in diesem Modus sind die Steifigkeit, Dämpfung, maximale erlaubte Kraft und maximale erlaubte Abweichung für jedes der sieben Gelenke.

Im Gravity-Compensation Modus werden nur die Momentensensoren verwendet. Bei richtiger Kalibrierung kompensiert der LWR die ausgeübte Gravitation, die auf je-

des Gelenk einwirkt und verharrt in seiner aktuellen Stellung. Bei externer Kraftausübung auf die Gelenke können diese in die gewünschte Position gebracht werden. Der Gravity-Compensation Modus kann nicht im Automatik Modus, sondern nur im T1 Modus betrieben werden. Das bedeutet, der LWR kann im Gravity-Compensation Modus nicht mit Programmen gesteuert werden [KUKA 2011, S. 107ff.], [Pedersen 2000, S. 49].

Für die Armsteuerung wurde zunächst der öffentlich zugängliche LWR Controller von Konrad Banachowicz aus dem `lwr_fri` package im `lwr_hardware` Stack verwendet. Das ROS Package ist mit dem Orocos Real Time Toolkit integriert, welches harte Echtzeitkommunikation zwischen den Kommunikationspartnern garantiert Orocos [2014]. Während der ersten Experimente sind jedoch zahlreiche Nachteile aufgetreten. Einerseits besitzt das `lwr_fri` Package zahlreiche Paketabhängigkeiten, wodurch der Installationsaufwand erhöht und die Integration an neu erscheinende ROS Distributionen erschwert wird. Die Integration des Packages unter ROS Hydro ist nach zahlreichen Versuchen nicht gelungen. Weiterhin wurde ein Programmierfehler beim Trajektorieabbruchbefehl entdeckt. Der Controller bricht zwar den Vorgang ab, jedoch werden anschließend Anfragen ignoriert, weil der Controller nicht mehr in den Bereitschaftsmodus zurückkehrt.

Weiterhin sind die Funktionen des Controllers verglichen mit den tatsächlichen Möglichkeiten des LWR stark eingeschränkt. Beispielsweise kann mit dem Controller nur der Joint-Position-Controller des LWR angesprochen werden. Erstrebenswert ist es jedoch auch den Cartesian-Impedance-Controller und den Joint-Impedance-Controller des LWR zu nutzen. Zudem ist es nicht möglich mit dem Package auf Sensorinformationen wie z.B. die TCP Kraftwerte zuzugreifen. Deshalb wurde die Entwicklung eines auf der FRI Library basierenden ROS Packages mit dem Namen `ros_fri` beschlossen. Mit `ros_fri` wird eine ROS Schnittstelle für fast alle Funktionen der FRI Library angeboten, welche wiederum nahezu das komplette Potenzial des LWR nutzt. Die Hauptaufgabe von `ros_fri` ist das Agieren als LWR Controller. Deshalb wurde `ros_fri` als ein Server implementiert, welcher auf Trajektorien vom Typ `FollowJointTrajectory` lauscht und bei Anfrage des Clients ausführt. Der Server ist genauer ein `ActionServer` der auf dem `actionlib` Package basiert `actionlib` [2014]. Das `actionlib` Package bietet unterstützende Funktionen für die Entwicklung eines Servers, welcher z.B. die Ausführung von Trajektorien abrechnen kann. Die Kommunikation zwischen `ActionClient` und `ActionServer` findet über das Action Protokoll statt, welches über ROS Topics in einem festgelegten ROS namespace den Nachrichtenaustausch ermöglicht. Die Schnittstellen des `ActionClient`s und `ActionServer`s sind in Abbildung 3.17 dargestellt. Hierbei haben die ausgetauschten ROS Nachrichten die folgenden Bedeutungen:

Action Interface

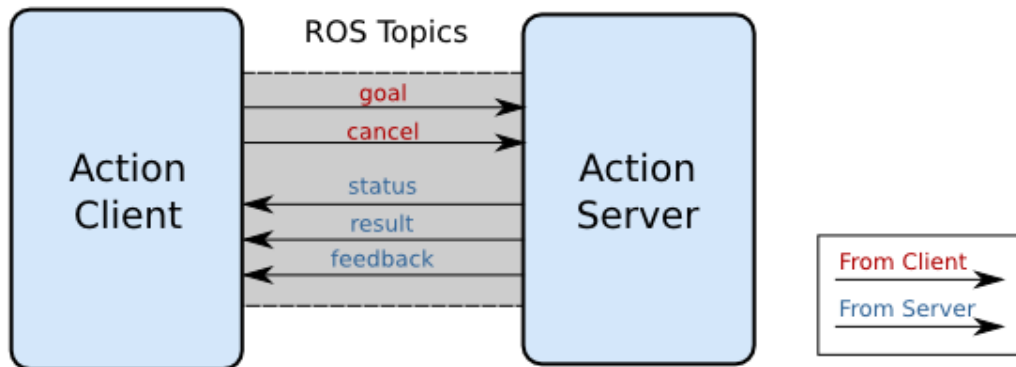


Abbildung 3.17: Abgebildet sind die Schnittstellen des Action Clients und Action Servers. Abbildung entnommen aus [actionlib](#) [2014].

- goal - Wird benutzt, um ein neues Ziel an den Server zu senden;
- cancel - Wird benutzt, um einen Abbruchbefehl an den Server zu senden;
- status - Wird benutzt, um dem Client den aktuellen Bearbeitungsstand des Ziels mitzuteilen;
- feedback - Wird benutzt, um dem Client periodisch Informationen während der Bearbeitung des Ziels mitzuteilen;
- result - Wird benutzt, um dem Client Informationen nach Abschluss des Ziels mitzuteilen.

Das Verständnis von stattfindenden Interaktionen zwischen dem Client und Server ist für die erfolgreiche Implementierung des ActionServers notwendig. In der Abbildung 3.18 ist der Zustandsautomat eines ActionServers mit allen möglichen Übergängen dargestellt.

Der ActionServer hat Zustandsübergänge, die von der Client Seite und der Server Seite ausgelöst werden können. Von der Client Seite kann ein CancelRequest an den ActionServer gehen, der die Bearbeitung des Ziels unterbrechen soll. Von der Server Seite können die folgenden Zustandsübergänge hervorgerufen werden [actionlib](#) [2014]:

- setAccepted - Nach Überprüfung des Ziels wird entschieden, die Bearbeitung zu beginnen;
- setRejected - Nach Überprüfung des Ziels wird entschieden, das Ziel zu verwerfen, weil es ungültig bzw. nicht erlaubte Angaben enthält;

Server State Transitions

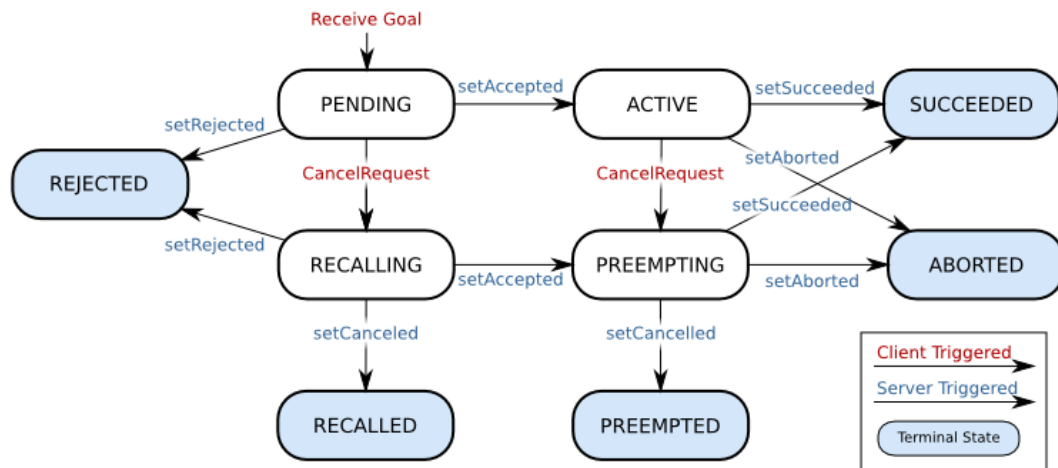


Abbildung 3.18: Abgebildet ist der Zustandsautomat eines ActionServers. Abbildung entnommen aus [actionlib \[2014\]](#).

- setSucceeded - Benachrichtigung über die erfolgreiche Bearbeitung des Ziels;
- setAborted - Benachrichtigung, dass ein Fehler während der Bearbeitung aufgetreten ist und das Ziels abgebrochen wurde;
- setCanceled - Benachrichtigung, dass die Bearbeitung des Ziels unterbrochen wurde, weil ein CancelRequest vom Client angekommen ist.

Aus den Zustandsübergängen ergeben sich die folgenden temporären Serverzustände:

- Pending - Das Ziel wird gleich vom ActionServer bearbeitet;
- Active - Das Ziel wird augenblicklich vom ActionServer bearbeitet;
- Recalling - Das Ziel wurde noch nicht bearbeitet und ein CancelRequest vom Client ist eingetroffen, aber der ActionServer hat das Ziel noch nicht abgebrochen;
- Preempting - Das Ziel wird bearbeitet und ein CancelRequest vom Client ist eingetroffen, aber der ActionServer hat das Ziel noch nicht abgebrochen.

Abschließend gehen die temporären Serverzustände in die folgenden endgültigen Serverzustände über:

- Rejected - Das Ziel wurde vom ActionServer vor der Bearbeitung verworfen und kein CancelRequest ist vom Client eingetroffen;
- Succeeded - Das Ziel wurde erfolgreich vom ActionServer erreicht;

- Aborted - Das Ziel wurde vom ActionServer unvollendet beendet und kein CancelRequest ist vom Client eingetroffen;
- Recalled - Das Ziel wurde durch das Eintreffen eines neuen Ziels abgebrochen oder durch einem CancelRequest bevor mit der Bearbeitung des Ziel begonnen wurde;
- Preempted - Die Bearbeitung des Ziels wurde abgebrochen durch das Eintreffen eines neuen Ziels oder einem CancelRequest.

Nach der Implementierung des ActionServers ist der ActionClient des move_group Knoten von MoveIt! in der Lage, mit diesem zu kommunizieren. Die Funktionalitäten von MoveIt! und ros_fri können kombiniert werden, um beispielsweise mit MoveIt! für eine beliebige Zielpose eine Trajektorie zu generieren, welche an ros_fri übergeben wird, um im Joint-Position-Mode, Joint-Impedance-Mode oder im Cartesian-Impedance-Mode ausgeführt zu werden.

Der Zustandsautomat des ros_fri ist in Abbildung 3.19 illustriert. In den Betriebsmodi JOINT_POSITION_CONTROL, JOINT_IMPEDANCE_CONTROL und CART_IMPEDANCE_CONTROL werden ankommende Trajektorien ausgeführt. Die Trajektorien können beispielsweise mit einem Bahnplaner aus der OMPL generiert werden und haben den Vorteil, dass diese bereits auf Kollisionen überprüft wurden. In den Betriebsmodi JntVelGoal und JntPosGoal können die gewünschte Endgeschwindigkeiten oder Endpositionen direkt ros_fri mitgeteilt werden, welcher die passende Trajektorie mit der Type II Reflexes Motion Library berechnet und ausführt. Die berechneten Trajektorien werden jedoch nicht auf Kollisionen mit der Umgebung und auf Selbstkollisionen mit dem montierten Greifer überprüft. Diese Betriebsmodi eignen sich vor allem um auf sensorbasierte Ereignisse schnellstmöglich reagieren zu können. Beispielsweise kann nach der Detektion eines Hindernisses unmittelbar ein Abbruch der Trajektorie eingeleitet werden.

Ein oft auftretendes Problem ist, die berechnete Trajektorie der Bewegungsplanungsalgorithmen für viele Industriemanipulatoren, welche häufig zu wenig Trajektoriepunkte enthalten, wodurch der Manipulator während der Ausführung der Trajektorie gelegentlich ruckelt. Dieses Problem wurde mit einem Trajektorienfilter gelöst, welcher unter MoveIt! in einem planning request adapter zur Verfügung gestellt wird. Hierfür wurde in die ompl_planning_pipeline.launch.xml der UniformSampleFilter mit einer sample_duration von 10 ms eingefügt (siehe Abbildung 3.20).

Der eingefügte Trajektorienfilter nimmt zur Laufzeit die geplante Trajektorie vom Bewegungsplanungsalgorithmus entgegen und gibt eine modifizierte Trajektorie zurück. Die modifizierte Trajektorie enthält alle 10 ms Ausführungslaufzeit einen Trajektoriepunkt. Hierbei wurden 10 ms aufgrund der eingestellten Taktraten zwischen

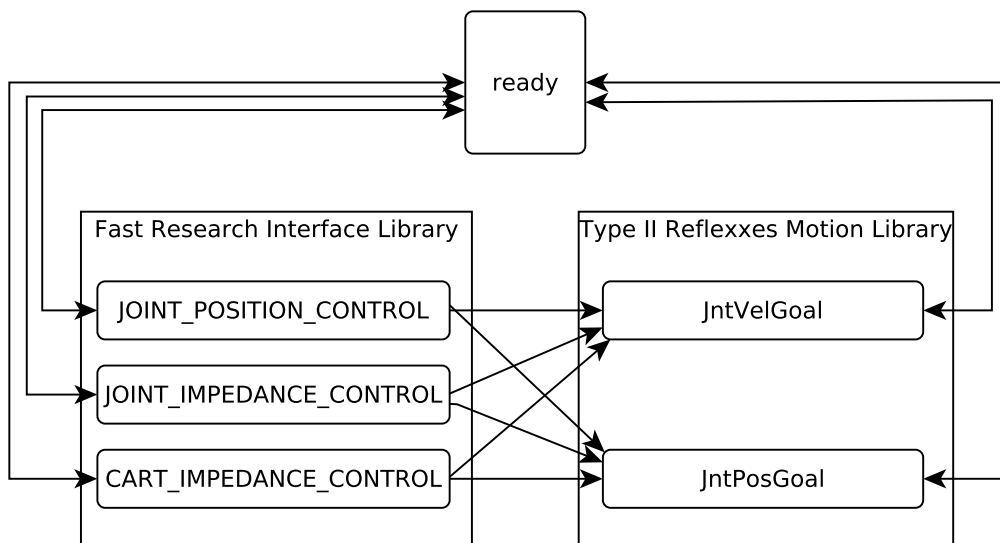


Abbildung 3.19: Abgebildet ist der Zustandsautomat des ros_fri.

```

<launch>

  <!-- OMPL Plugin for MoveIt! -->
  <arg name="planning_plugin" value="ompl_interface/OMPLPlanner" />
  <!-- The request adapters (plugins) used when planning with OMPL.
  ORDER MATTERS →
  <arg name="planning_adapters" value="
industrial_trajectory_filters/UniformSampleFilter
  default_planner_request_adapters/AddTimeParameterization
  default_planner_request_adapters/FixWorkspaceBounds
  default_planner_request_adapters/FixStartStateBounds
  default_planner_request_adapters/FixStartStateCollision
  default_planner_request_adapters/FixStartStatePathConstraints" />

  <arg name="start_state_max_bounds_error" value="0.1" />
  <param name="planning_plugin" value="$(arg planning_plugin)" />
  <param name="request_adapters" value="$(arg planning_adapters)" />
  <param name="start_state_max_bounds_error" value="$(arg start_state_max_bounds_error)" />
  <param name="sample_duration" value="0.01" />
  <roscpp command="load" file="$(find hydro_kuka_wsg50_moveit)/config/ompl_planning.yaml"/>

</launch>

```

Abbildung 3.20: Inhalt der Datei ompl_planning_pipeline.launch.xml. Ergänzungen sind fett markiert.

ros_fri bzw. FRI und dem KRC ausgewählt. Der KRC kann die Punkte der Trajektorie nur einzeln entgegennehmen, weshalb bei jedem Takt ein Trajektoriepunkt übergeben werden muss. Daher sollte eine Trajektorie idealerweise alle 10 ms Ausführungszeit einen Trajektoriepunkt enthalten. Bei einer höheren Ausführungszeit kann der Manipulator während der Ausführung Ruckeln, weil zwischen den einzelnen Trajektoriepunkten mal größere bzw. kleinere Distanzunterschiede vorliegen und zu einer erhöhten positiven bzw. negativen Beschleunigung führen. Hingegen verhält sich der Manipulator mit der gefilterten Trajektorie während der Ausführung geschmeidig. Nach der FRI Library kann der LWR mit einer Taktrate von 1-100 ms angesprochen werden. Das ros_fri wurde nur mit einer Taktrate von 10 ms betrieben. Der Betrieb sollte auch mit geringeren Taktraten funktionieren, wurde jedoch nicht getestet.

Wird während der Trajektorieausführung eine drohende Kollision detektiert, muss die Trajektorie abgebrochen werden. Hierfür wird ein Abbruchbefehl von move_group an ros_fri geschickt, welcher die Trajektorieabbruchphase initialisiert. Das bloße Einstellen des Sendevorgangs mit den Trajektoriepunkten an FRI führt durch die Auslösung einer Sicherheitsfunktion zu einem unmittelbaren Stoppen der Motoren. Daraufhin wird die Verbindung zwischen dem KRC und dem entfernten Rechner abgebrochen und die Ausführung von weiteren Operationen ist nicht mehr möglich. Deshalb ist die Berechnung und Ausführung einer neuen Trajektorie für das Stoppen des Manipulators in der Trajektorieabbruchphase erforderlich. Für die Trajektorieberechnung bleiben aufgrund derzeitiger Konfigurationen nur etwa 10 ms zur Verfügung, ansonsten wird das Stoppen der Motoren seitens des KRC eingeleitet. Die Planung der Trajektorie mit MoveIt! benötigt wesentlich höhere Berechnungszeiten und ist somit ungeeignet. Die Anforderung kann mit dem geschwindigkeitsbasierten Online-Trajektoriegenerator aus der Reflexxes Type II Motion Library erfüllt werden, welcher lediglich eine Berechnungszeit von maximal einer Millisekunde pro Kontrollzyklus benötigt. Die Eingabe- und Ausgabeparameter des Algorithmus sind in Abbildung 3.21 dargestellt. Neben der aktuellen Position und Geschwindigkeit der Gelenke werden als Eingabeparameter der Soll-Geschwindigkeitsvektor und der erlaubte Beschleunigungsparameter übergeben. Für den Abbremsvorgang wird als Soll-Geschwindigkeitsvektor der Nullvektor übergeben. Die Dauer und Positionsabweichung während des Abbremsvorgangs ist vom Beschleunigungsparameter abhängig. Je größer der erlaubte Beschleunigungsparameter, desto stärker erfolgt der Abbremsvorgang, welches wiederum zu einem kürzeren Abbremsweg bzw. einer kürzeren Abbremsdauer führt. Nach erfolgreichem Trajektorieabbruch besteht die Möglichkeit der Neuplanung der Trajektorie zur Erreichung der Zielkonfiguration. Dazu muss in MoveIt! nur das allow replanning flag gesetzt werden. Hierdurch ist es beispielsweise möglich, während der Trajektorieausführung in die Umgebung ein-

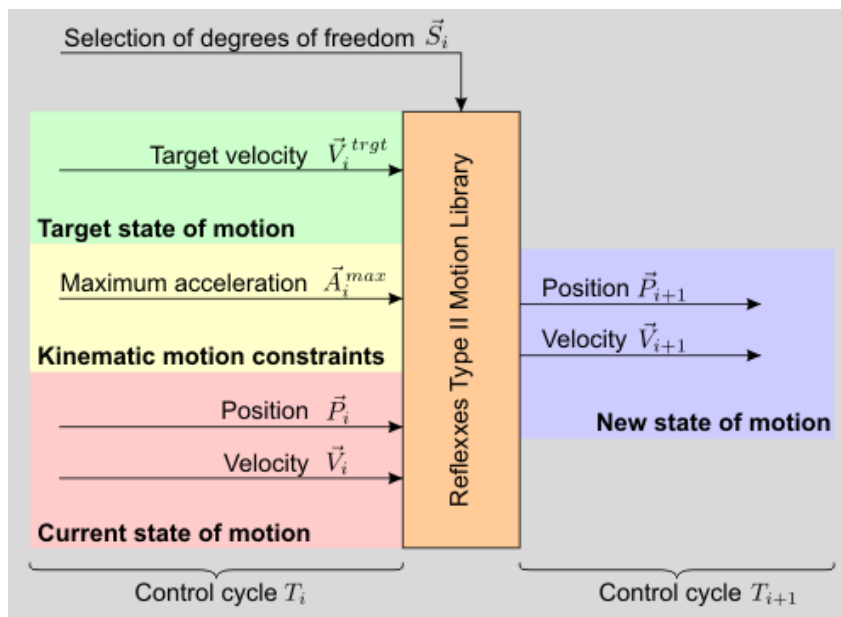


Abbildung 3.21: Dargestellt sind die Eingabe- und Ausgabeparameter des Geschwindigkeitsbasierten Online Trajektoriegenerators Algorithmus der Reflexxes Type II Motion Library. Abbildung entnommen aus [Reflexxes \[2014\]](#).

zugreifen und eine drohende Kollision zu verursachen. Nach der Aktualisierung der OctoMap wird die Trajektorie unverzüglich abgebrochen und mit der Neuplanung einer Trajektorie zur Erreichen des ursprünglichen Ziels begonnen.

Mit der Übernahme des Funktionsumfangs der FRI Library kann mit `ros_fri` auf notwendige Sensorinformationen für die Experimente zurückgegriffen werden. Beispielsweise kann mit den Momentensensoren des LWR die Ermittlung des Objektgewichtes erfolgen. Die externen Kräfte und Drehmomente, die auf das definierte Werkzeugkoordinatensystem des Manipulators einwirken, können dazu ausgelesen werden. Durch den Vergleich der gemessenen Kraftwerte in Z-Richtung vor und nach dem Greifen des Objektes, kann das Objektgewichtes mit einer Genauigkeit von ungefähr ± 0.1 kg berechnet werden. Überdies können mit den Momentensensoren die ausgeübten Kräfte des Menschen während der Objektübergabe gemessen werden.

3.7 Greifen und Platzieren

Das autonome Greifen und Platzieren von Objekten gehört zu den notwendigen elementaren Fähigkeiten eines Roboters. Im diesem Abschnitt werden die Zustände der entwickelten Pick & Place Demo näher erläutert.

