

Karlsruher Schriften
zur Anthropomatik

Band 39



Angelika Zube

**Bewegungsregelung mobiler Manipulatoren
für die Mensch-Roboter-Interaktion mittels
kartesischer modellprädiktiver Regelung**

Angelika Zube

**Bewegungsregelung mobiler Manipulatoren
für die Mensch-Roboter-Interaktion mittels
kartesischer modellprädiktiver Regelung**

Karlsruher Schriften zur Anthropomatik

Band 39

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe
erschienenen Bände finden Sie am Ende des Buchs.

Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung

von
Angelika Zube

Dissertation, Karlsruher Institut für Technologie
KIT-Fakultät für Informatik

Tag der mündlichen Prüfung: 20. Juli 2018
Gutachter Prof. Dr.-Ing. habil. Jürgen Beyerer
Prof. Dr.-Ing. Dr. h. c. Heinz Wörn

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding the cover, pictures and graphs – is licensed
under a Creative Commons Attribution-Share Alike 4.0 International License
(CC BY-SA 4.0): <https://creativecommons.org/licenses/by-sa/4.0/deed.en>*



*The cover page is licensed under a Creative Commons
Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0):
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>*

Print on Demand 2018 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489

ISBN 978-3-7315-0855-7

DOI 10.5445/KSP/1000086157

Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung

zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften

der KIT-Fakultät für Informatik
des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Angelika Zube

aus Karlsruhe

Tag der mündlichen Prüfung: 20. Juli 2018
Erster Gutachter: Prof. Dr.-Ing. habil. Jürgen Beyerer
Zweiter Gutachter: Prof. Dr.-Ing. Dr. h. c. Heinz Wörn

Kurzfassung

Gemeinsame Arbeitsräume von Mensch und Roboter sowie die physische Interaktion von Mensch und Roboter sind vielversprechende Möglichkeiten, um Produktionsabläufe flexibler zu gestalten. Nicht nur in der Produktion sondern auch in anderen Lebensbereichen steigt das Interesse an Robotern, die in der Lage sind, physisch mit Menschen zu interagieren, wie beispielsweise Assistenzroboter im Haushalt und in der Pflege. Grundlegende Voraussetzung hierfür ist zunächst, dass die Sicherheit der Menschen im Roboterumfeld stets gewährleistet ist. Ebenso entscheidend für die Akzeptanz eines kollaborierenden Roboters ist, dass sich Menschen in der unmittelbaren Umgebung des Roboters wohlfühlen.

Um diese Ziele zu erreichen, muss der Roboter in der Lage sein, seine komplexe, dynamische Umgebung wahrzunehmen und Hindernisse (z. B. Menschen, andere Roboter, sonstige Objekte) zu detektieren. Die Roboterbewegung muss unter Berücksichtigung der detektierten Hindernisse und der beabsichtigten oder unbeabsichtigten Kontakte des Roboters mit seiner Umgebung geregelt werden, um Kollisionen zu vermeiden und angemessen auf Kontakte zu reagieren. Die vorliegende Arbeit befasst sich mit diesen beiden Aufgabenstellungen im Hinblick auf die Mensch-Roboter-Interaktion mobiler Manipulatoren im industriellen Umfeld.

Für die Bewegungsregelung stationärer oder mobiler Manipulatoren wird eine kartesische modellprädiktive Regelung erarbeitet und untersucht. Diese regelt die Endeffektorpose, d. h. die 3D-Position und Orientierung, des Manipulators zur Verfolgung von zeitabhängigen Solltrajektorien oder rein

geometrischen Sollpfaden. Die modellprädiktive Regelung beruht auf der Lösung eines Optimierungsproblems, welches das prädierte Roboterverhalten innerhalb eines vorgegebenen Zeithorizonts bewertet und zusätzliche Nebenbedingungen berücksichtigt. Mobile Plattform und Manipulator werden als eine kinematische Kette modelliert, sodass keine getrennte Regelung von Plattform und Manipulator notwendig ist. Die Regelung ist auf verschiedenartige Roboterkinematiken anwendbar.

Mobile Manipulatoren besitzen im Allgemeinen viele Freiheitsgrade, häufig mehr als für die Ausführung der kartesischen Aufgabe notwendig sind. Durch Erweiterung der Kostenfunktion und der Nebenbedingungen der modellprädiktiven Regelung wird diese Redundanz des Roboters ausgenutzt, um Kollisionen zu verhindern, ohne die kartesische Aufgabenausführung zu unterbrechen. Dies erfolgt unter Berücksichtigung der gesamten Robotergeometrie – nicht nur des Endeffektors – und beinhaltet sowohl die Vermeidung von Eigenkollisionen als auch von Kollisionen mit statischen und dynamischen Hindernissen im Arbeitsraum des Roboters. Dabei lassen sich unterschiedliche Reaktionsstrategien auf verschiedene Hindernisarten realisieren.

Falls ein Kontakt des Roboters mit seiner Umgebung unvermeidbar oder erwünscht ist, soll der Roboter häufig nachgiebig auf diesen Kontakt reagieren. Eine zusätzliche Erweiterung der kartesischen modellprädiktiven Regelung nutzt die Redundanz des Roboters, um nachgiebige Reaktionen des Roboters auf Kontakte zu ermöglichen, ohne die Aufgabenausführung unterbrechen zu müssen. Die Reaktion beruht auf den gemessenen externen Drehmomenten oder der geschätzten Kontaktkraft und setzt kein exaktes dynamisches Robotermodell voraus. Es werden sowohl Kontakte am Endeffektor als auch Kontakte am Roboterarm berücksichtigt.

Für die zur Kollisionsvermeidung notwendige Erfassung der Roboterumgebung wird ein Verfahren zur Erzeugung einer 3D-Repräsentation von dynamischen Hindernissen vorgeschlagen, welches die Information mehrerer heterogener Abstandssensoren fusioniert. Das Verfahren erfüllt die Anforderungen an die Arbeitsraumüberwachung mobiler Manipulatoren. Aus Sicherheitsgründen werden Verdeckungen (abgeschattete Bereiche) der

Hindernisse und des Roboters berücksichtigt. Dies ist besonders wichtig, wenn sich ein Objekt zwischen Roboter und Sensor befindet. Um diese Verdeckungen zu reduzieren und den erfassten Bereich zu vergrößern, wird die Information mehrerer Abstandssensoren unter Verwendung einer Octree-Datenstruktur fusioniert. Die Abstandssensoren können sowohl im Arbeitsraum als auch auf dem Roboter montiert sein. Letzteres ermöglicht die Erfassung großer Arbeitsräume mobiler Roboter mit wenigen Sensoren. Es können Abstandssensoren mit unterschiedlichen Messprinzipien eingesetzt werden (z. B. Laserscanner, Tiefenkameras nach dem Laufzeitprinzip, Tiefenkameras nach dem Triangulationsprinzip). Die 3D-Arbeitsraumerfassung wird auf die für die Kollisionsvermeidung besonders relevante nahe Roboterumgebung beschränkt. So können trotz 3D-Verarbeitung die notwendigen Datenverarbeitungsdatenraten erzielt werden. Die entwickelte Methode ist nicht an eine spezielle Sensoranordnung gekoppelt und vielfältig einsetzbar – für mobile und stationäre Manipulatoren, für mobile Plattformen, im Gebäude ebenso wie im Freien.

Die entwickelten Ansätze werden in der Simulation und auf einem realen mobilen Manipulator mit zehn Freiheitsgraden bestehend aus einer omnidirektionalen Plattform und einem Leichtbauarm analysiert und validiert. Das Konzept der kartesischen modellprädiktiven Regelung wird erfolgreich zur Verfolgung von Trajektorien und Pfaden der Endeffektorpose angewendet. Durch Erweiterung der Kostenfunktion und der Beschränkungen lassen sich mehrere Anforderungen an das Roboterverhalten in einer Regelung vereinigen. So können durch Ausweichbewegungen Kollisionen mit statischen und dynamischen Hindernissen vermieden werden, während der Endeffektor einem Sollpfad folgt. Die Hindernisse werden zur Laufzeit durch die Arbeitsraumüberwachung erfasst. Gleichzeitig reagiert der Roboter nachgiebig auf erwünschte oder nicht vermeidbare Kontakte. Das resultierende Roboterverhalten leistet einen Beitrag zu einer intuitiven Zusammenarbeit von Mensch und Roboter in gemeinsamen Arbeitsräumen.

Abstract

Shared human-robot workspaces and physical interaction between humans and robots are promising possibilities in order to build flexible production processes. But not only in production, also in other areas of daily living there is an increasing interest in physically collaborating robots, for example as assistant robots at home or in care for the elderly and disabled. The main prerequisite for such collaborating robots is to guarantee human safety in the robot's environment. But for the acceptance of physical human-robot interaction it is equally important that humans feel comfortable while working close to the robot.

Therefore, the robot has to be able to perceive its complex dynamic environment and to detect obstacles (e.g., humans, other robots, other objects). The robot motion has to be controlled considering the detected obstacles and the desired or undesired contacts of the robot with its environment so that the robot is able to avoid undesired collisions and react adequately to contacts. This thesis deals with these tasks regarding their application to human-robot interaction of mobile manipulators in an industrial environment.

A Cartesian model predictive control approach is developed for motion control of stationary or mobile manipulators. The robot's end-effector pose consisting of 3D position and orientation is controlled in order to follow a time-dependent reference trajectory or a geometric reference path. In each control step of the model predictive controller, an optimization problem is solved which evaluates the predicted robot behavior within a given time

horizon considering additional constraints. The controller uses a unified robot model of the mobile platform and the manipulator creating one kinematic chain. That way, it is not necessary to control platform and manipulator separately. The controller is applicable to different robot kinematics.

Mobile manipulators usually have many degrees of freedom, often more than what is needed for the Cartesian task execution. By extending the cost function and the constraints of the model predictive controller, redundancy is exploited to avoid collisions while at the same time the execution of the Cartesian task continues uninterrupted. The approach considers the whole robot geometry, not only the end-effector and includes both self-collision avoidance and collision avoidance with static as well as dynamic obstacles in the robot's environment. Different reaction strategies to different obstacle types are possible.

A further extension of the Cartesian model predictive controller exploits the robot's redundancy in order to allow for compliant reactions to contacts of the robot with its environment. The reaction is based on either the measured external joint torques or on the estimated contact force and does not require an exact dynamic robot model. contacts both at the end-effector and at the robot body are considered.

A prerequisite for collision avoidance is the perception of the robot's environment. Therefore, a method to construct a 3D representation of dynamic obstacles is proposed, which fuses the information of multiple heterogeneous depth sensors. The approach meets the requirements of workspace monitoring for a mobile manipulator. For safety reasons, occlusions of the obstacles and of the robot have to be considered. This is especially important if an object is located between the robot and the sensor. In order to reduce the occluded space and to enlarge the observed space, the information of multiple depth sensors is fused using an octree data structure. The depth sensors may be mounted in the workspace or on the robot itself. Onboard mounting allows for monitoring of large workspaces of mobile robots with only a comparatively low amount of sensors. Depth sensors with different measurement principles may be applied (e.g., 2D or 3D laser scanners, time-of-flight depth cameras, triangulation-based depth cameras). The full 3D

workspace representation is restricted to the surroundings next to the robot, which are most relevant for collision avoidance, in order to achieve the necessary data processing rates despite the time consuming 3D procedures. The workspace monitoring approach is not restricted to a special sensor setup and can be used in various applications – on mobile and stationary manipulators, on mobile platforms, in indoor and outdoor scenarios.

The proposed approaches are analyzed in simulation and on a real mobile manipulator with ten degrees of freedom consisting of an omnidirectional mobile platform and a lightweight arm. The Cartesian model predictive control concept is successfully applied to follow end-effector paths and trajectories. By extending the cost function and the constraints, multiple control goals are combined in one controller. While executing the Cartesian task, collisions with static and dynamic obstacles are avoided. The dynamic obstacles are detected online using the proposed 3D monitoring method. At the same time the robot reacts compliantly to non-avoidable contacts. The resulting robot behavior enables an intuitive collaboration of human and robot in shared workspaces.

Danksagung

Diese Arbeit entstand während meiner Tätigkeit als wissenschaftliche Mitarbeiterin am Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB). An dieser Stelle möchte ich allen danken, die zum Gelingen meiner Arbeit beigetragen haben.

Herrn Prof. Dr.-Ing. habil. Jürgen Beyerer danke ich für die Betreuung der Arbeit, hilfreiche Diskussionsbeiträge und die Möglichkeit, die Arbeit mit eigenen Ideen gestalten zu können. Herrn Prof. Dr.-Ing. Dr. h. c. Heinz Wörn danke ich für die Übernahme des Korreferats.

Ich bedanke mich bei allen Kollegen der Abteilung Mess-, Regel- und Diagnosesysteme (MRD) für zahlreiche größere und kleinere Unterstützungen und die angenehme Atmosphäre. Ein besonderer Dank geht an Christian Frese. Er hat durch die gute Zusammenarbeit in SAPHARI und diversen anderen Projekten, zahlreiche konstruktive Diskussionen, seine Arbeit am Demonstrator und geduldiges Korrekturlesen maßgeblich zu dieser Arbeit beigetragen. Janko Petereit und Thomas Emter standen immer für Fragen jeglicher Art zur Verfügung (Hardware, Videos, \LaTeX , ...). Christian Frey danke ich für sein Vertrauen in meine Arbeit und die Freiräume zur Durchführung dieser Arbeit.

Inhaltsverzeichnis

Abbildungsverzeichnis	xvii
Akronyme	xix
Symbole	xxi
1 Einleitung	1
1.1 Motivation	1
1.2 Aufgabenstellung und Ziele	4
1.3 Eigene Beiträge	5
1.4 Gliederung	6
2 Stand der Forschung	9
2.1 Arbeitsraumüberwachung	9
2.1.1 Sensortypen	10
2.1.2 Sensorplatzierung	11
2.1.3 Arbeitsraumüberwachung mit Intensitätskameras	12
2.1.4 Arbeitsraumüberwachung mit Abstandssensoren	12
2.2 Kartesische Regelung	13
2.2.1 Regelung stationärer Manipulatoren	14
2.2.2 Regelung mobiler Manipulatoren	15
2.2.3 Regelung mobiler Plattformen	17
2.3 Kollisionsvermeidung	18

2.4 Kontaktregelung	19
2.4.1 Kraft- und Momentenschätzung	20
2.4.2 Regelungsstrategien	21
2.5 Modellprädiktive Regelung	22
2.6 Zusammenfassung und Bewertung	24
3 Arbeitsraumüberwachung	27
3.1 Anforderungen	27
3.2 Unterscheidung zwischen Nah- und Weitbereich	30
3.3 Erzeugung einer 3D-Hindernisdarstellung mit Verdeckungsanalyse	32
3.3.1 Octree-Datenstruktur	34
3.3.2 Sensormodell	34
3.3.3 Roboterfilter	35
3.3.4 Sensordatenvorverarbeitung	36
3.3.5 Verdeckungsanalyse	36
3.3.6 Sensorfusion	38
3.4 Objektklassifikation	43
3.5 Zusammenfassung	44
4 Robotermodell	45
4.1 Begriffsklärung	45
4.2 Kinematisches Robotermodell	46
4.2.1 Darstellung der Orientierung	47
4.2.2 Ermittlung der direkten Kinematik	48
4.2.3 Anmerkungen zu mobilen Plattformen	50
4.2.4 Jacobi-Matrix	50
4.2.5 Hesse-Matrix	51
4.2.6 Diskretisierung des kinematischen Robotermodells	51
4.3 Dynamisches Robotermodell	52
5 Kartesische modellprädiktive Roboterregelung	55
5.1 Problemstellung	55
5.1.1 Trajektorienverfolgung	56

5.1.2	Pfadverfolgung	57
5.2	Grundlagen der modellprädiktiven Regelung	59
5.2.1	Prinzip der modellprädiktiven Regelung	59
5.2.2	Formulierung des NMPC-Optimierungsproblems	61
5.2.3	Diskretisiertes NMPC-Optimierungsproblem	62
5.2.4	Lösung nichtlinearer Optimierungsprobleme mit Beschränkungen	64
5.3	Verfolgung von Solltrajektorien der Endeffektorpose	69
5.3.1	Nebenbedingungen	70
5.3.2	Gütekriterium	71
5.3.3	Wahl der Startwerte	73
5.4	Verfolgung von Sollpfaden der Endeffektorpose	74
5.4.1	Verwendung der maximalen kartesischen Geschwindigkeiten	75
5.4.2	Pfadoptimierung	78
5.5	Modifikationen der kartesischen Regelung	82
5.6	Zusammenfassung	83
6	Modellprädiktive Kollisionsvermeidung	85
6.1	Anforderungen	85
6.2	Ansatz	86
6.3	Vermeidung von Eigenkollisionen	88
6.4	Kartesische Roboterbeschränkungen	89
6.5	Vermeidung von Kollisionen mit statischen Hindernissen	89
6.6	Vermeidung von Kollisionen mit dynamischen Hindernissen	90
6.7	Reduzierung der Anzahl der Nebenbedingungen zur Kollisionsvermeidung	93
6.7.1	Reduzierung der Anzahl der Nebenbedingungen am Beispiel eines mobilen Manipulators mit zehn Freiheitsgraden	94
6.8	Kombination von Arbeitsraumüberwachung und Kollisionsvermeidung	98
6.9	Zusammenfassung	99

7 Modellprädiktive Kontaktregelung	101
7.1 Anforderungen	101
7.2 Kontaktschätzung	102
7.3 Regelung von Kontakten am Roboterarm	103
7.3.1 Momentenbasierte Kontaktregelung	104
7.3.2 Kraftbasierte Kontaktregelung	105
7.4 Regelung von Kontakten am Endeffektor	107
7.5 Kombination von Kollisionsvermeidung und Kontaktregelung	108
7.6 Zusammenfassung	108
8 Ergebnisse	111
8.1 Demonstrator	111
8.1.1 Versuchsplattform	112
8.1.2 Implementierung	115
8.1.3 Simulation	117
8.2 Kartesische Roboterregelung	118
8.2.1 Fehlermaße	119
8.2.2 Verfolgen von Sollpfaden	121
8.3 Kollisionsvermeidung	123
8.3.1 Arbeitsraumüberwachung	124
8.3.2 Kollisionsvermeidung mit dynamischem Hindernis	130
8.3.3 Kollisionsvermeidung mit statischen und dynamischen Hindernissen	133
8.4 Kontaktregelung	138
8.4.1 Momentenbasierte Kontaktregelung mit Kontakt am Roboterarm	140
8.4.2 Kraftbasierte Kontaktregelung mit Kontakt am Roboterarm	141
8.5 Kombinierte Kollisionsvermeidung und Kontaktregelung	145
8.6 Zusammenfassung	152
9 Zusammenfassung und Ausblick	155
9.1 Zusammenfassung der Beiträge	155
9.2 Ausblick	157

A Quaternionen	159
B Denavit-Hartenberg-Parameter der Versuchsplattform	163
Literatur	165
Liste der Veröffentlichungen	175

Abbildungsverzeichnis

3.1	Prinzip der Verdeckungen.	30
3.2	Visualisierung des erreichbaren Raums.	32
3.3	Ablaufdiagramm der Arbeitsraumüberwachung.	33
3.4	Sensormodell.	35
3.5	Prinzip der 3D-Hindernisdarstellung (Beispiel 1).	40
3.6	Prinzip der 3D-Hindernisdarstellung (Beispiel 2).	41
3.7	Prinzip der 3D-Hindernisdarstellung (Beispiel 3).	42
4.1	Parameter der Denavit-Hartenberg-Konvention.	49
5.1	Prinzip der Trajektorienverfolgung	56
5.2	Prinzip der Pfadverfolgung	58
5.3	Prinzip des gleitenden Horizonts.	60
6.1	Approximationsbeispiel der Robotergeometrie mit Kugeln.	87
6.2	Simulationsszenario zur Kollisionsvermeidung.	94
6.3	Simulationsergebnisse der Kollisionsvermeidung.	97
8.1	Versuchsplattform <i>IOSB.amp O2</i>	112
8.2	Gesamtsystem zur Evaluation der entwickelten Verfahren.	116
8.3	Simulationsergebnisse der kartesischen modellprädiktiven Regelung mit verschiedenen Ansätzen zur Pfadverfolgung.	122
8.4	Visualisierung der 3D-Hindernisdarstellung.	126
8.5	Bildfolge der Kollisionsvermeidung mit dynamischen Hindernissen.	131
8.6	Plattformposition des mobilen Manipulators mit und ohne Vermeidung eines dynamischen Hindernisses.	132

8.7	Gelenkkonfiguration mit Vermeidung eines dynamischen Hindernisses	132
8.8	Ergebnisse der Kollisionsvermeidung mit einem dynamischen Hindernis.	134
8.9	Schnappschüsse der Kollisionsvermeidung mit statischen und dynamischen Hindernissen.	136
8.10	Ergebnisse der Kollisionsvermeidung mit statischen und dynamischen Hindernissen.	137
8.11	Versuchsaufbau des stationären Manipulators mit Stift.	139
8.12	Bildfolge der momentenbasierten Kontaktregelung.	141
8.13	Ergebnisse der momentenbasierten Kontaktregelung.	142
8.14	Bildfolge der kraftbasierten Kontaktregelung.	143
8.15	Ergebnisse der kraftbasierten Kontaktregelung.	144
8.16	Bildfolge des parallelen Einsatzes von Kollisionsvermeidung und kraftbasierter Kontaktregelung.	147
8.17	Endeffektorposition des parallelen Einsatzes von Kollisionsvermeidung und kraftbasierter Kontaktregelung.	148
8.18	Ergebnisse des parallelen Einsatzes von Kollisionsvermeidung und kraftbasierter Kontaktregelung.	150
8.19	Visualisierung der statischen und dynamischen Hindernisse.	151
B.1	Denavit-Hartenberg (DH)-Koordinatensysteme der Versuchsplattform <i>IOSB.amp O2</i>	164

Akronyme

DH	Denavit-Hartenberg
EE	Endeffektor
GPU	Grafikprozessor (engl. <i>graphics processing unit</i>)
IFR	International Federation of Robotics
KKT	Karush-Kuhn-Tucker-Bedingungen
NLP	nichtlineares Optimierungsproblem (engl. <i>Nonlinear Programming</i>)
NMPC	nichtlineare modellprädiktive Regelung (engl. <i>Nonlinear Model Predictive Control</i>)
QP	quadratisches Optimierungsproblem (engl. <i>Quadratic Programming</i>)
ROS	Robot Operating System
SAPHARI	Safe and Autonomous Physical Human-Aware Robot Interaction
SQP	sequentielle quadratische Programmierung (engl. <i>Sequential Quadratic Programming</i>)
URDF	Unified Robot Description Format

Symbole

Generelle Notationsvereinbarungen

x, y, \dots	Skalar
$\mathbf{x}, \mathbf{y}, \dots$	Vektor, $\mathbf{x} = [x_1 \ x_2 \ \dots]^\top$
$\mathbf{A}, \mathbf{B}, \dots$	Matrix
$\mathbf{A}^\top, \mathbf{x}^\top$	transponierte Matrix, transponierter Vektor
\mathbf{I}	Identitätsmatrix
diag	Diagonalmatrix
$\mathcal{A}, \mathcal{B}, \dots$	Mengen, Folgen

Römische Buchstaben

\mathbf{a}	Vektor der Gelenkbeschleunigungen (von Rotations- und Translationsgelenken)
\mathbf{a}_{\max}	Vektor der maximalen Beträge der Gelenkbeschleunigungen
a_{DH}	Denavit-Hartenberg-Parameter: Translation entlang x -Achse
b	Octree-Knoten
c	Beschränkungen
C	Matrix der Coriolis- und Zentrifugalkräfte
d	minimaler Abstand zwischen Roboter und Hindernis
d_a	Abstand zum Pfadende, bei dem der Abbremsvorgang beginnt
d_{DH}	Denavit-Hartenberg-Parameter: Translation entlang z -Achse
d_{end}	Abstand zum Pfadende

$d_{\text{dyn}, i, j}$	Abstand zwischen den Mittelpunkten der Roboterkugel i und der Hinderniskugel j eines dynamischen Hindernisses
$d_{\text{stat}, i, j}$	Abstand zwischen den Mittelpunkten der Roboterkugel i und der Hinderniskugel j eines statischen Hindernisses
$d_{\text{dyn}, i, j, \text{min}}$	minimal erlaubter Abstand zwischen den Mittelpunkten der Roboterkugel i und der Hinderniskugel j eines dynamischen Hindernisses
d_{pfad}	minimaler Abstand zwischen Endeffektorposition und Pfad
$d_{\text{R}, i, j}$	Abstand zwischen den Mittelpunkten der Roboterkugeln i und j
$d_{\text{S}, \text{max}, i}$	maximale Reichweite des Sensors i
d_{safety}	Sicherheitsabstand zwischen Roboter und Hindernis
e_{p}	Abstand zur aktuellen Sollposition
\mathbf{e}_{pfad}	Pfadverfolgungsfehler
\mathbf{e}_{traj}	Trajektorienverfolgungsfehler
E	Endbestrafung des Gütekriteriums
E_{C}	Endbestrafung zur Kontaktregelung
E_{K}	Endbestrafung zur Pfad- oder Trajektorienverfolgung
E_{O}	Endbestrafung zur Kollisionsvermeidung
f_{C}	Gewichtung der Endbestrafung zur Kontaktregelung
f_{O}	Gewichtung der Endbestrafung zur Kollisionsvermeidung
F	Kostenfunktion des Gütekriteriums
F_{C}	Kostenfunktion zur Kontaktregelung
\mathbf{F}_{ext}	Vektor der externen Kontaktkräfte
F_{K}	Kostenfunktion zur Pfad- oder Trajektorienverfolgung
F_{O}	Kostenfunktion zur Kollisionsvermeidung
\mathbf{g}	Gravitationsvektor
$\mathbf{h}(\mathbf{q}, \mathbf{w})$	nicht-holonome Beschränkung
$\mathbf{H}_i(\mathbf{q})$	Hesse-Matrix des i -ten Elements der Endeffektorpose
J	Gütekriterium
$\mathbf{J}(\mathbf{q})$	Jacobi-Matrix der Endeffektorpose
$\mathbf{J}_{\text{C}}(\mathbf{q})$	Jacobi-Matrix des Kontaktpunktes
k	Zeitindex

$\mathbf{k}(\mathbf{q})$	direkte Kinematik
L	Lagrange-Funktion
M	Dimension des Aufgabenraums
\mathbf{M}	Massenträgheitsmatrix
N	Anzahl der Robotergelenke
N_L	Anzahl der Roboterglieder
N_O	Anzahl der Hindernispunkte bzw. Anzahl der Kugeln zur Approximation der Hindernisse
N_p	Anzahl der Prädiktionsschritte
N_R	Anzahl der Roboterpunkte bzw. Anzahl der Kugeln zur Approximation der Robotergeometrie
N_S	Anzahl der Strahlen eines Sensors
\mathbf{o}	Quaternion
\mathbf{o}_{EE}	Endeffektororientierung
\mathbf{o}_{pfad}	Pfad der Endeffektororientierung
\mathbf{o}_{traj}	Trajektorie der Endeffektororientierung
\mathbf{p}_C	Position des Kontaktpunkts
\mathbf{p}_{EE}	Endeffektorposition
\mathbf{p}_O	Hindernispunkt
$\mathbf{p}_{O,\text{dyn},j}$	Mittelpunkt der Hinderniskugel eines dynamischen Hindernisses
$\mathbf{p}_{O,\text{stat},j}$	Mittelpunkt der Hinderniskugel eines statischen Hindernisses
\mathbf{p}_{pfad}	Pfad der Endeffektorposition
\mathbf{p}_R	Roboterpunkt bzw. Position des Mittelpunkts der Kugeln zur Approximation der Robotergeometrie
\mathbf{p}_{traj}	Trajektorie der Endeffektorposition
\mathbf{q}	Vektor der Gelenkpositionen (von Rotations- und Translationsgelenken)
\mathbf{q}_m	Gelenkposition in der Mitte des zulässigen Gelenkbereichs
\mathbf{q}_{max}	maximale Gelenkpositionen
\mathbf{q}_{min}	minimale Gelenkpositionen
Q_C	Gewichtungsmatrix für die Kostenfunktion zur Kontaktregelung

Q_e	Gewichtungsmatrix des Pfad- bzw. Trajektorienverfolgungsfehlers für die Kostenfunktion
Q_q	Gewichtungsmatrix der Gelenkpositionsterme für die Kostenfunktion
Q_w	Gewichtungsmatrix der Gelenkgeschwindigkeitsterme für die Kostenfunktion
Q_θ	Gewichtung des Pfadparameters für die Kostenfunktion
$Q_{\hat{\theta}}$	Gewichtung der Pfadparametergeschwindigkeit für die Kostenfunktion
$\mathbf{r}_{i,j}$	normierter Richtungsvektor des j -ten Strahls des Sensors i
r_R	Radius der Kugel zur Approximation der Robotergeometrie
$r_{O,dyn,j}$	Radius der Hinderniskugel eines dynamischen Hindernisses
$r_{O,stat,j}$	Radius der Hinderniskugel eines statischen Hindernisses
$r_{V,j}$	Kantenlänge des Voxels j
\mathbf{R}	Rotationsmatrix
R_e	Gewichtungsmatrix des Pfad- bzw. Trajektorienverfolgungsfehlers für die Endbestrafung
R_q	Gewichtungsmatrix der Gelenkpositionsterme für die Endbestrafung
R_θ	Gewichtung des Pfadparameters für die Kostenfunktion
s^*	Vorzeichen der Sollquaternion, $s^* \in \{-1,1\}$
s_u	Sensorursprung
S	Sensorstrahl
S_O	hinter dem Hindernispunkt \mathbf{p}_O liegender Anteil des Sensorstrahls
S_R	hinter dem Roboterpunkt \mathbf{p}_R liegender Anteil des Sensorstrahls
t	Zeit
T_j^i	homogene Transformation zur Abbildung des Koordinatensystems i in das Koordinatensystem j
T_p	Prädiktionshorizont
T_s	Abtastzeit
\mathbf{u}	Stellgrößenvektor
v	translatorische Plattformgeschwindigkeit

$v_{C,d}$	Sollgeschwindigkeit des Kontaktpunkts
v_{EE}	translatorische Endeffektorgeschwindigkeit
v_{pfad}	translatorische Pfadgeschwindigkeit
\mathbf{w}	Vektor der Gelenkgeschwindigkeiten (von Rotations- und Translationsgelenken)
w_{max}	maximaler Betrag der Gelenkgeschwindigkeiten
\mathbf{x}	Zustandsgrößenvektor
\mathbf{y}	Ausgangsgrößenvektor
\mathbf{y}_{EE}	Endeffektorpose (bestehend aus Position \mathbf{p}_{EE} und Orientierung \mathbf{o}_{EE})
\mathbf{y}_{pfad}	Pfad der Endeffektorpose (bestehend aus Position \mathbf{p}_{pfad} und Orientierung \mathbf{o}_{pfad})
\mathbf{y}_{traj}	zeitabhängige Trajektorie der Endeffektorpose (bestehend aus Position \mathbf{p}_{traj} und Orientierung \mathbf{o}_{traj})
\mathbf{z}	Vektor der Optimierungsvariablen

Griechische Buchstaben

α_a	Winkel zwischen aktueller Orientierung und Orientierung am Pfadende, bei dem der Abbremsvorgang beginnt
α_{DH}	Denavit-Hartenberg-Parameter: Rotation um x -Achse
α_{end}	Winkel zwischen der aktuellen Endeffektororientierung und der Orientierung am Pfadende
α_{pfad}	minimaler Winkel zwischen Endeffektororientierung und Pfad
β	Dämpfungsparameter des Gütekriteriums für die Gelenkgeschwindigkeit
γ	Dämpfungsparameter des Gütekriteriums für die Pfadparametergeschwindigkeit
ϵ	Orientierungsfehler des Endeffektors
ϵ_{pfad}	minimale Winkelabweichung zwischen Endeffektororientierung und Pfadorientierung
δ	Suchrichtung
θ	Pfadparameter
θ_{DH}	Denavit-Hartenberg-Parameter: Rotation um z -Achse

$\dot{\theta}$	Pfadparametergeschwindigkeit
λ	Lagrange-Faktor
λ_{F}	Verstärkungsfaktor der kraftbasierten Kontaktregelung
$\lambda_{\text{O},i}$	Gewichtung des Hindernisabstands zur Roboterkugel i in der Kostenfunktion
λ_{T}	Verstärkungsfaktor der momentenbasierten Kontaktregelung
τ	Antriebsmoment
τ_{ext}	externe Drehmomente der Roboterjelenke
$\dot{\psi}$	rotatorische Plattformgeschwindigkeit
ω_{pfad}	rotatorische Pfadgeschwindigkeit

Sonstige

\mathcal{A}	Aufgabenraum des Roboters
\mathcal{B}	Octree
\mathcal{C}	Gelenkraum des Roboters
\mathcal{E}	Menge der Gleichheitsbeschränkungen
\mathcal{F}	Menge der freien Octree-Knoten im Gesichtsfeld eines Sensors
\mathcal{H}	Menge der durch ein Hindernis verdeckten Octree-Knoten eines Sensors
\mathcal{I}	Menge der Ungleichheitsbeschränkungen
\mathcal{N}	Menge der Octree-Knoten innerhalb des Nahbereichs des Roboters
\mathcal{O}	Menge der in der 3D-Hindernisdarstellung belegten Octree-Knoten
\mathcal{P}	Menge der Octree-Knoten, die einen von einem Sensor gemessenen Hindernispunkt enthalten
\mathcal{Q}	Folge von N_{p} Gelenkpositionen \mathbf{q}
\mathcal{R}	Menge der durch den Roboter belegten oder verdeckten Octree-Knoten eines Sensors
\mathcal{S}_i	Menge der Octree-Knoten im Gesichtsfeld des Sensors i
\mathcal{T}	Folge von N_{p} Pfadparametern θ
\mathcal{V}	Folge von N_{p} Pfadparametergeschwindigkeiten $\dot{\theta}$
\mathcal{W}	Folge von N_{p} Gelenkgeschwindigkeiten \mathbf{w}

Kapitel 1

Einleitung

1.1 Motivation

Die ersten Industrieroboter mit numerischer Steuerung¹ wurden in den 1960er Jahren in der Automobilindustrie eingesetzt [Jac04]. Seitdem ist die Zahl der im industriellen Umfeld eingesetzten Roboter enorm gestiegen. Im Jahr 2015 wurden laut Statistik der International Federation of Robotics (IFR) weltweit über 250 000 Industrieroboter verkauft [IFR16]. Industrieroboter arbeiten typischerweise in einer bekannten, strukturierten Umgebung mit fest programmierten oder durch Sensordaten beeinflussten Bewegungen. Die Sicherheit des Menschen wird dabei durch eine strikte Trennung von Mensch und Roboter erreicht. Die Arbeitsräume der Roboter werden durch Zäune, Lichtvorhänge oder ähnliche Sicherheitsmaßnahmen begrenzt. Betritt ein Mensch den Arbeitsraum, wird der Roboter sofort angehalten.

Aktuelle Untersuchungen zeigen jedoch, dass Roboter, die sich den Arbeitsraum mit Menschen teilen oder sogar Hand in Hand mit Menschen arbeiten, viele Vorteile mit sich bringen [Blo16]. Beispielsweise kann der Platzbedarf reduziert werden, da keine Sicherheitszäune benötigt werden

1. Numerische Steuerungen setzen Steuerbefehle, die als Code auf Datenträgern vorliegen, in Arbeits- oder Bewegungsabläufe um. Als Datenträger dienten in den 1960er Jahren beispielsweise Lochkarten, Lochstreifen oder Magnetbänder.

[Bog16]. Zudem wird erwartet, dass sich kollaborierende Roboter flexibler einsetzen lassen und sich somit Änderungen im Produktionsablauf oder auch kleine Chargen mit vertretbarem Aufwand realisieren lassen. Es wurden Prozesse untersucht, in denen die Qualität oder die Effizienz der Arbeit gesteigert werden konnte, wenn Mensch und Roboter zusammenarbeiten [Blo16]. Der Grund dafür ist, dass sich die Fähigkeiten von Mensch und Roboter gegenseitig ergänzen. Roboter eignen sich besonders für repetitive, ermüdende und kraftintensive Tätigkeiten während Menschen komplexe und kognitiv anspruchsvolle Aufgaben besser erfüllen.

Um Roboter, die sicher und intuitiv mit Menschen interagieren, im industriellen Umfeld einsetzen zu können, müssen verschiedene Aspekte der Mensch-Roboter-Interaktion untersucht und neue Konzepte entwickelt werden. Dieses Ziel hatte das EU-Projekt Safe and Autonomous Physical Human-Aware Robot Interaction (SAPHARI). Inhalt dieses Projekts waren biomechanische Sicherheitsanalysen, ein menschenfreundliches Hardwaredesign, die Wahrnehmung der Umwelt zur Verfolgung, zum Verstehen und Präzisieren menschlicher Bewegungen, Regelungsstrategien, Bewegungs- und Aufgabenplanung in Gegenwart von Menschen und die Kommunikation zwischen Mensch und Roboter. Der Hauptteil der vorliegenden Arbeit ist im Rahmen dieses Projekts entstanden.

Voraussetzung dafür, dass die Mensch-Roboter-Interaktion gelingen kann, ist zunächst die Sicherheit des Menschen zu gewährleisten. Hierzu gibt es mehrere Ansätze [Hei03]: Die Sicherheit kann durch die Konstruktion des Roboters erzielt werden, indem der Roboter leicht, schwach und nachgiebig mit einer weichen Oberfläche gebaut wird. Mit einem solchen sicheren Design ist der Roboter allerdings für viele Aufgaben, z. B. das Bewegen von größeren Lasten, nicht mehr einsetzbar. Eine weitere Möglichkeit ist die Detektion von stattfindenden Kontakten, z. B. basierend auf Kraft-Momenten-Sensoren am Endeffektor, Momentenmessungen in den Gelenken oder anderen taktilen Sensoren. Wird ein Kontakt detektiert, wird die Sicherheit durch eine angemessene Reaktion wie Anhalten oder ein nachgiebiges Verhalten realisiert. Ein anderer Ansatz ist es, die Kollision von Mensch und Roboter zu vermeiden, bevor sie stattfindet. Hierzu wird

die Roboterumgebung mit Sensoren überwacht, z.B. mit Kameras oder Abstandssensoren wie Laserscannern. Wird ein Objekt nahe des Roboters detektiert, hält der Roboter an oder führt eine Ausweichbewegung durch. In vielen Anwendungsfällen ist es jedoch schwierig, die Sicherheit mit einem dieser Ansätze alleine zu gewährleisten. Stattdessen werden zwei oder alle drei Strategien miteinander kombiniert: Der Roboter wird so leicht wie möglich und möglichst ohne scharfe Kanten und Ecken konstruiert. Basierend auf der sensoriiellen Überwachung des Arbeitsraums werden Kollisionen soweit möglich vermieden. Auf dennoch stattfindende bzw. erwünschte Kontakte muss sicher reagiert werden.

Auf dem Markt verfügbare Robotersysteme für die Mensch-Roboter-Interaktion, wie z. B. der KR 5 SI von MRK-Systeme oder die APAS-Familie von Bosch, gewährleisten die Sicherheit des Menschen im Allgemeinen durch Anhalten des Roboters, sobald ein Kontakt oder eine drohende Kollision detektiert wird [Bog16]. Dies kann zu häufigen Unterbrechungen des Roboters bei der Ausführung seiner Aufgabe führen und schränkt die Möglichkeiten der Kooperation zwischen Mensch und Roboter ein.

Um die Effizienz zu steigern und die Einsatzmöglichkeiten zu verbessern, beschäftigen sich aktuelle Forschungsarbeiten damit, unerwünschte Kollisionen durch geeignete Roboterbewegungen zu vermeiden und nachgiebig auf stattfindende Kontakte zu reagieren, um so die Aufgabenausführung trotz stattfindender Interaktion sicher fortsetzen zu können. Für die Vermeidung von Kollisionen ist die Überwachung des Arbeitsraums des Roboters mittels Sensoren notwendig, um statische und dynamische Hindernisse zu detektieren. Die Roboterumgebung ist in Anwesenheit von Menschen im Gegensatz zu klassischen Industrieanlagen immer als teilweise unstrukturiert zu betrachten, da Menschen Gegenstände verschieben, entfernen oder neue Gegenstände abstellen können und auch selbst dynamische Hindernisse darstellen. Für eine sichere, nachgiebige Reaktion auf stattfindende Kontakte müssen diese sensoriiell detektiert werden, beispielsweise mittels Kraft- oder Momentensensoren oder einer künstlichen Haut. Die Durchführung kollisionsfreier bzw. nachgiebiger Bewegungen ist Aufgabe der Planungs- und Regelungsalgorithmen, ebenso wie die Ausführung

der eigentlichen Roboter Aufgabe. Während bei mobilen Plattformen mit wenigen Freiheitsgraden die Bewegungsplanung unter Berücksichtigung dynamischer Hindernisse in Echtzeit erfolgen kann (z. B. [Pet13]) und vom Regler lediglich die Umsetzung der geplanten Bewegung auf dem Roboter erfordert, ist dies bei stationären und mobilen Manipulatoren mit vielen Freiheitsgraden in der Regel nicht möglich. Stattdessen erfolgt eine globale Planung, die eine kollisionsfreie Bewegung bis zur Zielpose plant, mit einer längeren Zykluszeit (z. B. [Sey16]). Die Bewegungsregelung regelt den Roboter mit höherer Taktrate entsprechend der geplanten Trajektorie oder des geplanten Pfads, muss dabei aber Hindernisse und Kontakte lokal berücksichtigen. So lassen sich schnelle Reaktionen auf die dynamische Umgebung ohne Unterbrechung der Aufgabe realisieren.

1.2 Aufgabenstellung und Ziele

In dieser Arbeit wird ein neues Konzept zur Bewegungsregelung für die Mensch-Roboter-Interaktion untersucht. Der Roboter teilt sich seinen Arbeitsraum mit Menschen und es können beabsichtigte und unbeabsichtigte physische Interaktionen stattfinden. Die Umgebung ist teilweise unstrukturiert.

Die Bewegungsregelung regelt die Pose, d. h. die Position und Orientierung, des Endeffektors und soll für zusätzliche Aufgaben wie die Kollisionsvermeidung und die nachgiebige Reaktion auf Kontakte erweitert werden. Es wird vorausgesetzt, dass ein befahrbarer Pfad oder eine befahrbare Trajektorie der Endeffektorpose aus einer übergeordneten Planungsschicht vorliegt.

Die Regelung soll universell einsetzbar sein: auf verschiedenen Roboterkinematiken von stationären oder mobilen Manipulatoren und muss für unterschiedliche Zielvorgaben parametrierbar sein. Mobile Manipulatoren besitzen im Allgemeinen viele Freiheitsgrade, häufig mehr als für die Durchführung der eigentlichen Aufgabe notwendig ist. Ihr Arbeitsraum ist größer und sie besitzen eine höhere Komplexität als stationäre Manipulatoren. Doch gerade diese Redundanz kann ausgenutzt werden, um neben der

eigentlichen Roboter Aufgabe auch die Anforderungen der Mensch-Roboter-Interaktion zu erfüllen.

Als Voraussetzung für die Kollisionsvermeidung ist ein Verfahren zur Arbeitsraumüberwachung erforderlich. Für die Mensch-Roboter-Interaktion müssen die Hindernisse in der Roboterumgebung dreidimensional dargestellt werden. Dabei müssen die durch Hindernisse verdeckten Bereiche berücksichtigt werden, die für die Sensoren nicht einsehbar sind. Der Algorithmus soll generisch mit unterschiedlichen Sensoren und verschiedenen Sensoranordnungen einsetzbar sein.

1.3 Eigene Beiträge

Der Hauptbeitrag dieser Arbeit liegt in der kartesischen Bewegungsregelung stationärer und mobiler Manipulatoren zur Mensch-Roboter-Interaktion. Des Weiteren wird ein Verfahren zur 3D-Arbeitsraumüberwachung erarbeitet. Die einzelnen Beiträge sind:

- **Kartesische modellprädiktive Regelung:** Übertragung des Konzepts der nichtlinearen modellprädiktiven Regelung (engl. *Nonlinear Model Predictive Control*, NMPC) auf die Regelung von stationären oder mobilen Manipulatoren im kartesischen Aufgabenraum (Verfolgung von zeitbehafteten Solltrajektorien oder geometrischen Sollpfaden der Endeffektorpose) basierend auf einem allgemeinen kinematischen Robotermodell [Zub15a]
- **Flexibilität des Regelungskonzepts:** Nachweis, dass sich die zuvor genannte kartesische modellprädiktive Regelung eignet, um auf einfache Art unterschiedliche Anforderungen an das Roboter Verhalten in einer Regelung zu vereinen (z. B. Kollisionsvermeidung und Kontaktreaktion) [Zub15b; Zub16]
- **Kontaktreaktion:** Erweiterung der kartesischen modellprädiktiven Regelung zur Erzielung nachgiebiger Reaktionen auf stattfindende Kontakte, ohne dass hierfür ein exaktes dynamisches Robotermodell notwendig ist [Zub16]

- **Kollisionsvermeidung:** Definition und Analyse geeigneter Kostenfunktionen und Beschränkungen für die kartesische modellprädiktive Regelung zur Ausführung von Ausweichbewegungen zur Vermeidung drohender Kollisionen, Berücksichtigung der gesamten Robotergeometrie (nicht nur des Endeffektors) für Fremd- und Eigenkollisionen, unterschiedliche Ausweichbewegungen bei statischen und dynamischen Hindernissen [Zub15b; Zub15c]
- **3D-Arbeitsraumüberwachung für mobile Manipulatoren:** Informationsfusion mehrerer heterogener Abstandssensoren, für auf dem Roboter oder im Arbeitsraum installierte Sensoren, Berücksichtigung von Verdeckungen, Beschränkung auf den aktuellen Nahbereich [Fet13; Fet14; Zub14; Zub15c]

Nach Wissen der Autorin wurde die NMPC zuvor noch nicht zur Regelung der vollständigen Pose eines stationären oder mobilen Manipulators eingesetzt. Im Vergleich zu aktuellen Verfahren der Bewegungsregelung stationärer oder mobiler Manipulatoren (siehe Kapitel 2) bietet die NMPC den Vorteil, dass Probleme mit singulären Gelenkstellungen ausgeschlossen sind. Auch lassen sich Anforderungen an das Roboterverhalten sehr intuitiv sowohl in Form von Kostenfunktionen als auch in Form von Beschränkungen formulieren. So lassen sich beispielsweise unterschiedliche Reaktionen auf statische und dynamische Hindernisse umsetzen, was in der Literatur nach Kenntnisstand der Autorin bislang nicht untersucht wurde. Kollisionsvermeidung, Eigenkollisionsvermeidung und Kontaktregelung können in ein einheitliches Regelungskonzept integriert werden.

Die Evaluation der Verfahren erfolgt in der Simulation sowie auf einem realen mobilen Manipulator mit zehn Freiheitsgraden und einem stationären Manipulator mit sieben Freiheitsgraden.

1.4 Gliederung

Im Folgenden wird in Kapitel 2 zunächst ein Überblick über den Stand der Forschung der für diese Arbeit relevanten Bereiche gegeben. Gängige

Ansätze zur kartesischen Regelung stationärer und mobiler Manipulatoren werden zusammengefasst, sowie die für die Mensch-Roboter-Interaktion grundlegenden Bereiche der Kollisionsvermeidung und der Reaktion auf stattfindende Kontakte betrachtet. Als Voraussetzung für die Kollisionsvermeidung und Kontaktreaktion werden bestehende Verfahren zur sensorischen Überwachung des Arbeitsraums und zur Erfassung von stattfindenden Kontakten zusammengefasst. In Kapitel 3 wird eine Methode zur 3D-Arbeitsraumüberwachung des Roboters vorgestellt. Sie fusioniert die Information mehrerer heterogener Abstandssensoren unter Berücksichtigung von Verdeckungen und stellt die Voraussetzung für kollisionsfreie Roboterbewegungen dar. Als Grundlage der folgenden Bewegungsregelung wird in Kapitel 4 ein allgemeingültiges kinematisches und dynamisches Robotermodell definiert. Basierend auf dem kinematischen Robotermodell werden in Kapitel 5 zunächst die Problemstellungen der Pfad- und Trajektorienverfolgung formal definiert. Es folgt die Übertragung des Prinzips der modellprädiktiven Regelung auf die Bewegungsregelung stationärer oder mobiler Manipulatoren im kartesischen Arbeitsraum zur Lösung dieser Problemstellungen. In Kapitel 6 wird die zuvor erarbeitete kartesische modellprädiktive Regelung erweitert, um sowohl Eigen- als auch Fremdkollisionen vermeiden zu können. Die einfache Anpassung der Regelung an spezifische Anforderungen wird demonstriert, indem unterschiedliche Reaktionen auf statische und dynamische Hindernisse untersucht werden. Eine zusätzliche Erweiterung der Regelung wird in Kapitel 7 vorgestellt. Sie dient der nachgiebigen Reaktion auf stattfindende Kontakte, während der die kartesische Roboteranfrage weiter ausgeführt werden kann. Es werden zwei Ansätze betrachtet: die momentenbasierte Kontaktregelung und die kraftbasierte Kontaktregelung. Die Analyse der Ergebnisse folgt in Kapitel 8. Es wird zunächst der Aufbau des Demonstrators vorgestellt. Es folgen Analysen der kartesischen Roboterregelung, der Arbeitsraumüberwachung und Kollisionsvermeidung sowie der Kontaktregelung. Die Arbeit schließt mit einer Zusammenfassung und einem Ausblick in Kapitel 9.

Kapitel 2

Stand der Forschung

In diesem Kapitel wird ein kurzer Überblick über den Stand der Forschung in den Bereichen der der Arbeitsraumüberwachung und der Regelung von stationären und mobilen Manipulatoren im Hinblick auf die Mensch-Roboter-Interaktion gegeben. Zunächst werden Ansätze zur Detektion von Hindernissen basierend auf unterschiedlichen Sensortypen diskutiert. Es folgen grundlegende Regelungsansätze für die kartesische Regelung stationärer und mobiler Manipulatoren sowie Methoden zur Kollisionsvermeidung und Reaktion auf stattfindende Kontakte. Ein weiterer Abschnitt stellt bisherige Anwendungsfelder des Konzepts der modellprädiktiven Regelung bei der Regelung von Robotern vor. Abschließend erfolgt eine Zusammenfassung und Bewertung der vorgestellten Methoden.

2.1 Arbeitsraumüberwachung

Ziel der Arbeitsraumüberwachung ist die Detektion von Hindernissen im Hinblick auf die Kollisionsvermeidung. In den folgenden Abschnitten werden einsetzbare Sensoren und ihre Platzierung sowie bekannte Ansätze für die Arbeitsraumüberwachung diskutiert.

2.1.1 Sensortypen

Die für die Überwachung des Arbeitsraums von stationären oder mobilen Manipulatoren relevanten Sensoren lassen sich entsprechend der ermittelten Information in mehrere Typen unterteilen:

- *Taktile Sensoren* detektieren einen direkten physikalischen Kontakt. Beispiele sind eine künstliche Haut oder Drehmomentsensoren.
- *Anwesenheitssensoren* geben an, ob sich innerhalb eines bestimmten Bereichs ein Objekt befindet. Das Objekt kann entweder durch einen physikalischen Kontakt (z. B. mit einem Bumper oder einer Kontaktmatte) oder ohne physikalischen Kontakt (z. B. mit Lichtschranken oder Lichtvorhängen) detektiert werden.
- *Abstandssensoren* liefern Informationen über den Abstand zwischen dem detektierten Objekt und dem Sensor. Beispiele sind 2D/3D-Laserscanner, Tiefenkameras nach dem Laufzeitprinzip (engl. *time-of-flight*), Tiefenkameras mit einem aktiven Triangulationsverfahren oder Stereokamerasysteme mit einem passiven Triangulationsverfahren.
- *Intensitätskameras* liefern Farb- oder Grauwertinformationen im sichtbaren oder nicht sichtbaren Frequenzbereich, z. B. im Infrarotbereich zur Detektion von Menschen.

Taktile Sensoren finden ihre Anwendung bei der Kontaktdetektion und zur Ermittlung einer geeigneten Reaktion auf stattfindende Kontakte. Anwesenheitssensoren sind bei der Überwachung von Roboterzellen, in denen keine Interaktion mit Menschen erwünscht ist, weit verbreitet. Zur Arbeitsraumüberwachung in Hinblick auf die Kollisionsvermeidung bei erwünschter Mensch-Roboter-Interaktion werden sie nur selten eingesetzt, da die reine Anwesenheitsinformation ohne Abstandsangaben häufig nicht ausreicht, um eine enge Mensch-Roboter-Interaktion zuzulassen. Stattdessen basiert die Arbeitsraumüberwachung in diesem Fall im Allgemeinen auf Intensitätskameras oder Abstandssensoren. Diese Ansätze werden in den Abschnitten 2.1.3 und 2.1.4 näher erläutert.

2.1.2 Sensorplatzierung

Die Platzierung von Intensitätskameras oder Abstandssensoren zur Überwachung des Arbeitsraums des Roboters ist insbesondere bei Anwesenheit von dynamischen Hindernissen wie Menschen ein nicht-triviales Problem.

Der Arbeitsraum stationärer Manipulatoren wird meist mit in der Roboterzelle montierten Sensoren überwacht. Die optimale Abdeckung mit Sensoren in zwei oder drei Dimensionen wurde in [Gon09] und [Bec09] untersucht. Diese Ansätze berücksichtigen jedoch weder dynamische Hindernisse noch die Roboterbewegungen. Heuristische Ansätze schlagen vor, die Sensoren in den oberen Ecken einer Roboterzelle zu platzieren, um eine gute Abdeckung zu erzielen [Ryb12]. In [Fla10] wird ein Algorithmus zur optimalen Platzierung von Anwesenheits- und Abstandssensoren unter Berücksichtigung dynamischer Hindernisse vorgeschlagen.

Im Gegensatz zu stationären Manipulatoren werden mobile Plattformen häufig mit auf der Plattform installierten Sensoren überwacht (z. B. in [Emt14; Hus16]). Dies liegt an dem im Allgemeinen wesentlich größeren Arbeitsraum des Roboters, der eine Vielzahl von externen Sensoren zur Überwachung benötigen würde. Auf Grund der geometrischen Verhältnisse kann der für die Kollisionsvermeidung entscheidende Raum zwischen der Plattform und der Umgebung dagegen häufig mit wenigen, auf der Plattform montierten Sensoren erfasst werden. Des Weiteren ist in diesem Fall der ermittelte Abstand zwischen Roboter und Hindernis unabhängig von etwaigen Lokalisierungsfehlern.

Die Sensorplatzierung zur Überwachung mobiler Manipulatoren wird in der Literatur nur selten diskutiert. Bei kleinen, beschränkten Arbeitsräumen lassen sich dieselben Kriterien wie für stationäre Manipulatoren anwenden (z. B. [Ryb12]). Es ist jedoch zu beobachten, dass viele mobile Manipulatoren mit auf dem Roboter montierten Sensoren ausgestattet sind (z. B. [Boh11; Her13; Hol15]). Dies bietet wie bei mobilen Plattformen den Vorteil, dass sich die mobilen Manipulatoren auch in größeren Arbeitsräumen flexibler einsetzen lassen und der ermittelte Roboter-Hindernis-Abstand nicht von möglichen Lokalisierungsfehlern beeinflusst wird.

2.1.3 Arbeitsraumüberwachung mit Intensitätskamas

Für die Detektion von Hindernissen im Arbeitsraum des Roboters existieren mehrere Ansätze, die auf Intensitätskamas basieren. So werden in [Hen08] Objekte, die sich im Arbeitsraum eines stationären Manipulators befinden, mittels mehrerer Grauwertkamas rekonstruiert. Hierzu wird zunächst der bekannte Hintergrund aus den Bildern entfernt. Größe und Position von Vordergrundobjekten können dann rekonstruiert werden, wenn diese aus verschiedenen Perspektiven von mindestens zwei Kamas erfasst werden.

In [Tan09] wird die mit dem Roboter interagierende Person mit Markern ausgestattet. Diese werden mittels zweier Farbkamas detektiert, sodass die Körperhaltung und die Bewegung des Menschen geschätzt werden kann.