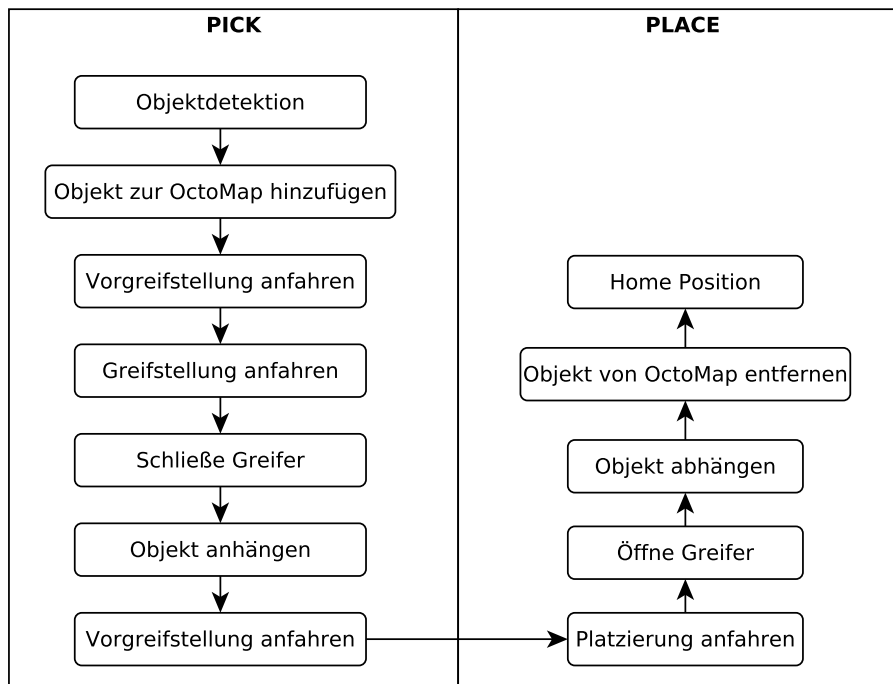


Abbildung 3.22: Abgebildet ist der Zustandsautomat der Pick & Place Demo.

In Abbildung 3.22 ist der Zustandsautomat der Pick & Place Demo illustriert. Für die Realisierung des Greifens muss das Objekt zuerst identifiziert werden. Anschließend muss eine kollisionsfreie Trajektorie zum Objekt gefunden werden, um das Objekt mit einem bestimmten Griff greifen zu können. Die Objektidentifikation findet mit der in Abschnitt 3.4 beschriebenen GStreamer Pipeline statt. Die gelieferten Objektinformationen beinhalten die Pose des Objektes und seine zuvor definierte Größe. Diese Daten reichen ohne die Berücksichtigung von Hindernissen in der Umgebung aus, um das Objekt zu greifen. Die Umgebung wird von MoveIt! mit einer periodisch aktualisierenden OctoMap modelliert. Das identifizierte Objekt wird in der OctoMap als Hindernis erkannt, wodurch jegliche Bahnplanung zur Pose des Objektes fehlschlägt. Um dies zu verhindern müssen die belegten Regionen im und um das erkannte Objekt in der OctoMap auf unbelegt gesetzt werden, welches automatisch durch das manuelle Einfügen des Objektes in die OctoMap geschieht. Hierdurch kann die Bewegungsplanung vor dem Greifen Hindernisse in einer dynamischen Umgebung berücksichtigen.

Vor dem Greifen eines Objektes wird der Greifer zunächst in eine Vorgreifstellung gefahren und danach in die Greifstellung. Die Vorgreifstellung ist notwendig damit die Finger des Greifers das Objekt nicht während der Trajektorieausführung berühren und zu einer leichten Verschiebung des Objektes führen. Eine Ursache dafür ist die leicht schwankende Pose des detektierten Objektes aus der GStreamer Pipeline.

Deshalb wird bei zukünftiger Weiterentwicklung des Robotersystems der Einsatz der Kinect mit ihrem Tiefensensor zur Objektdetektion empfohlen.

Das System ist mit dem verwendeten Programmcode von Dr. Hannes Bistry aus einer ähnlichen Arbeit in der Lage das Objekt mit verschiedenen Griffstellungen zu greifen. Hierfür wird der gewünschte Griff bzw. Pose des Greifers manuell festgelegt.

Nach dem Greifvorgang muss das Objekt an die kinematische Kette angehängt werden, damit der Bewegungsplaner das Objekt für die weitere Planung mitberücksichtigt, um Kollisionen mit Hindernissen zu vermeiden. Auf der MoveIt! Webseite wird zur Lösung des Problems Programmcode bereitgestellt, welcher jedoch nur in der Simulation mit RViz funktioniert hat. Recherchen haben ergeben, dass es sich um einen Fehler in der MoveIt! Software handelt. Weitere Versuche, das Problem zu lösen, wurden aufgegeben, weil MoveIt! sich noch in der Alpha Version 0.5 befindet und das Problem wahrscheinlich mit den nächsten Updates behoben wird.

Die Objektplatzierung zu vordefinierten Posen ist mit den bereits beschriebenen Implementierungen für das Greifen des Objektes sehr ähnlich und wird deshalb nicht weiter erläutert.

3.8 Sicherheitsmechanismen

In diesen Abschnitt werden die vorhandenen und neu hinzugefügten Sicherheitsmechanismen des Robotersystems beschrieben.

Der Hersteller KUKA liefert mit dem KRC einen Leistungsbegrenzungsschalter und einen Notausschalter. Der Leistungsbegrenzungsschalter begrenzt die Leistung der Motoren und der Notausschalter stoppt die Motoren bei Aktivierung. Der Anwender sollte letzteren immer Griffbereit halten, um auf unvorhersehbaren Ereignissen schnellstmöglich reagieren zu können. Weiterhin werden vom KRC die Motoren gestoppt, sobald die Kommunikation zwischen dem entfernten Rechner und dem KRC für kurze Zeit unterbrochen wird.

Eine weitere Sicherheitsmaßnahme ist die Reduzierung der Geschwindigkeits- und Beschleunigungswerte des KUKA LWR. Mit dem KRC kann die Geschwindigkeit bzw. Beschleunigung auf einen prozentualen Wert limitiert werden, welcher bei Überschreitung die Motoren des LWR stoppt. Um den Sicherheitsmechanismus jedoch erst gar nicht auszulösen, können mit MoveIt! für die Trajektoriengenerierung Geschwindigkeits- und Beschleunigungshöchstwerte festgelegt werden. Diese Limitierungen können für jedes einzelne Gelenk in der `joint_limits.yaml` der MoveIt! Konfigurationsdateien vorgenommen werden. Der LWR wird momentan mit einer Geschwindigkeit von 0.5 Rad/s und einer Beschleunigung von 0.2 Rad/s betrieben.

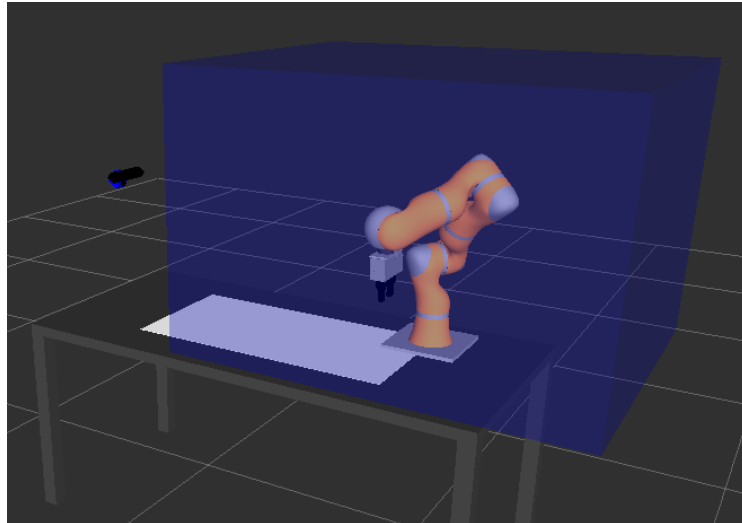


Abbildung 3.23: Abgebildet ist im blauen Quader der Arbeitsraum des LWR.

Obwohl die Bewegungen des Manipulators mit den Werten relativ schnell erscheinen, entspricht dies gerade einmal 15 % der maximalen Geschwindigkeit und 10 % der maximalen Beschleunigung des LWR. Von höheren Geschwindigkeiten wird aus Sicherheitsgründen abgeraten, weil dadurch die Reaktionszeit zur Betätigung des Notschalter herabgesetzt wird.

Weiterhin ist die Arbeitsraumbeschränkung des Manipulators sinnvoll, um den Bewegungsspielraum für den Menschen vorhersehbar zu machen. In [Abbildung 3.23](#) ist der Arbeitsraum des Manipulators als blauer Quader festgelegt worden. Die Arbeitsraumparameter werden dem Planer bei jeder Bewegungsplanungsanfrage mitgegeben, welcher anschließend versucht den Manipulator während der gesamten Zeit innerhalb des Arbeitsraumes zu halten [MoveIt!, 2014](#) [d].

Zudem müssen Sicherheitsvorkehrungen für den Fall getroffen werden, dass der Manipulator während der Trajektorienausführung mit dem Menschen zusammenstößt. Hierbei können Kollisionen entweder mit den Kamerasensoren oder den Armsensoren detektiert werden. Mit den Sensordaten der Kinect wird die OctoMap regelmäßig aktualisiert. MoveIt! überprüft nicht nur seine geplanten Trajektorien auf Kollisionen in der OctoMap vor der Trajektorienausführung, sondern auch währenddessen. Sobald eine Kollision droht, wird ein Abbruchbefehl an den Armcontroller `ros_fri` geschickt, um unmittelbar die Trajektorie zu stoppen. Diese Sicherheitsfunktion offenbart allerdings eine Schwachstelle bei der regelmäßigen Aktualisierung der OctoMap. Bei hoher Auslastung des Rechners wurde beobachtet, dass die Aktualisierungszeit der Umgebung manchmal mehrere Sekunden betragen hat. Woraufhin keine Hindernis entdeckt wurden und Kollisionen auftraten. Gelöst wurde das Problem durch die Verlagerung von Anwendungsprozessen auf einen zweiten Rechner. Hierdurch wur-

de zudem die Stabilität der Verbindung zwischen dem KRC und dem entfernten Rechner mit der FRI Software erhöht, welche ebenfalls aufgrund der Auslastung manchmal abgebrochen wurde.

Die limitierte Sichtweite des Systems ist ein weiteres Problem durch welches manchmal keine Hindernisse erkannt werden. Beispielsweise werden Hindernisse, die sich direkt hinter dem Manipulator befinden nicht detektiert, wodurch manchmal die darauf folgenden Trajektorien zu Kollisionen mit den Hindernissen geführt haben. Die Sensorfusion von mehrerer Sensoren aus verschiedenen Blickwinkel könnte für die Generierung eines 3D Umgebungsmodells verwendet werden, um damit die Kollisionsvermeidung weiter zu verbessern.

Mit den Armsensoren des LWR kann eine Kollision während der Trajektorienausführung durch den Vergleich der Soll- und Ist-Gelenkwerte feststellen werden. Falls die Abweichung einen bestimmten Schwellenwert überschreitet, ist wahrscheinlich eine Kollision aufgetreten. Während der Experimente hat sich gezeigt, dass der KRC die Motoren im Joint-Impedance-Mode und im Cartesian-Impedance-Mode bei vordefinierten Positionsabweichungen abschaltet. Im Joint-Position-Mode wurde der Sicherheitsmechanismus nicht beobachtet, weshalb seine Implementierung in das `ros_fri` Package angestrebt wurde. Bei den ersten Tests hat sich der Manipulator schon bei kleinsten Kollisionen ausgeschaltet, jedoch auch manchmal während der kollisionsfreien Trajektorienausführung. Der Grund für dieses Verhalten ist der gewählte Schwellenwert für die Positionsabweichung. Je geringer der Schwellenwert, desto empfindlicher reagiert der LWR auf Positionsabweichungen, welches zu einer früheren Kollisionsdetektion führt. Gleichzeitig erhöht sich jedoch die Wahrscheinlichkeit, dass der Mechanismus während der kollisionsfreien Trajektorienausführung vom KRC fälschlicherweise ausgelöst wird. Nach ein paar Versuchen wurde ein Schwellenwert gefunden, bei dem das gewünschte Verhalten ohne Beeinträchtigung der anderen Funktionalitäten auftrat. Allerdings gilt der Schwellenwert nur für die aktuelle Traglast und die festgelegten Beschleunigungs- und Geschwindigkeitswerte. Eine Abweichung müsste wahrscheinlich die Neuberechnung des Schwellenwertes erfordern, weshalb die Verwendung der beschriebenen Sicherheitsmaßnahme nur in bestimmten Situationen sinnvoll erscheint.

4 Experimentelle Ergebnisse

In Zusammenarbeit mit Dr. Norman Hendrich, Dr. Hannes Bistry und Prof. Dr. Jianwei Zhang wurden die ersten Experimente dieses Kapitels durchgeführt, um eine für den Menschen natürlich wirkende Interaktion mit dem Roboter zu schaffen. Die experimentellen Ergebnisse wurden unter dem Titel "Multi-modal clues for efficient human-robot object handover: a case study with elderly users" auf der RO-MAN 2014 Konferenz und der IROS 2014 Konferenz unter dem Titel "Natural Robot-Human Handover Combining Force and Tactile Sensors" veröffentlicht [Hendrich \[2014 a\]](#), [Hendrich \[2014 b\]](#). Im ersten Experiment liegt der Fokus auf den Einflussfaktoren Kraftaufwand und Kraftrichtung zur Objektübergabe bei Stillstand. Anschließend wird der Zusammenhang zwischen Objektgewicht und Auslösekraft bei unterschiedlichen Übergabestellungen, sowie mit und ohne Nachgiebigkeit untersucht. Im letzten Abschnitt wird die Objektübergabe im Laufe des Bewegungsablaufs kurz betrachtet. Die Abbildung [4.1](#) gewährt einen Überblick über die verwendeten Versuchsobjekte, Übergabestellungen und Übergabestrategien bei den Experimenten.

4.1 Objektübergabe bei Stillstand

In diesem Abschnitt wird die Objektübergabe während des Bewegungsstillstandes untersucht. Die Imitation der menschlichen Objektübergabe seitens des Roboters sollte eine für den Menschen natürlich wirkende Objektübergabe hervorrufen. In dieser Arbeit wird der Roboter nur in der Rolle des stationären bzw. mobilen Gebers und der Mensch in der Rolle des mobilen Nehmers betrachtet. Für den Fall des stationären Gebers sollte der Roboter idealerweise nach dem menschlichen Vorbild den Freigabezeitpunkt vor dem Kontakt mit dem Nehmer einleiten. Hierzu ist die Echtzeitdetektion des Objektes in der menschliche Hand erforderlich. Aufgrund des verhältnismäßig kleinen Nutzens von maximal 60 ms (siehe Abschnitt [2.7](#)) und dem sehr hohen technischen Einsatz für die Realisierung des Ansatzes, wird dieser verworfen. Weiterhin wird in dieser Arbeit auf den Einsatz von Kameras zur Unterstützung der Objektübergabe verzichtet.

Für die Ermittlung der ausgeübten Kräfte des Menschen während der Objektübergabe können die Momentensensoren des LWR und die taktilen Sensormatrizen des

Experiment	Objektübergabe bei Bewegungsstillstand	Untersuchung des Zusammenhangs zwischen Objektgewicht und Greifkraft	Objektübergabe im Laufe des Bewegungsablaufs
Versuchsobjekt	Nescafe Dose (70 g) Holzwürfel (20 g) Stahlscheibe (800 g) Bleibatterie (1600 g)	Nescafe Dose (70 g) gefüllt mit Gewichten zwischen 0.1 und 1 kg Holzwürfel (20 g)	Nescafe Dose (70 g)
Übergabestellung	P1	P1 & P2	-
Übergabestrategie	Joint-Position-Mode	Joint-Position-Mode Joint-Impedance-Mode	Joint-Position-Mode

Abbildung 4.1: Überblick der verwendeten Versuchsobjekte, Übergabestellungen und Übergabestrategien bei den Experimenten.

WSG-DSA verwendet werden. Letztere können bei einem gegriffenen Objekt allerdings nur für das Detektieren einer definierten unbekanntem Kraftausübungen eingesetzt werden. Die Bestimmung der Kraftstärke und Krafrichtung ist nicht möglich. Mit den Momentensensoren des LWR können die extern anliegenden Drehmomente aller Gelenke ausgelesen werden. Definiert man eine Übergabepose, so können die bekannten und neuen Messwerte miteinander verglichen werden, um zumindest grob die Krafrichtungen und Kraftstärken zu bestimmen. Wesentlich bessere Ergebnisse können mit der verfügbaren Funktion `GetEstimatedExternalCartForcesAndTorques(...)` aus der FRI Library erreicht werden, welche die externen Kräfte und Drehmomente für das definierte Werkzeugkoordinatensystem berechnet. Damit ist die Ermittlung der Krafrichtungen und Kraftstärken mit einer hohen Genauigkeit möglich, weshalb die letzte Messmethode für die Objektübergabeexperimente bei Stillstand eingesetzt wird.

Der Zustandsautomat für das Objektübergabeexperimente ist in [Abbildung 4.2](#) illustriert. Es gilt die Annahme, dass der Roboter das Objekt zuvor gegriffen und in eine für den Menschen erreichbare Position gebracht hat. Die Objektübergabe wird durch entsprechende Kraftausübung durch die Testpersonen ausgelöst. In diesem Experiment wird der benötigte Kraftaufwand und die Krafrichtungen für eine natürlich wirkende Objektübergabe untersucht. Der Roboter in der Rolle des Gebers ist nach dem menschlichem Vorbild für die Sicherheit während der Objektübergabe verantwortlich. Deshalb wird eine Sprachausgabe "Open the gripper now" verwendet, um die Versuchspersonen auf die bevorstehende Objektfreigabe hinzuweisen.

Für die ersten Experimente wurde ein fester Kraftwert eingesetzt um das Objekt (Nescafe Dose) freizugeben. Nach ein paar Testversuchen ist aufgefallen, dass zum Zeitpunkt der Objektfreigabe ein zu hoch eingestellter Kraftwert zu einer unmittel-

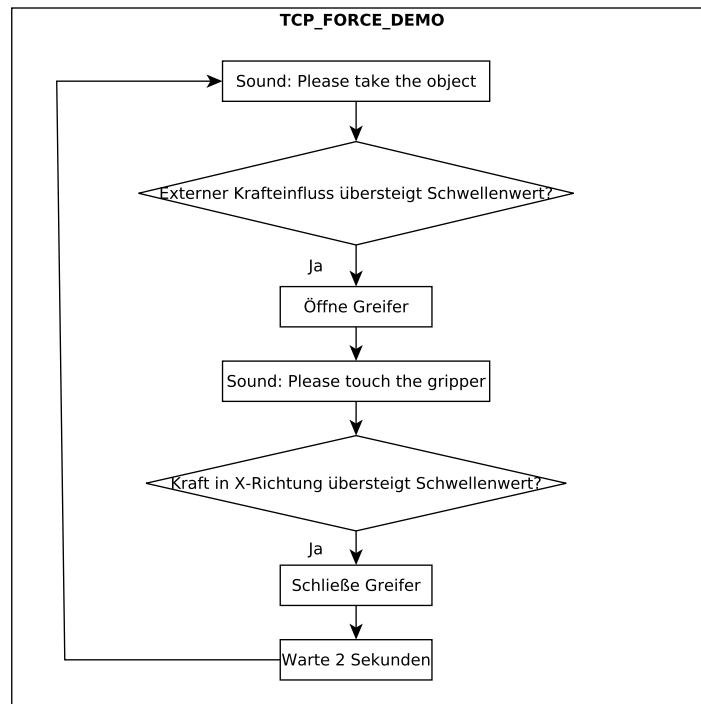


Abbildung 4.2: Zustandsautomat der TCP_FORCE_DEMO.

baren Stoßbewegung zur Krafrichtung führt. Ein zu niedrig eingestellter Kraftwert führt hingegen zu einem vorzeitigen Loslassen des Objektes, weil das Objektgewicht schon alleine den gewählten Schwellenwert übersteigt. Demnach muss der Kraftwert für jedes Objektgewicht angepasst werden. Diese Abhängigkeit kann durch das Messen des Objektgewichtes mit den Momentensensoren des Manipulators aufgelöst werden. Das Objektgewicht wird durch den Vergleich der Kraftwerte vor und nach dem Greifvorgang bestimmt. In den Experimenten wurden Messungenauigkeiten von ungefähr ± 0.1 kg beobachtet, wobei 2 Sekunden für den Messvorgang gewährt wurden. Bei einer kürzeren Messdauer traten manchmal erhebliche Messfehler auf. Der benötigte Kraftaufwand für die Objektfreigabe ergibt sich aus der Addition der Belastungskraft des Objektes und einer fest definierten Kraft von einem Newton.

In Abbildung 4.3 ist die Objektübergabe in zwei Phasen mit verschiedenen Objekten dargestellt. In der ersten Phase (grasp phase) übergibt der Mensch das Objekt an den Roboter, welcher die ausgeübten kartesischen Kräfte misst und bei Überschreitung des definierten Schwellenwertes, den Greifer schließt. In der zweiten Phase (release phase) nimmt der Mensch das Objekt vom Robotergreifer. Der Roboter ermittelt den Objektfreigabezeitpunkt ebenfalls durch die Auswertung der kartesischen Kräfte am TCP. In der Abbildung ist die kartesische Kraft in x,y,z -Richtung mit rot, blau und grün dargestellt. Deutlich erkennbar ist die Abhängigkeit zwischen dem Objektgewicht und dem benötigten Kraftaufwand um die verschiedenen Objekte

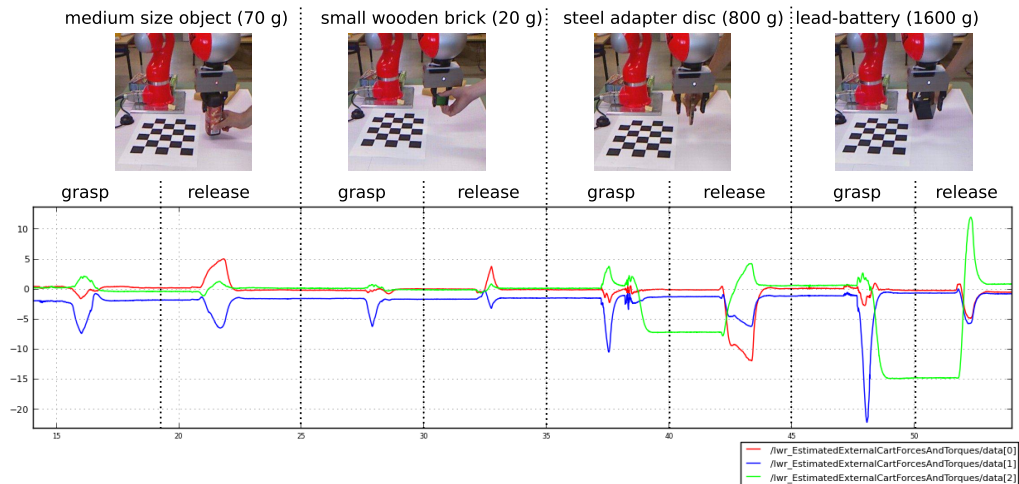


Abbildung 4.3: In der Abbildung wird in einer grasp und release Phase die Objektübergabe mit verschiedenen Objekten dargestellt. Die gemessenen kartesischen Kräfte mit der Maßeinheit Newton werden in x-Richtung (rot), y-Richtung (blau) und z-Richtung (grün) illustriert. Abbildung entnommen aus [Hendrich \[2014 a\]](#).

freizugeben. Für die beiden schwereren Objekte kann eine deutliche Übereinstimmung der aufgewendeten Kräfte in z-Richtung mit dem Objektgewicht beobachtet werden. Weiterhin ist die ausgeübte Kraftstärke und kartesische Kraftverteilung abhängig von der Art des Greifvorgangs des Menschen.