Die auf Intensitätskamas basierenden Methoden sind jedoch im Allgemeinen auf spezielle Anwendungen zugeschnitten wie im Fall von [Tan09] oder setzen voraus, dass wie in [Hen08] der Hintergrund aus den Sensordaten entfernt wird. Somit sind sie für die Anwendung mit Sensoren auf mobilen Robotern nicht geeignet.

2.1.4 Arbeitsraumüberwachung mit Abstandssensoren

Abstandssensoren bieten gegenüber Intensitätskamas den Vorteil, dass sie direkt eine räumliche Information über die detektierten Objekte liefern.

Basierend auf einer einzelnen Tiefenkamera wird in [Fla12] ein Ansatz zur Berechnung des minimalen Abstands zwischen Roboter und Hindernissen unter Berücksichtigung von Verdeckungen vorgeschlagen. Die Methode arbeitet direkt im Tiefenraum des Sensors. So kann eine sehr schnelle Verarbeitung der Sensordaten erreicht werden, die jedoch auf einen Sensor beschränkt ist.

Unter Verwendung von zwei Tiefenkamas wird der minimale Abstand zwischen Roboter und Hindernis in [Wan12] berechnet. Um die Daten in Echtzeit verarbeiten zu können, wird zunächst der bekannte Hintergrund aus den Tiefenbildern entfernt. Die verbleibenden Messungen werden als Punktwolken dargestellt, in das Roboterbasiskoordinatensystem transformiert und zusammengeführt. Der minimale Abstand zwischen Roboter und

der Punktwolke der Vordergrundobjekte wird ohne Berücksichtigung von Verdeckungen berechnet.

In [Boh11] werden die Daten eines rotierenden Laserscanners und einer Stereokamera fusioniert. Die Fusion verwendet eine Voxelrepräsentation, die auf eine Bounding Box um den Roboter beschränkt ist, sodass keine Entfernung des Hintergrunds notwendig ist, um eine ausreichend schnelle Datenverarbeitung zu erzielen.

Die Informationen mehrerer im Raum installierter 3D-Abstandssensoren werden in [Ryb12] mit Hilfe einer Voxeldarstellung fusioniert, wobei zwischen freien, belegten und unbekanntem Voxeln unterschieden wird. Der belegte Raum wird als Hintergrund, Roboter oder Mensch klassifiziert, wobei angenommen wird, dass der Hintergrund a-priori bekannt ist und alle Vordergrundobjekte Menschen sind.

In [Fis09] wird ebenfalls die Information mehrerer im Raum installierter Tiefenkameras fusioniert. Die sich im überwachten Bereich befindenden Objekte werden dabei mittels konvexer Hüllen approximiert, wobei Verdeckungen durch den Roboter und durch die Hindernisse berücksichtigt werden. Der überwachte Bereich ist der Raum, der von allen verfügbaren Sensoren erfasst wird.

Die Darstellung der Roboterumgebung mittels eines Octree wird in [Hor13] vorgeschlagen. Die Zielstellung dieses Ansatzes besteht in der Kartierung großer Umgebungen. Der Ansatz unterscheidet zwischen freiem, belegtem und unbekanntem Raum. Es werden nicht nur die aktuellen Sensordaten, sondern auch zeitlich zurückliegende Informationen berücksichtigt. Auf ähnliche Weise erfolgt die Repräsentation der Umgebung in [Her13] in einem Voxelgitter, wobei zur Verkürzung der Verarbeitungszeit das Voxelgitter auf dem Grafikprozessor (engl. *graphics processing unit*, GPU) berechnet wird.

2.2 Kartesische Regelung

Die Aufgabe eines stationären oder mobilen Manipulators wird häufig im sogenannten kartesischen Raum oder Aufgabenraum definiert: Der Endef-

fektor soll entweder eine bestimmte Pose, bestehend aus 3D-Position und Orientierung, anfahren. Oder er soll einem geometrischen Sollpfad folgen, der beispielsweise durch eine Folge von Posen definiert ist. Oder er soll einer Solltrajektorie der Endeffektorpose folgen, die beispielsweise durch eine Folge zeitabhängiger Sollposen gegeben ist. Die Aufgabe der kartesischen Regelung besteht darin, geeignete Stellgrößen zu berechnen, sodass die im kartesischen Raum gestellte Roboteranfrage erfüllt wird. Mögliche Stellgrößen für Manipulatoren sind Gelenkpositionen, -geschwindigkeiten oder -momente. Diese können von den im Allgemeinen durch den Roboterhersteller bereitgestellten internen Reglern umgesetzt werden. Als Stellgrößen für mobile Plattformen dienen häufig Plattformgeschwindigkeiten, Lenkwinkel oder Lenkwinkelgeschwindigkeiten.

2.2.1 Regelung stationärer Manipulatoren

Bei stationären Manipulatoren ist ein gängiger Ansatz zur Lösung dieser Problemstellung, die Sollpose des Roboters in eine Sollgelenkkonfiguration zu überführen, indem das Problem der inversen Kinematik gelöst wird, und Regelungsstrategien im Gelenkraum anzuwenden [Sic08]. Alternativ werden die Sollgeschwindigkeiten der Endeffektorpose mit Hilfe der (pseudo-)inversen Jacobi-Matrix in Gelenkgeschwindigkeiten umgerechnet [Sic08]. Die ermittelten Gelenkgeschwindigkeiten können entweder direkt auf den Roboter angewendet werden oder der Ansatz wird kombiniert mit gängigen Reglern wie PD- oder PID-Reglern. Im Fall von redundanten Robotern existiert jedoch keine eindeutige Lösung für das Problem der inversen Kinematik. Lösungen aus der (gewichteten) Pseudo-Invertierung der direkten Kinematik sind häufig nicht optimal in Hinblick auf die Aufgabenausführung. Auch werden keine weiteren Aufgaben oder Beschränkungen berücksichtigt, wie z. B. die Vermeidung von Eigen- und Fremdkollisionen, die in Hinblick auf die Mensch-Roboter-Interaktion eine wichtige Rolle spielen.

Das Problem der Redundanz des Roboters kann gelöst werden, indem exakt so viele zusätzliche kinematische Gleichungen definiert werden, wie

redundante Freiheitsgrade vorhanden sind, sodass eine eindeutige Invertierung der Jacobi-Matrix möglich ist. Die zusätzlichen kinematischen Gleichungen können zusätzliche Aufgaben wie Beschränkungen umsetzen. Dieses Vorgehen wird als Methode der zusätzlichen Aufgaben (engl. *additional task method*) bezeichnet [Sic90].

Mehrere simultane Anforderungen an den Roboter können durch die Methode der priorisierten Aufgaben (engl. *priority task control*, *multi-priority control*) berücksichtigt werden [Sic90]. Dabei werden die Anforderungen an den Roboter in einzelne Aufgaben unterteilt. Beispielsweise ist die Bewegung des Endeffektors mit einer gewünschten kartesischen Geschwindigkeit die erste Aufgabe. Für diese Aufgabe wird eine geeignete Stellgröße berechnet, beispielsweise durch Pseudo-Inversion der Jacobi-Matrix. Für die zweite Aufgabe wird eine gewünschte Roboterbewegung im Gelenkraum des Roboters berechnet. Diese Bewegung wird dann in den Nullraum der ersten Aufgabe projiziert. Der Nullraum einer Jacobi-Matrix ist die Menge der Gelenkgeschwindigkeiten, die durch die Jacobi-Matrix auf einen Nullvektor im Ausgangsraum abgebildet werden. Der Nullraum enthält also alle Gelenkgeschwindigkeiten, die keine Geschwindigkeit des Endeffektors (bzw. der Pose, auf die sich die Jacobi-Matrix bezieht) bewirken [Fah09]. So wird die zweite Aufgabe soweit wie möglich erfüllt, ohne die erste Aufgabe zu behindern. Weitere Aufgaben können mit niedrigerer Priorität analog hinzugefügt werden und es lassen sich mehrere Anforderungen an den Roboter gleichzeitig erfüllen.

2.2.2 Regelung mobiler Manipulatoren

Bei der kartesischen Regelung mobiler Manipulatoren kann der Roboter entweder als ein Gesamtsystem oder als zwei getrennte Subsysteme betrachtet werden. Beispiele für die getrennte Regelung von mobiler Plattform und Manipulator sind in [Ngo07; Wan08; Pen14] zu finden. Wird für die mobile Plattform eine separate Regelung verwendet, werden auch eigene Sollgrößen für die Plattformposition oder -geschwindigkeit benötigt. Diese zu bestimmen, wenn nur die Sollpose des Endeffektors bekannt ist, ist kein

einfaches Problem. Häufig werden hierfür speziell auf die konkrete Aufgabenstellung zugeschnittene Lösungen verwendet, die nicht immer optimal für das Verhalten des Gesamtsystems sind.

Diese Schwierigkeit tritt nicht auf, wenn die mobile Plattform und der Manipulator als ein Gesamtsystem geregelt werden. Wird die Kinematik des Gesamtsystems mit einer Jacobi-Matrix modelliert, können die für stationäre Manipulatoren gebräuchlichen Verfahren auch auf den mobilen Manipulator angewendet werden. Dabei lassen sich auch mögliche nicht-holonome Beschränkungen der Plattform in einer erweiterten Jacobi-Matrix abbilden [Sal12].

Die in [Pap00; Sal12] vorgeschlagenen Regelungen beruhen auf der Pseudo-Invertierung der Jacobi-Matrix des Gesamtsystems bestehend aus einer mobilen Plattform und einem Manipulator. Die Regelung in [Sal12] berücksichtigt dabei die Sollbeschleunigung, -geschwindigkeit und -position des Endeffektors in Form eines PD-Reglers mit beschleunigungsabhängigem Vorsteuerterm. In [Pap00] wird der gleiche Regelungsansatz verfolgt, jedoch werden Solltrajektorien sowohl für den Endeffektor als auch für die Plattform vorausgesetzt.

Die Methode der zusätzlichen Aufgaben wird beispielsweise in [Ser98] angewendet. Es werden zusätzliche anwendungsspezifische kinematische Gleichungen definiert, die z. B. Bewegungsbeschränkungen, Ungleichheitsbeschränkungen oder eine Optimierung der Ausführung bewirken, sodass keine redundanten Freiheitsgrade mehr vorhanden sind. Die Regelung basiert auf der nun möglichen Invertierung der Jacobi-Matrix, wobei neben der gewünschten Geschwindigkeit auch die Positionsabweichung von der gewünschten Endeffektorposition berücksichtigt wird.

In [Bay02] werden zusätzliche Roboteraufgaben sowohl zur Erweiterung der Jacobi-Matrix wie in [Ser98] eingesetzt, als auch für die Ausführung sogenannter interner Bewegungen. Die internen Bewegungen stellen Bewegungen der Robotergelenke dar, die die Endeffektorpose nicht verändern. Sie werden mittels Nullraumprojektion ermittelt und entsprechen dem Ansatz der priorisierten Aufgaben.

In [And12a] wird ein kaskadierter Regelungsansatz verfolgt. Die Bewegungsregelung des mobilen Manipulators erfolgt mittels des Ansatzes der priorisierten Aufgaben. Unterlagert wird ein Regler zur Kompensation der Roboterdynamik, der die Aufgabe hat, die Geschwindigkeitsfehler der einzelnen Roboterelkenke zu reduzieren. Als zusätzliche Aufgaben werden die Hindernisvermeidung der mobilen Plattform und die Vermeidung singularärer Roboterstellungen betrachtet.

Auch im Bereich mobiler Manipulatoren existieren Arbeiten, in denen die Antriebsmomente als Stellgröße dienen. Beispielsweise wird in [Omr03] das Konzept der priorisierten Aufgaben umgesetzt, wobei als Stellgrößen des Manipulators die Gelenkmomente dienen und die Plattform geschwindigkeitsgeregelt wird. Voraussetzung hierfür sind ein dynamisches Robotermodell und bekannte Lastdaten.

2.2.3 Regelung mobiler Plattformen

Für die Regelung der Position und Orientierung mobiler Plattformen existieren sehr viele Ansätze. In diesem Abschnitt wird lediglich ein kurzer Überblick über die am meisten verbreiteten Methoden gegeben.

Eine Möglichkeit für die Regelung mobiler Plattformen besteht darin, das zugrundeliegende Systemmodell bzw. das Fehlermodell um den aktuellen Arbeitspunkt zu linearisieren. Die Regelung erfolgt dann mittels Zustandsrückführungen [DeL98; Kla05] oder mittels einer linearen modellprädiktiven Regelung [Kan09].

Unter den nichtlinearen Reglern ist der Ansatz des *Sliding Mode Control* weit verbreitet [Utk91; Xue17]. Der Regelung kann dabei sowohl ein kinematisches Robotermodell als auch ein dynamisches Robotermodell zugrunde liegen. Der Vorteil des *Sliding Mode Control* liegt in der Robustheit gegenüber Parameterunsicherheiten. Ein Nachteil besteht darin, dass Beschränkungen der Stellgrößen nicht direkt berücksichtigt werden können.

Ein Ansatz, der dieses Problem löst, ist die nichtlineare modellprädiktive Regelung (engl. *Nonlinear Model Predictive Control*, NMPC). Arbeiten

zur Regelung mobiler Plattformen mittels NMPC werden in Abschnitt 2.5 vorgestellt.

2.3 Kollisionsvermeidung

In Hinblick auf die Mensch-Roboter-Interaktion ist eine wichtige Fähigkeit des Roboters, dass er in der Lage ist, unerwünschte Kollisionen mit Menschen oder anderen Objekten in seiner Umgebung zu vermeiden. Deshalb existieren zahlreichen Arbeiten, die sich mit Regelungsstrategien zur Kollisionsvermeidung stationärer oder mobiler Manipulatoren befassen.

Eine Kollisionsvermeidungsstrategie für geteilte Arbeitsräume von Mensch und Roboter wird in [Wan12] vorgeschlagen. Dabei wird der Abstand zwischen einem Manipulator und Menschen in seiner Umgebung überwacht. Mit kleiner werdendem Abstand wird die Geschwindigkeit des Roboters verlangsamt und der Roboter schließlich angehalten, wobei die Ausführung der Roboteraufgabe unterbrochen wird. Nimmt der Abstand weiter ab, bewegt der Roboter den Endeffektor vom Menschen weg.

Ein verbreiteter Ansatz zur Kollisionsvermeidung ohne Unterbrechung der Aufgabenausführung ist die Methode der künstlichen Potentialfelder (engl. *artificial potential field method*), die 1985 von Khatib eingeführt wurde [Kha85]. Die Roboteraufgabe wird durch anziehende Kräfte beschrieben, die auf den Endeffektor wirken. Hindernisse werden in abstoßende Kräfte überführt, die den Roboter von den Hindernissen abstoßen. Diese abstoßenden Kräfte werden in Gelenkmomente konvertiert, die in den Nullraum des Roboters projiziert werden. Auf mobile Manipulatoren wird die Potentialfeldmethode in [Tan03] und [Wu13] angewendet. In [Tan03] kann damit die Vermeidung von Kollisionen mit statischen Hindernissen unter Berücksichtigung der nicht punktförmigen Geometrie des Roboters erreicht werden. Dynamische Hindernisse werden in [Wu13] berücksichtigt. Da die Potentialfeldmethode generell sehr rechenaufwändig ist [Kit96], wird die Berücksichtigung der dynamischen Hindernisse auf einen Bereich um den Endeffektor begrenzt.

Ähnlich zur Potentialfeldmethode wird in [Pol14] eine abstoßende Bewegung für stationäre Manipulatoren basierend auf sogenannten *Kinetostatic Safety Fields* berechnet. Diese Sicherheitsfelder fusionieren Informationen über Position, Geschwindigkeit, Form und Größe von Objekten mit denen der Roboter möglicherweise kollidieren könnte. Um das Sicherheitsfeld zu generieren, wird ein Modell eines Menschen bestehend aus mehreren Quadern in die Hindernismessungen einer Tiefenkamera gefittet. Kollisionsfreie Bewegungen werden durch Nullraumprojektion bestimmt.

In [Fla12] wird ein Regelungsalgorithmus zur Kollisionsvermeidung stationärer Manipulatoren vorgeschlagen, der zwischen möglichen Kollisionen des Endeffektors und des Roboterarms unterscheidet. Kollisionen des Endeffektors werden ähnlich zur Potentialfeldmethode vermieden. Um Kollisionen des Roboterarms zu vermeiden, werden die zulässigen Gelenkgeschwindigkeiten abhängig vom minimalen Abstand zwischen Punkten auf dem Roboter und den Hindernissen beschränkt.

In [Zha12] wird eine Methode zur Kollisionsvermeidung für einen Manipulator, der auf einer teleoperierten mobilen Basis installiert ist, vorgestellt. Um mögliche Kollisionen zu detektieren, werden die Hindernismessungen und Punkte des Manipulators auf eine Ebene vor dem Roboter projiziert. Es wird dabei angenommen, dass die mobile Basis auf einem kollisionsfreien Pfad bewegt wird. Zur Kollisionsvermeidung wird unterschieden, ob eine Kollision des Endeffektors oder des Ellbogens verhindert werden muss. Droht eine Kollision des Endeffektors, wird dieser mittels der pseudo-inversen Jacobi-Matrix in eine kollisionsfreie Position bewegt. Droht eine Kollision mit dem Ellbogen wird mittels Nullraumprojektion eine Ausweichbewegung für den Endeffektor berechnet.

2.4 Kontaktregelung

Die Kontaktregelung von Manipulatoren erfolgt im Allgemeinen entweder basierend auf den durch den Kontakt hervorgerufenen externen Drehmomenten der einzelnen Gelenke oder basierend auf der geschätzten Kontaktkraft. Im Folgenden werden zunächst Methoden zur Schätzung der externen

Momente und der Kontaktkraft vorgestellt. Anschließend werden existierende Ansätze zur Kontaktregelung diskutiert.

2.4.1 Kraft- und Momentenschätzung

Externe Drehmomente und Kontaktkräfte können prinzipiell mit einem Kraft-Momenten-Sensor gemessen werden. Im Allgemeinen ist es jedoch nur möglich den Endeffektor mit einem derartigen Sensor auszustatten, d. h. es können keine Kontakte am Roboterarm oder der Roboterplattform erfasst werden.

Für die Schätzung der externen Drehmomente aller Manipulatorgelenke ist das in [DeL06] vorgeschlagene residuenbasierte Verfahren weit verbreitet. Es setzt voraus, dass die Gelenkmomente gemessen werden. Diese setzen sich wie in Abschnitt 4.3 dargestellt aus den Motormomenten und den externen Momenten zusammen. Üblicherweise stellen Leichtbauarme, die typischerweise zur Mensch-Roboter-Interaktion eingesetzt werden, Momentenmessungen zur Verfügung. Des Weiteren benötigt die Methode ein exaktes dynamisches Robotermodell. Dann kann aus dem Vergleich der erwarteten und gemessenen Drehmomente auf die externen Momente geschlossen werden.

Eine Möglichkeit zur Messung der Kontaktkraft ist die Anbringung einer künstlichen Haut auf dem Roboter (z. B. [Cir12]). Die künstliche Haut misst direkt die Amplitude und Richtung der einwirkenden Kontaktkraft. Diese Methode hat den Vorteil, dass sie kein Wissen über das dynamische Robotermodell oder die Messung der Gelenkmomente benötigt und auch auf mobile Plattformen anwendbar ist. Der Nachteil besteht darin, dass der gesamte Roboter mit der künstlichen Haut abgedeckt werden muss.

Alternative Ansätze zur Kontaktkraftschätzung beruhen auf den externen Drehmomenten der einzelnen Robotergelenke. In [Mag14] wird der Kontaktpunkt auf dem Roboter mit Hilfe einer Tiefenkamera ermittelt. Ist der Kontaktpunkt bekannt, kann die Kontaktkraft aus den externen Drehmomenten mittels Pseudo-Inversion der Jacobi-Matrix berechnet werden (siehe Abschnitt 4.3). In [Lik14] wird eine Methode vorgestellt, die keine weiteren

externen Sensoren benötigt. Diese Methode nimmt an, dass nur ein Kontakt auf einmal stattfindet und dass die Kraft senkrecht auf das Roboterglied wirkt, das durch einen Zylinder approximiert wird. Mit Hilfe des kinematischen Robotermodells (bzw. der Jacobi-Matrix) können aus einer Kontaktkraft und einem Kontaktpunkt die resultierenden externen Momente berechnet werden. Das Verfahren schätzt die Kontaktkraft und die Kontaktposition durch Minimierung der Differenz zwischen den basierend auf dem Robotermodell in Abhängigkeit der geschätzten Kraft und Position berechneten Momenten und den tatsächlichen externen Momenten.

2.4.2 Regelungsstrategien

Für die Reaktion auf stattfindende Kontakte existieren verschiedene Ansätze. In [DeL06] und [Had08] werden die durch den Kontakt entstehenden externen Momente geschätzt und verschiedene Reaktionsstrategien verglichen: Anhalten des Roboters, die Bewegung des Roboters entsprechend der externen Momente, wobei lediglich der Einfluss der Gravitation kompensiert wird, und eine verstärkte Bewegung des Roboters in Richtung der einwirkenden Momente, sodass eine Überreaktion hervorgerufen wird. Bei der letzten Strategie werden für alle Gelenke Sollmomente (Impedanzregelung) oder Sollgeschwindigkeiten (Admittanzregelung) berechnet, indem ein Verstärkungsfaktor auf die externen Momente angewendet wird. Mit allen Strategien muss der Roboter die gewünschte Bewegung des Endeffektors aufgeben, sobald ein Kontakt auftritt.

In [DeL08] wird die zuvor beschriebene Impedanzregelung erweitert, indem die auf Grund des Kontakts berechneten Sollmomente in den Nullraum der Endeffektoraufgabe projiziert werden. So muss die ursprüngliche Aufgabe des Roboters nicht unbedingt unterbrochen werden, wenn ein Kontakt auftritt. Ist es jedoch nicht möglich sowohl die gewünschte Reaktion auf den stattfindenden Kontakt als auch die ursprüngliche Aufgabe auszuführen, hat die Kontaktreaktion eine höhere Priorität. Ein ähnlicher Ansatz ist der in [Sad12] vorgeschlagene Nullraum-Impedanzregler mit Störgrößen-

beobachter. Dieser erlaubt ebenfalls ein nachgiebiges Verhalten während gleichzeitig der Trajektorienverfolgungsfehler minimiert wird.

Speziell für Kontakte am Endeffektor wird in [Pla11] ein Impedanzregler mit mehreren Aufgaben vorgeschlagen. Die erste Aufgabe berücksichtigt die kartesische Impedanz des Endeffektors für nachgiebige Reaktionen auf Kontakte. Die zweite Aufgabe betrachtet den Gelenkraum. Sie projiziert die Impedanz des Gelenkraums in den Nullraum der kartesischen Impedanz des Endeffektors, um die Redundanz des Roboters während des Kontakts lösen zu können.

Anders als in den oben genannten Regelungsstrategien, die alle ein dynamisches Robotermodell benötigen, wird in [Li11] eine passive, modellfreie Impedanzregelung vorgeschlagen. Dabei muss die ursprüngliche Roboter-aufgabe jedoch aufgegeben werden, wenn ein Kontakt auftritt.

Basierend auf der geschätzten Kontaktkraft und der Kontaktposition anstelle der externen Momente wird in [Mag14] eine Admittanzregelung des Kontaktpunkts vorgeschlagen. Aus der geschätzten Kontaktkraft wird die gewünschte Bewegung des Kontaktpunkts ermittelt. Diese Bewegung wird mittels der pseudo-inversen Jacobi-Matrix in Gelenkgeschwindigkeiten des Roboters transformiert. Wird ein zweiter Kontakt detektiert, wird die daraus resultierende Bewegung in den Nullraum des ersten Kontakts projiziert, sodass die gewünschte Bewegung auf Grund des ersten Kontakts nicht behindert wird (Methode der priorisierten Aufgaben). Für diesen Ansatz wird kein dynamisches Robotermodell benötigt.

2.5 Modellprädiktive Regelung

Das Prinzip der nichtlinearen modellprädiktiven Regelung (engl. *Nonlinear Model Predictive Control*, NMPC) besteht darin, basierend auf dem Systemmodell optimale Stellgrößen für einen bestimmten Zeihorizont zu ermitteln. Diese optimalen Stellgrößen werden für einen Abtastschritt auf das System angewendet. Unter Berücksichtigung aktueller Messwerte werden dann neue optimale Stellgrößen ermittelt und so die Reglerschleife geschlossen. Weitere Details zur NMPC sind in Kapitel 5 zu finden.

In der Robotik findet die NMPC Anwendung zur Lösung des Trajektorienverfolgungsproblems mobiler Plattformen [Kla07; Gu09; Mau11]. Üblicherweise werden dabei basierend auf einem kinematischen Robotermodell die zweidimensionale Plattformposition sowie der Richtungswinkel geregelt. Die Solltrajektorie der Plattformposition wird als gegeben angenommen. Neben der Lösung des Trajektorienverfolgungsproblems dient die NMPC auch der Kollisionsvermeidung [Pri17].

In Hinblick auf stationäre Manipulatoren wird die NMPC häufig zur Regelung der Gelenkpositionen oder -geschwindigkeiten eingesetzt [Poi00; Viv05; Hed05; Duc08; Mer08; Bdi12]. Dabei werden basierend auf dem dynamischen Robotermodell optimale Drehmomente berechnet, sodass der Manipulator einer Solltrajektorie im Gelenkraum folgt. In einigen Fällen wird die Solltrajektorie im Gelenkraum mittels der inversen Kinematik aus einer Solltrajektorie im kartesischen Raum gewonnen. Andernfalls wird die Solltrajektorie im Gelenkraum als bekannt vorausgesetzt.

In [Fau13] wird die NMPC im Gelenkraum des Roboters dahingehend erweitert, dass das Pfadverfolgungsproblem für gegebene Pfade der Gelenkposition ohne Zeitabhängigkeit gelöst werden kann. Hierfür wird die Geschwindigkeit entlang des Pfads zusammen mit den Drehmomenten der Gelenke optimiert. Die Methode wird auf eine zweigliedrige planare Armkonfiguration angewendet.

In [Cas09] ist statt einer Solltrajektorie oder eines Sollpfads eine einzelne Zielkonfiguration im Gelenkraum gegeben. Der Manipulator wird mit dem Ziel geregelt, diese Gelenkkonfiguration zu erreichen und dabei Kollisionen mit anderen Robotern zu vermeiden. Die Robotergeometrien werden mit *Line-Swept Sphere Volumes* modelliert. Die Kollisionsvermeidung wird durch zusätzliche Beschränkungen des Optimierungsproblems realisiert.

Für die Regelung von mobilen oder stationären Manipulatoren im kartesischen Raum existieren nur sehr wenige modellprädiktive Ansätze. In [Ide11] wird die Endeffektorposition eines mobilen Manipulators mittels NMPC geregelt um eine gegebene Zielposition zu erreichen. Bei dem betrachteten mobilen Manipulator handelt es sich um eine omnidirektionale mobile Plattform mit zwei planaren Robotergliedern. In [Moh01] wird

ebenfalls die Endeffektorposition eines mobilen Manipulators mit zwei planaren Gliedern geregelt, wobei hier zusätzlich Hindernisse berücksichtigt werden. Die Kollisionsvermeidung wird durch zusätzliche Beschränkungen des Optimierungsproblems erreicht, die den Abstand zwischen sphärischen Hindernissen und den Mittelpunkten der Plattform bzw. der Roboterglieder betrachten.

2.6 Zusammenfassung und Bewertung

Aktuelle Methoden der Arbeitsraumüberwachung zur Detektion von Hindernissen basieren auf Abstandssensoren, die je nach betrachteter Anwendung im Raum oder auf dem Roboter installiert sind. Eine direkte Verarbeitung der Sensordaten zur Abstandsermittlung zwischen Roboter und Hindernis erfolgt nur für einen einzelnen Sensor oder in Kombination mit Hintergrundentfernung. Hintergrundentfernung eignet sich jedoch nur für stationäre Sensoren. Stattdessen werden die Sensordaten mit Voxeligittern, Octrees oder konvexen Hüllen repräsentiert, um den benötigten Rechenaufwand zu reduzieren. Einige Ansätze berücksichtigen zusätzlich auch durch Hindernisse verdeckte Bereiche. Die Methoden sind jedoch meist auf stationäre Sensoren ausgelegt und häufig werden nur kleine Arbeitsräume betrachtet. Damit erfüllen diese Methoden nicht alle Anforderungen an die Ermittlung einer 3D-Hindernisdarstellung für mobile Manipulatoren.

Für die kartesische Regelung stationärer und mobiler Manipulatoren sind neben anwendungsspezifischen Ansätzen Methoden, die auf der Pseudoinversion der Jacobi-Matrix und der Nullraumprojektion basieren, am weitesten verbreitet. Eine grundsätzliche Schwierigkeit dieses Verfahrens ist der Umgang mit Singularitäten der Jacobi-Matrix. Für die Kollisionsvermeidung werden meist entweder die Potentialfeldmethode und Varianten davon eingesetzt, die in der Nähe von Hindernissen ein abstoßendes Verhalten erzeugen, aber keine harten Beschränkungen berücksichtigen. Alternativ werden Methoden eingesetzt, die rein auf Beschränkungen zur Kollisionsvermeidung basieren, jedoch keine weiteren Anforderungen an die Kollisionsvermeidung ermöglichen, wie z. B. einen möglichst großen

Abstand zum Hindernis einzuhalten. Zur Kontaktregelung werden häufig Impedanzregelungen selten auch Admittanzregelungen mit Nullraumprojektion eingesetzt.

Bisherige Arbeiten berücksichtigen die Orientierung des Endeffektors meistens nicht und vermeiden so Fragestellungen der mathematischen Beschreibung der Orientierung. Insbesondere bei mobilen Manipulatoren beruhen die Methoden oft auf roboterspezifischen Modellen. Meist betrachten die Arbeiten die eigentlich notwendige Vermeidung von Eigenkollisionen nicht und berücksichtigen auch keine weiteren Randbedingungen wie Gelenkbeschränkungen oder kartesische Beschränkungen des Arbeitsraums. Viele Ansätze zur Kollisionsvermeidung beschränken sich auf Kollisionen mit dem Endeffektor oder mit vorab bekannten Hindernissen. Die Arbeiten untersuchen keine gleichzeitige Anwendung von Kollisionsvermeidung und Kontaktreaktion.

Die Möglichkeit der Übertragung des Ansatzes der NMPC auf die Regelung der Endeffektorpose stationärer und mobiler Manipulatoren wurde bisher kaum untersucht. Es existieren lediglich einfache Ansätze zur Regelung der 2D-Position (ohne Orientierung) einer mobilen Plattform mit zwei planaren Gliedern ohne oder mit eingeschränkter Kollisionsvermeidung, jedoch keine Regelung der Endeffektorpose bestehend aus 3D-Position und Orientierung.

Kapitel 3

Arbeitsraumüberwachung

Voraussetzung für eine sichere und intuitive Mensch-Roboter-Kooperation in einer schwach strukturierten Umgebung ist die Erfassung des Roboterumfelds mittels Sensoren, um Kollisionen mit Menschen oder anderen Hindernissen verhindern zu können. In diesem Kapitel wird eine Methode zur 3D-Erfassung der aktuellen Hindernisse im nahen Roboterumfeld vorgestellt.

3.1 Anforderungen

Das Ziel der Arbeitsraumüberwachung ist die Erfassung von Hindernissen in der Umgebung des Roboters. Als Hindernisse werden dabei alle Menschen und Objekte aufgefasst, mit denen der Roboter nicht kollidieren darf. Diese Hindernisse können z. B. interagierende oder unbeteiligte Personen sein. Aber auch Tische, Wände oder auf einem Tisch liegende Gegenstände sind in vielen Szenarien auftretende Hindernisse. Die Hindernisse können sich sowohl statisch als auch dynamisch verhalten.

Um Kollisionen mit dynamischen Hindernissen vermeiden zu können, ist es von Vorteil, mögliche Gefahrensituationen frühzeitig zu erkennen und vorausschauend darauf zu reagieren. Hierfür muss ein möglichst großer Bereich um den Roboter sensoruell erfasst werden, um die Bewegung dyna-

mischer Hindernisse präzisieren zu können. Jedoch können nicht alle Hindernisbewegungen zuverlässig präzisiert werden. So sind beispielsweise Armbewegungen interagierender Personen kaum präzifizierbar, da sie sehr hohe Beschleunigungen und schnelle Richtungswechsel aufweisen können. Deshalb müssen gleichzeitig zur Gewährleistung der Sicherheit plötzlich auftretende Gefahrensituationen schnell erkannt werden. Hierfür ist im Fall von stationären und mobilen Manipulatoren eine 3D-Darstellung der aktuellen statischen und dynamischen Hindernisse in der nahen Roboterumgebung notwendig. Mit einer derartigen Darstellung der Hindernisse können unmittelbar bevorstehende Kollisionen erkannt werden, sodass eine schnelle Reaktion möglich ist. Diese Arbeit konzentriert sich auf die dreidimensionale Erfassung der aktuellen Hindernislage, die für die direkte Mensch-Roboter-Interaktion unerlässlich ist. Für die Prädiktion dynamischer Hindernisse für die Mensch-Roboter-Interaktion sei auf [Fre14; Fet13] verwiesen.

Für die Überwachung der Roboterumgebung in Hinblick auf die Kollisionsvermeidung kommen unterschiedliche Sensortypen in Frage (siehe Abschnitt 2.1.1). Während Intensitätskameras und Anwesenheitssensoren lediglich auf die Bildebene projizierte Informationen bereitstellen, liefern Abstandssensoren direkt eine räumliche Information über die Roboterumgebung. Für die Arbeitsraumüberwachung zur Kollisionsvermeidung eignen sich deshalb Abstandssensoren am Besten, weshalb die Ermittlung der 3D-Hindernisdarstellung auf Abstandssensordaten basieren soll.

Grundsätzlich können die Abstandssensoren sowohl auf dem Roboter als auch in der Arbeitszelle des Roboters angebracht werden. Die Auswahl und Platzierung der Sensoren ist eine nicht triviale Aufgabe, da die Anzahl der einsetzbaren Sensoren durch die Anschaffungskosten, ihren Energieverbrauch (besonders auf mobilen Robotern) und die benötigte Rechenkapazität begrenzt ist. Gleichzeitig soll eine möglichst gute Abdeckung des Arbeitsraums erzielt werden. Werden die Sensoren in der Arbeitszelle des Roboters installiert, kann der Hauptarbeitsbereich des Roboters durch mehrere Sensoren mit unterschiedlichen Blickwinkeln abgedeckt werden, um so das Risiko für Verdeckungen zu reduzieren. Dieser Ansatz ist für stationäre Roboterarme und mobile Roboter in kleinen, begrenzten Arbeitszellen

geeignet. Um große Arbeitsräume mobiler Roboter vollständig abzudecken, erfordert dieser Ansatz jedoch eine große Anzahl an Sensoren mit großem Gesichtsfeld und großer Reichweite. Des Weiteren stellt dieser Ansatz hohe Anforderungen an die Genauigkeit der Roboterlokalisierung, da der Roboter, wie in Abschnitt 3.3.3 dargestellt, aus den Sensordaten entfernt werden muss. Werden die Sensoren auf dem Roboter angebracht, sind die Platzierungsmöglichkeiten der Sensoren deutlich eingeschränkter. Vorteile sind jedoch, dass der zu überwachende Bereich auf einen kleineren Bereich um den Roboter beschränkt werden kann und die Sensorplatzierung unabhängig von der Größe der Arbeitszelle erfolgen kann. Je nach Anwendungsfall überwiegen die Vorteile stationärer Sensoren in der Arbeitszelle oder mobiler Sensoren auf dem Roboter. Deshalb sollen die Methoden der Arbeitsraumüberwachung sowohl auf Sensoren in der Arbeitszelle als auch auf Sensoren auf dem Roboter angewendet werden können.

Aus Sicherheitsgründen darf die zu ermittelnde 3D-Darstellung der Hindernisse nicht nur die für die Sensoren sichtbaren Teile der Hindernisoberfläche beinhalten. Stattdessen müssen die durch Hindernisse oder den Roboter verdeckten Bereiche ebenfalls berücksichtigt werden. Dies ermöglicht eine konservative Abschätzung der Hindernisgröße. Insbesondere wenn sich ein Hindernis zwischen dem Roboter und dem Sensor befindet (siehe Abbildung 3.1) ist dies notwendig, um den Abstand zwischen Roboter und Hindernis und damit das Kollisionsrisiko nicht zu unterschätzen.

Um Verdeckungen durch den Roboter und durch die Hindernisse zu reduzieren und den überwachten Bereich zu vergrößern, muss die Information mehrerer Sensoren fusioniert werden. Es kann sich hierbei um heterogene Sensoren handeln, die beispielsweise 2D- und 3D-Informationen liefern. Die Verarbeitung der Sensordaten muss in Echtzeit erfolgen, um die schnelle Detektion von Gefahrensituationen zu gewährleisten.

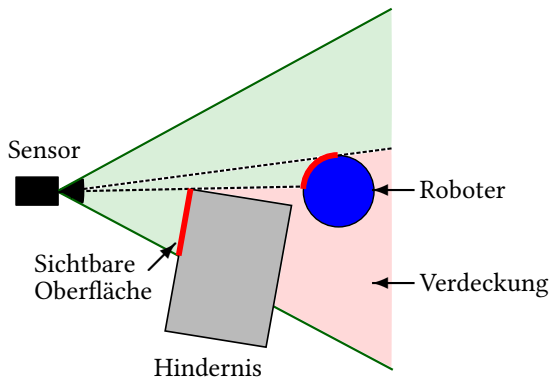


Abbildung 3.1: Prinzip der Verdeckungen.

3.2 Unterscheidung zwischen Nah- und Weitbereich

Wie in Abschnitt 3.1 dargestellt, muss die Arbeitsraumüberwachung Echtzeitanforderungen erfüllen, um schnelle Reaktionen auf plötzliche Gefahren zu ermöglichen. Gleichzeitig ist die Verarbeitung der 3D-Informationen mehrerer Abstandssensoren im Allgemeinen sehr rechenaufwändig. Um dennoch die Anforderungen bezüglich der Rechenzeit erfüllen zu können, wird der Umstand ausgenutzt, dass die 3D-Hindernisdarstellung auf die nahe Roboterumgebung und kurze Zeithorizonte, den sogenannten Nahbereich, beschränkt werden kann. Im weiteren Roboterumfeld, dem sogenannten Weitbereich, können gleichzeitig weniger rechenaufwändige $2\frac{1}{2}D$ -Methoden eingesetzt werden [Fre14].

Der Nahbereich des Roboters wird hier definiert als der Bereich, den der Roboter basierend auf den maximalen Gelenkgeschwindigkeiten und dem aktuellen Roboterzustand innerhalb eines gegebenen Zeithorizonts mit mindestens einem Roboterglied erreichen kann, und wird durch die im Folgenden dargestellte Erreichbarkeitsanalyse bestimmt.

Zur Ermittlung des erreichbaren Raums eines Manipulators wird in [And12b] die Berechnung eines Erreichbarkeitsgitters (engl. *reachability*

grid) vorgeschlagen. Die Geometrie eines jeden Manipulatorglieds wird durch eine Punktwolke repräsentiert. Beginnend mit dem letzten Glied (dem Endeffektorglied) wird die das Glied beschreibende Punktwolke um das dazugehörige Gelenk rotiert. Dabei wird das innerhalb des gegebenen Zeithorizonts maximal mögliche Intervall des Gelenkwinkels überstrichen. Dieses wird neben dem Zeithorizont durch den aktuellen Gelenkwinkel, die minimalen und maximalen Gelenkwinkel sowie die maximale Gelenkgeschwindigkeit bestimmt. Für jeden rotierten Punkt (d. h. für alle erreichbaren Punkte) wird die minimale Zeit berechnet, die benötigt wird, um diesen Punkt mit dem Roboter zu erreichen. Die resultierende Punktwolke wird reduziert, indem sie auf ein Voxelgitter abgebildet wird. Die reduzierte Punktwolke wird der Punktwolke des nächsten Roboterglieds hinzugefügt und die vorherigen Schritte werden wiederholt. Auf Grund der Reduzierung der Punktwolke durch Abbildung auf ein Voxelgitter nach jedem Rotations-schritt erreicht die Methode ausreichend kurze Rechenzeiten, auch für Manipulatoren mit vielen Freiheitsgraden.

Dieser Ansatz lässt sich leicht auf einen mobilen Manipulator mit omnidirektionaler Plattform übertragen. Die omnidirektionale Plattform wird als ein Roboterglied mit einem rotatorischen Gelenk (Drehung um die Hochachse) und zwei translatorischen Gelenken (Bewegung vorwärts/rückwärts und nach rechts/links) modelliert. Die Berechnung des Erreichbarkeitsgitters erfolgt wie im Fall des Manipulators, wobei zusätzlich die maximalen Plattformgeschwindigkeiten zu berücksichtigen sind. Die Gelenkwinkel und -positionen der Plattform sind unbegrenzt.

Zwei Beispiele der Erreichbarkeitsanalyse für den in Abschnitt 8.1 beschriebenen Roboter sind in Abbildung 3.2 zu sehen. Es wird dabei ein Zeithorizont von 0,5 s betrachtet und es werden die vom Hersteller angegebenen maximalen Geschwindigkeiten berücksichtigt.

Bei der Erreichbarkeitsanalyse werden die Beschleunigungsbeschränkungen der Robotergelenke vernachlässigt. Der Fehler auf Grund der Annahme von unendlich hohen Beschleunigungen führt lediglich zur Überschätzung des erreichbaren Raums. Dies ist für die Definition des Nahbereichs vertretbar. Da die Methode zur Erreichbarkeitsanalyse alle Gelenke

unabhängig voneinander betrachtet, ist es nicht möglich, ungültige Roboterkonfigurationen (z. B. auf Grund von Eigenkollisionen) auszuschließen oder nicht-holonome Eigenschaften einer mobilen Plattform exakt widerzuspiegeln. Wird für eine nicht-holonome Plattform das Bewegungsmodell einer omnidirektionalen Plattform mit beschränkten Geschwindigkeiten zugrunde gelegt, führt dies jedoch wie die anderen Vereinfachungen lediglich zu einer Überschätzung des erreichbaren Raums.

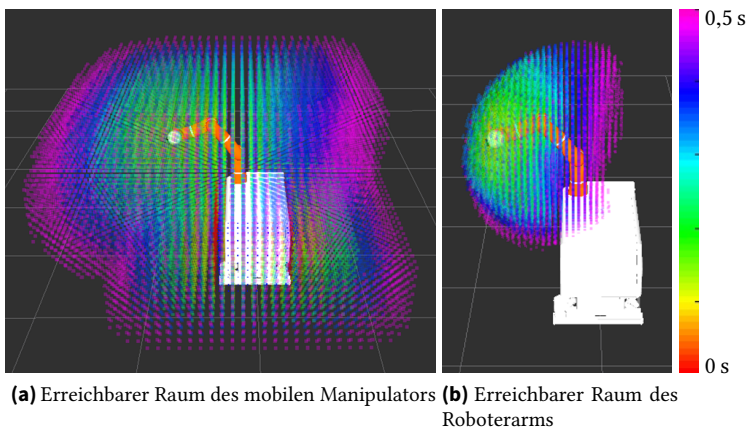


Abbildung 3.2: Visualisierung des erreichbaren Raums innerhalb eines Zeithorizonts von 0,5 s (aus [Fet14]). Farbliche Codierung der Mindestzeit bis zum Erreichen des jeweiligen Punktes.

3.3 Erzeugung einer 3D-Hindernisdarstellung mit Verdeckungsanalyse

Basierend auf den Daten mehrerer Abstandssensoren wird eine 3D-Repräsentation der Hindernisse im Nahbereich des Roboters ermittelt, welche die in Abschnitt 3.1 dargestellten Anforderungen erfüllt. Hierfür wird zunächst für jeden Sensor der freie Raum sowie der durch Hindernisse bzw. den Roboter belegte oder verdeckte Raum bestimmt. Diese Verdeckungsana-

lyse und die dafür notwendigen Vorverarbeitungsschritte werden für jeden Sensor unabhängig in parallelen Prozessen durchgeführt. Dann werden die Informationen der einzelnen Sensoren fusioniert. Die resultierende 3D-Hindernisdarstellung beinhaltet alle durch Hindernisse belegten oder verdeckten Bereiche im Nahbereich des Roboters, die sich im Gesichtsfeld der Sensoren befinden. Die Repräsentation des Raums erfolgt mit Hilfe einer Octree-Datenstruktur. Der grundlegende Ablauf ist in Abbildung 3.3 zu sehen, die einzelnen Schritte werden in den folgenden Abschnitten detailliert erläutert.

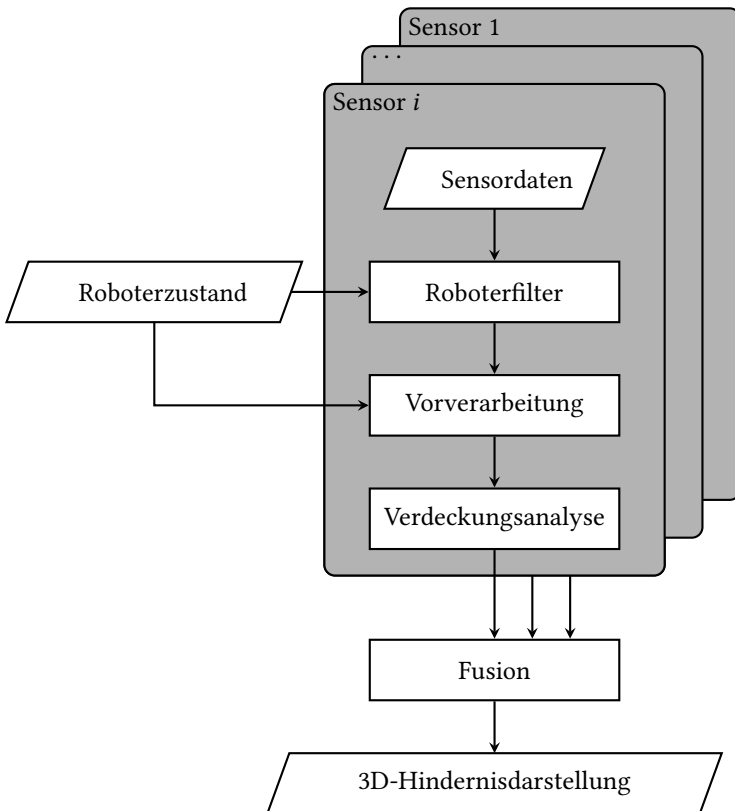


Abbildung 3.3: Ablaufdiagramm der Arbeitsraumüberwachung.

3.3.1 Octree-Datenstruktur

Ein Octree ist ein gerichteter Baum, dessen Knoten mit Ausnahme der Blätter jeweils acht Nachfolger besitzen. Zur Repräsentation des dreidimensionalen Raums mit einem Octree wird der Raum in Voxel unterteilt. Jedes Voxel wird rekursiv in acht Subvoxel unterteilt, bis die minimale Voxelgröße erreicht ist. Jedes Voxel entspricht einem Octree-Knoten [Jac80; Hor13; Sic08].

Octrees ermöglichen eine schnelle Kollisionsprüfung und effiziente Strahlverfolgungsmethoden. Deshalb eignen sie sich für die 3D-Repräsentation von Hindernissen im Roboternahbereich.

Für die Arbeitsraumüberwachung wird der Octree \mathcal{B} im Basiskoordinatensystem des Roboters spezifiziert. Der Octree hat eine begrenzte Ausdehnung, wobei die Begrenzung so gewählt wird, dass sie den Nahbereich des Roboters immer umfasst.

3.3.2 Sensormodell

Die Abstandssensoren werden unabhängig vom zugrunde liegenden Messprinzip mit Hilfe eines allgemeinen strahlenbasiertes Sensormodells modelliert (siehe Abbildung 3.4). Das Sensormodell des Sensors i besteht aus einer Menge von Strahlen, die im Sensorursprung $\mathbf{s}_{u,i}$ beginnen. Die Anzahl $N_{S,i}$ und Richtung der Strahlen ergibt sich aus dem Öffnungswinkel und der Auflösung des Sensors. Die Länge der Strahlen ist durch die maximale Reichweite des Sensors $d_{S,\max,i}$ gegeben. Damit gilt für die Strahlen j ($j = 1, \dots, N_{S,i}$) des Sensors i

$$S_{i,j} = \{ \mathbf{s}_{u,i} + s \mathbf{r}_{i,j} \mid s \in [0, d_{S,\max,i}] \} , \quad (3.1)$$

wobei $\mathbf{r}_{i,j}$ der normierte Richtungsvektor des Strahls j ist. Der Sensor misst die Abstände zwischen dem Sensorursprung und den Punkten, an denen die Strahlen das erste Objekt treffen. Diese Punkte werden im Folgenden als Objektpunkte bezeichnet.

Mathematisch korrekt handelt es sich aufgrund der begrenzten Sensorreichweite um ein Streckenmodell des Sensors. Da jedoch die Bezeichnung

Sensorstrahl statt Sensorstrecke üblich ist, wird die Bezeichnung Strahl in diesem Kontext beibehalten.

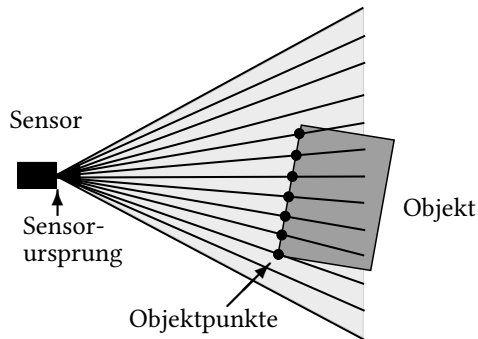


Abbildung 3.4: Sensormodell.

Es wird vorausgesetzt, dass die Sensoreigenschaften (Öffnungswinkel, maximale Reichweite, Auflösung) bekannt sind. Des Weiteren wird angenommen, dass die Transformation zwischen dem Sensorkoordinatensystem und dem Basiskoordinatensystem des Roboters, das auch das Bezugskordinatensystem für die 3D-Hindernisdarstellung ist, bekannt ist. Im Fall einer statischen Transformation (z. B. bei stationärem Roboter und stationären Sensoren oder Sensoren auf einer mobilen Roboterplattform) erfordert dies eine einmalige extrinsische Kalibrierung. Im Fall eines mobilen Roboters und stationärer Sensoren in der Arbeitszelle ist zusätzlich die Lokalisierung des Roboters notwendig.

3.3.3 Roboterfilter

Befindet sich der Roboter im Gesichtsfeld des Sensors, können die vom Sensor erfassten Objektpunkte sowohl Punkte auf Hindernissen in der Roboterumgebung als auch Punkte auf dem Roboter selbst sein. Zur Klassifizierung dieser Hindernis- und Roboterpunkte wird ein Filter eingesetzt, das die gemessenen Sensordaten mit einem geometrischen Modell des Roboters vergleicht. Das Ergebnis sind zwei Sensordatensätze, die jeweils nur die

Hindernis- bzw. die Roboterpunkte enthalten. Es existieren mehrere Ansätze für derartige Filter, beispielsweise das *Realtime URDF Filter* [Blo17] für Tiefenbilder oder das *Robot Self Filter* von *MoveIt!* [Suc17] für Punktwolken.

3.3.4 Sensordatenvorverarbeitung

Die Sensordaten können in unterschiedlichen Datenformaten vorliegen. Typische Datenformate sind Tiefenbilder (z. B. von Tiefenkameras), Listen von Abstandsmessungen (z. B. von Laserscannern) oder Punktwolken. Um eine sensorunabhängige Darstellung zu erhalten, werden die (gefilterten) Sensordaten zunächst in eine Roboter- und Hindernispunktwolke im Sensorkoordinatensystem umgewandelt. Dabei werden Punkte, die sich außerhalb des Octree \mathcal{B} befinden, teilweise entfernt. Hierfür wird der maximale Abstand zwischen dem Sensorursprung und der Octree-Begrenzung berechnet. Alle Messungen mit einem größeren Abstand zum Sensorursprung liegen sicher nicht im interessierenden Nahbereich des Roboters. Die verbleibenden Hindernis- und Roboterpunkte werden in das Basiskoordinatensystem des Roboters transformiert. Die restlichen Punkte außerhalb der Octree-Begrenzung werden ausgefiltert.

3.3.5 Verdeckungsanalyse

Ziel der Verdeckungsanalyse ist es, für jeden Sensor die Bereiche im Gesichtsfeld des Sensors zu ermitteln, die durch Hindernisse oder den Roboter belegt bzw. verdeckt sind, sowie den freien Bereich, der weder ein Hindernis noch den Roboter enthält.

Das Gesichtsfeld des Sensors wird durch die Öffnungswinkel, die Auflösung sowie die maximale Reichweite bestimmt. Basierend auf diesen Sensoreigenschaften und der bekannten Transformation zwischen Sensor- und Roboterkoordinaten wird für jeden Sensor i die Menge $\mathcal{S}_i(t)$ der Octree-Knoten, die sich im Gesichtsfeld des Sensors befinden,

$$\mathcal{S}_i(t) = \{b \in \mathcal{B} \mid \exists j \in [1, N_{s,i}] : S_{i,j} \text{ schneidet } b\} \quad (3.2)$$

für den aktuellen Zeitpunkt t bestimmt. Dabei ist \mathcal{B} ein Octree im Basiskoordinatensystem des Roboters und b ist ein Octree-Knoten. Im Fall einer statischen Transformation zwischen Sensor- und Roboterkoordinatensystem ist \mathcal{S}_i zeitinvariant und muss nur einmal berechnet werden.

Basierend auf den vorverarbeiteten Hindernispunktwolken $\{\mathbf{p}_{O,i,j}\}$ mit N_O Hindernispunkten werden die Bereiche innerhalb des Octrees ermittelt, welche die gemessene Hindernisoberfläche darstellen. Diese Bereiche werden durch die Menge $\mathcal{P}_i(t)$ der Octree-Knoten beschrieben, die mindestens einen gemessenen Hindernispunkt enthalten:

$$\mathcal{P}_i(t) = \{b \in \mathcal{B} \mid \exists j \in [1, N_O] : \mathbf{p}_{O,i,j} \text{ liegt in } b\} . \quad (3.3)$$

Mittels Strahlverfolgung (engl. *ray tracing*) wird die Menge $\mathcal{H}_i(t)$ der Octree-Knoten, die durch ein Hindernis verdeckt sind, ermittelt. Sie enthält alle Octree-Knoten, die von einem hinter einem Hindernispunkt liegenden Strahlenanteil des Sensormodells geschnitten werden. Die durch Hindernisse verdeckten Strahlenanteile des Sensors i sind

$$S_{O,i,j} = \left\{ \mathbf{p}_{O,i,j} + s \frac{\mathbf{p}_{O,i,j} - \mathbf{s}_{u,i}}{\|\mathbf{p}_{O,i,j} - \mathbf{s}_{u,i}\|} \mid s \in \left[0, d_{S,\max,i} - \|\mathbf{p}_{O,i,j} - \mathbf{s}_{u,i}\| \right] \right\} \quad (3.4)$$

für $j = 1, \dots, N_O$. Die Menge der durch Hindernisse verdeckten Octree-Knoten ist

$$\mathcal{H}_i(t) = \{b \in \mathcal{B} \mid \exists j \in [1, N_O] : S_{O,i,j} \text{ schneidet } b\} . \quad (3.5)$$

Analog wird unter Verwendung der vorverarbeiteten Roboterpunktwolke $\{\mathbf{p}_{R,i,j}\}$ mit N_R , i Roboterpunkten die Menge $\mathcal{R}_i(t)$ der durch den Roboter verdeckten Octree-Knoten berechnet:

$$\mathcal{R}_i(t) = \{b \in \mathcal{B} \mid \exists j \in [1, N_R, i] : S_{R,i,j} \text{ schneidet } b\} \quad (3.6)$$

mit

$$S_{R,i,j} = \left\{ \mathbf{P}_{R,i,j} + s \frac{\mathbf{P}_{R,i,j} - \mathbf{s}_{u,i}}{\|\mathbf{P}_{R,i,j} - \mathbf{s}_{u,i}\|} \mid s \in \left[0, d_{S,\max,i} - \|\mathbf{P}_{R,i,j} - \mathbf{s}_{u,i}\| \right] \right\}. \quad (3.7)$$

Der hindernisfreie Bereich im Gesichtsfeld eines Sensors ist durch die Menge \mathcal{F}_i der Octree-Knoten gegeben, die weder durch den Roboter noch durch Hindernisse verdeckt sind:

$$\mathcal{F}_i(t) = S_i(t) \setminus (\mathcal{H}_i(t) \cup \mathcal{R}_i(t)). \quad (3.8)$$

Dies stellt eine konservative Abschätzung des hindernisfreien Bereichs im Gesichtsfeld eines Sensors dar. Die konservative Abschätzung ist notwendig, da über die durch Hindernisse und den Roboter verdeckten Bereiche keine Informationen vorliegen.