Für die in der Abbildung 4.3 dargestellten Übergabestellung wurde aus Sicherheitsbedenken in den ersten Experimenten nur die Kraftaufwendung in Richtung des Greifers erlaubt. Damit sollte die Objektfreigabe erst erfolgen, sobald der Mensch bereits genug Kraft aufgewendet hat, um das gesamte Objektgewicht halten zu können. In den Experimenten haben die Testpersonen jedoch meistens das Objekt bei der Objektfreigabe mit einer starken Stoßbewegung in Richtung des Greifers bewegt und diesen manchmal dabei berührt.

Für dieses Verhalten wurden drei Ursachen identifiziert. Einerseits benötigt der Roboter eine gewisse Berechnungszeit um die Bedingungen für die Objektfreigabe auszuwerten. Die Verzögerungszeit wird noch durch die zusätzliche definierte Zeit für die Sprachausgabe "Open the gripper now" verstärkt, weil mit der Objektübergabe gewartet wird bis die Sprachausgabe beendet ist. Der Mensch erhöht jedoch in der kurzen Zeitspanne seinen Kraftaufwand weiter und es kommt zur Überregelung. Ein weiterer Punkt scheint die Limitierung auf nur eine Krafrichtung zu sein. Der Mensch ist nicht in der Lage seine Kraft exakt in nur eine Richtung auszuüben, wodurch bei der Objektübergabe auch Kräfte auf die Finger des Greifers ausgeübt werden. Dadurch übt der Mensch mehr Kraftaufwand als notwendig aus. Anschließend

führt bei der Objektfreigabe ein Teil der aufgewendeten Kraft auf die Greiferfinger zu einer seitlichen Bewegung des Objektes. Der restliche Kraftteilwert addiert sich zur ausgeübten Kraft in Krafrichtung des Greifers, wodurch das Objekt ungewollt zum Greifer bewegt wird. Überdies muss noch berücksichtigt werden das der Roboter bei der Objektfreigabe das Objekt unverzüglich loslässt. Ein menschlicher Objektgeber hingegen gibt das Objekt kontinuierlich frei, wodurch ein leichtes Nachgeben des Objektes zur Krafrichtung auftritt. Der Objektnehmer registriert das Nachgeben und kann vor der abgeschlossenen Objektfreigabe seinen Kraftaufwand stabilisieren und somit ein Überschwenken des Objektes vermeiden.

In weiteren Versuchen sollten die Testpersonen die Objektübergabe mit einer, zwei und drei erlaubten Krafrichtungen miteinander vergleichen. Die meisten Testpersonen empfanden eine Limitierungen in den Krafrichtungen im Joint-Position-Mode als störend, weshalb in den weiteren Experimenten die Krafrichtungslimitierung nicht eingesetzt wurde. Im Joint-Impedance-Modus befürworteten hingegen einige Testpersonen eine Beschränkung der Richtungen, um die Richtung zum Auslösen der Objektfreigabe zu erfahren.

Überdies wurden die Verzögerungen während der Objektübergabe als störend empfunden. Die Hauptursache ist die abzuwartende Zeit für die Beendigung der Sprachausgabe vor der Objektübergabe. Diese wurde aus Sicherheitsüberlegungen implementiert um den Menschen vor der Objektfreigabe zu warnen. Jedoch wurde daraufhin die Verwendung des Kommunikationsmittels Sprache für die Objektübergabe in Frage gestellt. Die Sprachausgabe stellt für den Menschen spätestens nach der zweiten Objektübergabe keinerlei Informationsgewinn mehr dar, weil die auszugebende Nachricht bereits bekannt ist und der Mensch schon davor für die Objektübergabe bereit war. Das Kommunikationsmittel Sprache wird somit als überflüssig und störend empfunden. Die Erkenntnis kann durch die Beobachtung der Mensch zu Mensch Objektübergabe bestätigt werden, welche in der Regel ohne den Einsatz des vergleichsweise langsamen Kommunikationsmittels Sprache erfolgt.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslösekraft

In diesem Abschnitt wird der Zusammenhang zwischen dem Objektgewicht und der Auslösekraft einer natürlich wirkenden Objektübergabe untersucht. Hierbei werden für die Experimente die in Abbildung 4.4 dargestellten Objektübergabestellungen P1 und P2 verwendet. Beide Übergabestellungen unterscheiden sich lediglich in der Orientierung. Bei der Übergabestellung P2 wurden aus Sicherheitsüberlegungen die Greiferfinger parallel zum Boden ausgerichtet, damit bei der Objektfreigabe das

gegriffene Objektgewicht auf einen Greiferfinger verlagert wird. Dadurch sinkt die Wahrscheinlichkeit bei der Objektfreigabe das Objekt durch unterschätzen des Objektgewichtes oder durch vorzeitiges Loslassen fallen zu lassen.

Die Positionierung des Objektes durch den Objektgeber ist ein wesentlicher Einflussfaktor auf die darauf folgende Objektübergabe. In der Publikation "Human Preferences for Robot-Human Hand-over Configurations" haben die Autoren bezüglich der relativen Position der Objekte zum Menschen folgende Präferenzen festgestellt. Der Mensch bevorzugt höhere über unteren, linke über rechte und nahe über ferne Positionen. Die höhere Erreichbarkeit und Sichtbarkeit des Objektes ist vermutlich die Ursache für diese Präferenz. Bezüglich der Orientierung haben die Autoren empfohlen die am häufigsten vorkommende Orientierung des Objektes aus dem Alltag zu verwenden [Cakmak et al., 2011 \[a\]](#). Die Empfehlungen wurden in den folgenden Experimenten weitgehend berücksichtigt.

Für das erste Experiment wurde die bereits beschriebene TCP_FORCE_DEMO im Joint-Position-Modus eingesetzt. Als Versuchsobjekte wurde ein 20 Gramm (g) schwerer grüner Würfel, sowie eine 70 g Nescafe Dose verwendet, die mit Gewichten zwischen 100 g und 1000 g gefüllt wurde (siehe [Abbildung 4.4](#)). Für die Freigabe der Objekte muss eine Gesamtkraft bestehend aus der Belastungskraft des Objektes und einer definierten Kraft von 0.1, 0.5, 1, 2, 5, 7, 10, 12, 15 oder 20 N aufgebracht werden. Für jede Kraft und Objektkombination wurde die Objektübergabe von den Versuchspersonen immer vier mal durchgeführt. Während der Experimente sollten die Natürlichkeit der Objektübergabe beurteilt und in eine der drei Kategorien "gut", "akzeptabel" oder "schlecht" eingeordnet werden. Für die Auswertung der Daten werden im Weiteren mehrere Darstellungsarten verwendet. Die Experimentdaten sind in den [Abbildungen von 4.5 bis 4.8](#) tabellarisch dargestellt. In den [Abbildungen von 4.9 bis 4.11](#) werden die Versuchsdaten jeder Testperson einzeln in der Übergabestellung P1 im Joint-Position Modus illustriert, um Abweichungen zwischen diesen aufzuzeigen. Die durchschnittlichen Ergebnisse werden in den [Abbildungen von 4.12 bis 4.15](#) gezeigt und der Vergleich der Übergabestellungen und Übergabestrategien wird in den [Abbildungen 4.16 bis 4.19](#) illustriert. Anschließend werden die drei Kategorien "gut", "akzeptabel" und "schlecht" mit schlecht = 0 Punkte, akzeptabel = 0.5 Punkte und gut = 1.0 Punkte in numerische Scores umgerechnet und als Boxplots für jeden Benutzer einzeln und für alle zusammen in den [Abbildungen von 4.24 bis 4.23](#) dargestellt. Zudem werden in den [Abbildungen 4.25 bis 4.29](#) die numerischen Scores, die Gewichte und die Greifkräfte in dreidimensionalen Diagrammen abgebildet. Jedes Diagramm illustriert zusätzlich auf der -3 Achse des Score, eine zweidimensionale Darstellung der Daten. Die Versuchsreihen für 0.1 N wurden nicht illustriert, weil aufgrund von technischen Messschwankungen häufig das Objekt vorzeitig losgelassen und als Fehlversuch klassifiziert wurde. Allerdings

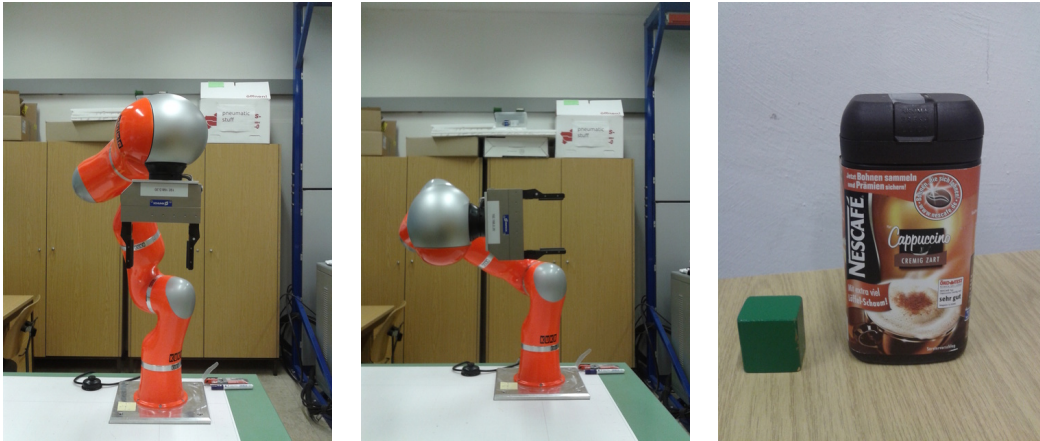


Abbildung 4.4: Dargestellt ist auf der linken Seite die Übergabestellung P1 und in der mittleren Abbildung die Übergabestellung P2. Beide Übergabestellungen unterscheiden sich nur in der Orientierung. In der rechten Abbildung sind die beiden Versuchsobjekte, ein grüner Würfel und eine Nescafé Dose abgebildet.

wurde beobachtet, dass die Versuchsreihen immer als "gut" klassifiziert wurden. Einige Versuchspersonen äußerten jedoch manchmal, dass der benötigten Kraftaufwand für Sie zu gering sei. Der Effekt trat allerdings ab den 0.5 N Versuchsreihen nicht mehr auf und wurde daher nicht weiter untersucht.

4 Experimentelle Ergebnisse

	gut										schlecht										akzeptabel								
05N	24g*	100g	200g	300g	400g	600g	800g	1000g		05N	24g*	100g	200g	300g	400g	600g	800g	1000g		05N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	3	3	3	2	3	3	1	0		T1	0	0	0	0	0	1	0	0		T1	1	1	0	2	1	0	3	3	
T2	4	4	4	4	4	4	4	4		T2	0	0	0	0	0	0	0	0		T2	0	0	0	0	0	0	0	0	
T3	4	4	4	4	4	4	4	4		T3	0	0	0	0	0	0	0	0		T3	0	0	0	0	0	0	0	0	
T4	4	4	4	4	4	4	4	4		T4	0	0	0	0	0	0	0	0		T4	0	0	0	0	0	0	0	0	
1N	24g*	100g	200g	300g	400g	600g	800g	1000g		1N	24g*	100g	200g	300g	400g	600g	800g	1000g		1N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	3	4	2	2	3	2	4	3		T1	0	0	0	2	1	0	0	0		T1	1	0	1	0	0	2	0	1	
T2	4	4	4	4	4	4	4	4		T2	0	0	0	0	0	0	0	0		T2	0	0	0	0	0	0	0	0	
T3	4	4	4	4	4	4	4	4		T3	0	0	0	0	0	0	0	0		T3	0	0	0	0	0	0	0	0	
T4	4	4	4	3	4	4	4	4		T4	0	0	0	0	0	0	0	0		T4	0	0	0	1	0	0	0	0	
2N	24g*	100g	200g	300g	400g	600g	800g	1000g		2N	24g*	100g	200g	300g	400g	600g	800g	1000g		2N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	1	2	3	3	4	3	3	3		T1	0	0	0	0	0	0	1	1		T1	3	2	1	1	0	1	0	0	
T2	4	4	4	4	4	4	4	4		T2	0	0	0	0	0	0	0	0		T2	0	0	0	0	0	0	0	0	
T3	3	3	4	3	3	4	4	4		T3	0	0	0	0	0	0	0	0		T3	1	1	0	1	1	0	0	0	
T4	4	4	2	4	2	1	4	4		T4	0	0	0	0	0	0	0	0		T4	0	0	2	0	2	3	0	0	
5N	24g*	100g	200g	300g	400g	600g	800g	1000g		5N	24g*	100g	200g	300g	400g	600g	800g	1000g		5N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	1	1	0	0	0	1	3	4		T1	1	0	0	1	0	0	0	0		T1	2	3	4	3	4	3	1	0	
T2	0	1	3	4	3	4	4	4		T2	0	0	0	0	0	0	0	0		T2	4	3	1	0	1	0	0	0	
T3	2	1	4	0	0	4	4	4		T3	0	0	0	0	0	0	0	0		T3	2	3	0	4	4	0	0	0	
T4	0	2	1	4	0	4	2	4		T4	0	0	0	0	0	0	0	0		T4	4	2	3	0	4	0	2	0	
7N	24g*	100g	200g	300g	400g	600g	800g	1000g		7N	24g*	100g	200g	300g	400g	600g	800g	1000g		7N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	0	0	0	0	0	0	0	0		T1	0	3	1	3	3	1	2	2		T1	3	1	3	1	1	3	2	2	
T2	0	0	0	2	2	2	2	2		T2	0	3	0	0	0	0	0	0		T2	4	1	4	2	2	2	2	2	
T3	1	2	4	3	0	3	3	3		T3	0	0	0	0	3	0	0	0		T3	3	2	0	1	1	1	1	1	
T4	0	1	2	0	4	1	2	4		T4	0	0	0	0	0	0	0	0		T4	4	3	2	4	0	3	2	0	
10N	24g*	100g	200g	300g	400g	600g	800g	1000g		10N	24g*	100g	200g	300g	400g	600g	800g	1000g		10N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	0	0	0	0	0	0	0	0		T1	2	4	4	4	4	1	3	2		T1	2	0	0	0	0	3	1	2	
T2	0	0	0	0	0	0	1	0		T2	1	4	3	0	0	1	0	0		T2	3	0	1	4	4	3	3	4	
T3	0	0	0	0	0	0	0	0		T3	0	4	2	0	0	1	0	0		T3	4	0	2	4	4	3	4	4	
T4	0	0	0	0	1	0	0	4		T4	1	0	0	0	0	0	0	0		T4	3	4	4	4	3	4	4	0	
12N	24g*	100g	200g	300g	400g	600g	800g	1000g		12N	24g*	100g	200g	300g	400g	600g	800g	1000g		12N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	0	0	0	0	0	0	0	0		T1	4	4	4	4	4	4	4	4		T1	0	0	0	0	0	0	0	0	
T2	0	0	0	0	0	0	0	0		T2	4	4	4	2	4	2	0	0		T2	0	0	0	2	0	2	4	4	
T3	0	0	0	0	0	0	0	0		T3	1	4	4	1	3	3	0	1		T3	3	0	0	3	1	1	4	3	
T4	0	0	0	0	1	0	0	3		T4	1	4	0	0	0	0	0	0		T4	3	0	4	4	3	4	4	1	
15N	24g*	100g	200g	300g	400g	600g	800g	1000g		15N	24g*	100g	200g	300g	400g	600g	800g	1000g		15N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	0	0	0	0	0	0	0	0		T1	4	4	4	4	4	4	4	4		T1	0	0	0	0	0	0	0	0	
T2	0	0	0	0	0	0	0	0		T2	4	4	4	4	4	4	1	4		T2	0	0	0	0	0	0	3	0	
T3	0	0	0	0	0	0	0	0		T3	2	4	4	1	4	3	1	1		T3	2	0	0	3	0	1	3	3	
T4	0	0	0	0	0	0	0	3		T4	3	0	0	0	0	0	0	0		T4	1	4	4	4	4	4	4	1	
20N	24g*	100g	200g	300g	400g	600g	800g	1000g		20N	24g*	100g	200g	300g	400g	600g	800g	1000g		20N	24g*	100g	200g	300g	400g	600g	800g	1000g	
T1	0	0	0	0	0	0	0	0		T1	4	4	4	4	4	4	4	4		T1	0	0	0	0	0	0	0	0	
T2	0	0	0	0	0	0	0	0		T2	4	4	4	4	4	4	4	4		T2	0	0	0	0	0	0	0	0	
T3	0	0	0	0	0	0	0	0		T3	4	4	4	4	4	4	4	3		T3	0	0	0	0	0	0	0	1	
T4	0	0	0	0	0	0	0	0		T4	4	4	4	1	1	4	1	0		T4	0	0	0	3	3	0	3	4	

Abbildung 4.5: Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P1 im Joint-Position Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslöskraft

	gut					schlecht					akzeptabel			
0.5 N	24g*	100g	300g	600g	0.5 N	24g*	100g	300g	600g	0.5 N	24g*	100g	300g	600g
T1	4	4	4	4	T1	0	0	0	0	T1	0	0	0	0
T2	4	4	4	4	T2	0	0	0	0	T2	0	0	0	0
T3	1	2	4	4	T3	0	0	0	0	T3	3	2	0	0
T4	4	4	4	4	T4	0	0	0	0	T4	0	0	0	0
1 N	24g*	100g	300g	600g	1 N	24g*	100g	300g	600g	1 N	24g*	100g	300g	600g
T1	4	1	4	4	T1	0	0	0	0	T1	0	3	0	0
T2	4	4	4	4	T2	0	0	0	0	T2	0	0	0	0
T3	2	4	3	4	T3	0	0	0	0	T3	2	0	1	0
T4	4	4	4	4	T4	0	0	0	0	T4	0	0	0	0
2 N	24g*	100g	300g	600g	2 N	24g*	100g	300g	600g	2 N	24g*	100g	300g	600g
T1	4	0	2	3	T1	0	2	0	0	T1	0	2	2	1
T2	4	4	4	4	T2	0	0	0	0	T2	0	0	0	0
T3	0	0	0	2	T3	2	0	0	0	T3	2	4	4	2
T4	0	0	4	4	T4	0	0	0	0	T4	4	4	0	0
5 N	24g*	100g	300g	600g	5 N	24g*	100g	300g	600g	5 N	24g*	100g	300g	600g
T1	4	0	0	0	T1	0	0	1	0	T1	0	4	3	4
T2	4	2	2	2	T2	0	0	0	0	T2	0	2	2	2
T3	0	0	1	1	T3	4	4	2	1	T3	0	0	1	2
T4	0	0	0	4	T4	0	0	0	0	T4	4	4	4	0
7 N	24g*	100g	300g	600g	7 N	24g*	100g	300g	600g	7 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	0	3	1	0	T1	4	1	3	4
T2	0	0	0	0	T2	1	0	0	0	T2	3	4	4	4
T3	0	0	0	0	T3	4	4	4	3	T3	0	0	0	1
T4	0	0	0	0	T4	0	4	0	0	T4	4	0	4	4
10 N	24g*	100g	300g	600g	10 N	24g*	100g	300g	600g	10 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	4	0	0	T1	0	0	4	4
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	3	T3	0	0	0	1
T4	0	0	0	0	T4	0	4	0	0	T4	4	0	4	4
12 N	24g*	100g	300g	600g	12 N	24g*	100g	300g	600g	12 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	4	3	1	T1	0	0	1	3
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	3	4	0	1	T4	1	0	4	3
15 N	24g*	100g	300g	600g	15 N	24g*	100g	300g	600g	15 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	4	4	1	T1	0	0	0	3
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	4	4	3	4	T4	0	0	1	0
20 N	24g*	100g	300g	600g	20 N	24g*	100g	300g	600g	20 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	4	4	4	T1	0	0	0	0
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	4	4	4	4	T4	0	0	0	0

Abbildung 4.6: Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P2 im Joint-Position Modus.