3.3.6 Sensorfusion

Die Filterung, die Sensordatenvorverarbeitung und die Verdeckungsanalyse werden für jeden Sensor unabhängig durchgeführt, sodass eine parallele Verarbeitung möglich ist. Um die 3D-Hindernisdarstellung zu erhalten, werden die Informationen über freie und verdeckte Bereiche der einzelnen Sensoren fusioniert. Hierzu werden zunächst für alle Sensoren die Anteile der durch Hindernisse verdeckten Bereiche entfernt, die von anderen Sensoren als frei erkannt werden:

$$\tilde{\mathcal{H}}_i(t) = \mathcal{H}_i(t) \setminus \left(\bigcup_{j,j \neq i} \mathcal{F}_j(t) \right). \quad (3.9)$$

Im resultierenden Octree

$$\mathcal{O}(t) = \bigcup_i \left(\mathcal{N}(t) \cap \left(\mathcal{P}_i(t) \cup \tilde{\mathcal{H}}_i(t) \right) \right) \quad (3.10)$$

werden alle Knoten als mögliches Hindernis interpretiert, die direkt gemessene Hindernispunkte enthalten oder die durch ein Hindernis verdeckt sind

und nicht von einem anderen Sensor als frei erkannt werden. Dabei ist $\mathcal{N}(t)$ die Menge aller Octree-Knoten, die innerhalb des interessierenden Nahbereichs des Roboters liegen.

Das bedeutet, dass der überwachte Raum die Vereinigung aller Sensorgesichtsfelder innerhalb des Nahbereichs des Roboters ist. Unüberwachter Raum wird nicht als Hindernis betrachtet, da angenommen wird, dass die Sensoranordnung eine ausreichende Abdeckung bietet. Diese Annahme ist notwendig, da die Sensoranordnung insbesondere bei mobilen Robotern häufig keine vollständige Abdeckung der Roboterumgebung ermöglichen kann. So wird beispielsweise in der in Abschnitt 8.1 vorgestellten Sensoranordnung der Raum um eine mobile Plattform lediglich in einer Ebene erfasst. Das dadurch verbleibende Risiko wird aber als akzeptabel eingestuft, da falls ein Menschen neben der Plattform steht, zumindest Teile des Menschen durch die Sensoren detektiert werden. Bereiche, die nur durch den Roboter aber nicht durch ein Hindernis verdeckt werden, werden ebenfalls nicht als Hindernis interpretiert.

Da die einzelnen Sensoren asynchron arbeiten und unterschiedliche Messraten aufweisen können, wird der Fusionsschritt durchgeführt, sobald ein neues Ergebnis der Verdeckungsanalyse für mindestens einen Sensor vorliegt. Für alle anderen Sensoren werden die letzten verfügbaren Informationen zugrunde gelegt. So werden plötzlich auftretende Hindernisse schnellstmöglich detektiert.

Das vorgeschlagene Fusionsprinzip wird in mehreren Skizzen am Beispiel von je zwei Abstandssensoren illustriert. Abbildung 3.5 zeigt eine Szene mit dem bekannten Roboter (blaues Objekt) sowie einem Hindernis (rotes Objekt), das sich aus Sicht der Sensoren neben dem Roboter befindet. Die beiden oberen Skizzen zeigen für jeden der beiden Sensoren die freien Octree-Knoten im Sensorgesichtsfeld (grün), die durch das Hindernis belegten oder verdeckten Knoten (rot) sowie die durch den Roboter belegten oder verdeckten Knoten (blau). Die untere Skizze zeigt das Fusionsergebnis, wobei die roten Zellen der resultierenden Hindernisdarstellung entsprechen. Die Abbildung veranschaulicht, wie durch die Fusion der Information mehrerer Sensoren der überwachte Raum vergrößert wird. Des Weiteren ist zu sehen,

dass durch die Fusion der durch den Roboter und das Hindernis verdeckte Raum verkleinert wird, ohne dass hierbei die Ausdehnung des Hindernisses unterschätzt wird.

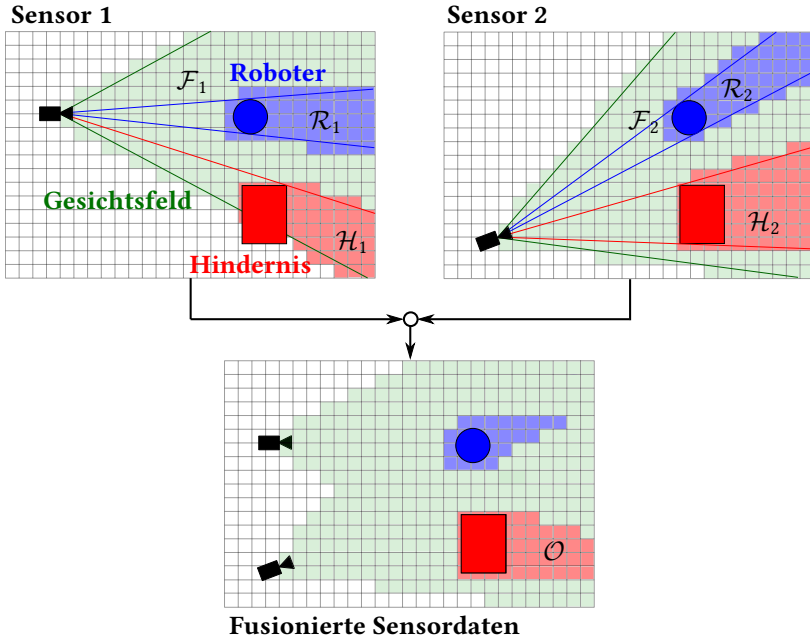


Abbildung 3.5: Prinzip der 3D-Hindernisdarstellung mit zwei Sensoren: Beispiel mit Hindernis neben dem Roboter (nach [Fet14]).

Das zweite Beispiel in Abbildung 3.6 zeigt eine ähnliche Situation. In diesem Beispiel befindet sich das Hindernis jedoch zwischen den Sensoren und dem Roboter. Es zeigt die Notwendigkeit, den durch das Hindernis verdeckten Raum zu berücksichtigen. Auf Grund der Sensordaten kann nicht ausgeschlossen werden, dass das Hindernis bis nahe an den Roboter heran reicht. Deshalb ist aus Sicherheitsgründen eine konservative Abschätzung der Hindernisgröße erforderlich, indem durch Hindernisse verdeckte Bereiche, die nicht von einem anderen Sensor als frei erkannt werden, als Hindernis interpretiert werden.

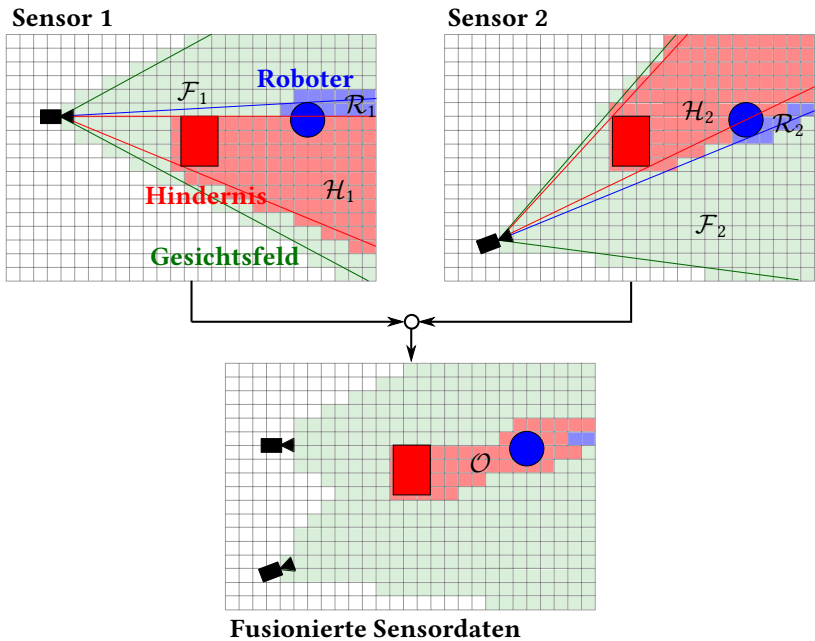


Abbildung 3.6: Prinzip der 3D-Hindernisdarstellung mit zwei Sensoren: Beispiel mit Hindernis zwischen Sensor und Roboter (nach [Fet14]).

Auch das letzte Beispiel in Abbildung 3.7 zeigt die Notwendigkeit, die durch Roboter und Hindernis verdeckten Bereiche zu berücksichtigen, wobei besonderes Augenmerk auf die Berücksichtigung der Verdeckung durch den Roboter gelegt wird. Wie die vorherigen Beispiele enthält die dargestellte Szene einen Roboter und ein Hindernis, die mit zwei Abstandssensoren beobachtet werden. In diesem Beispiel befindet sich das Hindernis für einen der beiden Sensoren hinter dem Roboter, sodass ein Teil des Hindernisses durch den Roboter verdeckt ist und nicht von diesem Sensor erfasst werden kann. Es wird deutlich, dass der durch den Roboter verdeckte Bereich nicht als frei angenommen werden kann, da in diesem Fall die Ausdehnung des Hindernisses als zu klein abgeschätzt würde.

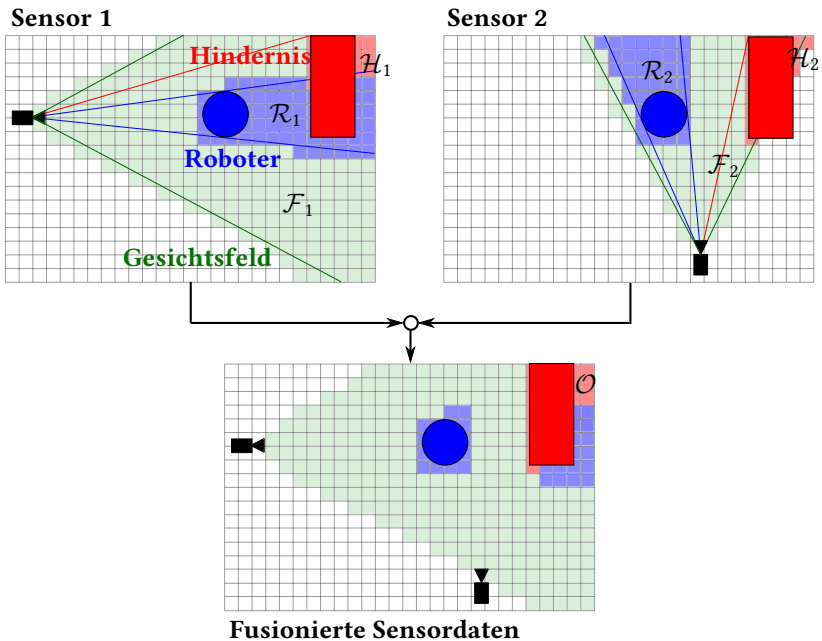


Abbildung 3.7: Prinzip der 3D-Hindernisdarstellung mit zwei Sensoren: Beispiel mit Hindernis hinter dem Roboter.

3.4 Objektklassifikation

Im bisherigen Teil wurde angenommen, dass die Objekte im Bereich der Arbeitsraumüberwachung entweder Teile des Roboters oder zu detektierende Hindernisse sind. Für die Kollisionsvermeidung entspricht das einer Worst-Case-Annahme. In Hinblick auf die Mensch-Roboter-Interaktion ist jedoch eine feinere Unterscheidung der Objekte im Arbeitsraum des Roboters sinnvoll. In vielen Anwendungsszenarien gibt es neben unbekanntem statischen oder dynamischen Hindernissen auch a-priori bekannte statische Hindernisse. Beispielsweise kann ein Umgebungsmodell vorhanden sein, das Wände und feststehende Einrichtungsgegenstände enthält. Dieses Vorwissen kann genutzt werden, um den Aufwand für die 3D-Arbeitsraumüberwachung zu reduzieren. Auch ist es nicht notwendig die Größe von bekannten Hindernissen auf Grund von Verdeckungen im Sensorsichtfeld zu überschätzen. Des Weiteren können neben Robotern und Hindernissen (d. h. Objekten, mit denen eine Kollision vermieden werden muss) auch Objekte existieren, mit denen ein Kontakt erlaubt oder gefordert ist. Derartige Objekte sind beispielsweise Gegenstände, die vom Roboter manipuliert werden sollen. Werden diese Objekte als Hindernis in der Arbeitsraumüberwachung detektiert, ist es für den Roboter nicht möglich seine Aufgabe auszuführen.

Deshalb werden folgende Ergänzungen der in Abschnitt 3.3 beschriebenen Methode zur Ermittlung einer 3D-Hindernisdarstellung eingeführt: Liegt eine separate Darstellung der bekannten statischen Hindernisse vor, werden diese Hindernisse bei Erzeugung der 3D-Hindernisdarstellung der unbekanntem (dynamischen) Hindernisse wie der Roboter behandelt. Das bedeutet, dass die durch diese bekannten statischen Hindernisse belegten oder verdeckten Bereiche nicht als Hindernis in die Darstellung eingehen. Voraussetzung ist, dass die Bewegungsplanung und -regelung des Roboters diese statischen Hindernisse ebenfalls zur Kollisionsvermeidung berücksichtigt (siehe auch Abschnitt 6.8).

Analog werden auch zu manipulierende Objekte in der Arbeitsraumüberwachung wie Teile des Roboters behandelt. Auch sie gehen nicht in die

3D-Darstellung der unbekanntes Hindernisse ein. Voraussetzung ist, dass Vorwissen über Größe und Pose dieser Objekte existiert oder dieses Wissen mit Hilfe von Objektdetektionsalgorithmen gewonnen wird.

3.5 Zusammenfassung

In Hinblick auf die Kollisionsvermeidung in der Mensch-Roboter-Interaktion wurde eine Methode zur Ermittlung einer 3D-Darstellung der (dynamischen) Hindernisse in der nahen Roboterumgebung vorgeschlagen. Der aktuelle Nahbereich des Roboters wird basierend auf dem durch den Roboter erreichbaren Raum definiert. Innerhalb dieses Nahbereichs wird der durch Hindernisse und durch den Roboter belegte oder verdeckte Raum mit Hilfe einer Octree-Datenstruktur dargestellt. Dabei wird die Information mehrerer heterogener Abstandssensoren fusioniert. Aus Sicherheitsgründen erfolgt eine konservative Abschätzung der Hindernisausdehnung.

Kapitel 4

Robotermodell

In diesem Kapitel wird ein allgemeines kinematisches Modell zur Modellierung von feststehenden und mobilen Manipulatoren vorgestellt, wie es in der kartesischen Roboterreglung (siehe Kapitel 5) eingesetzt werden kann. Des Weiteren werden die Grundlagen der Modellierung der Roboterdynamik angegeben. Für weitere Details sei auf [Sci00] und [Sic08] verwiesen.

4.1 Begriffsklärung

In dieser Arbeit werden Roboter betrachtet, die sich als eine Kette von steifen Robotergliedern beschreiben lassen. Die einzelnen Roboterglieder sind durch Rotations- oder Translationsgelenke verbunden. Zusätzlich können nicht-holonome Beschränkungen gelten. Auf diese Art können generell Roboterarme mit steifen Gliedern modelliert werden. Aber auch mobile Plattformen sowie mobile Manipulatoren (Manipulatoren auf mobilen Plattformen) lassen sich so beschreiben.

Das letzte Glied der kinematischen Kette des Roboters wird als Endeffektor bezeichnet. Dies entspricht der für Manipulatoren üblichen Namensgebung, bei welcher der Endeffektor meist einen Greifer oder ein Werkzeug darstellt. Es erfolgt jedoch keine Unterscheidung zwischen unterschiedli-

chen Robotertypen, sodass beispielsweise im Fall einer mobilen Plattform der Endeffektor auch der Plattform selbst entsprechen kann.

Bei der Modellierung eines Roboters wird zwischen der Roboterkinematik und der Roboterdynamik unterschieden. Die Kinematik beschreibt die Bewegung der Roboterglieder unter Berücksichtigung von Geschwindigkeiten und Beschleunigungen. Die Roboterdynamik betrachtet die dabei auftretenden Kräfte und Drehmomente. Sie umfasst sowohl die internen Kräfte und Drehmomente, die in den Gelenken aufgebracht werden, um eine Bewegung zu erzeugen, als auch die extern auf den Roboter einwirkenden Kräfte und Drehmomente wie Gewichtskräfte oder Kollisionskräfte.

4.2 Kinematisches Robotermodell

Ein Roboter, der als eine Kette steifer Roboterglieder modelliert werden kann, die durch N Rotations- oder Translationsgelenke verbunden sind, wird durch das folgende kinematische Zustandsmodell beschrieben:

$$\begin{bmatrix} \dot{\mathbf{q}}(t) \\ \mathbf{w}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}(t) \\ \mathbf{w}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{a}(t) \end{bmatrix} \quad (4.1)$$

$$\mathbf{y}_{EE}(t) = \mathbf{k}(\mathbf{q}(t)) . \quad (4.2)$$

Dabei gibt $\mathbf{q}(t)$ den N -dimensionalen Vektor der Gelenkpositionen der Rotations- und Translationsgelenke, $\mathbf{w}(t)$ den N -dimensionalen Vektor der Gelenkgeschwindigkeiten, $\mathbf{a}(t)$ den N -dimensionalen Vektor der Gelenkbeschleunigungen (Eingangsgrößen) und $\mathbf{y}_{EE}(t)$ den M -dimensionalen Vektor der Endeffektorpose (Ausgangsgröße) an. Die Endeffektorpose setzt sich aus der Position $\mathbf{p}_{EE}(t)$ und Orientierung $\mathbf{o}_{EE}(t)$ des Endeffektors zusammen:

$$\mathbf{y}_{EE}(t) := \begin{bmatrix} \mathbf{p}_{EE}(t) \\ \mathbf{o}_{EE}(t) \end{bmatrix} . \quad (4.3)$$

Die Variablen zur Beschreibung des Robotersystems sind abhängig von der Zeit t . Im Folgenden wird jedoch häufig auf eine explizite Darstellung der

Zeitabhängigkeit verzichtet, um eine bessere Lesbarkeit der Gleichungen zu erreichen.

Die Menge aller möglichen Gelenkkonfigurationen $\mathcal{C} \subseteq \mathbb{R}^N$ ist der Gelenkraum des Roboters (engl. *joint space*). Die Menge aller erreichbaren Endeffektorposen $\mathcal{A} \subseteq \mathbb{R}^M$ ist der Aufgabenraum (engl. *task space*). Dieser Raum wird in der Robotik auch als kartesischer Raum bezeichnet.

Die nichtlineare Funktion $\mathbf{k} : \mathcal{C} \rightarrow \mathcal{A}$ gibt an, welche Pose \mathbf{y}_{EE} der Endeffektor bei einer gegebenen Gelenkkonfiguration \mathbf{q} einnimmt, und wird als die direkte Kinematik bezeichnet. Sie ist nur im Fall spezieller Roboterstrukturen eindeutig invertierbar.

4.2.1 Darstellung der Orientierung

Ohne Beschränkung der Allgemeinheit wird hier davon ausgegangen, dass sich der Roboter im dreidimensionalen Raum bewegt. Dann setzt sich die Endeffektorpose aus einem dreidimensionalen Positionsvektor \mathbf{p}_{EE} und einem Vektor \mathbf{o}_{EE} , der die Orientierung im dreidimensionalen Raum angibt, zusammen. Für die Darstellung der Orientierung bestehen verschiedene Möglichkeiten. Ein gängiger Ansatz ist, die Orientierung durch drei Eulerwinkel zu beschreiben, woraus ein sechsdimensionaler Aufgabenraum $\mathcal{A} \subseteq \mathbb{R}^M$ mit $M = 6$ resultiert. Eulerwinkel haben den Vorteil, dass sie eine minimale Repräsentation der Orientierung darstellen und sehr anschaulich sind. Die Darstellung der Orientierung mit Eulerwinkeln ist jedoch singularitätsbehaftet. So gibt es Orientierungen, die sich durch unendlich viele Kombinationen von Eulerwinkeln beschreiben lassen. Dies stellt für die zu entwickelnde Regelung ein schwerwiegendes Problem dar, weshalb stattdessen die Orientierung mit Hilfe von Einheitsquaternionen dargestellt wird. Diese ermöglichen eine singularitätsfreie Repräsentation der Orientierung [Sic08]. Eine Einheitsquaternion bildet einen vierdimensionalen Vektor $\mathbf{o}_{EE} := [o_x, o_y, o_z, o_w]^T$, womit sich ein Aufgabenraum der Dimension $M = 7$ ergibt. Hierbei sind die vier Quaternionenelemente jedoch nicht unabhängig und es gilt $\|\mathbf{o}_{EE}\| = 1$.

4.2.2 Ermittlung der direkten Kinematik

Die direkte Kinematik $k : \mathcal{C} \rightarrow \mathcal{A}$ kann mit Hilfe der Denavit-Hartenberg-Konvention (weitere Details siehe [Sci00]) bestimmt werden. Dabei wird jedem Roboterglied ein Koordinatensystem mit Ursprung \mathbf{O}_i zugeordnet (siehe Abbildung 4.1). Die Beziehung zweier aufeinanderfolgender Koordinatensysteme wird durch die Denavit-Hartenberg-Parameter $\alpha_{\text{DH}}, \theta_{\text{DH}}, a_{\text{DH}}$ und d_{DH} angegeben, die wie folgt definiert sind:

- Der Winkel $\theta_{\text{DH},i}$ beschreibt die Rotation um die z_{i-1} -Achse, sodass die x_{i-1} -Achse und die x_i -Achse parallel verlaufen.
- $d_{\text{DH},i}$ gibt die Verschiebung des Ursprungs \mathbf{O}_{i-1} entlang der z_{i-1} -Achse bis zum Schnittpunkt \mathbf{S}_i der z_{i-1} -Achse und der x_i -Achse an.
- $a_{\text{DH},i}$ gibt die Verschiebung des Schnittpunkts \mathbf{S}_i entlang der x_i -Achse bis zum Koordinatenursprung \mathbf{O}_i an.
- Der Winkel $\alpha_{\text{DH},i}$ beschreibt die Rotation um die x_i -Achse, sodass die z_{i-1} -Achse und die z_i -Achse parallel verlaufen.

Die homogene Transformation \mathbf{T}_{i-1}^i zwischen den Koordinatensystemen der Glieder i und $i-1$, die durch das Gelenk i verbunden sind, ist gegeben durch

$$\mathbf{T}_{i-1}^i = \begin{bmatrix} c_{\theta,i} & -c_{\alpha,i}s_{\theta,i} & s_{\alpha,i}s_{\theta,i} & a_{\text{DH},i}c_{\theta,i} \\ s_{\theta,i} & c_{\alpha,i}c_{\theta,i} & -s_{\alpha,i}c_{\theta,i} & a_{\text{DH},i}s_{\theta,i} \\ 0 & s_{\alpha,i} & c_{\alpha,i} & d_{\text{DH},i} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4.4)$$

Zur Verbesserung der Übersichtlichkeit werden hierbei die Abkürzungen $c_{\theta,i} := \cos(\theta_{\text{DH},i})$, $s_{\theta,i} := \sin(\theta_{\text{DH},i})$, $c_{\alpha,i} := \cos(\alpha_{\text{DH},i})$ und $s_{\alpha,i} := \sin(\alpha_{\text{DH},i})$ verwendet. Im Falle eines Rotationsgelenks ist $\theta_{\text{DH},i}$ abhängig von der Gelenkposition mit $\theta_{\text{DH},i}(q_i) = \theta_{\text{DH},i,0} + q_i$ mit dem konstanten Winkel $\theta_{\text{DH},i,0}$. Im Falle eines Translationsgelenks gilt $d_{\text{DH},i}(q_i) = d_{\text{DH},i,0} + q_i$ mit

der konstanten Verschiebung $d_{\text{DH},i,0}$. Alle anderen Denavit-Hartenberg-Parameter sind konstant. Die homogene Transformationsmatrix zwischen dem Koordinatensystem des Endeffektors und der Roboterbasis ergibt sich aus der Verkettung der den Gelenken $i = 1, \dots, N$ zugeordneten Transformationen:

$$\mathbf{T}_0^N(\mathbf{q}) = \mathbf{T}_0^1(q_1) \cdot \mathbf{T}_1^2(q_2) \cdot \dots \cdot \mathbf{T}_{N-1}^N(q_N). \quad (4.5)$$

Diese Transformationsmatrix beinhaltet den Positionsvektor des Endeffektors sowie die Rotationsmatrix \mathbf{R}_0^N , welche die Orientierung des Endeffektors im Basiskoordinatensystem angibt:

$$\mathbf{T}_0^N(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_0^N & \mathbf{p}_{\text{EE}} \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (4.6)$$

Die Position des Endeffektors ergibt sich aus

$$\begin{bmatrix} \mathbf{p}_{\text{EE}} \\ 1 \end{bmatrix} = \mathbf{T}_0^N(\mathbf{q}) \cdot \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^\top. \quad (4.7)$$

Die Einheitsquaternion zur Beschreibung der Endeffektororientierung in Basiskoordinaten $\mathbf{o}_{\text{EE}} = [o_x, o_y, o_z, o_w]^\top$ kann wie in Anhang A dargestellt aus der Rotationsmatrix \mathbf{R}_0^N ermittelt werden.

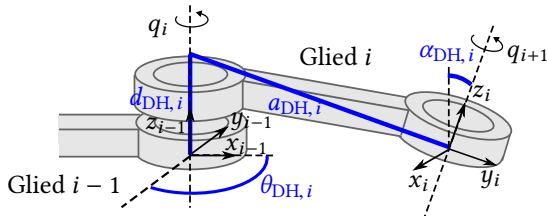


Abbildung 4.1: Parameter der Denavit-Hartenberg-Konvention.

4.2.3 Anmerkungen zu mobilen Plattformen

Für feststehende Manipulatoren ist es weit verbreitet, die Denavit-Hartenberg-Konvention zur Bestimmung der direkten Kinematik zu verwenden. Details zur Wahl der Parameter sind beispielsweise in [Sci00] zu finden. Der Ansatz lässt sich jedoch leicht auf mobile Manipulatoren erweitern. Eine mobile Plattform, die sich auf einer Ebene bewegt, kann mit Hilfe von zwei Translationsgelenken und einem Rotationsgelenk für die Drehung um die Hochachse des Fahrzeugs modelliert werden. Besitzt das Fahrzeug nicht-holonome Eigenschaften, werden diese zusätzlich durch nicht-holonome Beschränkungen der Form

$$\mathbf{h}(\mathbf{q}, \mathbf{w}) = \mathbf{0} \quad (4.8)$$

abgebildet.

Somit steht ein analytisches kinematisches Modell zur Verfügung, das sich sowohl für feststehende Manipulatoren, mobile Plattformen als auch mobile Manipulatoren anwenden lässt.

4.2.4 Jacobi-Matrix

Die analytische Jacobi-Matrix der direkten Kinematik ist die $M \times N$ Matrix der partiellen Ableitungen der direkten Kinematik nach den Gelenkpositionen

$$\mathbf{J}(\mathbf{q}) := \frac{\partial \mathbf{k}}{\partial \mathbf{q}}(\mathbf{q}). \quad (4.9)$$

Mit ihr gilt für die Ableitung der Endeffektorpose nach der Zeit

$$\dot{\mathbf{y}}_{EE} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}. \quad (4.10)$$

Wie die direkte Kinematik ist die Jacobi-Matrix nur in Spezialfällen eindeutig invertierbar.