4 Experimentelle Ergebnisse

	gut										schlecht										akzeptabel								
0.5N	24g*	100g	200g	300g	400g	600g	800g	1000g	0.5N	24g*	100g	200g	300g	400g	600g	800g	1000g	0.5N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	1	0	1	4	3	4	4	4	T1	0	0	0	0	0	0	0	0	T1	3	4	3	0	0	0	0	0			
T2	4	4	4	4	4	4	4	4	T2	0	0	0	0	0	0	0	0	T2	0	0	0	0	0	0	0	0			
T3	4	4	4	4	4	4	4	4	T3	0	0	0	0	0	0	0	0	T3	0	0	0	0	0	0	0	0			
T4	4	4	4	4	4	4	4	4	T4	0	0	0	0	0	0	0	0	T4	0	0	0	0	0	0	0	0			
1N	24g*	100g	200g	300g	400g	600g	800g	1000g	1N	24g*	100g	200g	300g	400g	600g	800g	1000g	1N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	2	4	4	4	4	2	4	T1	1	0	0	0	0	0	0	0	T1	3	2	0	0	0	0	2	0			
T2	3	4	4	4	4	4	4	4	T2	0	0	0	0	0	0	0	0	T2	1	0	0	0	0	0	0	0			
T3	4	4	4	4	4	4	4	4	T3	0	0	0	0	0	0	0	0	T3	0	0	0	0	0	0	0	0			
T4	2	4	3	4	0	4	4	4	T4	0	0	0	0	0	0	0	0	T4	2	0	1	0	4	0	0	0			
2N	24g*	100g	200g	300g	400g	600g	800g	1000g	2N	24g*	100g	200g	300g	400g	600g	800g	1000g	2N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	1	0	0	3	3	3	4	4	T1	1	1	0	0	1	1	0	0	T1	2	3	4	1	0	0	0	0			
T2	4	4	4	4	4	4	4	4	T2	0	0	0	0	0	0	0	0	T2	0	0	0	0	0	0	0	0			
T3	3	1	3	4	0	3	4	4	T3	0	0	0	0	0	0	0	0	T3	1	3	1	0	4	0	0	0			
T4	0	2	3	4	2	3	2	2	T4	0	0	0	0	0	0	0	0	T4	4	2	1	0	2	1	2	2			
5N	24g*	100g	200g	300g	400g	600g	800g	1000g	5N	24g*	100g	200g	300g	400g	600g	800g	1000g	5N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	0	1	0	3	2	2	2	T1	2	0	0	0	0	0	0	0	T1	2	3	3	4	1	2	2	2			
T2	0	1	0	0	2	2	0	2	T2	2	0	0	0	0	0	0	0	T2	2	3	4	4	2	2	4	2			
T3	2	0	0	0	0	4	3	3	T3	0	1	1	0	0	0	0	0	T3	2	3	3	4	4	0	1	1			
T4	0	0	1	2	1	4	0	4	T4	0	0	0	0	0	0	0	0	T4	4	4	3	2	3	0	4	0			
7N	24g*	100g	200g	300g	400g	600g	800g	1000g	7N	24g*	100g	200g	300g	400g	600g	800g	1000g	7N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	0	0	2	0	4	2	1	T1	3	1	2	0	1	0	1	0	T1	1	3	2	2	3	0	1	3			
T2	0	0	0	0	0	0	0	2	T2	3	3	4	3	0	1	0	0	T2	1	1	0	1	4	3	4	2			
T3	0	0	0	0	0	2	0	1	T3	0	4	3	2	2	0	0	0	T3	4	0	1	2	2	2	4	3			
T4	0	0	0	0	1	0	0	0	T4	1	0	0	0	0	0	0	0	T4	3	4	4	4	3	4	4	4			
10N	24g*	100g	200g	300g	400g	600g	800g	1000g	10N	24g*	100g	200g	300g	400g	600g	800g	1000g	10N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	0	0	0	0	0	1	0	T1	4	4	3	1	1	4	1	0	T1	0	0	1	3	2	0	2	4			
T2	0	0	0	0	0	0	0	0	T2	4	4	4	4	4	4	3	0	T2	0	0	0	0	0	0	1	4			
T3	0	0	0	0	0	1	0	0	T3	1	4	4	4	4	0	0	0	T3	3	0	0	0	0	3	4	4			
T4	0	0	0	0	0	0	0	4	T4	4	3	0	0	0	0	4	0	T4	0	1	4	4	4	4	0	0			
12N	24g*	100g	200g	300g	400g	600g	800g	1000g	12N	24g*	100g	200g	300g	400g	600g	800g	1000g	12N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	0	0	0	0	0	0	0	T1	4	4	4	4	4	4	4	4	T1	0	0	0	0	0	0	0	0			
T2	0	0	0	0	0	0	0	0	T2	4	4	4	4	4	4	4	2	T2	0	0	0	0	0	0	0	2			
T3	0	0	0	0	0	0	0	0	T3	2	4	4	4	4	4	3	3	T3	2	0	0	0	0	0	1	1			
T4	0	0	0	0	0	0	0	1	T4	4	4	4	4	4	0	4	0	T4	0	0	0	0	0	4	0	3			
15N	24g*	100g	200g	300g	400g	600g	800g	1000g	15N	24g*	100g	200g	300g	400g	600g	800g	1000g	15N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	0	0	0	0	0	0	0	T1	4	4	4	4	4	4	4	4	T1	0	0	0	0	0	0	0	0			
T2	0	0	0	0	0	0	0	0	T2	4	4	4	4	4	4	4	4	T2	0	0	0	0	0	0	0	0			
T3	0	0	0	0	0	0	0	0	T3	4	4	4	4	4	4	4	3	T3	0	0	0	0	0	0	0	1			
T4	0	0	0	0	0	0	0	0	T4	4	4	4	4	4	4	4	0	T4	0	0	0	0	0	0	0	4			
20N	24g*	100g	200g	300g	400g	600g	800g	1000g	20N	24g*	100g	200g	300g	400g	600g	800g	1000g	20N	24g*	100g	200g	300g	400g	600g	800g	1000g			
T1	0	0	0	0	0	0	0	0	T1	4	4	4	4	4	3	4	4	T1	0	0	0	0	0	0	0	0			
T2	0	0	0	0	0	0	0	0	T2	4	4	4	4	4	4	4	4	T2	0	0	0	0	0	0	0	0			
T3	0	0	0	0	0	0	0	0	T3	4	4	4	4	4	4	4	4	T3	0	0	0	0	0	0	0	0			
T4	0	0	0	0	0	0	0	0	T4	4	4	4	4	4	4	4	4	T4	0	0	0	0	0	0	0	0			

Abbildung 4.7: Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P1 im Joint-Impedance Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslöskraft

	gut					schlecht					akzeptabel			
0.5 N	24g*	100g	300g	600g	0.5 N	24g*	100g	300g	600g	0.5 N	24g*	100g	300g	600g
T1	2	4	4	4	T1	0	0	0	0	T1	2	0	0	0
T2	4	4	4	4	T2	0	0	0	0	T2	0	0	0	0
T3	3	4	4	3	T3	0	0	0	0	T3	1	0	0	1
T4	4	4	4	4	T4	0	0	0	0	T4	0	0	0	0
1 N	24g*	100g	300g	600g	1 N	24g*	100g	300g	600g	1 N	24g*	100g	300g	600g
T1	1	2	2	3	T1	0	0	0	0	T1	3	2	2	1
T2	4	4	4	4	T2	0	0	0	0	T2	0	0	0	0
T3	0	0	2	3	T3	0	0	0	0	T3	4	4	2	1
T4	4	4	4	4	T4	0	0	0	0	T4	0	0	0	0
2 N	24g*	100g	300g	600g	2 N	24g*	100g	300g	600g	2 N	24g*	100g	300g	600g
T1	3	0	2	0	T1	0	0	0	0	T1	1	4	2	4
T2	4	4	4	4	T2	0	0	0	0	T2	0	0	0	0
T3	2	2	0	2	T3	1	2	0	0	T3	1	0	4	2
T4	3	1	4	4	T4	0	0	0	0	T4	1	3	0	0
5 N	24g*	100g	300g	600g	5 N	24g*	100g	300g	600g	5 N	24g*	100g	300g	600g
T1	0	0	0	4	T1	0	0	2	0	T1	4	4	2	0
T2	2	0	0	0	T2	0	0	2	0	T2	2	4	2	4
T3	0	0	1	0	T3	4	4	1	4	T3	0	0	2	0
T4	0	1	1	4	T4	0	0	0	0	T4	4	3	3	0
7 N	24g*	100g	300g	600g	7 N	24g*	100g	300g	600g	7 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	0	0	0	T1	0	4	4	4
T2	0	0	0	0	T2	0	4	4	0	T2	4	0	0	4
T3	0	0	0	0	T3	4	4	4	3	T3	0	0	0	1
T4	0	0	0	1	T4	0	0	0	0	T4	4	4	4	3
10 N	24g*	100g	300g	600g	10 N	24g*	100g	300g	600g	10 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	0	0	3	T1	0	4	4	1
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	4	4	0	0	T4	0	0	4	4
12 N	24g*	100g	300g	600g	12 N	24g*	100g	300g	600g	12 N	24g*	100g	300g	600g
T1	0	1	0	0	T1	4	0	4	4	T1	0	3	0	0
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	4	4	4	4	T4	0	0	0	0
15 N	24g*	100g	300g	600g	15 N	24g*	100g	300g	600g	15 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	4	4	4	T1	0	0	0	0
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	4	4	4	4	T4	0	0	0	0
20 N	24g*	100g	300g	600g	20 N	24g*	100g	300g	600g	20 N	24g*	100g	300g	600g
T1	0	0	0	0	T1	4	4	4	4	T1	0	0	0	0
T2	0	0	0	0	T2	4	4	4	4	T2	0	0	0	0
T3	0	0	0	0	T3	4	4	4	4	T3	0	0	0	0
T4	0	0	0	0	T4	4	4	4	4	T4	0	0	0	0

Abbildung 4.8: Abgebildet sind die tabellarischen Experimentdaten für die Übergabestellung P2 im Joint-Impedance Modus.

4 Experimentelle Ergebnisse

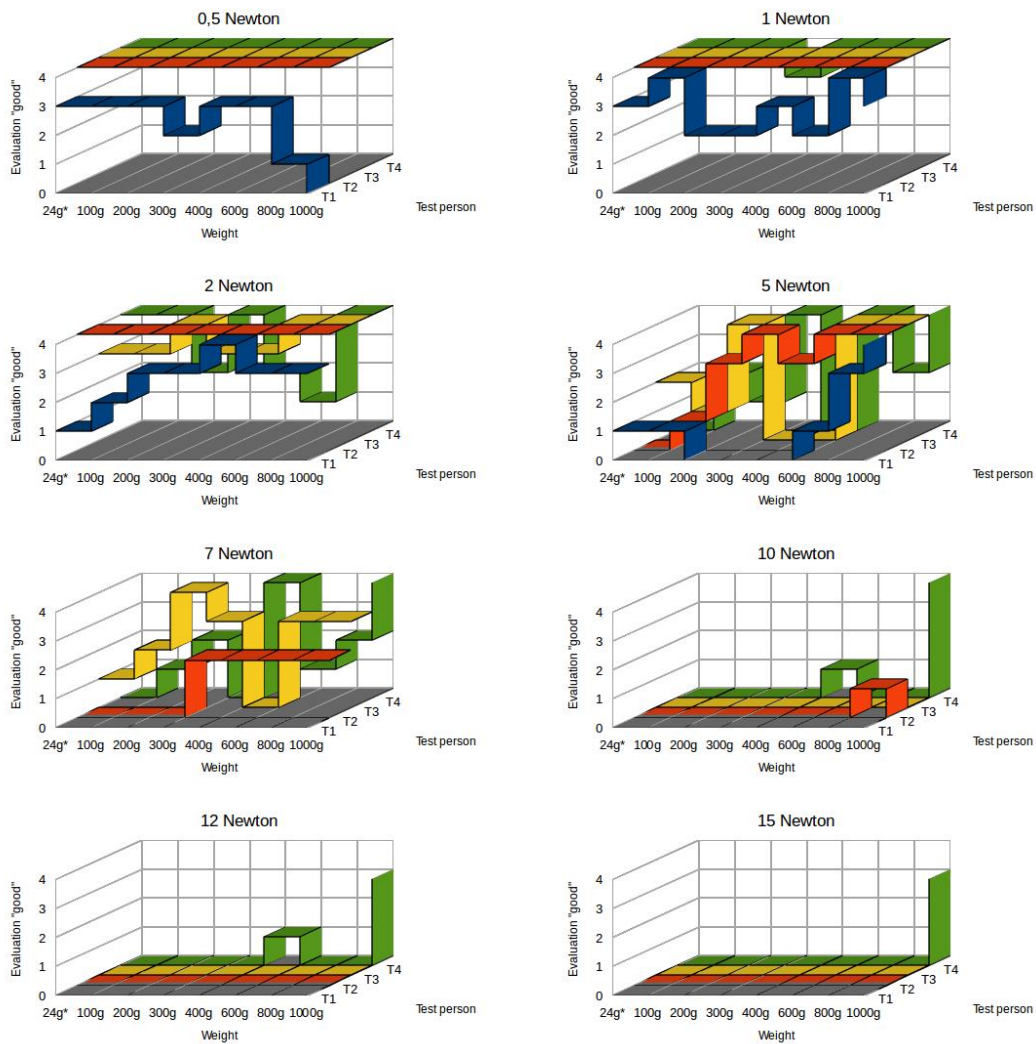


Abbildung 4.9: Abgebildet sind die mit "gut" bewerteten Versuchsreihen für die Übergabestelle P1 im Joint-Position Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslösekraft

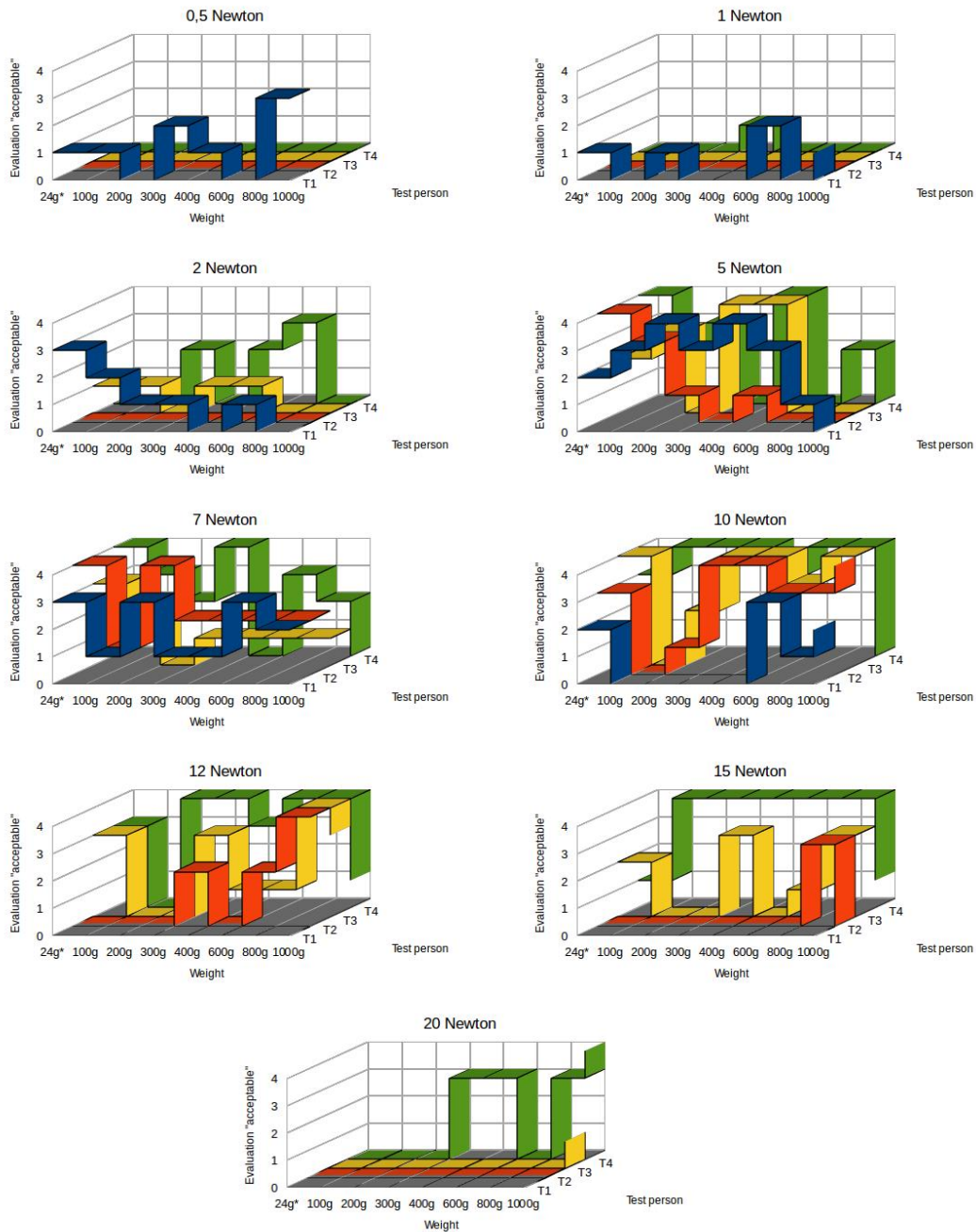


Abbildung 4.10: Abgebildet sind die mit "akzeptabel" bewerteten Versuchsreihen für die Übergabestelle P1 im Joint-Position Modus.

4 Experimentelle Ergebnisse

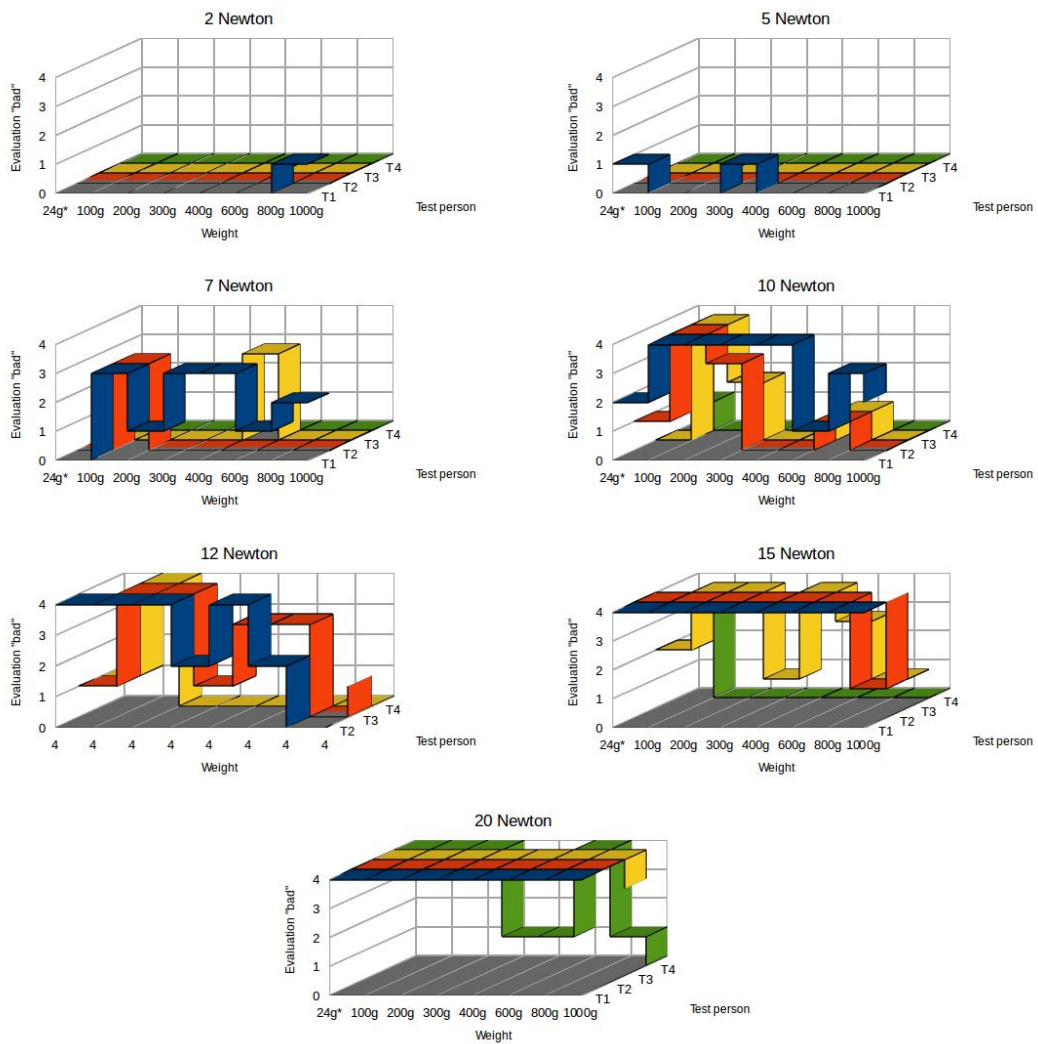


Abbildung 4.11: Abgebildet sind die mit "schlecht" bewerteten Versuchsreihen für die Übergabestelle P1 im Joint-Position Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslöskraft

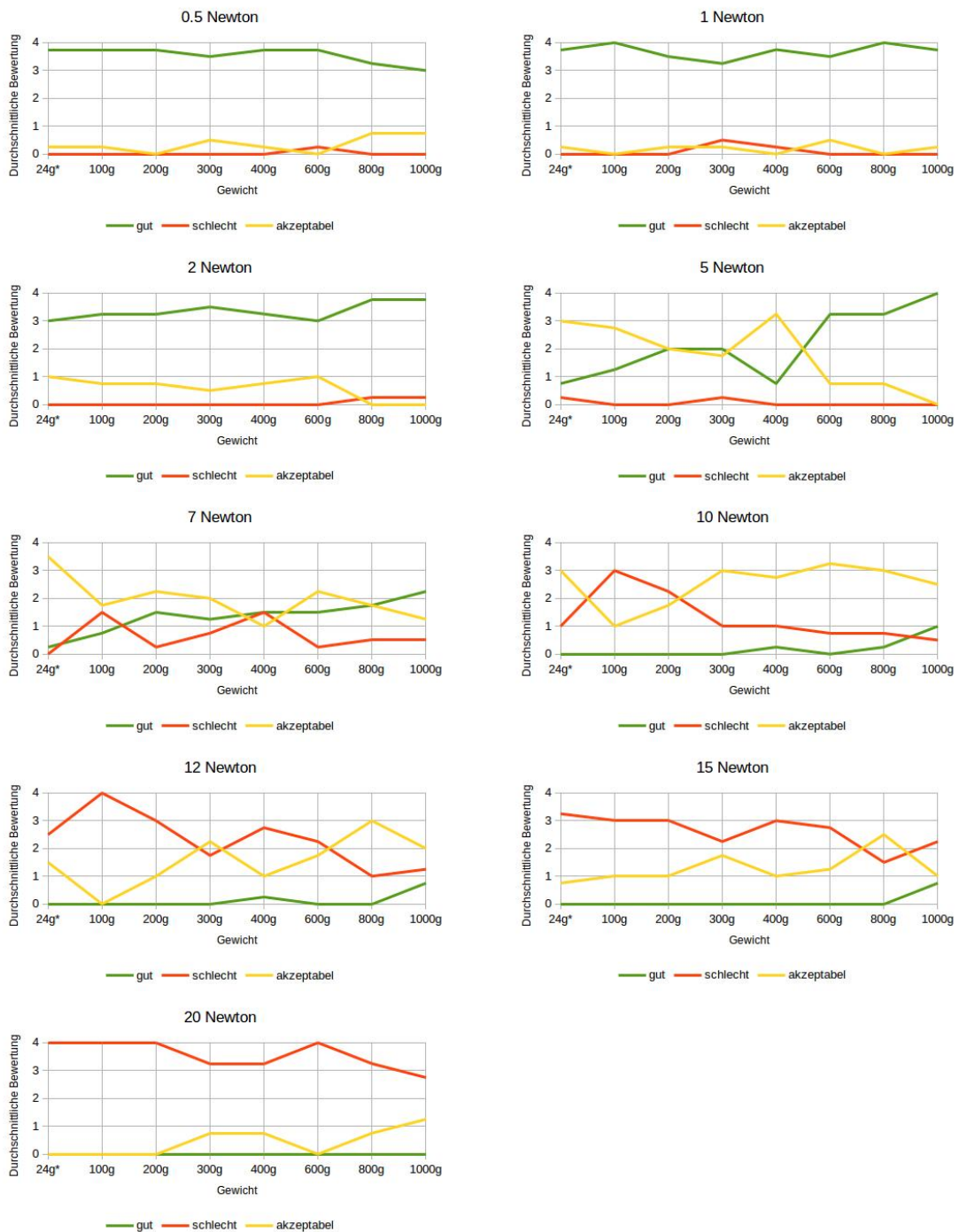


Abbildung 4.12: Abgebildet sind die Versuchsergebnisse für die Übergabestelle P1 im Joint-Position Modus. Dargestellt sind in den Diagrammen die durchschnittlichen Bewertungen gut (grün), akzeptabel (gelb) und schlecht (rot) für alle Objektgewichte. Für jede Greifkraft ist ein Diagramm illustriert.

4 Experimentelle Ergebnisse

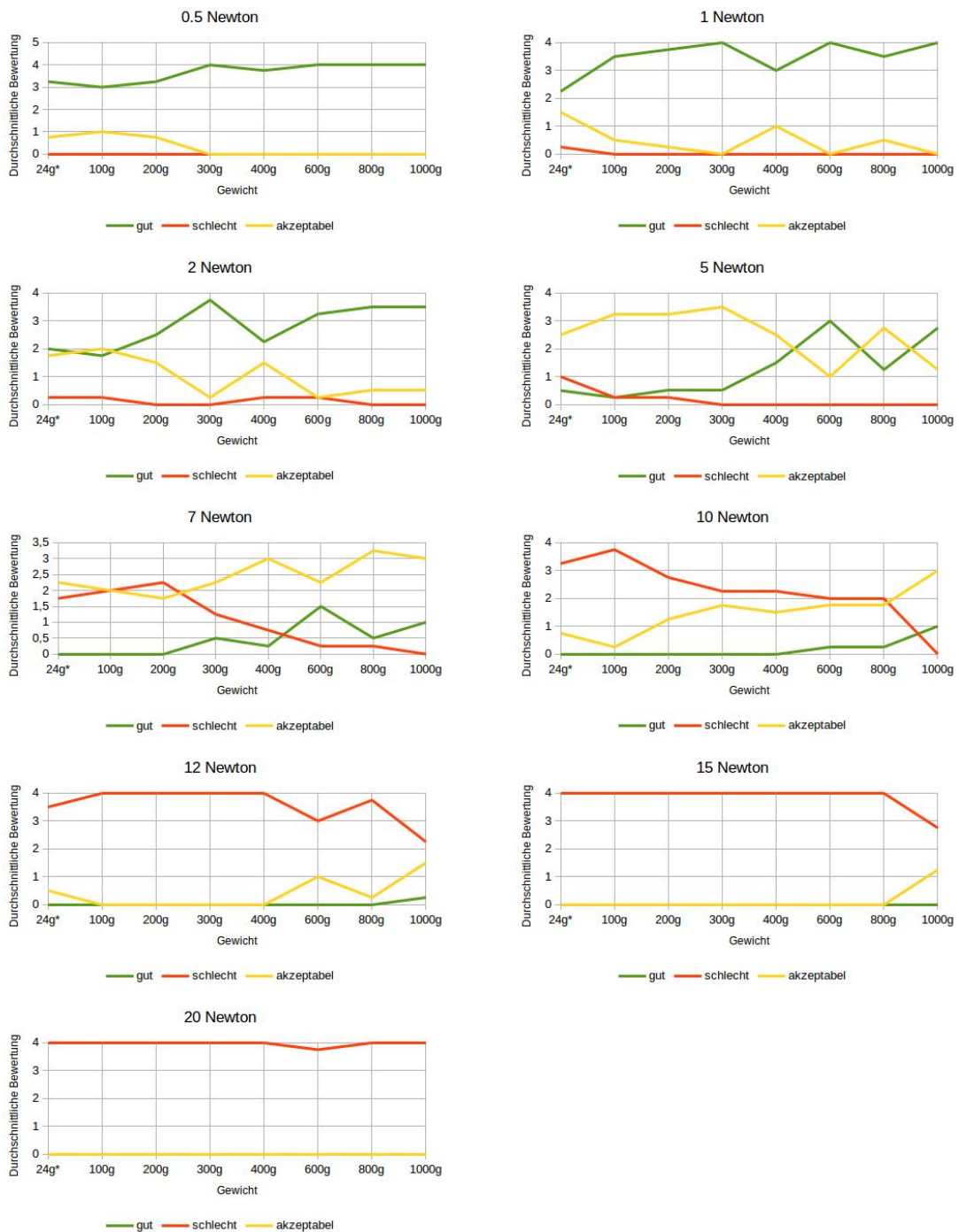


Abbildung 4.13: Abgebildet sind die Versuchsergebnisse für die Übergabestelle P1 im Joint-Impedance Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslöskraft

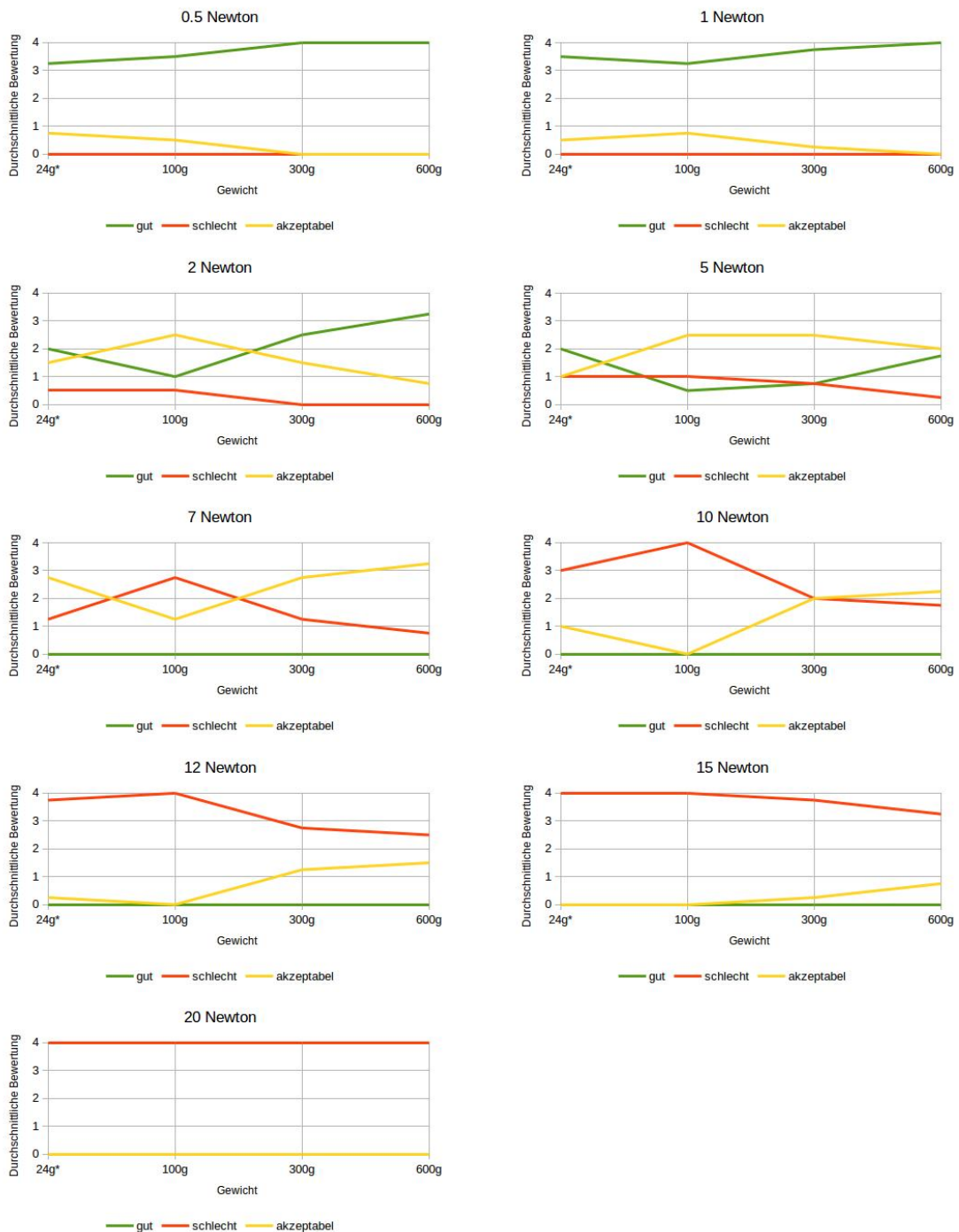


Abbildung 4.14: Abgebildet sind die Versuchsergebnisse für die Übergabestelle P2 im Joint-Position Modus.

4 Experimentelle Ergebnisse

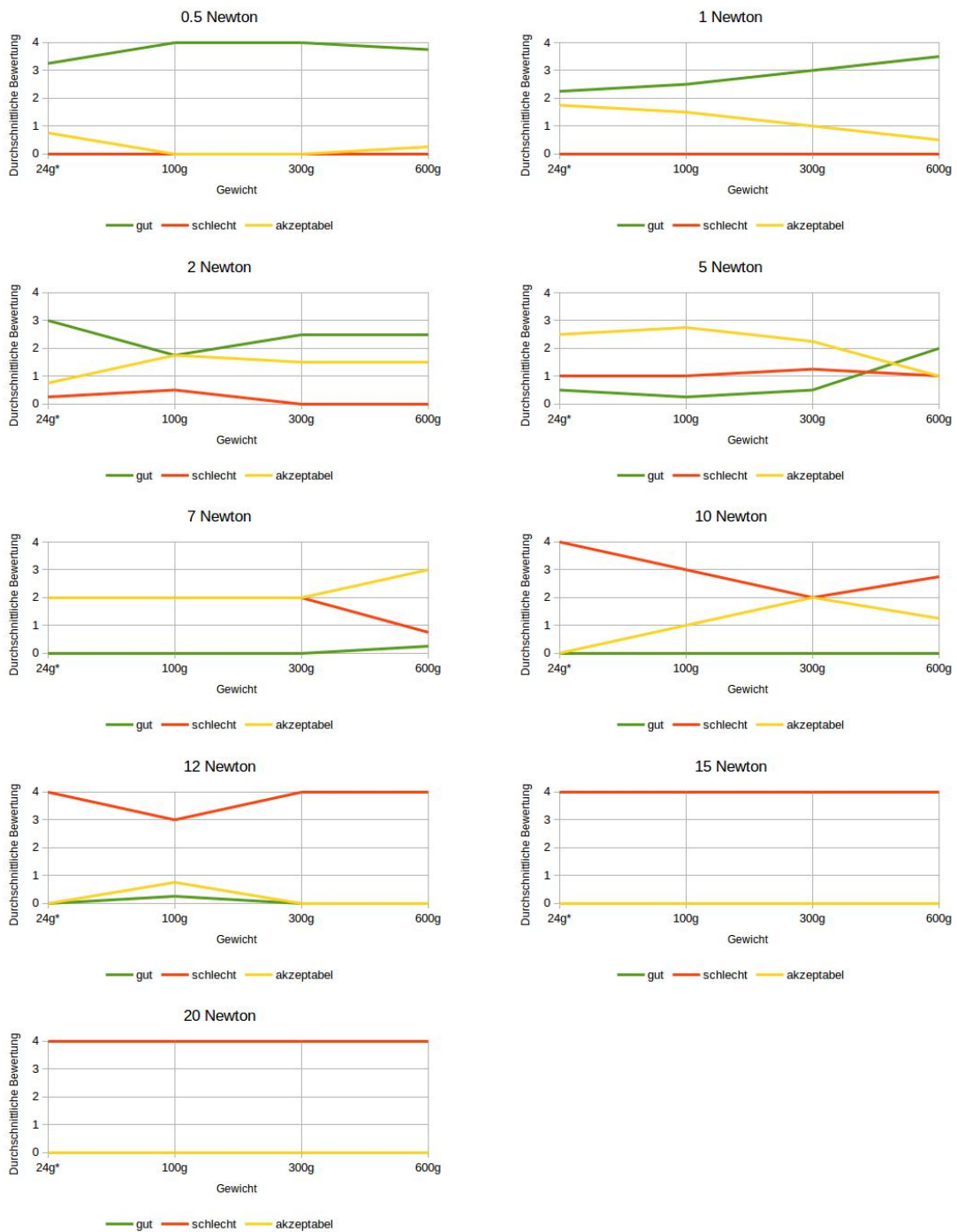


Abbildung 4.15: Abgebildet sind die Versuchsergebnisse für die Übergabestelle P2 im Joint-Impedance Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslöskraft

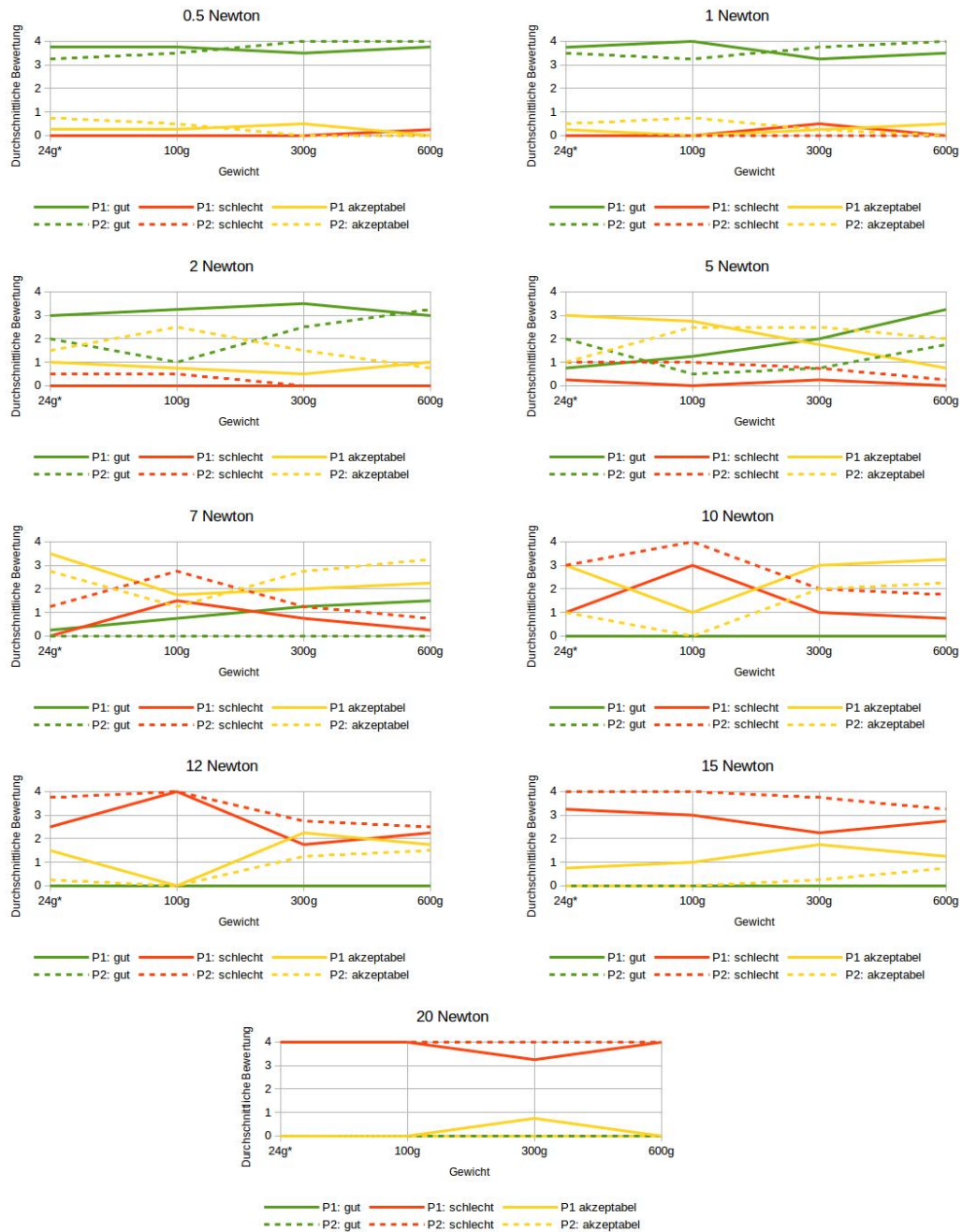


Abbildung 4.16: Vergleich der Versuchsergebnisse für die Übergabestelle P1 & P2 im Joint-Position Modus.

4 Experimentelle Ergebnisse

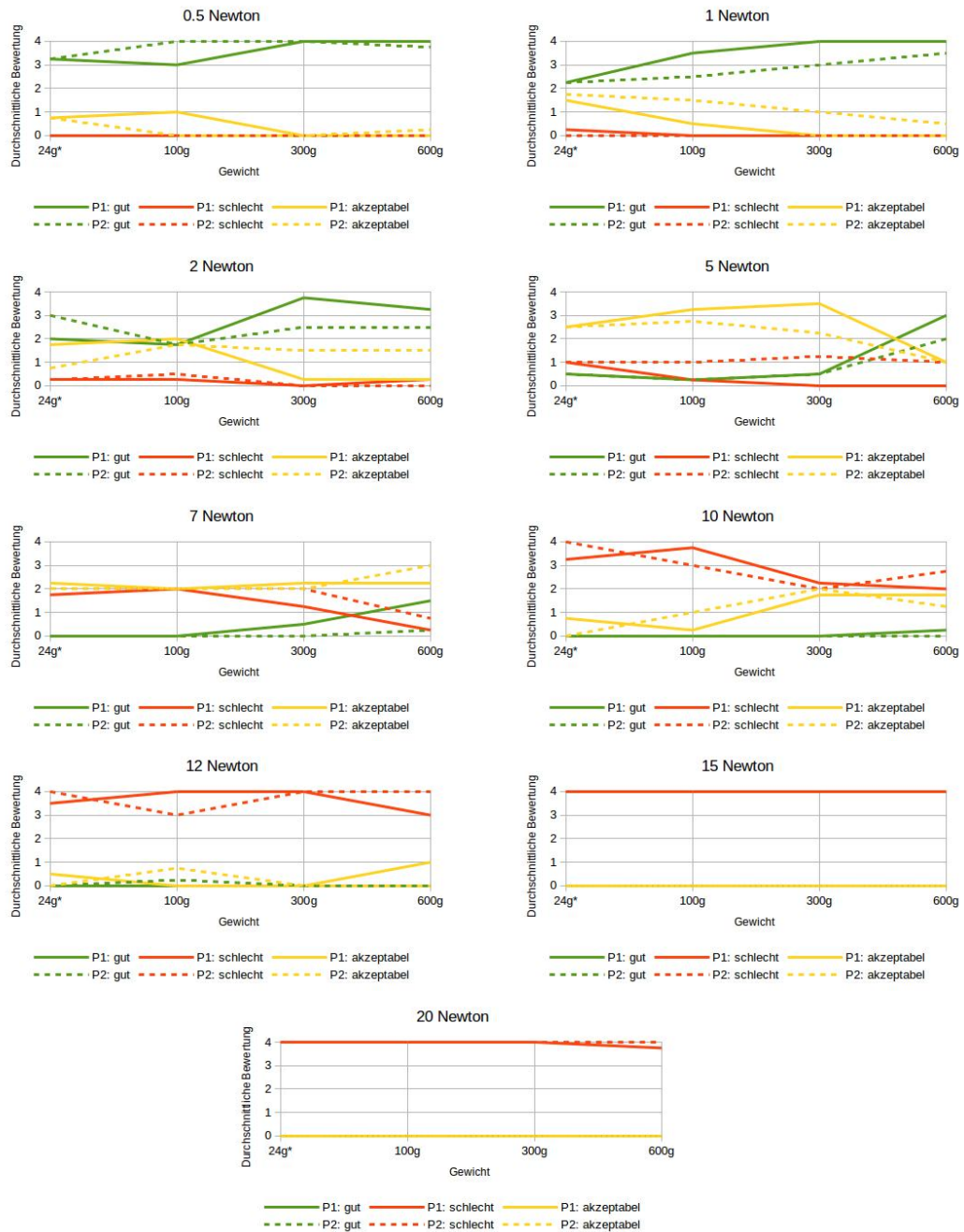


Abbildung 4.17: Vergleich der Versuchsergebnisse für die Übergabestelle P1 & P2 im Joint-Impedance Modus.

4.2 Untersuchung des Zusammenhangs zwischen Objektgewicht und Auslösekraft

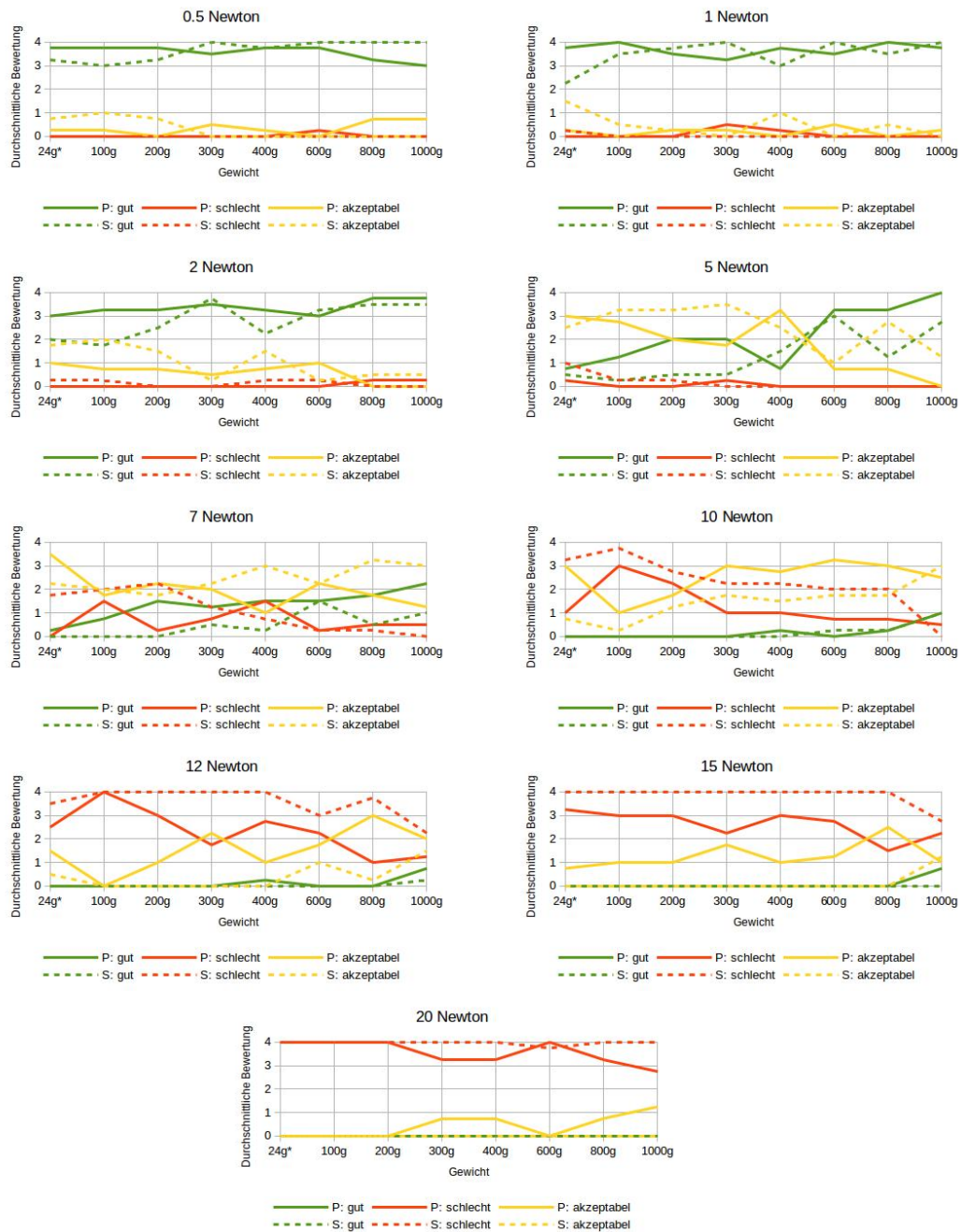


Abbildung 4.18: Vergleich der Versuchsergebnisse zwischen dem Joint-Position Modus (P) und dem Joint-Impedance Modus (S) in der Übergabestelle P1.