Eine schnelle Berechnung der Jacobi-Matrix für die aktuelle Gelenkkonfiguration ist unter Anwendung der Kettenregel auf die Vorschriften zur Berechnung der Position und Orientierung aus der Transformationsmatrix

$T_0^N(\mathbf{q})$ und Ermittlung der partiellen Ableitungen von Gleichung (4.5)

$$\frac{\partial}{\partial q_j} T_0^N(\mathbf{q}) = \prod_{i=1}^{j-1} T_{i-1}^i(q_i) \cdot \frac{\partial}{\partial q_j} T_{j-1}^j(q_j) \cdot \prod_{i=j+1}^N T_{i-1}^i(q_i) \quad (4.11)$$

möglich. Wird sowohl die direkte Kinematik als auch die Jacobi-Matrix benötigt, kann die Rechenzeit beschleunigt werden, indem Zwischenergebnisse wie die Werte der einzelnen Transformationsmatrizen oder das Produkt der Transformationsmatrizen $\prod_{i=1}^{j-1} T_{i-1}^i(q_i)$ für $j = 2, \dots, N - 1$ zwischengespeichert und für beide Berechnungen genutzt werden.

Analog kann die Jacobi-Matrix statt für den Endeffektor auch für jeden anderen Punkt auf dem Roboter bestimmt werden, indem statt der direkten Kinematik des Endeffektors die direkte Kinematik des jeweiligen Roboterpunktes zugrunde gelegt wird. Im Folgenden werden Jacobi-Matrizen, die sich nicht auf die Endeffektorpose beziehen, durch einen Index gekennzeichnet, der den entsprechenden Bezugspunkt angibt.

4.2.5 Hesse-Matrix

Die Hesse-Matrizen der direkten Kinematik sind die $N \times N$ Matrizen der zweiten partiellen Ableitungen der Elemente der Endeffektorpose

$$\mathbf{H}_l(\mathbf{q}) := \left(\frac{\partial^2 k_l}{\partial q_i \partial q_j}(\mathbf{q}) \right)_{i,j=1,\dots,N} \quad (4.12)$$

mit $l = 1, \dots, M$.

Die Hesse-Matrizen können somit wie die Jacobi-Matrix und die direkte Kinematik analytisch für das allgemeine kinematische Robotermodell berechnet werden.

4.2.6 Diskretisierung des kinematischen Robotermodells

Unter Verwendung der Euler-Approximation lässt sich das kontinuierliche kinematische Robotermodell (4.1)–(4.2) in das diskrete kinematische

Robotermodell

$$\mathbf{q}(k+1) = \mathbf{q}(k) + T_s \mathbf{w}(k) \quad (4.13)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + T_s \mathbf{a}(k) \quad (4.14)$$

$$\mathbf{y}_{EE}(k) = \mathbf{k}(\mathbf{q}(k)) \quad (4.15)$$

überführen. Hierbei bezeichnet T_s die Abtastzeit. Der Zeitindex $k \in \mathbb{N}$ gibt den Abtastzeitpunkt $t(k) = kT_s$ an.

Zur Verbesserung der Lesbarkeit wird im Folgenden häufig auf die explizite Darstellung der Abhängigkeit der Variablen vom Zeitindex k verzichtet.

4.3 Dynamisches Robotermodell

Für die Dynamik eines Roboters bestehend aus $N + 1$ steifen Gliedern und N Gelenken gilt unter Vernachlässigung der Reibungskräfte

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{ext}}. \quad (4.16)$$

Hierbei ist $\mathbf{M} \in \mathbb{R}^{N \times N}$ die Massenträgheitsmatrix, $\mathbf{C} \in \mathbb{R}^{N \times N}$ die Matrix der Coriolis- und Zentrifugalkräfte und $\mathbf{g} \in \mathbb{R}^N$ der Gravitationsvektor. Das Gesamtmoment setzt sich aus dem Antriebsmoment $\boldsymbol{\tau} \in \mathbb{R}^N$ und den externen Momenten $\boldsymbol{\tau}_{\text{ext}} \in \mathbb{R}^N$ zusammen, die durch das Einwirken von externen Kontaktkräften auf den Roboter entstehen [Sic08].

Zwischen einer externen Kontaktkraft und den resultierenden externen Momenten besteht folgender Zusammenhang: Wirkt die Kontaktkraft \mathbf{F}_{ext} auf den Endeffektor resultiert der Momentenvektor

$$\boldsymbol{\tau}_{\text{ext}} = \mathbf{J}_p^T(\mathbf{q}) \cdot \mathbf{F}_{\text{ext}}. \quad (4.17)$$

Hierbei bezeichnet \mathbf{J}_p die Jacobi-Matrix der direkten Kinematik der Endeffektorposition und ist damit eine $3 \times N$ Matrix.

Wirkt die Kontaktkraft auf einen beliebigen anderen Punkt des Roboters, ist der Zusammenhang durch die zu dem Kontaktpunkt gehörende

Jacobi-Matrix $\mathbf{J}_C(\mathbf{q})$ gegeben. In diesem Fall sind alle externen Momente der Gelenke, die in der kinematischen Kette oberhalb des Kontaktpunktes liegen, gleich Null.

Kapitel 5

Kartesische modellprädiktive Roboterregelung

In diesem Kapitel werden zunächst die Problemstellungen der Verfolgung von Pfaden und Trajektorien der Endeffektorpose formal definiert. Es folgen Grundlagen der modellprädiktiven Regelung. Dieses Prinzip wird dann auf die Bewegungsregelung (mobiler) Manipulatoren basierend auf dem in Kapitel 4 eingeführten allgemeinen kinematischen Robotermodell übertragen.

5.1 Problemstellung

In dieser Arbeit wird die Regelung von Robotern im Aufgabenraum des Roboters betrachtet. Der Roboter kann ein stationärer oder mobiler Manipulator oder auch eine mobile Plattform sein (zur Begriffsklärung siehe Abschnitt 4.1). Als Sollgröße der Regelung sind Sollwerte der Endeffektorpose y_{EE} gegeben, die die Position und Orientierung des Endeffektors des Roboters im kartesischen Raum angibt.

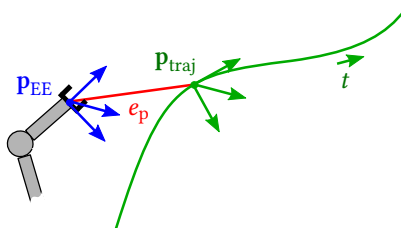


Abbildung 5.1: Trajektorienverfolgung mit Endeffektorposition p_{EE} , Sollposition p_{traj} und Positionsfehler e_p .

Abhängig von der Art der Sollgröße lässt sich die Regelung der Endeffektorpose in unterschiedliche Aufgabenstellungen unterteilen. Stellt die Sollpose eine stationäre Pose dar, liegt das regelungstechnische Problem der Arbeitspunktstabilisierung vor. Hat die Endeffektorpose dagegen einer dynamischen Sollpose zu folgen, wird eine Folgeregelung benötigt. Die Folgeregelung lässt sich wiederum in die Verfolgung von zeitabhängigen Solltrajektorien (engl. *trajectory tracking*) und die Pfadverfolgung (engl. *path following*) ohne a-priori gegebene Zeitabhängigkeit unterteilen [Fau09].

Unabhängig von der Aufgabenstellung besteht eine grundlegende Anforderung an die Regelung in der Einhaltung der Gelenkbeschränkungen des Roboters wie z. B. der maximalen Gelenkgeschwindigkeiten. Weitere Anforderungen, wie die Vermeidung von Kollisionen mit der Umgebung sowie die Vermeidung von Eigenkollisionen, werden in den folgenden Kapiteln als Erweiterung der kartesischen Regelung behandelt.

5.1.1 Trajektorienverfolgung

Die Aufgabe der Trajektorienverfolgung besteht darin, dass die Endeffektorpose $y_{EE}(t)$ der zeitabhängigen Solltrajektorie $y_{traj}(t)$ im Aufgabenraum $\mathcal{A} \subseteq \mathbb{R}^M$ möglichst gut folgt (siehe Abbildung 5.1). Die Aufgabe der Regelung besteht somit in der Minimierung des Trajektorienverfolgungsfehlers, der die Abweichung der tatsächlichen Endeffektorpose von der Sollpose angibt. Es wird dabei vorausgesetzt, dass die Trajektorie für den Roboter befahrbar ist.

Für die Endeffektorpose bestehend aus Positionsvektor \mathbf{p}_{EE} und Einheitsquaternion \mathbf{o}_{EE} wird der Trajektorienverfolgungsfehler wie folgt definiert:

$$\mathbf{e}_{\text{traj}}(t) := \begin{bmatrix} \mathbf{p}_{EE}(t) - \mathbf{p}_{\text{traj}}(t) \\ \mathbf{o}_{EE}(t) - s^*(t) \cdot \mathbf{o}_{\text{traj}}(t) \end{bmatrix} \quad (5.1)$$

mit

$$s^*(t) = \arg \min_{s \in \{-1, 1\}} (\|\mathbf{o}_{EE}(t) - s(t) \cdot \mathbf{o}_{\text{traj}}(t)\|) . \quad (5.2)$$

Für den Trajektorienverfolgungsfehler wird das Vorzeichen s^* der Sollquaternion aufgrund der Eigenschaft von Quaternionen, dass die Quaternionen \mathbf{o} und $-\mathbf{o}$ dieselbe Orientierung im dreidimensionalen kartesischen Raum darstellen [Huy09], entsprechend Gleichung (5.2) gewählt. Das bedeutet, dass der Trajektorienverfolgungsfehler die Abweichung der Endeffektororientierung von der Sollorientierung mit dem kleineren euklidischen Abstand enthält. Das euklidische Abstandsmaß hat hierbei den Vorteil, dass es ein Abstandsmaß für Quaternionen mit geringem Rechenaufwand darstellt [Huy09] und eine einheitliche Behandlung des Abstandes von Positionen und Orientierungen ermöglicht.

Die Regelung der Endeffektorpose auf eine stationäre Sollpose \mathbf{y}_s kann als Spezialfall der Trajektorienverfolgung mit

$$\mathbf{y}_{\text{traj}}(t) = \mathbf{y}_s = \text{const} \quad (5.3)$$

behandelt werden.

5.1.2 Pfadverfolgung

Ein Pfadverfolgungsproblem liegt vor, wenn statt einer zeitabhängigen Solltrajektorie ein geometrischer Sollpfad ohne Zeitabhängigkeit gegeben ist. Es gibt a-priori keine Information, zu welchem Zeitpunkt eine bestimmte Pose eingenommen werden soll oder mit welcher Geschwindigkeit der Pfad abzufahren ist.

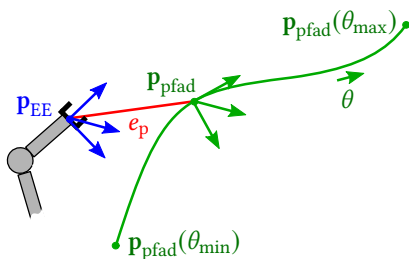


Abbildung 5.2: Pfadverfolgung mit Endeffektorposition \mathbf{p}_{EE} , Sollposition \mathbf{p}_{pfad} , Pfadparameter θ und Positionsfehler \mathbf{e}_p .

Der geometrische Referenzpfad der Endeffektorpose $\mathbf{y}_{pfad}(\theta)$ im Aufgabenraum $\mathcal{A} \subseteq \mathbb{R}^M$ wird in Abhängigkeit des Pfadparameters θ mit $\theta \in [\theta_{min}, \theta_{max}]$ angegeben, wobei der Pfadbeginn durch $\mathbf{y}_{pfad}(\theta_{min})$ und das Pfadende durch $\mathbf{y}_{pfad}(\theta_{max})$ gegeben sind (siehe Abbildung 5.2). Der Pfadparameter ist zeitabhängig, die zeitliche Entwicklung $\theta(t)$ ist jedoch a-priori nicht bekannt. Die Bewegungsrichtung auf dem Pfad ist durch $\dot{\theta} > 0$ gegeben.

Analog zum Trajektorienverfolgungsfehler wird der Pfadverfolgungsfehler definiert als

$$\mathbf{e}_{pfad}(t) := \begin{bmatrix} \mathbf{p}_{EE}(t) - \mathbf{p}_{pfad}(\theta(t)) \\ \mathbf{o}_{EE}(t) - s^*(t) \cdot \mathbf{o}_{pfad}(\theta(t)) \end{bmatrix} \quad (5.4)$$

mit

$$s^*(t) = \arg \min_{s \in \{-1, 1\}} (\|\mathbf{o}_{EE}(t) - s(t) \cdot \mathbf{o}_{pfad}(\theta(t))\|) . \quad (5.5)$$

Die Aufgabe des Pfadverfolgers besteht darin, den Pfadverfolgungsfehler zu minimieren, wobei das Voranschreiten auf dem Pfad mit $\dot{\theta} > 0$ gewährleistet werden muss.

In dieser Arbeit wird dabei vorausgesetzt, dass der Pfad durch den Roboter befahrbar und zweimal stetig differenzierbar ist. Ist der Pfad durch eine Folge von Sollposen gegeben, kann ein glatter Pfad durch die Interpolation mit kubischen Splines erzeugt werden (siehe hierzu auch Anhang A).

5.2 Grundlagen der modellprädiktiven Regelung

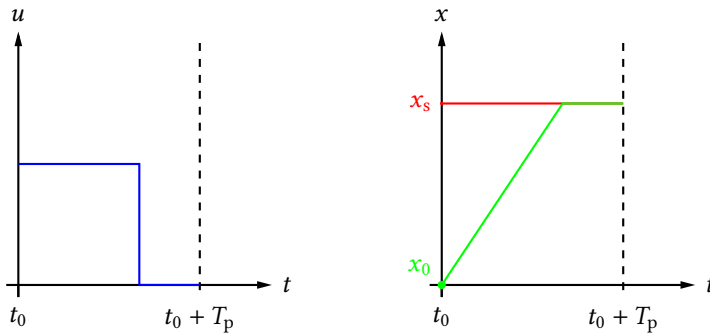
Die nichtlineare modellprädiktive Regelung (engl. *Nonlinear Model Predictive Control*, NMPC) ist ein geeigneter Ansatz für das vorliegende Regelungsproblem. Sie ermöglicht die Regelung nichtlinearer Systeme unter Berücksichtigung von harten Beschränkungen und lässt sich sowohl für die Arbeitspunktstabilisierung als auch zur Folgeregelung einsetzen. Im Folgenden wird ein kurzer Überblick über das Prinzip dieser Regelung gegeben. Weitere Details zur NMPC sind in [Fin04; Fau12] zu finden. Der Entwurf der Regelung für die vorgestellten Trajektorien- und Pfadverfolgungsprobleme folgt in den Abschnitten 5.3 und 5.4.

5.2.1 Prinzip der modellprädiktiven Regelung

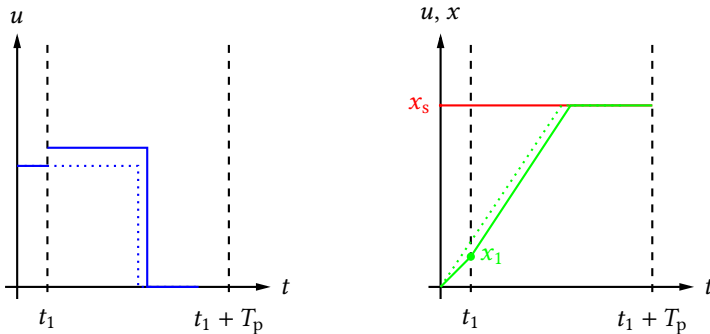
Ausgehend vom aktuellen Systemzustand prädiziert die NMPC das künftige Systemverhalten abhängig von den Stellgrößen des Systems. Die Prädiktion erfolgt über einen festen Zeithorizont und legt ein zeitkontinuierliches oder zeitdiskretes Systemmodell zugrunde. Es werden in Hinblick auf ein Gütekriterium optimale Stellgrößen für den gesamten Prädiktionshorizont bestimmt, wobei Beschränkungen der Stell- und Zustandsgrößen direkt berücksichtigt werden können. Aufgrund von externen Störeinflüssen und Modellfehlern können die prädizierten Zustände jedoch von den tatsächlichen Zuständen abweichen. Außerdem können die optimalen Stellgrößen nur für einen endlichen Prädiktionshorizont bestimmt werden. Deshalb werden die berechneten Stellgrößen nicht über den gesamten Prädiktionshorizont, sondern nur für den nächsten Reglerschritt verwendet. Dann wird entsprechend des Prinzips des gleitenden Horizonts der Systemzustand aktualisiert und es werden neue optimale Stellgrößen für den „verschobenen“ Prädiktionshorizont berechnet.

Dieses Vorgehen ist in Abbildung 5.3 für ein allgemeines System mit Zustandsgröße x und Stellgröße u dargestellt. Die beiden oberen Schaubilder zeigen den Verlauf der optimierten Stellgröße u (blau) und den Verlauf

der prädizierten Zustandsgröße x (grün) über den Prädiktionshorizont T_p für den ersten Reglerschritt zum Zeitpunkt t_0 . Die unteren Schaubilder zeigen die Situation nach Anwendung der optimierten Stellgröße auf das System und Ablauf eines Abtastschritts. Der tatsächliche Verlauf der Zustandsgröße (grün) weicht vom prädizierten Verlauf (grün gepunktet) ab. Basierend auf dem tatsächlich erreichten Zustand x_1 werden ein neuer optimaler Verlauf der Stellgröße (blau) und der dazugehörige prädizierte Verlauf der Zustandsgröße (grün) für den Prädiktionshorizont T_p berechnet.



(a) Erster Abtastschritt.



(b) Zweiter Abtastschritt.

Abbildung 5.3: Prinzip des gleitenden Horizonts der modellprädiktiven Regelung eines Systems mit Zustandsgröße x (grün) und Stellgröße u (blau).

5.2.2 Formulierung des NMPC-Optimierungsproblems

Der Entwurf nichtlinearer modellprädiktiver Regelungen ist sowohl für abgetastete kontinuierliche als auch für zeitdiskrete Strecken möglich. Roboter stellen abgetastete kontinuierliche Systeme dar.

Die zu regelnde Strecke wird allgemein durch das nichtlineare Zustandsmodell

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \quad (5.6)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \quad (5.7)$$

mit $\mathbf{x}(0) = \mathbf{x}_0$ beschrieben. Hierbei ist $\mathbf{x} \in \mathbb{R}^{N_x}$ der Zustandsvektor, $\mathbf{u} \in \mathbb{R}^{N_u}$ der Stellgrößenvektor und $\mathbf{y} \in \mathbb{R}^{N_y}$ der Ausgangsgrößenvektor. Für das System gelten die Beschränkungen der Stell- und Zustandsgrößen

$$\mathbf{c}(\mathbf{u}(t), \mathbf{x}(t)) \leq \mathbf{0} .^1 \quad (5.8)$$

Bei einem abgetasteten System wird das Stellsignal $\mathbf{u}(t)$ zu diskreten Zeitpunkten berechnet. Für einen Abtastschritt zum Zeitpunkt t_0 ist das auf das System aufgeschaltete Stellsignal die Lösung eines Optimalsteuerproblems der Form

$$\min_{\mathbf{u}(\cdot)} \int_{t_0}^{t_0+T_p} F(\mathbf{u}(t), \mathbf{x}(t)) dt + E(\mathbf{x}(t_0 + T_p)) \quad (5.9)$$

unter den Nebenbedingungen

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \forall t \in [t_0, t_0 + T_p] \quad (5.10)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (5.11)$$

$$\mathbf{c}(\mathbf{u}(t), \mathbf{x}(t)) \leq \mathbf{0}, \quad \forall t \in [t_0, t_0 + T_p] . \quad (5.12)$$

1. Vergleichsoperatoren für Vektoren werden in dieser Arbeit komponentenweise ausgewertet.

Das Gütekriterium des Optimierungsproblems setzt sich aus einem Integralanteil F , der über den gesamten Prädiktionshorizont T_p ausgewertet wird, und der Endzustandsbewertung E zusammen. Oft werden für den Integralanteil quadratische Terme der Stell- und Zustandsgrößen verwendet. Die Endzustandsbewertung, auch Mayer-Term genannt, bewertet den prädizierten Systemzustand am Ende des Prädiktionshorizonts. Sie dient der Stabilisierung des Regelkreises [Fin04]. Dies ist notwendig, da nur ein endlicher Prädiktionshorizont betrachtet wird. Die Nebenbedingungen des Optimierungsproblems dienen zur Einhaltung des Systemmodells (5.10) und der Beschränkungen der Stell-, und Zustandsgrößen (5.12). Dabei ist \mathbf{x}_0 der gemessene oder geschätzte Systemzustand zum Zeitpunkt t_0 .

In der Praxis wird statt des kontinuierlichen NMPC-Optimierungsproblems in (5.9)–(5.12) meist das diskretisierte Optimierungsproblem gelöst. Die Lösung des diskretisierten Optimierungsproblems stellt lediglich eine Approximation der Lösung des kontinuierlichen Optimierungsproblems dar. Das diskretisierte Optimierungsproblem hat jedoch den Vorteil, dass lediglich eine finite Anzahl von Optimierungsvariablen ermittelt werden muss und zur Lösung gängige Verfahren der statischen Optimierung eingesetzt werden können.

5.2.3 Diskretisiertes NMPC-Optimierungsproblem

Das diskretisierte NMPC-Optimierungsproblem beruht auf dem zeitdiskreten nichtlinearen Zustandsmodell des zu regelnden Systems

$$\mathbf{x}(k + 1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad (5.13)$$

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k)) , \quad (5.14)$$

wobei $\mathbf{x} \in \mathbb{R}^{N_x}$ der Zustandsvektor, $\mathbf{u} \in \mathbb{R}^{N_u}$ der Stellgrößenvektor und $\mathbf{y} \in \mathbb{R}^{N_y}$ der Ausgangsgrößenvektor ist. Der Anfangszustand ist $\mathbf{x}(0) = \mathbf{x}_0$. Der Zeitindex k gibt den Abtastzeitpunkt an. Für das System gelten die Beschränkungen der Stell- und Zustandsgrößen

$$\mathbf{c}(\mathbf{u}(k), \mathbf{x}(k + 1)) \leq \mathbf{0} \quad (5.15)$$

für alle $k \geq 0$.

In jedem Reglerabtastschritt berechnet die Regelung eine optimale Folge der Stell- und Zustandsgrößen \mathcal{U}^* und \mathcal{X}^* für den gesamten Prädiktionshorizont mit N_p Abtastschritten mit

$$\mathcal{U} := \{\mathbf{u}(0), \dots, \mathbf{u}(N_p - 1)\} \quad (5.16)$$

$$\mathcal{X} := \{\mathbf{x}(1), \dots, \mathbf{x}(N_p)\} . \quad (5.17)$$

Für eine übersichtliche Darstellung wird davon ausgegangen, dass $k = 0$ den Anfangszeitpunkt des aktuellen Reglerabtastschritts bestimmt.

Das NMPC-Optimierungsproblem eines Reglerabtastschritts ist die Minimierung des Gütekriteriums J

$$\min_{\mathcal{U}, \mathcal{X}} J(\mathcal{U}, \mathcal{X}) \quad (5.18)$$

unter den Nebenbedingungen

$$\mathbf{x}(k + 1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) , \text{ für } k = 0, \dots, N_p - 1 \quad (5.19)$$

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (5.20)$$

$$\mathbf{c}(\mathbf{u}(k), \mathbf{x}(k + 1)) \leq \mathbf{0} , \text{ für } k = 0, \dots, N_p - 1 . \quad (5.21)$$

Das Gütekriterium

$$J(\mathcal{U}, \mathcal{X}) := \sum_{k=0}^{N_p-1} F(\mathbf{u}(k), \mathbf{x}(k + 1)) + E(\mathbf{x}(N_p)) \quad (5.22)$$

setzt sich wie im kontinuierlichen Fall aus einer Kostenfunktion F , die für alle Prädiktionsschritte ausgewertet wird, und der Endzustandsbewertung E zusammen. Die Nebenbedingungen garantieren die Einhaltung des diskreten Systemmodells und der Beschränkungen der Stell- und Zustandsgrößen. Dabei ist \mathbf{x}_0 der gemessene oder geschätzte Systemzustand des aktuellen Abtastzeitpunkts.

Die optimale Stellgröße des ersten Prädiktionsschritts $\mathbf{u}^*(0)$ wird auf das System angewendet. Dann wird die Optimierung basierend auf den neuen gemessenen oder geschätzten Zustandsgrößen wiederholt.

Das diskretisierte NMPC-Optimierungsproblem stellt ein nichtkonvexes und nichtlineares Optimierungsproblem (engl. *Nonlinear Programming*, NLP) dar. Mögliche Ansätze zur Lösung des Optimierungsproblems werden im folgenden Absatz dargestellt.

5.2.4 Lösung nichtlinearer Optimierungsprobleme mit Beschränkungen

Das nichtlineare Optimierungsproblem mit Beschränkungen lässt sich allgemein darstellen als die Minimierung des Gütekriteriums J

$$\min_{\mathbf{z} \in \mathbb{R}^{N_z}} J(\mathbf{z}) \quad (5.23)$$

unter den Nebenbedingungen

$$\mathbf{c}_i(\mathbf{z}) = 0, \quad i \in \mathcal{E} \quad (5.24)$$

$$\mathbf{c}_i(\mathbf{z}) \leq 0, \quad i \in \mathcal{I}. \quad (5.25)$$

Dabei ist $\mathbf{z} \in \mathbb{R}^{N_z}$ der Optimierungsvektor. Im Fall der NMPC setzt er sich aus den Folgen der Zustands- und Stellgrößen zusammen. Die Nebenbedingungen werden in Gleichheitsbeschränkungen ($i \in \mathcal{E}$) und Ungleichheitsbeschränkungen ($i \in \mathcal{I}$) unterteilt. Im Fall der NMPC entsprechen die Gleichheitsbeschränkungen den Beschränkungen zur Einhaltung des Systemmodells (5.19). Es wird hier vorausgesetzt, dass sowohl das Gütekriterium als auch die Nebenbedingungen zweimal stetig differenzierbare reellwertige Funktionen sind.

Aktuell sind zwei vielversprechende Ansätze zur Lösung derartiger Optimierungsprobleme die Methode der sequentiellen quadratischen Programmierung (engl. *Sequential Quadratic Programming*, SQP) und die Methode der inneren Punkte (engl. *Interior Point Method*) [Noc06]. Beide Ansätze wer-

den im Folgenden kurz vorgestellt. Für weitere Details sei auf [Noc06; Die02; Wäc06] verwiesen.

Sequentielle Quadratische Programmierung

Die Lösung des nichtlinearen Optimierungsproblems mittels SQP beruht auf den Karush-Kuhn-Tucker-Bedingungen (KKT). Die KKT-Bedingungen stellen notwendige Bedingungen für den Fall dar, dass \mathbf{z}^* eine lokale Lösung des Optimierungsproblems (5.23)–(5.25) ist. Sind J und c_i kontinuierlich differenzierbar und die Gradienten der aktiven Beschränkungen linear unabhängig, dann existiert ein Lagrange-Faktor $\boldsymbol{\lambda}^*$ mit den Komponenten λ_i^* ($i \in \mathcal{E} \cup \mathcal{I}$), sodass folgende Bedingungen für \mathbf{z}^* und $\boldsymbol{\lambda}^*$ erfüllt sind:

$$\nabla_{\mathbf{z}} L(\mathbf{z}^*, \boldsymbol{\lambda}^*) = 0 \quad (5.26)$$

$$c_i(\mathbf{z}) = 0, \quad \forall i \in \mathcal{E} \quad (5.27)$$

$$c_i(\mathbf{z}) \leq 0, \quad \forall i \in \mathcal{I} \quad (5.28)$$

$$\lambda_i^* \geq 0, \quad \forall i \in \mathcal{I} \quad (5.29)$$

$$\lambda_i^* c_i(\mathbf{z}) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I} \quad (5.30)$$

Hierbei gibt L die Lagrange-Funktion

$$L(\mathbf{z}, \boldsymbol{\lambda}) := J(\mathbf{z}) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\mathbf{z}) \quad (5.31)$$

an.