4 Experimentelle Ergebnisse

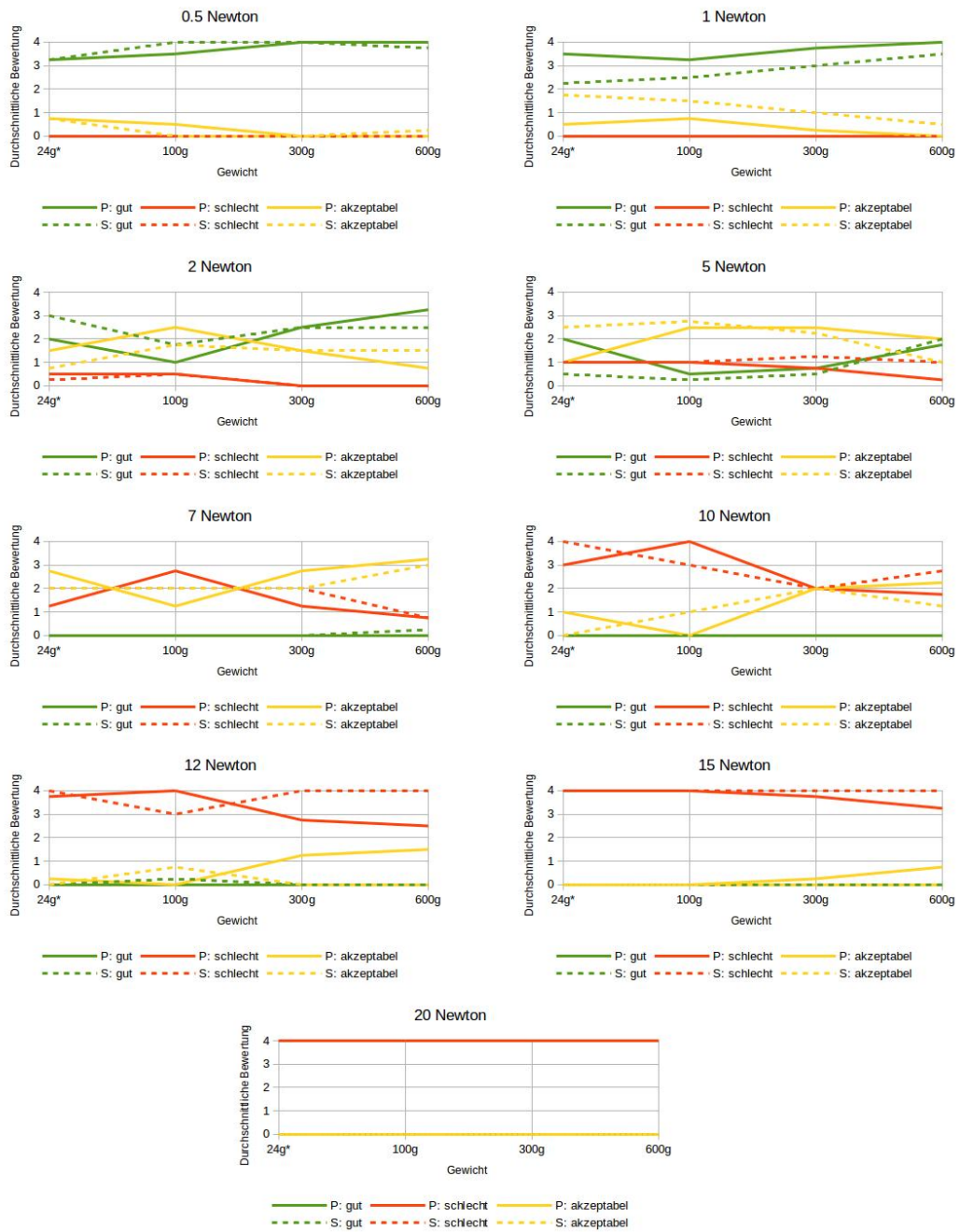


Abbildung 4.19: Vergleich der Versuchsergebnisse zwischen dem Joint-Position Modus (P) und dem Joint-Impedance Modus (S) in der Übergabelung P2.

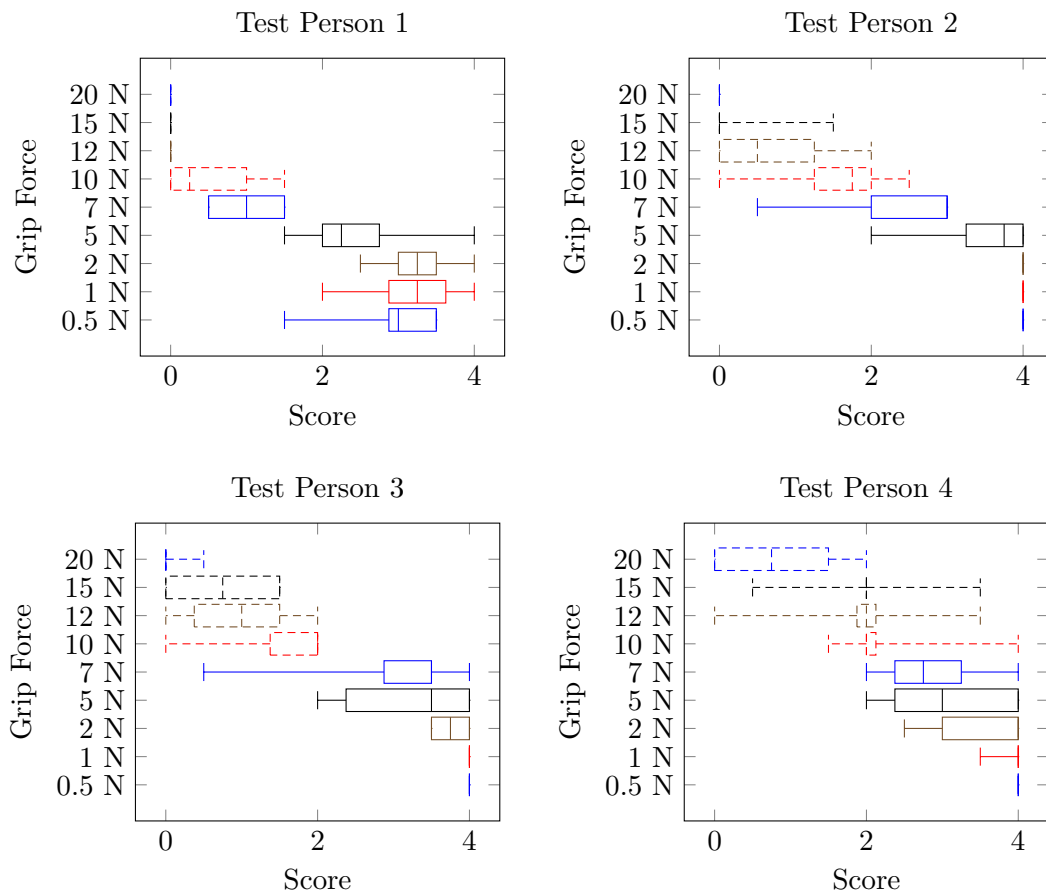


Abbildung 4.20: Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P1 im Joint-Position Modus.

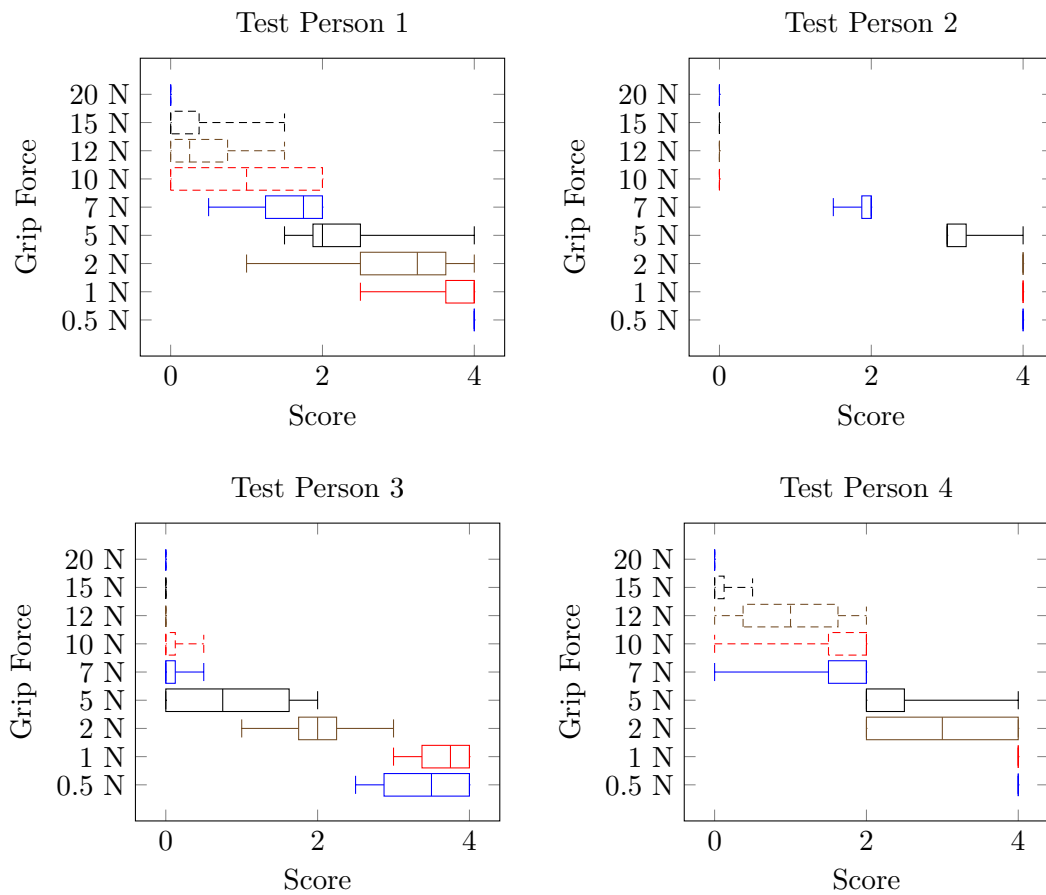


Abbildung 4.21: Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P2 im Joint-Position Modus.

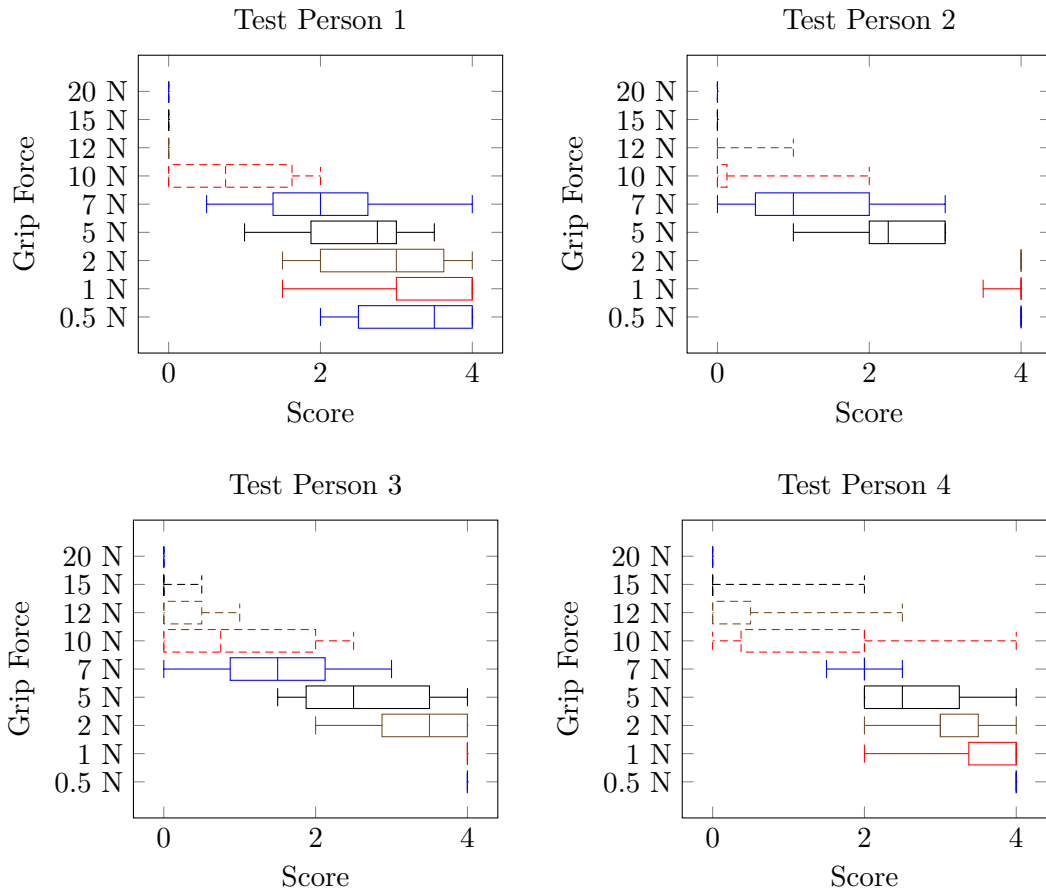


Abbildung 4.22: Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P1 im Joint-Impedance Modus.

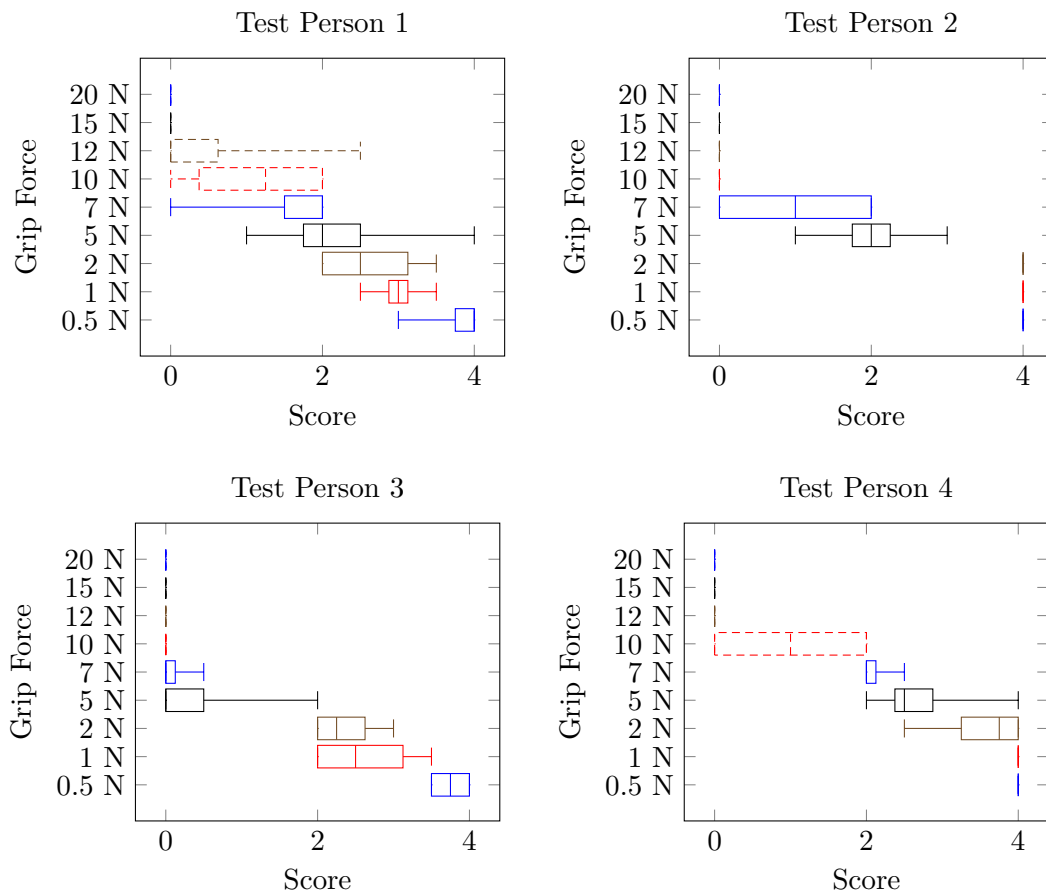


Abbildung 4.23: Abgebildet sind der numerische Score und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P2 im Joint-Impedance Modus.

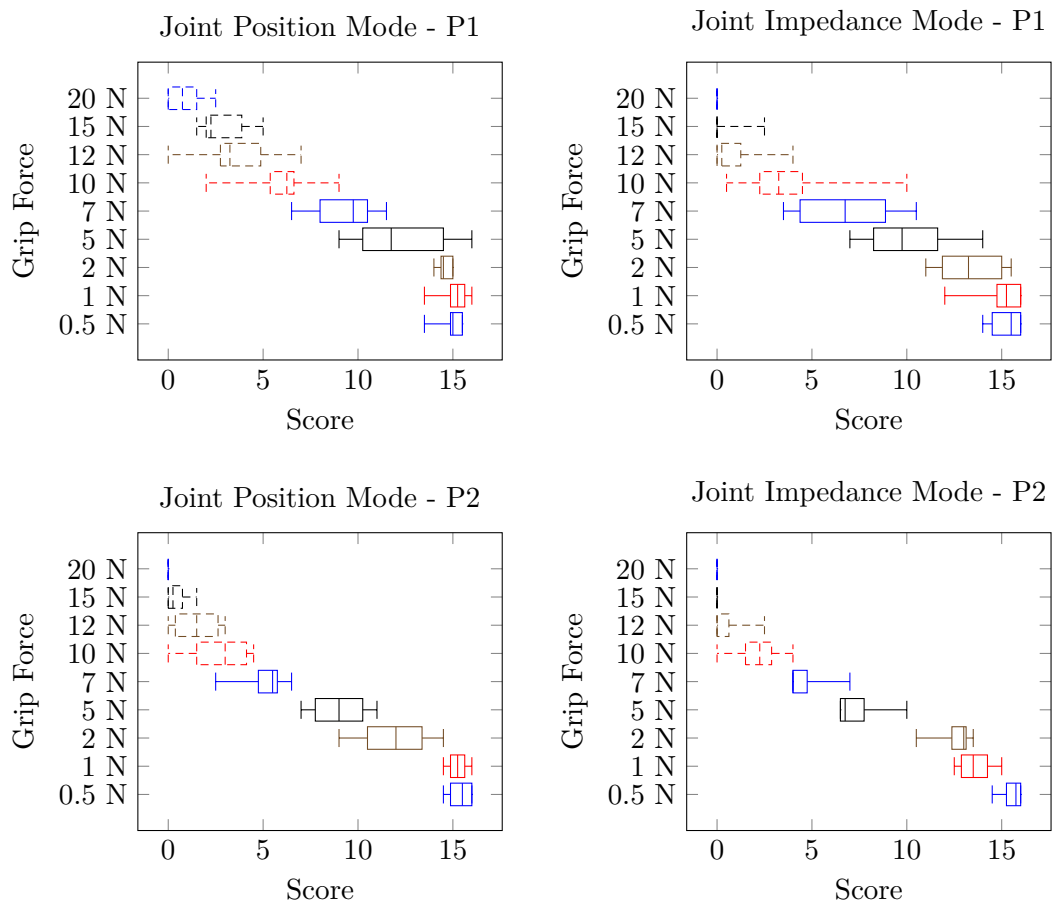


Abbildung 4.24: Abgebildet sind die durchschnittlichen Ergebnisse des numerischen Scores und der Greifkräfte für beide Übergabestellungen und Übergabestrategien.

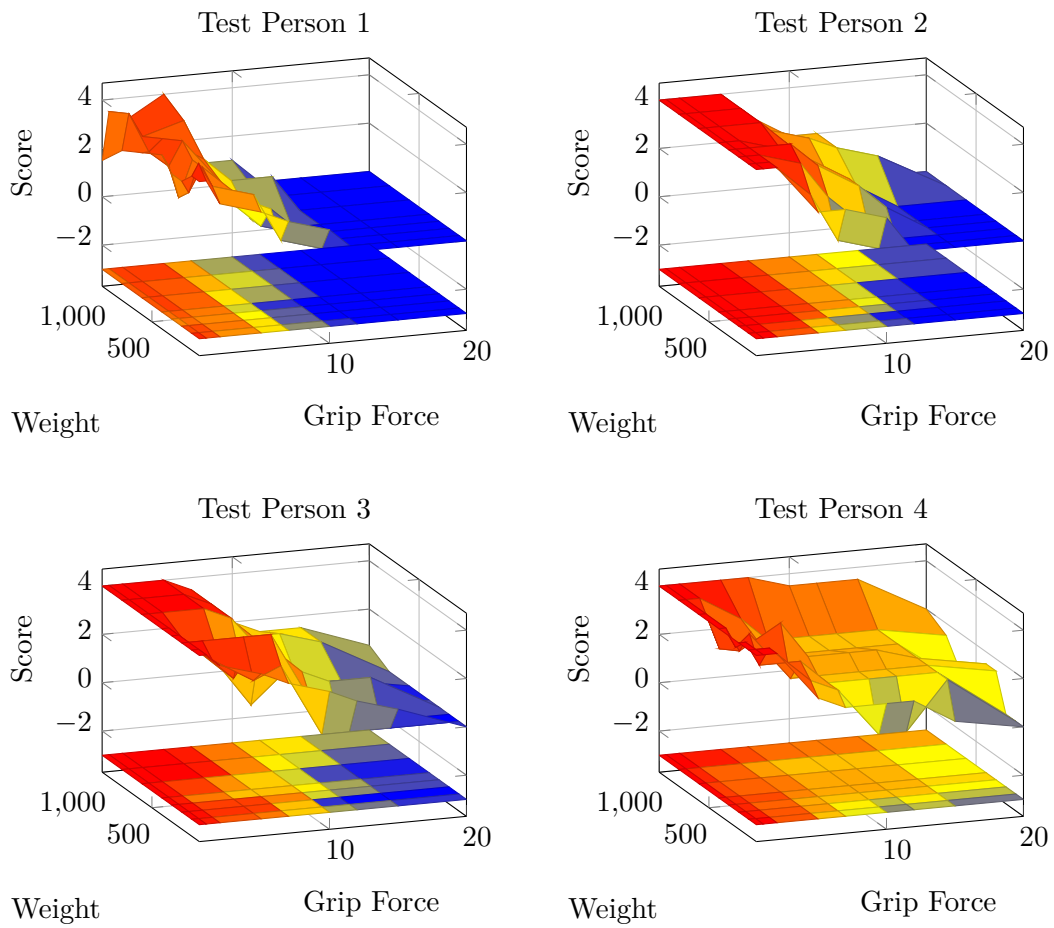


Abbildung 4.25: Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P1 im Joint-Position Modus.