SQP-Methoden approximieren das nichtlineare Optimierungsproblem iterativ durch ein quadratisches Optimierungsproblem (engl. *Quadratic Programming*, QP). Es wird eine Folge von Optimierungsvektoren (\mathbf{z}_k) mit

$$\mathbf{z}_{k+1} = \mathbf{z}_k + a_k \boldsymbol{\delta}_k^* \quad (5.32)$$

bestimmt, die gegen die Lösung des Optimierungsproblems \mathbf{z}^* konvergieren soll. $\boldsymbol{\delta}_k^*$ gibt die Suchrichtung an, a_k die Schrittweite. Die Suchrichtung $\boldsymbol{\delta}_k^*$

ist die Lösung des QP-Problems

$$\delta_k^* = \arg \min_{\delta_k} \left\{ J(\mathbf{z}_k) + \nabla J(\mathbf{z}_k)^\top \delta_k + \frac{1}{2} \delta_k^\top \nabla_{zz}^2 L(\mathbf{z}_k, \lambda_k) \delta_k \right\} \quad (5.33)$$

mit den Beschränkungen

$$\nabla c_i(\mathbf{z}_k)^\top \delta_k + c_i(\mathbf{z}_k) = 0, \quad i \in \mathcal{E} \quad (5.34)$$

$$\nabla c_i(\mathbf{z}_k)^\top \delta_k + c_i(\mathbf{z}_k) \geq 0, \quad i \in \mathcal{I}. \quad (5.35)$$

Die Schrittweite a_k wird basierend auf einer Bewertungsfunktion gewählt. Die Bewertungsfunktion setzt sich aus der Kostenfunktion und den Nebenbedingungen zusammen. Sie berücksichtigt die oft entgegengesetzten Ziele der Minimierung der Kostenfunktion und der Einhaltung der Beschränkungen.

Für die Lösung des QP-Problems gibt es mehrere bekannte Methoden. Eine häufig verwendeter Ansatz ist die *Active Set Method*. Hierbei werden, ausgehend von einer zulässigen Lösung, die aktiven Beschränkungen bestimmt. Mit den aktiven Beschränkungen wird eine Folge von zulässigen Punkten bestimmt, die zur Lösung konvergiert. Verletzt diese Lösung eine der zuvor nicht aktiven Beschränkungen, wird die Menge der aktiven Beschränkungen aktualisiert und das Verfahren iterativ fortgesetzt.

SQP-Methoden sind am effizientesten, wenn die Anzahl der aktiven Beschränkungen nahezu so groß ist wie die Anzahl der Optimierungsvariablen, d. h. wenn es wenige freie Variablen gibt. Sie sind vergleichsweise robust gegenüber schlecht skalierten Optimierungsproblemen [Noc06].

Methoden der inneren Punkte

Für die Methode der inneren Punkte werden die Ungleichheitsbedingungen (5.25) in Gleichheitsbedingungen

$$\mathbf{c}_I(\mathbf{z}) - \mathbf{s} = \mathbf{0} \quad (5.36)$$

$$\mathbf{s} \geq \mathbf{0} \quad (5.37)$$

mit dem Vektor der Schlupfvariablen \mathbf{s} umformuliert. Hierzu enthält im Folgenden $\mathbf{c}_I(\mathbf{z})$ die N_I Ungleichheitsbedingungen $c_i(\mathbf{z})$ mit $i \in \mathcal{I}$ und $\mathbf{c}_E(\mathbf{z})$ die N_E Gleichheitsbedingungen $c_i(\mathbf{z})$ mit $i \in \mathcal{E}$.

Statt des Optimierungsproblems (5.23)–(5.25) wird das sogenannte Barrier-Problem

$$\min_{\mathbf{z}, \mathbf{s}} J(\mathbf{z}) - \mu \sum_{i=1}^{N_I} \ln s_i \quad (5.38)$$

unter den Nebenbedingungen

$$\mathbf{c}_E(\mathbf{z}) = \mathbf{0} \quad (5.39)$$

$$\mathbf{c}_I(\mathbf{z}) - \mathbf{s} = \mathbf{0} . \quad (5.40)$$

gelöst. Durch Berücksichtigung der Schlupfvariablen im Gütekriterium kann auf die Ungleichheitsbedingung $\mathbf{s} \geq \mathbf{0}$ verzichtet werden, sodass das zu lösende Barrier-Problem nur noch Gleichheitsnebenbedingungen besitzt. μ ist ein positiver Parameter, der sogenannte Barrier-Parameter. Je kleiner μ ist, desto besser approximiert die Lösung des Barrier-Problems die Lösung des ursprünglichen NLP-Problems. Der Barrier-Ansatz besteht darin, das Barrier-Problem mit einer Folge von kleiner werdenden μ zu lösen und so die Approximation der Lösung des ursprünglichen NLP-Problems zu verbessern.

Die KKT-Bedingungen des Barrier-Problems lauten

$$\nabla_{\mathbf{z}} J(\mathbf{z}) - \boldsymbol{\kappa}^T \mathbf{J}_E(\mathbf{z}) - \boldsymbol{\lambda}^T \mathbf{J}_I(\mathbf{z}) = \mathbf{0} \quad (5.41)$$

$$-\mu \mathbf{e} + \mathbf{S} \boldsymbol{\lambda} = \mathbf{0} \quad (5.42)$$

$$\mathbf{c}_E(\mathbf{z}) = \mathbf{0} \quad (5.43)$$

$$\mathbf{c}_I(\mathbf{z}) - \mathbf{s} = \mathbf{0} . \quad (5.44)$$

Hierbei sind \mathbf{J}_E und \mathbf{J}_I die Jacobi-Matrizen der Funktionen \mathbf{c}_E und \mathbf{c}_I und $\boldsymbol{\kappa}$ und $\boldsymbol{\lambda}$ sind ihre Lagrange-Multiplikatoren. \mathbf{S} ist eine Diagonalmatrix, deren Diagonalelemente durch die Schlupfvariablen \mathbf{s} gegeben sind. Es gilt $\mathbf{e} := \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}^T$.

Unter Anwendung der Newton-Methode ergibt sich das Gleichungssystem für die Suchrichtung $\boldsymbol{\delta} := \left[\boldsymbol{\delta}_z^\top \quad \boldsymbol{\delta}_s^\top \quad \boldsymbol{\delta}_\kappa^\top \quad \boldsymbol{\delta}_\lambda^\top \right]^\top$

$$\begin{bmatrix} \nabla_{zz}^2 L & \mathbf{0} & -\mathbf{J}_E^\top(\mathbf{z}) & -\mathbf{J}_I^\top(\mathbf{z}) \\ \mathbf{0} & \Lambda & \mathbf{0} & \mathbf{S} \\ \mathbf{J}_E(\mathbf{z}) & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_I(\mathbf{z}) & -\mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\delta}_z \\ \boldsymbol{\delta}_s \\ \boldsymbol{\delta}_\kappa \\ \boldsymbol{\delta}_\lambda \end{bmatrix} = - \begin{bmatrix} \nabla_z L \\ \mathbf{S}\boldsymbol{\lambda} - \boldsymbol{\mu}\mathbf{e} \\ \mathbf{c}_E(\mathbf{z}) \\ \mathbf{c}_I(\mathbf{z}) - \mathbf{s} \end{bmatrix} \quad (5.45)$$

mit der Lagrange-Funktion des ursprünglichen NLP-Problems

$$L(\mathbf{z}, \mathbf{s}, \boldsymbol{\kappa}, \boldsymbol{\lambda}) := J(\mathbf{z}) + \boldsymbol{\kappa}^\top \mathbf{c}_E(\mathbf{z}) + \boldsymbol{\lambda}^\top (\mathbf{c}_I(\mathbf{z}) - \mathbf{s}) . \quad (5.46)$$

Hierbei ist Λ eine Diagonalmatrix, deren Diagonalelemente durch den Vektor $\boldsymbol{\lambda}$ gegeben sind. Zur Lösung des Barrier-Problems werden iterativ neue Suchrichtungen und Schrittweiten berechnet und mit diesen die Variablen \mathbf{z} , \mathbf{s} , $\boldsymbol{\kappa}$ und $\boldsymbol{\lambda}$ aktualisiert.

Methoden der inneren Punkte sind in Anwendungen mit hochdimensionalen Optimierungsvektoren meist schneller als SQP-Methoden. Ein Nachteil liegt in der Empfindlichkeit in Bezug auf die Wahl der Initialisierungswerte. Bei schlecht skalierten Optimierungsproblemen verhalten sie sich weniger robust als SQP-Methoden [Noc06].

Für die NMPC kommen beide Lösungsansätze in Frage. In Bezug auf hochdimensionale Optimierungsvektoren, die bei der NMPC häufig auftreten, bietet die Methode der inneren Punkte jedoch Vorteile. Bei Folgeproblemen wie Trajektorien- oder Pfadverfolgung können meist gute Initialisierungswerte gewählt werden, sodass die Empfindlichkeit in Bezug auf die Startwerte keine große Rolle spielt. Wichtig ist jedoch, dass der Optimierungsalgorithmus die spezielle Struktur der NMPC-Optimierungsprobleme – der Gradient der Nebenbedingungen ist eine dünnbesetzte Blockmatrix – für effiziente Berechnungen ausnutzt. Deshalb wird in dieser Arbeit ein Verfahren der inneren Punkte eingesetzt, das die Struktur der NMPC-Optimierungsprobleme ausnutzt [Wäc06].

5.3 Kartesische modellprädiktive Regelung zur Verfolgung von Solltrajektorien der Endeffektorpose

Um das in Abschnitt 5.1.1 dargestellte Problem der Verfolgung von kartesischen Solltrajektorien zu lösen, wird eine nichtlineare modellprädiktive Regelung entworfen [Zub15a]. Die Regelung basiert auf dem diskretisierten Robotermodell (4.13)–(4.15). Hierbei wird jedoch der Einfluss der Gelenkbeschleunigungen auf die Gelenkgeschwindigkeiten vernachlässigt, da typische Roboter (sowohl Manipulatoren als auch mobile Plattformen) keine Beschleunigungen, sondern Geschwindigkeiten als Stellgrößen verwenden. Die Vernachlässigung der Beschleunigung bietet den Vorteil, dass die Ordnung des Optimierungsproblems und damit auch der Rechenaufwand zur Lösung des Optimierungsproblems reduziert werden. Des Weiteren zeigt die Anwendung, dass durch die Vernachlässigung der Beschleunigung keine Probleme in der Regelung auftreten. Somit ist das Modell des zu regelnden Systems

$$\mathbf{q}(k+1) = \mathbf{q}(k) + T_s \mathbf{w}(k) \quad (5.47)$$

$$\mathbf{y}_{EE}(k) = \mathbf{k}(\mathbf{q}(k)) \quad (5.48)$$

mit Gelenkpositionen \mathbf{q} (Zustandsgrößen), Gelenkgeschwindigkeiten \mathbf{w} (Stellgrößen), Endeffektorpose \mathbf{y}_{EE} (Ausgangsgröße) und Abtastzeit T_s .

Die Regelung berechnet optimale Werte für die Folgen der Gelenkpositionen und -geschwindigkeiten

$$\mathcal{W}^* := \{\mathbf{w}^*(0), \dots, \mathbf{w}^*(N_p - 1)\} \quad (5.49)$$

$$\mathcal{Q}^* := \{\mathbf{q}^*(1), \dots, \mathbf{q}^*(N_p)\} \quad (5.50)$$

der nächsten N_p Abtastschritte. Die dabei zu berücksichtigenden Nebenbedingungen und das zugrunde gelegte Gütekriterium werden in den folgenden Abschnitten erläutert.

5.3.1 Nebenbedingungen

Zur Einhaltung des Robotermodells müssen bei der Optimierung die Gleichheitsbeschränkungen

$$\mathbf{q}(k+1) - \mathbf{q}(k) - T_s \mathbf{w}(k) = 0 \quad (5.51)$$

für $k = 0, \dots, N_p - 1$ berücksichtigt werden. Besitzt das kinematische Robotermodell zusätzlich nicht-holonome Beschränkungen (vgl. Abschnitt 4.2.3), bilden diese ebenfalls Gleichheitsbeschränkungen für die Optimierung.

Des Weiteren müssen die Gelenkbeschränkungen des Roboters, gegeben durch die minimalen und maximalen Gelenkpositionen \mathbf{q}_{\min} und \mathbf{q}_{\max} , durch den maximalen Betrag der Gelenkgeschwindigkeiten \mathbf{w}_{\max} sowie durch den maximalen Betrag der Gelenkbeschleunigungen \mathbf{a}_{\max} , eingehalten werden. Dies wird durch die Ungleichheitsbeschränkungen

$$\mathbf{q}_{\min} \leq \mathbf{q}(k+1) \leq \mathbf{q}_{\max} \quad (5.52)$$

$$\tilde{\mathbf{w}}_{\min}(k) \leq \mathbf{w}(k) \leq \tilde{\mathbf{w}}_{\max}(k) \quad (5.53)$$

für $k = 0, \dots, N_p - 1$ mit

$$\tilde{\mathbf{w}}_{\min}(k) := \max(-\mathbf{w}_{\max}, \mathbf{w}_0 - kT_s \mathbf{a}_{\max}) \quad (5.54)$$

$$\tilde{\mathbf{w}}_{\max}(k) := \min(\mathbf{w}_{\max}, \mathbf{w}_0 + kT_s \mathbf{a}_{\max}) \quad (5.55)$$

erreicht. Die aktuellen Gelenkgeschwindigkeiten zu Beginn des Reglerschritts sind hierbei durch \mathbf{w}_0 gegeben. Die Bestimmung der in einem Abtastschritt zulässigen minimalen und maximalen Geschwindigkeiten in (5.54) und (5.55) in Abhängigkeit der aktuellen Geschwindigkeit garantiert für den ersten Abtastschritt neben der Einhaltung der Geschwindigkeitsbeschränkungen auch die Einhaltung der Beschleunigungsbeschränkungen. Für alle weiteren Schritte kann die Einhaltung der Beschleunigungsbeschränkungen nicht garantiert werden, es wird lediglich eine optimistische Abschätzung getroffen. Dies ist jedoch unproblematisch, da die berechne-

ten Geschwindigkeiten nur für den ersten Abtastschritt auf den Roboter angewendet werden und dann eine neue Optimierung durchgeführt wird.

Die Wahl der Beschränkungen garantiert somit das Einhalten der kinematischen Eigenschaften des Roboters.

5.3.2 Gütekriterium

Um die eigentliche Aufgabe – das Verfolgen einer kartesischen Solltrajektorie $\mathbf{y}_{\text{traj}}(k)$ – zu erfüllen, wird ein geeignetes Gütekriterium

$$J(\mathcal{W}, \mathcal{Q}) := \sum_{k=0}^{N_p-1} F_K(\mathbf{w}(k), \mathbf{q}(k+1)) + E_K(\mathbf{q}(N_p)) \quad (5.56)$$

gewählt.

Die Hauptanforderung an das Gütekriterium ergibt sich aus dem Trajektorienverfolgungsproblem: Die Minimierung des Gütekriteriums muss eine Minimierung des Trajektorienverfolgungsfehlers \mathbf{e}_{traj} (siehe Abschnitt 5.1.1) bewirken. Da die Endeffektorpose des Roboters \mathbf{y}_{EE} keine (messbare) Zustandsgröße des Systems darstellt, wird zur Bestimmung des Trajektorienverfolgungsfehlers statt der tatsächlichen Endeffektorpose die auf Grundlage der direkten Kinematik ermittelte Endeffektorpose verwendet:

$$\mathbf{e}_{\text{traj}}(\mathbf{q}(k)) := \begin{bmatrix} \mathbf{p}_{\text{EE}}(k) - \mathbf{p}_{\text{traj}}(k) \\ \mathbf{o}_{\text{EE}}(k) - \mathbf{s}^*(k) \cdot \mathbf{o}_{\text{traj}}(k) \end{bmatrix} \quad (5.57)$$

$$= \mathbf{k}(\mathbf{q}(k)) - \begin{bmatrix} \mathbf{p}_{\text{traj}}(k) \\ \mathbf{s}^*(k) \cdot \mathbf{o}_{\text{traj}}(k) \end{bmatrix}. \quad (5.58)$$

Weitere grundlegende Aufgaben des Gütekriteriums sind die Vermeidung von Schwingungen der Stell- und Zustandsgrößen, sowie damit verbunden die Vermeidung unnötig hoher Stellenergie.

Diese Anforderungen können durch eine Vielzahl unterschiedlicher Gütekriterien erfüllt werden. Für eine zeiteffiziente Optimierung des Gütekriteriums werden jedoch üblicherweise quadratische Terme gewählt. Diese

entsprechen dem gewünschten Verhalten der Kostenfunktion, sind zweimal stetig differenzierbar und führen zu dünnbesetzten Hessematrizen des Gütekriteriums. Somit ist eine geeignete Wahl für die Kostenfunktion des Gütekriteriums

$$F_K(\mathbf{w}, \mathbf{q}) := \mathbf{e}_{\text{traj}}^\top(\mathbf{q}) \mathbf{Q}_e \mathbf{e}_{\text{traj}}(\mathbf{q}) + (\mathbf{w} - \beta \mathbf{w}_0)^\top \mathbf{Q}_w (\mathbf{w} - \beta \mathbf{w}_0). \quad (5.59)$$

Hierbei sind \mathbf{Q}_w und \mathbf{Q}_e positiv semidefinite Diagonalmatrizen, die zur Gewichtung der einzelnen quadratischen Terme dienen. Zum einen bestraft die Kostenfunktion den Trajektorienverfolgungsfehler, um die kartesische Aufgabe zu erfüllen. Zum anderen werden durch die Berücksichtigung der Gelenkgeschwindigkeitsänderung $\mathbf{w} - \beta \mathbf{w}_0$ Oszillationen der Gelenkgeschwindigkeiten und damit auch der Gelenkpositionen reduziert. Hierbei ist \mathbf{w}_0 der Vektor der aktuellen Gelenkgeschwindigkeiten. β stellt einen Dämpfungsfaktor im Bereich $0 \leq \beta < 1$ dar. Mit β nahe Eins bevorzugt die Optimierung des Gütekriteriums das Beibehalten der aktuellen (nur leicht reduzierten) Geschwindigkeit. Mit β nahe Null werden generell kleine Geschwindigkeiten bevorzugt.

Um Gelenkbewegungen zu bevorzugen, die sich von den Gelenkbegrenzungen fernhalten und zu jedem Zeitpunkt größtmögliche Änderungen in beide Bewegungsrichtungen ermöglichen, wird die Kostenfunktion um einen Term erweitert, der große Abweichungen vom Mittelpunkt des zulässigen Gelenkbereichs

$$\mathbf{q}_m := \frac{\mathbf{q}_{\max} - \mathbf{q}_{\min}}{2} \quad (5.60)$$

bestraft. Insgesamt ergibt sich dann für die Kostenfunktion

$$F_K(\mathbf{w}, \mathbf{q}) := \mathbf{e}_{\text{traj}}^\top(\mathbf{q}) \mathbf{Q}_e \mathbf{e}_{\text{traj}}(\mathbf{q}) + (\mathbf{w} - \beta \mathbf{w}_0)^\top \mathbf{Q}_w (\mathbf{w} - \beta \mathbf{w}_0) + (\mathbf{q} - \mathbf{q}_m)^\top \mathbf{Q}_q (\mathbf{q} - \mathbf{q}_m) \quad (5.61)$$

wobei \mathbf{Q}_q ebenfalls eine positiv semidefinite Diagonalmatrix ist.

Analog zur Kostenfunktion werden auch für die Endzustandsbewertung quadratische Terme gewählt.

$$E_K(\mathbf{q}) := (\mathbf{q} - \mathbf{q}_m)^\top \mathbf{R}_q (\mathbf{q} - \mathbf{q}_m) + \mathbf{e}_{\text{traj}}^\top(\mathbf{q}) \mathbf{R}_e \mathbf{e}_{\text{traj}}(\mathbf{q}) \quad (5.62)$$

Hierbei sind \mathbf{R}_q und \mathbf{R}_e positiv semidefinite Diagonalmatrizen.

5.3.3 Wahl der Startwerte

Die Wahl der Startwerte für die zu optimierenden Größen hat einen großen Einfluss darauf, wie viele Iterationen notwendig sind, um eine Lösung des Optimierungsproblems zu finden. Die zu optimierenden Größen sind hier die Gelenkgeschwindigkeiten und die Gelenkpositionen der nächsten N_p Abtastschritte. Aufgrund des Prinzips der NMPC, optimale Stell- und Zustandsgrößen für den gesamten Prädiktionshorizont zu berechnen, jedoch bereits nach Ausführen des ersten Prädiktionsschritts eine neue Optimierung durchzuführen, liegen für die meisten benötigten Startwerte bereits gute Werte aus der vorherigen Optimierung vor. Es muss dann lediglich eine Korrektur aufgrund von Modellierungsungenauigkeiten, des verschobenen Prädiktionshorizonts und sich gegebenenfalls ändernder Rahmenbedingungen durchgeführt werden. Dementsprechend werden die Startwerte des aktuellen Optimierungsproblems

$$\mathcal{W}_{\text{start}} := \{\mathbf{w}_{\text{start}}(0), \dots, \mathbf{w}_{\text{start}}(N_p - 1)\} \quad (5.63)$$

$$\mathcal{Q}_{\text{start}} := \{\mathbf{q}_{\text{start}}(1), \dots, \mathbf{q}_{\text{start}}(N_p)\} \quad (5.64)$$

auf Grundlage des Optimierungsergebnisses des letzten Reglerschritts

$$\mathcal{W}^* := \{\mathbf{w}^*(0), \dots, \mathbf{w}^*(N_p - 1)\} \quad (5.65)$$

$$\mathcal{Q}^* := \{\mathbf{q}^*(1), \dots, \mathbf{q}^*(N_p)\} \quad (5.66)$$

wie folgt gewählt:

$$\mathcal{W}_{\text{start}} = \{\mathbf{w}^*(1), \dots, \mathbf{w}^*(N_p - 1), \mathbf{w}^*(N_p - 1)\} \quad (5.67)$$

$$\mathcal{Q}_{\text{start}} = \{\mathbf{q}^*(2), \dots, \mathbf{q}^*(N_p), \mathbf{q}^*(N_p) + T_s \mathbf{q}^*(N_p - 1)\} . \quad (5.68)$$

Da noch keine Information über geeignete Werte für den letzten Prädiktionsschritt des neuen Optimierungsproblems vorliegt, wird für diesen Prädiktionsschritt eine Bewegung mit der gleichen Geschwindigkeit wie im vorherigen Prädiktionsschritt als Startwert verwendet.

Im ersten Reglerschritt liegen ebenfalls noch keine Informationen aus vorherigen Optimierungen vor, weshalb in diesem Fall die Startwerte zu $\mathbf{w}_{\text{start}}(k) = \mathbf{0}$ und $\mathbf{q}_{\text{start}}(k + 1) = \mathbf{q}_0$ für $k = 0, \dots, N_p - 1$ gewählt werden. \mathbf{q}_0 sind dabei die aktuellen Gelenkpositionen, sodass diese Wahl einem Stillstehen des Roboters entspricht.

5.4 Kartesische modellprädiktive Regelung zur Verfolgung von Sollpfaden der Endeffektorpose

Für die Verfolgung von geometrischen Sollpfaden der Endeffektorpose $y_{\text{pfad}}(\theta)$ ohne gegebene Zeitabhängigkeit (siehe Abschnitt 5.1.2) mittels der kartesischen modellprädiktiven Regelung werden zwei Ansätze untersucht. Im ersten Ansatz wird die zeitliche Entwicklung des Pfadparameters $\theta(t)$ unter Verwendung der gegebenen maximal erlaubten kartesischen Geschwindigkeiten des Endeffektors bestimmt. Im zweiten Ansatz wird die zeitliche Entwicklung des Pfadparameters als zusätzlicher Freiheitsgrad der Regelung gemeinsam mit den Stell- und Zustandsgrößen der Regelung optimiert. Diese beiden Ansätze werden im Folgenden vorgestellt, ein Vergleich folgt in Abschnitt 8.2.2.

5.4.1 Verwendung der maximalen kartesischen Geschwindigkeiten

Die Idee dieses Ansatzes besteht darin, den Sollpfad in kürzest möglicher Zeit bei gegebener maximaler translatorischer und rotatorischer Geschwindigkeit abzufahren. Hierzu wird in jedem Reglerabtastschritt die größte Pfadparametergeschwindigkeit $\dot{\theta}$ ermittelt, die weder die maximale translatorische noch die maximale rotatorische Geschwindigkeit überschreitet.

Der Betrag der translatorischen Pfadgeschwindigkeit ergibt sich aus der Ableitung der Sollposition nach der Zeit

$$v_{\text{pfad}} := \left\| \frac{d}{dt} \mathbf{p}_{\text{pfad}}(\theta) \right\| = \left\| \frac{\partial \mathbf{p}_{\text{pfad}}(\theta)}{\partial \theta} \cdot \dot{\theta} \right\|. \quad (5.69)$$

Da $\dot{\theta} > 0$ gefordert wird, ist die Pfadparametergeschwindigkeit, die zur gewünschten translatorischen Geschwindigkeit $v_{\text{pfad}} = v_{\text{pfad,max}}$ führt

$$\dot{\theta}_{\text{max,transl}}(\theta) = v_{\text{pfad}} \left\| \frac{\partial \mathbf{p}_{\text{pfad}}(\theta)}{\partial \theta} \right\|^{-1}. \quad (5.70)$$

Die Pfadorientierung ist durch die Einheitsquaternion $\mathbf{o}_{\text{pfad}}(\theta)$ gegeben. Die maximal erlaubte rotatorische Geschwindigkeit liegt jedoch üblicherweise als Winkelgeschwindigkeit vor. Um den Zusammenhang zwischen der Pfadparametergeschwindigkeit und der resultierenden Winkelgeschwindigkeit zu erhalten, wird die Winkelgeschwindigkeit mit Hilfe des Winkels zwischen zwei durch Quaternionen dargestellten Orientierungen approximiert. Der Winkel zwischen den durch die Quaternion $\mathbf{o}_{\text{pfad}}(\theta)$ und die Quaternion $\mathbf{o}_{\text{pfad}}(\theta + \Delta\theta)$ beschriebenen Orientierungen beträgt

$$\alpha_{\text{pfad}} := \arccos \left(2 \langle \mathbf{o}_{\text{pfad}}(\theta), \mathbf{o}_{\text{pfad}}(\theta + \Delta\theta) \rangle^2 - 1 \right), \quad (5.71)$$

wobei $\langle \cdot, \cdot \rangle$ das Skalarprodukt zweier Quaternionen angibt. Die Winkelgeschwindigkeit der Pfadorientierung wird approximiert, indem die durch die Pfadparameteränderung $\Delta\theta$ bewirkte Winkeländerung α_{pfad} auf die Pfadpa-

parameteränderung bezogen wird:

$$\omega_{\text{pfad}} \approx \frac{\alpha_{\text{pfad}}}{\Delta\theta} \cdot \dot{\theta}. \quad (5.72)$$

Damit wird die gewünschte rotatorische Pfadgeschwindigkeit $\omega_{\text{pfad}} = \omega_{\text{pfad,max}}$ erreicht mit

$$\dot{\theta}_{\text{max,rot}}(\theta) = \omega_{\text{pfad}} \cdot \left| \frac{\arccos \left(2 \langle \mathbf{o}_{\text{pfad}}(\theta), \mathbf{o}_{\text{pfad}}(\theta + \Delta\theta) \rangle^2 - 1 \right)}{\Delta\theta} \right|^{-1}. \quad (5.73)$$

Um den Pfad mit größtmöglicher Geschwindigkeit abzufahren, ohne die maximal zulässige translatorische oder rotatorische Pfadgeschwindigkeit zu überschreiten, wird die Pfadparametergeschwindigkeit gewählt als

$$\dot{\theta}(\theta) = \min \left(\dot{\theta}_{\text{max,transl}}(\theta), \dot{\theta}_{\text{max,rot}}(\theta) \right). \quad (5.74)$$

Damit wird der Sollpfad in eine Solltrajektorie mit maximaler kartesischer Geschwindigkeit überführt. Diese Solltrajektorie wird wie in Abschnitt 5.1.1 als Sollgröße der Regelung verwendet.

Um ein für die Mensch-Roboter-Interaktion geeignetes und intuitiv verständliches Verhalten zu erzeugen, wird die Wahl der Pfadparametergeschwindigkeit im Folgenden um zwei Heuristiken ergänzt. Diese dienen zum einen dem Abbremsen am Pfadende sowie der Reduzierung der kartesischen Geschwindigkeit in der Nähe von Hindernissen.

Abbremsen am Pfadende

Am Pfadende wird ein rampenförmiges Abbremsen abhängig vom Abstand der aktuellen Pfadposition zur Endposition des Pfads bzw. des Winkels zwischen der aktuellen Pfadorientierung und der Orientierung am Pfadende eingeführt.