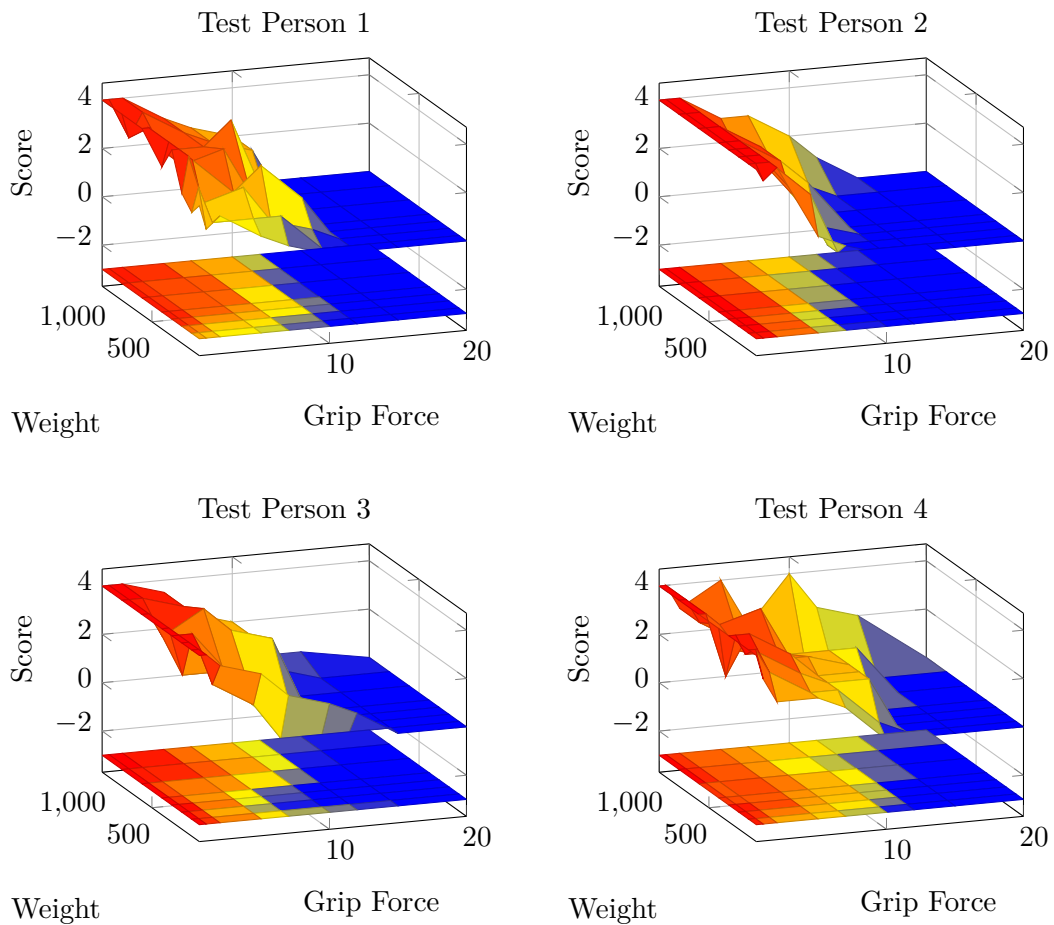


Abbildung 4.26: Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P1 im Joint-Impedance Modus.

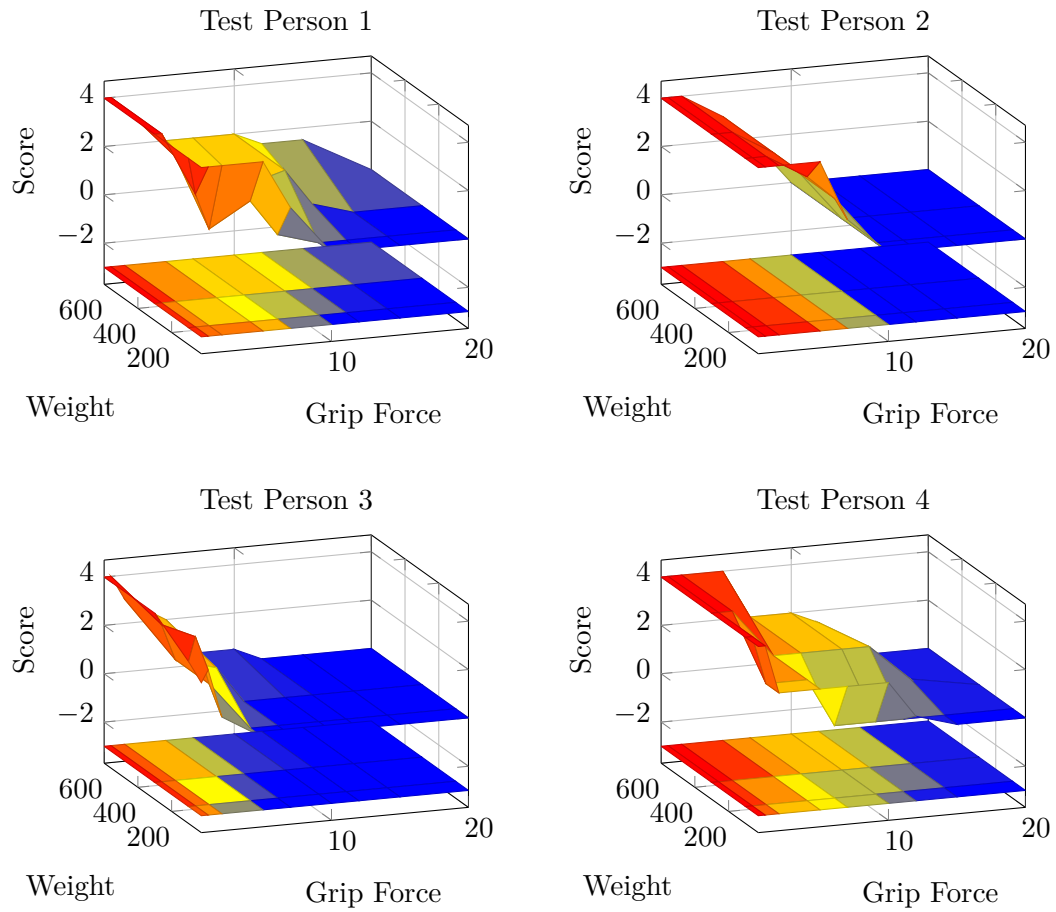


Abbildung 4.27: Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P2 im Joint-Position Modus.

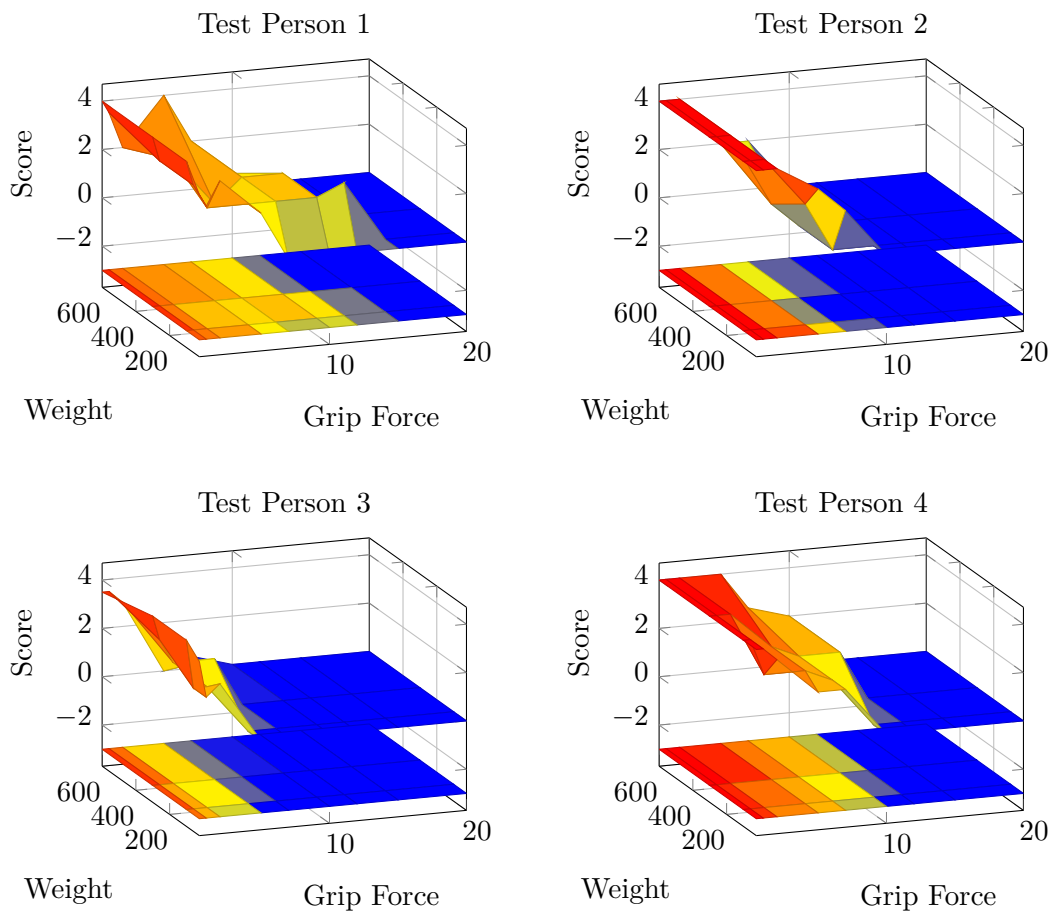


Abbildung 4.28: Abgebildet sind der numerische Score, das Objektgewicht und die Greifkräfte für die einzelnen Testpersonen in der Übergabestelle P2 im Joint-Impedance Modus.

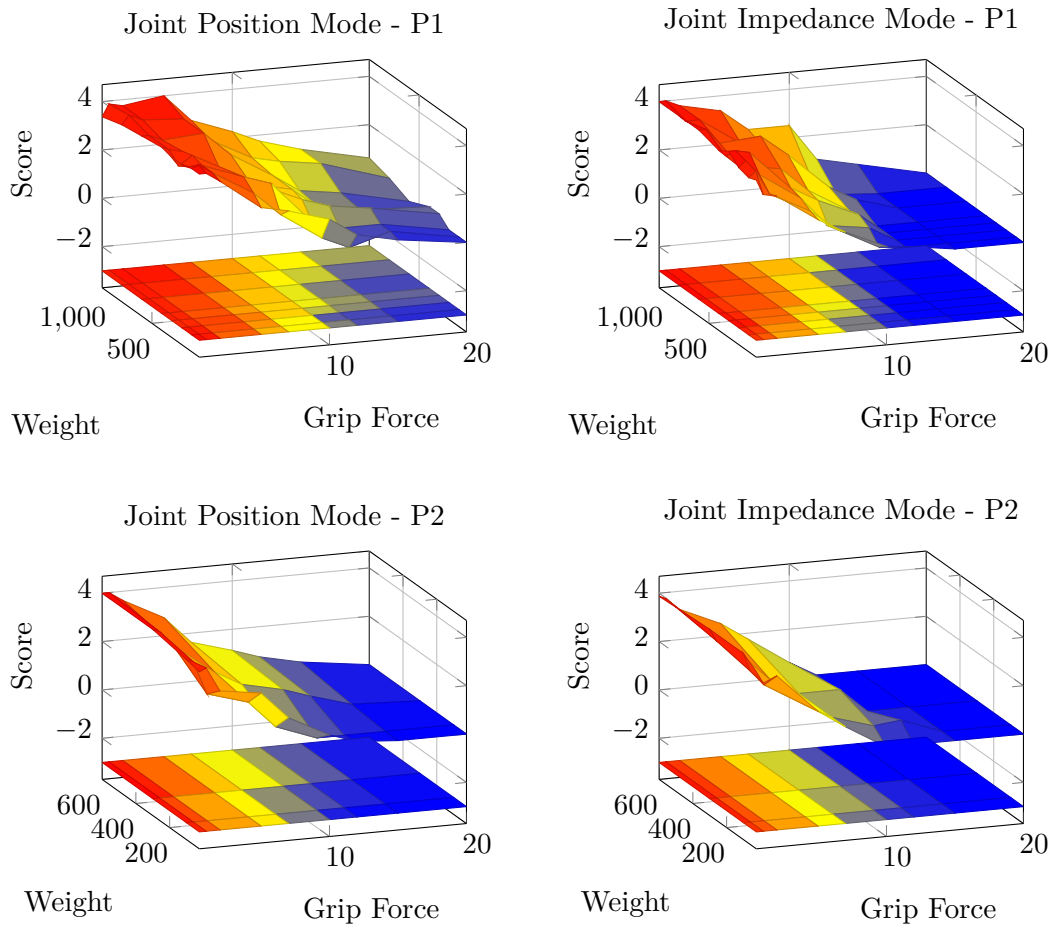


Abbildung 4.29: Abgebildet sind die durchschnittlichen Ergebnisse des numerischen Scores, des Objektgewichtes und der Greifkräfte für beide Übergabestrategien und Übergabestrategien.

Analysiert man die Abweichungen der einzelnen Testpersonen in den verschiedenen Darstellungsarten (4.9 bis 4.11, 4.20 bis 4.23 und 4.25 bis 4.29), werden Unterschiede in der Sensitivität erkennbar. Besonders sichtbar ist dies bei Testperson 4, welche bei allen Versuchen die höchsten Bewertungen bei den hohen Greifkräften abgegeben hat. Testperson 4 scheint somit vergleichsweise unsensibler zu sein.

In den Abbildungen 4.12, 4.13, 4.14 und 4.15 sind die durchschnittlichen Versuchsergebnisse für beide Übergabestellungen und Übergabestrategien dargestellt. Beim Betrachten der Daten wird einerseits ersichtlich, dass die Bewertungsanzahl "gut" umso größer wird, desto kleiner die Greifkräfte sind. Weiterhin wird in den Abbildungen ersichtlich, dass bei steigender Greifkraft die Anzahl der Bewertungen "gut" zuerst bei Objekten mit kleinem Gewicht abnimmt. Die Anzahl der Bewertungen "schlecht" steigt hingegen zuerst bei Objekten mit kleinem Gewicht an. Dies wird auch in der Abbildung 4.24 und besonders in 4.29 bestätigt. Daraus lässt sich Schlussfolgern, dass der Mensch die Objektübergabe mit dem möglichst geringsten Kraftaufwand bevorzugt.

Weiterhin wird unter anderem aus Abbildung 4.29 ersichtlich, dass bei gleichbleibender Greifkraft der Score sich mit einem Anstieg des Objektgewichtes erhöht. Je schwerer das Objektgewicht, desto stärker darf die Greifkraft während der Objektübergabe sein, ohne die natürlich wirkende Objektübergabe negativ zu beeinflussen. Für die genaue Ermittlung dieses Zusammenhangsfaktors müssen umfangreichere Experimente mit mehr Probanden durchgeführt werden. Die Bestimmung ist nicht mehr Bestandteil dieser Masterarbeit und wird dem Arbeitsbereich TAMS der Universität Hamburg herzlich empfohlen.

Überdies wird durch den Vergleich der beiden Übergabestellungen in den Abbildungen 4.16, 4.17, 4.23 und 4.29 ersichtlich, dass bei P2 früher kein "gut" und früher nur "schlecht" vergeben wird. Dies bedeutet das die Versuchspersonen empfindlicher bei P2 als bei P1 reagieren. Ursache hierfür ist vermutlich, dass bei der Objektfreigabe das Objektgewicht ganz oder teilweise auf den unteren Greiferfinger verlagert wird, wodurch ein anderes Auslösekraft zu Belastungskraft Verhältnis wahrgenommen wird bzw. dem Menschen mehr Zeit bleibt das Objekt anzuheben. Die Übergabestellung beeinflusst somit die Auslösekraft zur Erreichung einer natürlich wirkenden Objektübergabe.

Bei den durchgeführten Experimenten wurde während der Objektübergabe immer derselbe Griff und dieselbe Übergabestellung angewendet. Nebenbei wurden aber auch andere Griffarten und Übergaberichtungen miteinander verglichen. Eine Testperson bevorzugte beim Auslösen in P1 immer eine von zwei möglichen Richtungen entlang einer kartesischen Achse. Bei der bevorzugten Übergaberichtung wird laut der Testperson das Objekt erst noch auf einen der Finger gedrückt, wodurch sich das

Objekt stabilisiert. Bei den anderen Übergaberichtungen konnte die Testperson einen ziemlichen Ruck wahrnehmen, welches als unnatürlich wahrgenommen wurde. Die Aussagen wurde bei beiden Übergabestrategien gemacht. In P2 nahm die Testperson hingegen keine nennenswerten Unterschiede wahr. Vermutlich ist auch die soeben beschriebene Ursache verantwortlich für diese Wahrnehmung. Die Übergaberichtung und der Griff beeinflussen damit ebenfalls die natürlich wirkende Objektübergabe.

In den Abbildungen 4.18, 4.19, 4.23 und 4.29 kann der Joint-Position Modus und der Joint-Impedance Modus jeweils in P1 und P2 miteinander verglichen werden. Im Joint-Impedance Modus scheinen die Versuchspersonen empfindlicher als beim Joint-Position Modus zu sein. Die Nachgiebigkeit beeinflusst somit die Auslösekraft zur Erreichung einer natürlich wirkenden Objektübergabe. Möglicherweise würden anderen Ergebnissen mit anderen Einstellungen der Nachgiebigkeit auftreten.

Unter den Versuchspersonen trat allerdings keine eindeutige Präferenz für oder gegen die nachgiebige Objektübergabe auf. Eine Versuchsperson bevorzugte den Joint-Position Modus, weil dieser berechenbarer für Sie sei. Eine andere Versuchsperson bevorzugte die leichte Nachgiebigkeit, da diese auf Sie natürlicher wirke und ebenfalls bei der Mensch zu Mensch Objektübergabe auftrete. Des Weiterem wird vermutet, dass die nachgebende Bewegung des Manipulators einen Einfluss auf die ansteigende Auslösekraft des Nehmers haben kann, weil eine Rückmeldung durch die visuell wahrgenommene Bewegung erfolgt.

4.3 Objektübergabe im Laufe des Bewegungsablaufs

Die Objektübergabe bei der Mensch zu Mensch Interaktion findet manchmal im Laufe einer Bewegung statt. Dieses Verhalten wurde auch bei einigen Testpersonen in der Publikation von [Cakmak et al., 2011 \[b\]](#) mit Robotern beobachtet. Der Inhalt dieses Abschnitts ist die Implementierung der Objektübergabe im Laufe des Bewegungsablaufs. Für die Implementierung eignen sich die Armsensoren des LWR nicht mehr, weil die Momentensensoren in den Gelenken durch die Bewegung selbst beeinflusst werden bzw. nicht ausreichend Zeit erhalten, um die Kräfte am TCP mit einer akzeptablen Genauigkeit zu berechnen. Beim Testen der Objektübergabe mit dieser Messmethode, kam es durch die stark schwankenden Messwerte, häufig zu einem vorzeitigen Loslassen des Objektes. Daraufhin konnte zwar durch Erhöhung des Auslöseschwellenwertes das vorzeitige Loslassen verhindert werden, dies hatte jedoch zur Folge, dass erheblicher Kraftaufwand für die Einleitung der Objektfreigabe aufgebracht werden musste. Nichtsdestotrotz wird in Abbildung 4.30 gezeigt, dass es möglich ist, die Kräfte am TCP als Auslösekriterium für die Objektübergabe zu verwenden. Im Laufe der Bewegung sollten die dynamischen Kräfte des

Objektgewichtes zu der Berechnung der Greifkraft mitberücksichtigt werden, um eine natürlich wirkende Objektübergabe zu realisieren. Der Ansatz ist jedoch sehr aufwändig und wird in dieser Arbeit nicht weiter verfolgt.

Stattdessen wurden die taktilen Fingersensoren WSG-DSA zum Detektieren einer externen Kraft eingesetzt. Hierfür wird die Summe aller Pixel der Sensormatrix nach einem Greifvorgang berechnet und periodisch mit der aktuellen Summe verglichen. Bei fest definierter Abweichung wird der Greifer geöffnet und ein Befehl zum Abbrechen der Trajektorie an `move_group` geschickt, welcher diese wiederum an `ros_fri` weiterleitet. Anschließend wird die Trajektorie des LWR mit einer langsam werdenden Bewegung zum Stillstand gebracht. Mit den taktilen Sensoren können die Kraftstärke und Kraftrichtungen nicht ermittelt werden, wodurch die Bestimmung des Objektgewichtes unmöglich ist. Nach dem Greifen eines beliebigen Objektes mit unbekanntem Gewicht erfolgt die Objektübergabe durch die Auslösung eines vorher bestimmten unbekanntes Kraftaufwandes, welcher für jedes Objekt meistens erneut bestimmt werden muss. Eine systematische Analyse der Objektübergabe ist unter den beschriebenen Bedingungen sehr aufwändig, weshalb in diesem Abschnitt nur die Machbarkeit der Objektübergabe im Laufe des Bewegungsablaufs gezeigt wird.

Die Pick & Place Demo wurde mit den oben erwähnten Änderungen zur `TCP_CANCEL_ON_FORCE` Demo erweitert. Daraufhin sollten einige Testpersonen das Objekt zu einem beliebigen Zeitpunkt vom Greifer entnehmen und wurden auf störende Einflüsse während der Objektübergabe befragt. Die Personen kategorisierten die Objektübergabe als "Gut" bis "Akzeptabel", da jedoch keine Aussage über die eingesetzten Greifkräfte gemacht werden können, dienen die Implementierungen nur als Hilfestellungen für zukünftige Experimente.

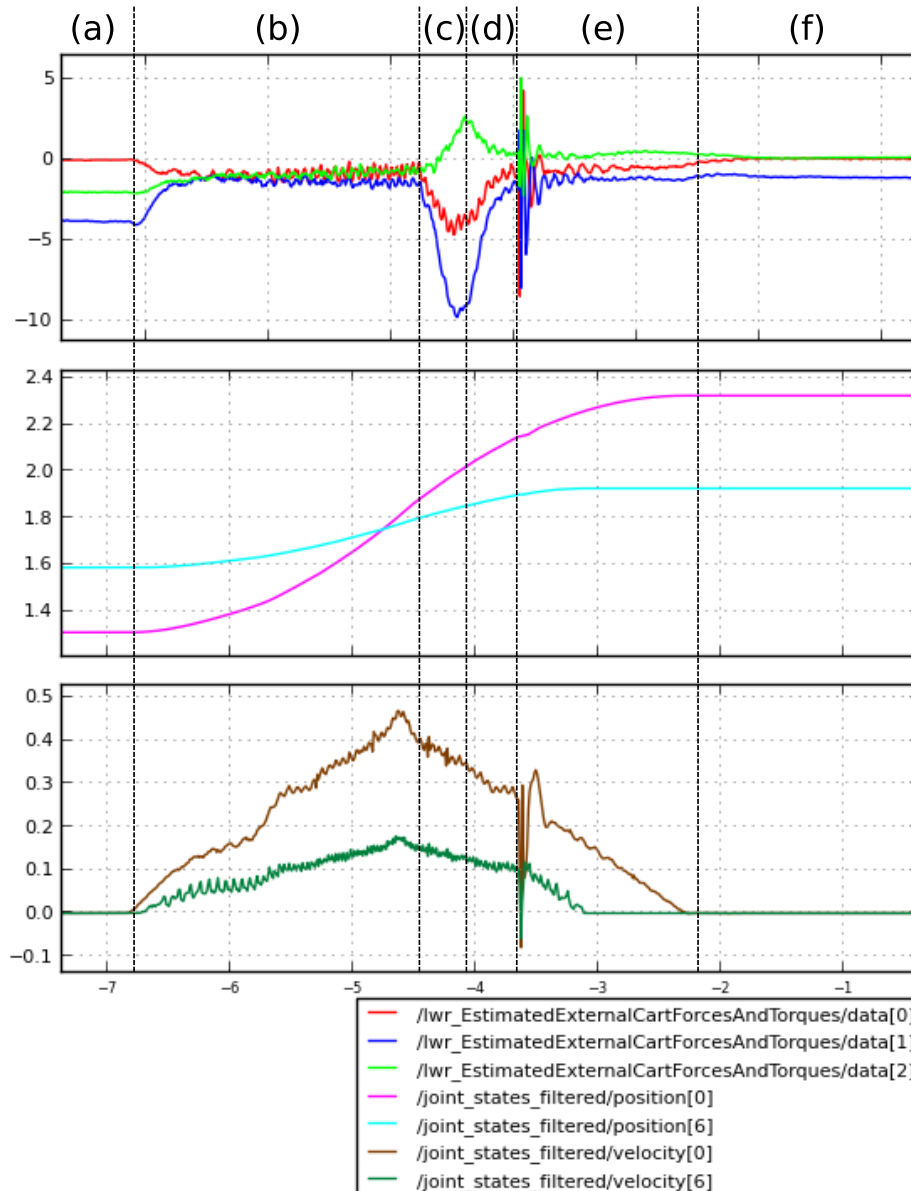


Abbildung 4.30: In der Abbildung sind die externen Kräfte, Gelenkwerte und Gelenkgeschwindigkeiten für eine Objektübergabe im Laufe der Bewegung dargestellt. Abbildung entnommen aus [Hendrich \[2014 a\]](#). In dem oberen Diagramm sind die externen Kräfte $[N]$ am TCP illustriert. In dem mittleren Diagramm sind die Gelenkwerte in $[rad]$ und in dem unteren Diagramm die Gelenkgeschwindigkeit in $[rad/s]$ dargestellt. Die unterschiedlichen Phasen der Objektübergabe können erkannt werden:

- (a): LWR ist untätig (keine Geschwindigkeit, konstante Gelenk- und Kräftewerte),
- (b): Zum Benutzer fahren (Rauschen in den Kraftwerten),
- (c): Benutzer greift das Objekt (Sichtbar anhand der Kraftwerte),
- (d): Schwellenwert überschritten, Greifer öffnet sich, wodurch die Kraftwerte wieder absinken,
- (e): Armcontroller stoppt den LWR (Ruck tritt auf),
- (f): LWR ist untätig.

5 Fazit und Ausblick

In diesem letzten Kapitel werden zuerst die erreichten Änderungen in Hinsicht auf die zu Beginn formulierten Ziele diskutiert. Anschließend werden weitere Entwicklungsmöglichkeiten beschrieben.

5.1 Fazit

Das erste Teilziel dieser Arbeit war die Entwicklung eines Robotersystems mit den zur Verfügung stehenden Hardware- und Softwarekomponenten. Die Integration der Hardware zu einem Robotersystem wurde mit MoveIt! erheblich vereinfacht. Jedoch mussten auch viele Konfigurationen manuell vorgenommen werden. Aufgrund dem Alpha-Entwicklungsstatus von MoveIt! waren die erforderlichen Informationen hierzu teilweise nicht dokumentiert bzw. veraltet gewesen. Daher können die beschriebenen Vorgehensweisen in dieser Arbeit als Orientierung für die Entwicklung von anderen Robotersystemen dienen.

Der verfügbare Controller für den LWR wird durch `ros_fri` ersetzt, welcher mit viel mehr Funktionen ausgestattet ist. Dieser hat den großen Vorteil, dass auf den Einsatz von Orocos verzichtet werden konnte und somit die Integration an die neue ROS Distribution Hydro möglich war. Weiterhin wurde der WSG 50 Controller unter anderem um das Auslesen der taktilen Sensordaten erweitert. Eine ältere Version des verbesserten Controllers wurde von Dr. Norman Hendrich an Weiss Robotics übermittelt, welche die Modifikationen unverändert übernommen haben.

Die Veröffentlichung des leichtgewichtigen Pakets `ros_fri` und des verbesserten `wsg50-ros-pkg` wird in naher Zukunft angestrebt, weil diese in anderen Robotersystemen wiederverwendet werden können und dadurch zum Voranschreiten des technologischen Fortschritts in der Robotik beitragen.

Der zweite Schwerpunkt dieser Arbeit lag in der Erreichung einer für den Menschen natürlich wirkenden Objektübergabe zwischen Menschen und Robotern. Anhand von Experimenten wurden neue Erkenntnisse zur Verbesserung der Objektübergabe gewonnen. Beispielsweise wurde von den meisten Testpersonen im Joint-Position Mode jegliche Arten von Restriktionen in den Kraftrichtungen bemerkt und als unnatürlich empfunden.

Weiterhin wird für die Erreichung einer natürlich wirkende Objektübergabe ein möglichst kleiner Kraftaufwand zum Initialisieren der Objektfreigabe empfohlen. Bei kleinen Greifkräften sollte allerdings beachtet werden, dass direkte oder indirekte Stöße gegen den Manipulator ebenfalls gemessen werden und zu einem vorzeitigen Loslassen führen. Das Sicherheitsrisiko kann durch die Erhöhung der Auslösekraft abhängig vom Objektgewicht reduziert werden, da der Einfluss des Objektgewichtes auf die benötigte Auslösekraft zur Erreichung einer natürlich wirkende Objektübergabe in dieser Arbeit nachgewiesen wurde. Je schwerer das Objektgewicht, desto stärker darf die Auslösekraft während der Objektübergabe sein, ohne die natürlich wirkende Objektübergabe negativ zu beeinflussen. Leider wurden in den Experimenten nicht ausreichend Daten für die genaue Bestimmung des Zusammenhangfaktors gesammelt, um eine unwiderlegbare Empfehlung abzugeben. Ohne die adaptive Erhöhung der Greifkraft wird zumindest ein Greifkraftschwellenwert von 0.5 N bis 1 N empfohlen, weil damit die höchsten Bewertungen während der Experimente erzielt wurden und keine vorzeitige Objektfreigabe auftrat.

Der Einfluss der Übergabestelle auf die natürlich wirkende Objektübergabe wurde ebenfalls nachgewiesen. Die Erforschung und der Vergleich weiterer Übergabestellen ist erstrebenswert, damit Bahnplaner über Listen mit vorrangigen Übergabestellen bezüglich der natürlichen Objektübergabe verfügen.

Der Einfluss des Joint-Impedance Modus gegenüber dem Joint-Position Modus ist in den Experimenten negativ hervorgetreten und wird deshalb für die Objektübergabe nicht empfohlen. Die Modifikation der Nachgiebigkeitswerte könnte jedoch zu anderen Ergebnissen führen und sollte weiter untersucht werden.

Die gewonnenen Erkenntnisse dieser Masterarbeit können für die Weiterentwicklung von Steuerungsalgorithmen und Bewegungsplanern eingesetzt werden.

5.2 Ausblick

In dieser Masterarbeit wurde nur ein kleiner Aspekt der komplexen Problemstellung behandelt, sodass wesentlich mehr Forschung erforderlich ist. Im Folgenden werden mehrere Möglichkeiten aufgeführt, deren Umsetzung das Robotersystem und die Mensch-Roboter-Interaktion verbessern könnten.

Beispielsweise könnte mit einer weiteren Kinect durch Sensorfusion der Arbeitsraum des LWR dreidimensional erfasst werden. Dies würde die Detektion von Hindernissen, die vom Manipulator verdeckt werden ermöglichen und somit die bestehende Kollisionsvermeidung verbessern.

Weiterhin wird der Einsatz der Kinect für die Objekterkennung empfohlen, um mit den zusätzlichen Tiefeninformationen genauere Ergebnisse bei der Objektlokalisierung zu erhalten. Zudem existiert für die Kinect bereits das sehr populäre ROS Paket `tabletop_object_detector`, welches die Objektsegmentation und Objekterkennung durchführt. Dieses verwendet für die Objekterkennung die Datenbank `household_objects_database`, die schon dreidimensionale Modelle von vielen Alltagsgegenständen enthält [Tabletop \[2014\]](#). Je nach geplanter Weiterentwicklung und Anwendung des Robotersystems könnte die Logitech HD Pro Webcam C910 stattdessen z.B. zur Gesichtsdetektion verwendet werden. Bei größeren Entfernungen könnte die Logitech HD Pro Webcam C910 wegen der höheren Auflösung besser für die Aufgabe geeignet sein.

Außerdem wird die Entwicklung einer Visualisierung für die gelieferten Daten der taktilen Sensormatrizen vom modifizierten `wsg50-ros-pkg` empfohlen. Damit könnte die Objektidentifikation von für die Kamera schwer bzw. nicht erkennbaren Objekten erfolgen.

Überdies werden komplexe Funktionen vom MoveIt! MoveGroup Interface z.B. die Pick oder Place Operation angeboten, welche erst nach der Integration des Greifercontrollers mit MoveIt! funktionsfähig wären. Hierfür könnte der bestehende Controller für den Greifer um den Typ `GripperCommand` erweitert werden oder der noch unzureichend getestete `gripper_adapter_server` weiterentwickelt werden.

Die natürliche Objektübergabe könnte durch die Weiterentwicklung des Controllers um das kontinuierliche Loslassen bezüglich Kraft und Zeit verbessert werden. Die Objektübergabe sollte nach menschlichem Vorbild ungefähr 500 ms dauern.

Weiterhin wurde in den Experimenten der Zusammenhang zwischen dem Kraftaufwand zum Initialisieren der Objektfreigabe und der Belastungskraft des Objektes nachgewiesen. Die genaue Bestimmung des noch unbekanntem Zusammenhangfaktors könnte für die Entwicklung eines Controllers genutzt werden, um das Sicherheitsrisiko der vorzeitigen Objektfreigabe zu reduzieren.

Literaturverzeichnis

- [**actionlib, 2014**] actionlib,
<http://wiki.ros.org/actionlib/DetailedDescription>, Zuletzt abgerufen:
2014-09-20
- [**Bischoff, 2010**] Bischoff, R.; Kurth, J.; Schreiber, G.; Köppe, R.; Albu-Schäffer, A.; Beyer, A.; Eiberger, O.; Haddadin, S.; Stemmer, A.; Grunwald, G.; Hirzinger, G.: The KUKA-DLR Lightweight Robot arm – a new reference platform for robotics research and manufacturing, International Symposium on Robotics (ISR2010), Munich, Germany, 2010. -ISBN 978-3-8007-3273-9. p. 1-8
- [**Bistry, 2013**] Bistry, Hannes.: Intelligente Kamerasysteme im Anwendungsfeld mobiler Service-Roboter, 2013.
- [**Cakmak et al., 2011(a)**] Cakmak, Maya; Srinivasa, Siddhartha; Lee, Min Kyung; Forlizzi, Jodi; Kiesler, Sara: "Human Preferences for Robot-Human Hand-over Configurations", IEEE/RSJ International Conference on Intelligent Robots and Systems, 25-30 Sept. 2011. -ISBN 978-1-61284-454-1. p. 1986 - 1993
- [**Cakmak et al., 2011(b)**] Cakmak, Maya; Srinivasa, Siddhartha; Lee, Min Kyung; Forlizzi, Jodi; Kiesler, Sara: "Using Spatial and Temporal Contrast for Fluent Robot-Human Hand-overs", 6th ACM/IEEE International Conference on Human-Robot Interaction, 8-11 March 2011. -ISBN 978-1-4673-4393-0. p. 489 - 496
- [**Carbone, 2012**] Carbone, Giuseppe: Grasping in Robotics - Berlin Heidelberg: Springer Science & Business Media, 2012. -ISBN 978-1-447-14664-3. p. 1-475
- [**Chan et al., 2012**] Chan, W. P.; Parker, C. A.; Loos, H. M. Van der; & Croft, E. A.: Grip forces and load forces in handovers: implications for designing human-robot handover controllers. Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference, 5-8 March 2012. -ISBN 978-1-4503-1063-5. p. 9-16
- [**Craig, 2005**] Craig, John J.: Introduction to Robotics - Mechanics and Control. 3rd ed. New Jersey: Pearson/Prentice Hall, 2005 -ISBN 0-13-123629-6. p. 1-408

- [**Corke, 2011**] Corke, Peter: Robotics, Vision and Control : Fundamental Algorithms in MATLAB. 1st ed. 2011. Berlin Heidelberg: Springer Science & Business Media, 2011. -ISBN 978-3-642-20143-1. p. 1-570
- [**Dehais et al., 2011**] Dehais, Frédéric; Sisbot, Emrah Akin; Alami, Rachid; Causse, Mickaël : Applied ergonomics : the journal of people's relationships with equipment, environments and work systems.- Oxford : Elsevier, ISSN 0003-6870, ZDB-ID 2164048. - Vol. 42.2011, 6, p. 785-792, 2011.
- [**DLR, 2006**] Deutsches Zentrum für Luft- und Raumfahrt e. V.: DLR-HIT-Hand: Multisensory 4-Finger-Hand, 2006.
- [**DLR, 2014**] Deutsches Zentrum für Luft- und Raumfahrt e. V.: Bild- und kraftgestütztes Fügen, http://www.dlr.de/rm/de/desktopdefault.aspx/tabid-4789/7945_read-12718/1, Zuletzt abgerufen: 2014-09-20
- [**DLR Hand, 2014**] The best five anthropomorphic robotic hands/arms, <http://mindtrans.narod.ru/hands/hands.htm>, Zuletzt abgerufen: 2014-09-20
- [**DSA, 2011**] WSG-DSA - Taktile Sensorik für Greifer der WSG-Serie, 2011.
- [**ELU-1 Hand, 2010**] The Elumotion Robotic Hand, <http://www.elumotion.com/eluhand.html>, Zuletzt abgerufen: 2014-09-20
- [**FM, 2014**] Einsatzmöglichkeiten einer 3D-Kamera in der Produktionstechnik am Beispiel der Kinect-Kamera, https://wiki.zimt.uni-siegen.de/fertigungsautomatisierung/index.php/Einsatzm%C3%B6glichkeiten_einer_3D-Kamera_in_der_Produktionstechnik_am_Beispiel_der_Kinect-Kamera, Zuletzt abgerufen: 2014-09-20
- [**Fraden, 2010**] Fraden, Jacob: Handbook of Modern Sensors - Physics, Designs, and Applications. 4rd ed. Berlin Heidelberg: Springer Science & Business Media, 2010. -ISBN 978-1-441-96466-3. p. 1-678
- [**FraunhoferIFF, 2014 (a)**] Entwicklung und Validierung taktile Sensorsysteme für die sichere Kollisionsdetektion bei der Mensch-Roboter-Interaktion (TAKSENS), <http://www.iff.fraunhofer.de/de/geschaeftsbereiche/robotersysteme/taksens.html>, Zuletzt abgerufen: 2014-09-20
- [**FraunhoferIFF, 2014 (b)**] Industrieroboter mit Annäherungsdetektion für sichere Mensch-Roboter-Kollaboration, <http://www.iff.fraunhofer.de/de/geschaeftsbereiche/robotersysteme/kapazitive-sensorik.html>, Zuletzt abgerufen: 2014-09-20

- [**FraunhoferIFF, 2014 (c)**] Sichere Personenlokalisierung
<http://www.iff.fraunhofer.de/de/geschaeftsbereiche/robotersysteme/taktile-sensorsysteme-drucksensitive-fussbodenbelaege.html>, Zuletzt abgerufen: 2014-09-20
- [**FRI, 2014**] Fast Research Interface Library, <http://cs.stanford.edu/people/tkr/fri/html/>, Zuletzt abgerufen: 2014-09-20
- [**GStreamer, 2014**] GStreamer Homepage, <http://gstreamer.freedesktop.org/1>, Zuletzt abgerufen: 2014-09-20
- [**Hendrich, 2014 a**] Hendrich, Norman; Bistry, Hannes; Liebrecht, Johannes; Zhang, Jianwei: Natural Robot-Human Handover Combining Force and Tactile Sensors, Workshop on Assistance and Service Robotics in a Human Environment, IROS 2014, Chicago, USA, 2014.
- [**Hendrich, 2014 b**] Hendrich, Norman; Bistry, Hannes; Liebrecht, Johannes; Zhang, Jianwei: Multi-modal clues for efficient human-robot object handover: a case study with elderly users, Workshop on New frontiers of service robotics for the elderly, RO-MAN 2014, Edinburgh, UK, 2014.
- [**Hesse, 2011**] Hesse, Stefan: Greifertechnik - Effektoren für Roboter und Automaten - München: Hanser, 2011. -ISBN 978-3-446-42422-7. S. 1-280
- [**Khoshelham, 2011**] Khoshelham, K.: ACCURACY ANALYSIS OF KINECT DEPTH DATA, 2011.
- [**Kinect, 2014**] Microsoft Kinect, <https://de.wikipedia.org/wiki/Datei:Xbox-360-Kinect-Standalone.png>, Zuletzt abgerufen: 2014-09-20
- [**Kuffner & Lavelle, 2010**] Kuffner Jr., James J.; Lavelle, Steven M.: RRT-Connect: An efficient approach to single-query path planning, Proc. IEEE International Conf. on Robotics and Automation, 2000, -ISBN 0-7803-5886-4. S.995-1001
- [**KUKA, 2011**] KUKA System Software 5.6 lr, 2010.
- [**Kroeger, 2011**] Kroeger, Torsten: Opening the Door to New Sensor-Based Robot Applications - The Reflexxes Motion Libraries. In Proc. of the IEEE International Conference on Robotics and Automation, Shanghai, China, May 2011. -ISBN 978-1-61284-386-5. p. 1-4
- [**Lowe, 1999**] Lowe, D.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision 2, 1999, -ISBN 0-7695-0164-8. p. 1150-1157

- [**MaschinenMarkt, 2014**] Sichere Mensch-Roboter-Interaktion dank neuer Sensorik, <http://www.maschinenmarkt.vogel.de/themenkanaele/automatisierung/robotik/articles/392669/>, Zuletzt abgerufen: 2014-09-20
- [**Mason & MacKenzie, 2005**] Mason, Andrea H.; MacKenzie, Christine L.: Grip forces when passing an object to a partner, *Experimental Brain Research* Volumen 163, Issue 2, pp 173-187, 2005. DOI 10.1007/s00221-004-2157-x.
- [**MoveIt!, 2014 (a)**] MoveIt! Homepage, <http://moveit.ros.org/>, Zuletzt abgerufen: 2014-09-20
- [**MoveIt!, 2014 (b)**] MoveIt!, <http://www.willowgarage.com/blog/2013/05/06/moveit>, Zuletzt abgerufen: 2014-09-20
- [**MoveIt!, 2014 (c)**] MoveIt! Robots, <http://moveit.ros.org/robots/>, Zuletzt abgerufen: 2014-09-20
- [**MoveIt!, 2014 (d)**] Motion Planning Overview, http://moveit.ros.org/wiki/Motion_Planning/Overview, Zuletzt abgerufen: 2014-09-20
- [**Möller, 2007**] Möller, Ralf: Vorlesung Robotik 1: Serielle Manipulatoren, Vorlesungsskript, Universität Bielefeld, 2007.
- [**OctoMap, 2012**] Hornung, Armin; Wurm, Kai M.; Bennewitz, Maren; Stachniss, Cyrill; Burgard, Wolfram: OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees, *Autonomous Robots*, 2013. DOI: 10.1007/s10514-012-9321-0.
- [**OMPL, 2011**] Open Motion Planning Library: A Primer, Kavraki Lab Rice University, 2014.
- [**Orocos, 2014**] Orocos RTT and ROS integrated, <https://www.willowgarage.com/blog/2009/06/10/orocos-rtt-and-ros-integrated>, Zuletzt abgerufen: 2014-09-20
- [**Pedersen, 2000**] Pedersen, Mikkel Rath: Integration of the KUKA Light Weight Robot in a mobile manipulator, Aalborg University, 2000.
- [**Pflegelandschaft, 2012**] Pflegelandschaft 2030, Eine Studie der Prognos AG im Auftrag der vbw – Vereinigung der Bayerischen Wirtschaft e. V., 2012.
- [**Pipeline, 2014**] GStreamer Pipeline, <http://wiki.ubuntuusers.de/GStreamer#Pipelinel>, Zuletzt abgerufen: 2014-09-20
- [**Ploner, 2000**] Ploner, Nico: Gestensteuerung für Powerwall-basierte Visualisierungen Universität Stuttgart, 2000.

- [**Prozessautomatisierung, 2014**] Prozessautomatisierung, <http://www.bma-worldwide.com/?id=2391&L=1>, Zuletzt abgerufen: 2014-09-20
- [**Reflexxes, 2014**] The Reflexxes Motion Libraries — An Overview, http://reflexxes.com/software/typeiirml/v1.2.6/docs/page_reflexxes_motion_libraries.html, Zuletzt abgerufen: 2014-09-20
- [**ROBOT-ERA, 2014**] ROBOT-ERA, <http://www.robot-era.eu/robotera/>, Zuletzt abgerufen: 2014-09-20
- [**ROS, 2014**] About ROS, <http://www.ros.org/about-ros/>, Zuletzt abgerufen: 2014-09-20
- [**Sandia Hand, 2012**] Lifelike, cost-effective robotic Sandia Hand can disable IEDs https://share.sandia.gov/news/resources/news_releases/robotic_hand/, Zuletzt abgerufen: 2014-09-20
- [**Schou et al., 2012**] Schou, Casper; Carøe, Christian; Hvilshøj, Mikkel: Intuitive Programming of AIMM Robot, Aalborg University, 2012.
- [**Shadow, 2009**] Shadow Dextrous Hand Research Systems, http://mindtrans.narod.ru/pdfs/Standard_Prices_aug_09.pdf, Zuletzt abgerufen: 2014-09-20
- [**SRDF, 2014**] Semantic Robot Description Format, <http://wiki.ros.org/srdf>, Zuletzt abgerufen: 2014-09-20
- [**Tabletop, 2014**] tabletop_object_detector, http://wiki.ros.org/tabletop_object_detector, Zuletzt abgerufen: 2014-09-20
- [**Tirpitz, 2012**] Tirpitz, Markus: Berechnung der visuellen Hülle mit mehreren Tiefensensoren, Karlsruher Institut für Technologie, 2012.
- [**VDI2860, 2014**] Industrieroboter - Definition nach VDI 2860, <http://www.forum-sondermaschinenbau.de/seiten/Industrieroboter.htm>, Zuletzt abgerufen: 2014-09-20
- [**Webcam, 2014**] Logitech HD Pro Webcam C910 , <http://www.chip.de/preisvergleich/146993/Datenblatt-Logitech-HD-Pro-Webcam-C910.html>, Zuletzt abgerufen: 2014-09-20
- [**Wilke, 2002**] Wilke, Wilhelm: Segmentierung und Approximation großer Punktwolken Universität Darmstadt, 2002.
- [**WSG-DSA, 2011 (a)**] Tactile Sensing Finger, User's Manual, Weiss Robotics, 2011.
- [**WSG-DSA, 2011 (b)**] Tactile Sensing for the WSG series of grippers, User's Manual, Weiss Robotics, 2011.

[**Zacharias, 2012**] Zacharias, Franziska: Knowledge Representations for Planning Manipulation Tasks. Berlin Heidelberg: Springer Science & Business Media, 2012. -ISBN 978-3-642-25182-5. p. 1-160

Danksagung

An dieser Stelle möchte ich mich bei Personen bedanken, die mir die Anfertigung dieser Arbeit ermöglicht haben.

Herrn Prof. Dr. J. Zhang danke ich dafür, dass ich an seinem Fachbereich meine Masterarbeit schreiben durfte, für die zur Verfügung gestellten Arbeitsmittel und für das Vertrauen, dass er mir durch die Vergabe der Masterarbeit entgegengebracht hat.

Herrn Dr. Norman Hendrich und Herrn Dr. Hannes Bistry danke ich für die unzähligen Tipps und unermüdliche Hilfsbereitschaft während der Anfertigung der vorliegenden Arbeit.

Eidesstattliche Erklärung

Ich versichere, dass ich die Masterarbeit im Studiengang Informatik selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.

Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereichs einverstanden.

Hamburg, den 18. Dezember 2014

JOHANNES LIEBRECHT