Der Abstand der aktuellen Pfadposition von der Endposition am Pfadende ist

$$d_{\text{end}} := \left\| \mathbf{p}_{\text{pfad}}(\theta) - \mathbf{p}_{\text{pfad}}(\theta_{\text{max}}) \right\|. \quad (5.75)$$

Abhängig von diesem Abstand wird die maximale translatorische Pfadgeschwindigkeit bei Unterschreiten des Schwellwerts d_a reduziert:

$$v_{\text{pfad}}(d_{\text{end}}) := \begin{cases} v_{\text{pfad,max}} & \text{für } d_{\text{end}} \geq d_a \\ v_{\text{pfad,max}} \frac{d_{\text{end}}}{d_a} & \text{für } d_a > d_{\text{end}} \geq 0. \end{cases} \quad (5.76)$$

Analog wird die maximale rotatorische Geschwindigkeit abhängig vom Winkel zwischen der aktuellen Pfadorientierung und der Orientierung am Pfadende

$$\alpha_{\text{end}} := \arccos \left(2 \langle \mathbf{o}_{\text{pfad}}(\theta), \mathbf{o}_{\text{pfad}}(\theta_{\text{max}}) \rangle^2 - 1 \right) \quad (5.77)$$

bei Unterschreiten des Schwellwerts α_a reduziert:

$$\omega_{\text{pfad}}(\alpha_{\text{end}}) := \begin{cases} \omega_{\text{pfad,max}} & \text{für } \alpha_{\text{end}} \geq \alpha_a \\ \omega_{\text{pfad,max}} \frac{\alpha_{\text{end}}}{\alpha_a} & \text{für } \alpha_a > \alpha_{\text{end}} \geq 0. \end{cases} \quad (5.78)$$

Durch das rampenförmige Abbremsen am Pfadende wird der Pfadverfolgungsfehler am Pfadende reduziert und ein Überschwingen vermieden.

Abstandsabhängige Geschwindigkeit nahe Hindernissen

Für die Mensch-Roboter-Interaktion ist es erwünscht, dass der Roboter seine kartesische Endeffektorgeschwindigkeit in der Nähe von dynamischen Hindernisse reduziert, um die Sicherheit von Menschen in seiner Umgebung nicht zu gefährden und ihr Wohlbefinden nicht einzuschränken. Um in der Nähe von dynamischen Hindernissen kleine kartesische Geschwindigkeiten zu erhalten, aber im freien Raum die Aufgabenausführung nicht zu verlangsamen, ist es zweckmäßig, die erlaubte translatorische und rotatorische Pfadgeschwindigkeit v_{pfad} und ω_{pfad} abhängig vom Abstand d zwischen dem Roboter und den dynamischen Hindernissen in seiner Umgebung zu wählen.

Der Abhängigkeit der Pfadgeschwindigkeiten vom Abstand können verschiedene Funktionen zugrunde gelegt werden. Es bieten sich z.B. lineare, quadratische oder exponentielle Zusammenhänge an. Im in dieser Arbeit analysierten Fall eines mobilen Manipulators zur Mensch-Roboter-Interaktion hat sich der folgende lineare Zusammenhang als geeignet erwiesen:

$$v_{\text{pfad}}(d) := \begin{cases} v_{\text{pfad,max}} & \text{für } d \geq d_o \\ v_{\text{pfad,min}} + \frac{v_{\text{pfad,max}} - v_{\text{pfad,min}}}{d_o - d_u} (d - d_u) & \text{für } d_o > d \geq d_u \\ v_{\text{pfad,min}} & \text{für } d_u > d > 0 \end{cases} \quad (5.79)$$

$$\omega_{\text{pfad}}(d) := \begin{cases} \omega_{\text{pfad,max}} & \text{für } d \geq d_o \\ \omega_{\text{pfad,min}} + \frac{\omega_{\text{pfad,max}} - \omega_{\text{pfad,min}}}{d_o - d_u} (d - d_u) & \text{für } d_o > d \geq d_u \\ \omega_{\text{pfad,min}} & \text{für } d_u > d > 0 \end{cases} \quad (5.80)$$

mit $0 \leq d_u < d_o$, $0 \leq v_{\text{pfad,min}} < v_{\text{pfad,max}}$ und $0 \leq \omega_{\text{pfad,min}} < \omega_{\text{pfad,max}}$. Hierbei stellen d_u und d_o untere und obere Schranken des Abstands dar. $v_{\text{pfad,min}}$ und $v_{\text{pfad,max}}$ sind die unteren und oberen Schranken der translatorischen Pfadgeschwindigkeit, $\omega_{\text{pfad,min}}$ und $\omega_{\text{pfad,max}}$ die der rotatorischen Pfadgeschwindigkeit.

Damit ist im freien Raum die Pfadgeschwindigkeit lediglich durch die vorgegebene translatorische und rotatorische Geschwindigkeit $v_{\text{pfad,max}}$ und $\omega_{\text{pfad,max}}$ begrenzt. Unterschreitet der Abstand zwischen Roboter und Hindernissen den Schwellwert d_o , werden die maximalen Geschwindigkeiten reduziert. Wird der Sicherheitsabstand d_u unterschritten, werden lediglich die Mindestgeschwindigkeiten $v_{\text{pfad,min}}$ und $\omega_{\text{pfad,min}}$ zugelassen. Typischerweise werden diese Mindestgeschwindigkeiten zu Null gewählt, es können aufgabenabhängig aber auch kleine Geschwindigkeiten größer Null sinnvoll sein.

5.4.2 Pfadoptimierung

Bei der Lösung des Pfadverfolgungsproblems stellt die zeitliche Entwicklung des Pfadparameters $\theta(t)$ einen zusätzlichen Freiheitsgrad beim Entwurf der

Regelung dar. Dieser zusätzliche Freiheitsgrad lässt sich bei einer modellprädiktiven Regelung nutzen, um eine optimale Entwicklung des Pfadparameters zu ermitteln [Fau12]. Hierzu wird zunächst das der Regelung zugrundeliegende Systemmodell um ein Modell der zeitlichen Entwicklung des Pfadparameters ergänzt. Hier wird analog zum kinematischen Robotermodell ein lineares Modell des Pfadparameters unter Verwendung der Pfadparametergeschwindigkeit $\dot{\theta}$ als Stellgröße verwendet:

$$\theta(k+1) = \theta(k) + T_s \dot{\theta}(k) \quad (5.81)$$

$\theta(0) = \theta_0$ ist hierbei der aktuelle Pfadparameter.

Die Aufgabe der Regelung besteht dann darin, neben der Folge der Stell- und Zustandsgrößen des Robotermodells \mathcal{W} und \mathcal{Q} auch die Folge der Stell- und Zustandsgrößen des Pfadparametermodells

$$\mathcal{V} := \{\dot{\theta}(0), \dots, \dot{\theta}(N_p - 1)\} \quad (5.82)$$

$$\mathcal{T} := \{\theta(1), \dots, \theta(N_p)\} \quad (5.83)$$

über den gesamten Prädiktionshorizont zu optimieren. Das Gütekriterium ist neben den Gelenkgeschwindigkeiten und -positionen auch vom Pfadparameter und dessen Geschwindigkeit abhängig.

$$J(\mathcal{W}, \mathcal{Q}, \mathcal{V}, \mathcal{T}) := \sum_{k=0}^{N_p-1} F_K(\mathbf{w}(k), \mathbf{q}(k+1), \dot{\theta}(k), \theta(k+1)) + E_K(\mathbf{q}(N_p), \theta(N_p)) \quad (5.84)$$

In den folgenden Abschnitten werden die Wahl des Gütekriteriums sowie die zu berücksichtigenden Nebenbedingungen dargestellt.

Nebenbedingungen

Bei der Minimierung des Gütekriteriums (5.84) muss zusätzlich zu den in Abschnitt 5.3.1 dargestellten Beschränkungen, die aus dem kinematischen Robotermodell resultieren, auch das Pfadparametermodell als Gleichheits-

beschränkung für $k = 0, \dots, N_p - 1$ berücksichtigt werden:

$$\theta(k+1) - \theta(k) - T_s \dot{\theta}(k) = 0 \quad (5.85)$$

Da der Pfadparameter so gewählt wird, dass sich der Pfadanzug bei θ_{\min} und das Pfadende bei θ_{\max} befindet, gelten für die Pfadparametergeschwindigkeit und den Pfadparameter für $k = 0, \dots, N_p - 1$ folgende Ungleichheitsbedingungen:

$$\dot{\theta}_{\min}(k) < \dot{\theta}(k) \leq \dot{\theta}_{\max}(k) \quad (5.86)$$

$$\theta_{\min} \leq \theta(k+1) \leq \theta_{\max} \quad (5.87)$$

Um ein Voranschreiten auf dem Pfad zu gewährleisten, muss $\dot{\theta}_{\min}(k) \geq 0$ gewählt werden. Die maximale Pfadparametergeschwindigkeit kann direkt als Parameter der Regelung vorgegeben sein. Typischerweise sind jedoch stattdessen die maximale translatorische und rotatorische Geschwindigkeit des Endeffektors gegeben. Dann wird die maximale Pfadparametergeschwindigkeit wie in Abschnitt 5.4.1 dargestellt aus den maximalen kartesischen Geschwindigkeiten ermittelt.

Gütekriterium

Das Gütekriterium zur Pfadverfolgung muss so gewählt werden, dass gleichzeitig die Abweichung der resultierenden Endeffektorpose von der Sollpose minimiert und das Erreichen des Pfadendes gewährleistet wird.

Dementsprechend wird der erste Teil der Kostenfunktion analog zur Kostenfunktion der Trajektorienverfolgung gewählt und der Pfadverfolgungsfehler

$$\mathbf{e}_{\text{pfad}}(\mathbf{q}(k), \theta(k)) := \mathbf{k}(\mathbf{q}(k)) - \begin{bmatrix} \mathbf{p}_{\text{pfad}}(\theta(k)) \\ \mathbf{s}^*(k) \cdot \mathbf{o}_{\text{pfad}}(\theta(k)) \end{bmatrix} \quad (5.88)$$

bestraft. Des Weiteren enthält er Terme zur Reduzierung von Schwingungen und zur Vermeidung von Gelenkstellungen nahe den Beschränkungen.

Im Vergleich zur Trajektorienverfolgung wird die Kostenfunktion um einen Term ergänzt, der den Abstand des Pfadparameters θ zum Wert

des Pfadparameters am Pfadende θ_{\max} bestraft. Das bedeutet, dass ein möglichst schnelles Annähern an das Pfadende bevorzugt wird. Eine konstante Gewichtung des quadratischen Pfadparameterabstands vom Pfadende $Q_\theta(\theta - \theta_{\max})^2$ mit $Q_\theta = \text{const}$ würde jedoch bewirken, dass zu Beginn ein schnelleres Voranschreiten bei größeren Pfadverfolgungsfehlern und zum Pfadende ein langsames Voranschreiten bei kleineren Pfadverfolgungsfehlern erzielt wird. Ein langsamer werdendes Annähern an das Pfadende ist jedoch erst in einem kleinen Abschnitt am Pfadende erwünscht. Deshalb wird eine Normierung des Terms auf den aktuellen Abstand des Pfadparameters vom Pfadende eingeführt:

$$Q_{\theta,N}(\theta_0) := \begin{cases} \frac{Q_\theta}{(\theta_0 - \theta_{\max})^2} & \text{für } |\theta_0 - \theta_{\max}| > 1 \\ Q_\theta & \text{sonst} \end{cases} \quad (5.89)$$

mit $Q_\theta > 0$.

Des Weiteren werden abhängig vom Dämpfungsparameter $\gamma \in [0,1)$ Abweichungen der Pfadparametergeschwindigkeit $\dot{\theta}$ von der aktuellen Pfadparametergeschwindigkeit $\dot{\theta}_0$ bestraft, um Schwingungen des Pfadparameters zu vermeiden.

Insgesamt ergibt sich für die Kostenfunktion zur Pfadverfolgung

$$\begin{aligned} F_K(\mathbf{w}, \mathbf{q}, \dot{\theta}, \theta) &:= (\mathbf{e}_{\text{pfad}}(\mathbf{q}, \theta))^\top \mathbf{Q}_e \mathbf{e}_{\text{pfad}}(\mathbf{q}, \theta) & (5.90) \\ &+ (\mathbf{w} - \beta \mathbf{w}_0)^\top \mathbf{Q}_w (\mathbf{w} - \beta \mathbf{w}_0) + (\mathbf{q} - \mathbf{q}_m)^\top \mathbf{Q}_q (\mathbf{q} - \mathbf{q}_m) \\ &+ Q_{\theta,N}(\theta_0)(\theta - \theta_{\max})^2 + Q_{\dot{\theta}}(\dot{\theta} - \gamma \dot{\theta}_0)^2 \end{aligned}$$

mit $Q_\theta > 0$ und $Q_{\dot{\theta}} \geq 0$.

Analog zur Kostenfunktion werden auch für die Endzustandsbewertung quadratische Terme der Form

$$\begin{aligned} E_K(\mathbf{q}, \theta) &:= (\mathbf{e}_{\text{pfad}}(\mathbf{q}, \theta))^\top \mathbf{R}_e \mathbf{e}_{\text{pfad}}(\mathbf{q}, \theta) + (\mathbf{q} - \mathbf{q}_m)^\top \mathbf{R}_q (\mathbf{q} - \mathbf{q}_m) & (5.91) \\ &+ R_{\theta,N}(\theta_0)(\theta - \theta_{\max})^2 \end{aligned}$$

gewählt mit

$$R_{\theta,N}(\theta_0) := \begin{cases} \frac{R_\theta}{(\theta_0 - \theta_{\max})^2} & \text{für } |\theta_0 - \theta_{\max}| > 1 \\ R_\theta & \text{sonst} \end{cases} \quad (5.92)$$

und $R_\theta > 0$.

Wahl der Startwerte

Analog zur Trajektorienverfolgung werden die Startwerte des Optimierungsproblems der Pfadverfolgung

$$\mathcal{W}_{\text{start}} := \{\mathbf{w}_{\text{start}}(0), \dots, \mathbf{w}_{\text{start}}(N_p - 1)\} \quad (5.93)$$

$$\mathcal{Q}_{\text{start}} := \{\mathbf{q}_{\text{start}}(1), \dots, \mathbf{q}_{\text{start}}(N_p)\} \quad (5.94)$$

$$\mathcal{V}_{\text{start}} := \{\dot{\theta}_{\text{start}}(0), \dots, \dot{\theta}_{\text{start}}(N_p - 1)\} \quad (5.95)$$

$$\mathcal{T}_{\text{start}} := \{\theta_{\text{start}}(1), \dots, \theta_{\text{start}}(N_p)\} \quad (5.96)$$

auf Grundlage des Optimierungsergebnisses des letzten Reglerschritts gewählt:

$$\mathcal{W}_{\text{start}} = \{\mathbf{w}^*(1), \dots, \mathbf{w}^*(N_p - 1), \mathbf{w}^*(N_p - 1)\} \quad (5.97)$$

$$\mathcal{Q}_{\text{start}} = \{\mathbf{q}^*(2), \dots, \mathbf{q}^*(N_p), \mathbf{q}^*(N_p) + T_s \dot{\mathbf{q}}^*(N_p - 1)\} \quad (5.98)$$

$$\mathcal{V}_{\text{start}} = \{\dot{\theta}^*(1), \dots, \dot{\theta}^*(N_p - 1), \dot{\theta}^*(N_p - 1)\} \quad (5.99)$$

$$\mathcal{T}_{\text{start}} = \{\theta^*(2), \dots, \theta^*(N_p), \theta^*(N_p) + T_s v_{\dot{\theta}}^*(N_p - 1)\} . \quad (5.100)$$

5.5 Modifikationen der kartesischen Regelung

In der praktischen Anwendung der kartesischen Bewegungsregelung können kleine Modifikationen des Gütekriteriums oder der Beschränkungen notwendig sein, die sich auf Grund der Flexibilität des Regelungskonzepts jedoch leicht realisieren lassen.

Nicht immer ist die vollständige Pose des Endeffektors vorgegeben. So kann beispielsweise beim Transport von Gegenstände eine vorgegebene ver-

tikale Orientierung bestehen (um den Inhalt einer Kiste nicht zu verlieren), während die Rotation um die Hochachse beliebig ist. Derartige Anforderungen können durch die Modifikation des Gütekriteriums bzw. der Gewichtungsmatrizen des Gütekriteriums erfüllt werden. So lässt sich beispielsweise die freie Rotation des Endeffektors um die Hochachse dadurch erreichen, dass die Elemente o_x und o_y der Quaternion zur Repräsentation der Endeffektororientierung im Gütekriterium unberücksichtigt bleiben.

Auch zusätzliche Beschränkungen der Endeffektorpose der Form

$$\mathbf{c}(\mathbf{y}_{EE}) = \mathbf{c}(\mathbf{k}(\mathbf{q})) \geq \mathbf{0} \quad (5.101)$$

können leicht eingeführt werden. Dies kann notwendig sein, wenn die Solltrajektorie bzw. der Sollpfad sich nahe eines Bereichs befindet, den der Endeffektor nicht betreten darf. Auch Fälle, in denen einzelne Elemente der Endeffektorpose nicht vorgegeben aber beschränkt sind, sind denkbar. Beschränkungen, die den gesamten Roboter betreffen, werden mit der Kollisionsvermeidung in Abschnitt 6.4 behandelt.

5.6 Zusammenfassung

In diesem Kapitel wurde eine kartesische modellprädiktive Regelung zur Verfolgung von zeitabhängigen Trajektorien der Endeffektorpose oder geometrischen Sollpfaden der Endeffektorpose ohne vorgegebene Zeitabhängigkeit vorgeschlagen. Die Regelung beruht auf der Minimierung des in den Abschnitten 5.1.1 und 5.1.2 definierten Trajektorien- bzw. Pfadverfolgungsfehlers. Das Prinzip der modellprädiktiven Regelung erlaubt die direkte Berücksichtigung der kinematischen Beschränkungen des Roboters.

Für die Lösung des Pfadverfolgungsproblems mittels der kartesischen modellprädiktiven Regelung wurden zwei Ansätze betrachtet: Im ersten Ansatz wird der Pfad mit der maximal zulässigen kartesischen Geschwindigkeit abgefahren. Am Pfadende und in der Nähe von Hindernissen wird die Geschwindigkeit reduziert. Im zweiten Ansatz wird die Pfadgeschwindigkeit

keit zusammen mit den Stell- und Zustandsgrößen des Roboters optimiert. Ein Vergleich der beiden Ansätze erfolgt in Abschnitt 8.2.2.

Kapitel 6

Modellprädiktive Kollisionsvermeidung

Die im vorherigen Kapitel eingeführt kartesische modellprädiktive Regelung wird nun erweitert, um sowohl Eigenkollisionen des Roboters mit sich selbst als auch Kollisionen des Roboters mit statischen und dynamischen Hindernissen in seiner Umgebung zu verhindern.

6.1 Anforderungen

Während der Roboter seine kartesische Aufgabe (die Verfolgung von Solltrajektorien oder -pfaden der Endeffektorpose) ausführt, müssen Kollisionen vermieden werden. Dies umfasst sowohl die Eigenkollisionen des Roboters mit sich selbst als auch Kollisionen mit Hindernissen in der Roboterumgebung. Die Redundanz des Roboters soll dabei ausgenutzt werden, um soweit möglich die Aufgabenausführung nicht unterbrechen zu müssen, während gleichzeitig Ausweichbewegungen zur Kollisionsvermeidung durchgeführt werden.

Es wird davon ausgegangen, dass der vorgegebene Sollpfad bzw. die vorgegebene Solltrajektorie kollisionsfrei befahren werden kann oder nur

kurzzeitig blockiert ist. Andernfalls ist eine Neuplanung oder Anpassung des Pfads bzw. der Trajektorie durch die überlagerte Planungseinheit notwendig. Da derartige globale Planungen für mobile Manipulatoren mit vielen Freiheitsgraden im Allgemeinen jedoch rechenaufwändig sind, soll die Integration einer lokalen Kollisionsvermeidungsstrategie in die Regelung schnelle Reaktionen auf die dynamische Umgebung ermöglichen.

Je nach Hindernistyp kann sich die notwendige Verhaltensweise des Roboters zur Kollisionsvermeidung unterscheiden: Bei statischen Hindernissen wie z. B. Tischen und Wänden muss zwar ein Kontakt des Roboters mit dem Hindernis vermieden werden, kleine Abstände zwischen Roboter und Hindernis stellen jedoch keine Gefahr dar, da sich das Hindernis nicht unerwartet bewegt. Es kann sogar notwendig sein, dass sich der Roboter einem derartigen Hindernis nähert, beispielsweise um einen Gegenstand von einem Tisch aufzunehmen. Im Gegensatz hierzu ist der Abstand zu dynamischen Hindernissen wie z. B. Menschen möglichst groß zu halten, um das Risiko einer Kollision zu verkleinern. Entsprechend dieser Anforderungen wird im Folgenden zwischen statischen und dynamischen Hindernissen unterschieden.

6.2 Ansatz

Um Fremd- und Eigenkollisionen des Roboters zu verhindern, werden das Gütekriterium und die Nebenbedingungen der kartesischen modellprädiktiven Regelung erweitert. Die Erweiterungen basieren auf den aktuellen und prädizierten Abständen des Roboters zu den Hindernissen bzw. auf den Abständen der einzelnen Roboterglieder untereinander.

Um den Rechenaufwand für die Optimierung möglichst gering zu halten und eine gute Konvergenz der Optimierung zu erzielen, ist es notwendig, dass die Anzahl der zu berücksichtigenden Abstände klein und die Abstandsberechnung zweimal stetig differenzierbar ist. Dies wird durch die Approximation der Robotergeometrie mit N_R Kugeln erreicht. Die Position des Mittelpunkts der i -ten Kugel $\mathbf{p}_{R,i}(\mathbf{q}) \in \mathbb{R}^3$ ist abhängig von den Gelenkpositionen \mathbf{q} . Der Kugelradius $r_{R,i}$ ist konstant und wird anhand des geometrischen

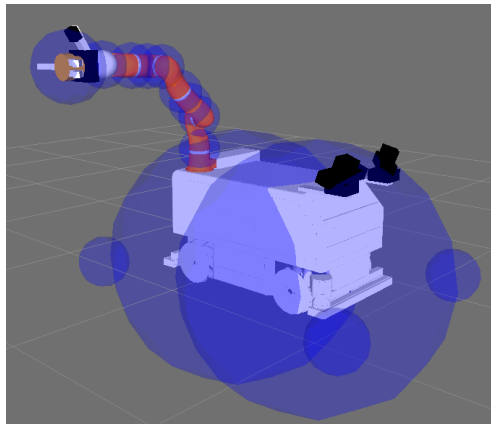


Abbildung 6.1: Beispiel für die Approximation der Robotergeometrie mit Kugeln.

Robotermodells festgelegt. Die Kugeln werden so gewählt, dass sie alle relevanten Roboterteile umschließen. Je größer die Anzahl der Kugeln ist, desto genauer ist die Approximation möglich. Mit einer größeren Anzahl von Roboterkugeln steigt aber auch der Optimierungsaufwand. Im Allgemeinen kann schon mit wenigen Kugeln eine für die Kollisionsvermeidung ausreichende Genauigkeit erzielt werden, wenn keine Annäherung des Roboters auf wenige Zentimeter an ein Hindernis notwendig ist.

Dies ist in dem in Abbildung 6.1 dargestellten Beispiel zu sehen, in dem ein mobiler Manipulator durch 15 Kugeln approximiert wird. Der Roboterarm wird mit acht Kugeln gleicher Größe und einer größeren Kugel für den Greifer approximiert. Für die Plattform werden zwei Kugeln mit sehr großem Radius sowie vier kleine Kugeln verwendet. Die Kugeln bilden die Plattform und ein herstellerseitig vorgeschriebenes Laserscannerschutzfeld um die Plattform ab. Dabei dienen die vier kleinen Kugeln zur Nachbildung der Eckbereiche der rechteckigen Grundfläche.

Analog zur Robotergeometrie wird auch die Geometrie der statischen Hindernisse in der Roboterumgebung mit $N_{O,stat}$ Kugeln und die Geometrie der dynamischen Hindernisse mit $N_{O,dyn}$ Kugeln approximiert. Dabei bezeichnet $\mathbf{p}_{O,stat,j} \in \mathbb{R}^3$ den Mittelpunkt der j -ten Hinderniskugel eines

statischen Hindernisses und $r_{O,stat,j}$ ihren Radius. Die j -te Hinderniskugel eines dynamischen Hindernisses besitzt den Mittelpunkt $\mathbf{p}_{O,dyn,j} \in \mathbb{R}^3$ und den Radius $r_{O,dyn,j}$.

6.3 Vermeidung von Eigenkollisionen

Einige Roboter schließen Eigenkollisionen, d. h. Kollisionen eines Roboterparts mit einem anderen, konstruktionsbedingt aus. Die Robotergeometrie und die anfahrbaren Gelenkstellungen lassen keine Eigenkollisionen zu. In vielen Fällen, insbesondere bei mobilen Manipulatoren, ist ein konstruktionsbedingter Ausschluss von Eigenkollisionen jedoch nicht möglich, ohne den erreichbaren Arbeitsraum des Roboters zu stark einzuschränken. Ein Ansatz zur Eigenkollisionsvermeidung ist es, mit Hilfe eines geometrischen Robotermodells die von der Regelung ermittelten Sollgelenkpositionen des Roboters auf Eigenkollisionen zu prüfen. Damit wird die Eigenkollisionsprüfung der Bewegungsregelung nachgeschaltet. Diese Vorgehensweise verhindert Eigenkollisionen, kann aber zum Scheitern der Aufgabenausführung führen. Stattdessen sollte die Eigenkollisionsvermeidung direkt von der Bewegungsregelung übernommen werden.

Mit Hilfe der in Abschnitt 6.2 eingeführten Approximation der Robotergeometrie wird die Forderung der Eigenkollisionsvermeidung überführt in die Forderung, dass sich Kugeln, die unterschiedliche Roboterglieder beschreiben, nicht überlappen dürfen. Hiervon ausgenommen sind Kugeln die zu zwei aufeinanderfolgenden Robotergliedern gehören. Die Kollision zweier aufeinanderfolgender Glieder kann ohne Einschränkung der weiteren Bewegungsmöglichkeiten durch Beschränkung der zulässigen Gelenkpositionen verhindert werden.

Diese konservativ approximierte Eigenkollisionsvermeidungsforderung kann direkt als zusätzliche Nebenbedingungen der in Kapitel 5 dargestellten kartesischen modellprädiktiven Regelung verwendet werden. Für die Abstände der Mittelpunkte aller Kugelpaare der Kugeln i und j , die nicht

zum selben oder zu aufeinanderfolgenden Gliedern gehören,

$$d_{R,i,j}(\mathbf{q}(k)) := \|\mathbf{p}_{R,i}(\mathbf{q}(k)) - \mathbf{p}_{R,j}(\mathbf{q}(k))\| \quad (6.1)$$

muss für alle Prädiktionsschritte $k = 1, \dots, N_p$

$$d_{R,i,j}(\mathbf{q}(k)) > r_{R,i} + r_{R,j} \quad (6.2)$$

gelten.

Durch die zusätzlichen Nebenbedingungen für die Optimierung der kartesischen modellprädiktiven Regelung führt keine der von der Regelung berechneten Gelenkkonfigurationen zu einer Eigenkollision des Roboters mit sich selbst und es ist keine nachgeschaltete Eigenkollisionsprüfung notwendig.

6.4 Kartesische Roboterbeschränkungen

Analog zur Eigenkollisionsvermeidung lassen sich basierend auf der Approximation der Robotergeometrie durch Kugeln auch kartesische Beschränkungen für den gesamten Roboter oder für einzelne Roboterteile der Form

$$c_i(\mathbf{p}_{R,i}(\mathbf{q}(k))) \geq 0 \quad (6.3)$$

einführen. Mit derartigen Beschränkungen lässt sich beispielsweise das Verlassen eines gegebenen Arbeitsraums durch Teile des Roboters verhindern.

6.5 Vermeidung von Kollisionen mit statischen Hindernissen

Ähnlich zur Eigenkollisionsvermeidung werden Kollisionen des Roboters mit statischen Hindernissen durch zusätzliche Nebenbedingungen für die kartesische modellprädiktive Regelung verhindert.

Zur Kollisionsvermeidung muss der minimale Abstand zwischen Roboter und statischem Hindernis immer größer als ein Sicherheitsabstand d_{safety} sein. Dieser Sicherheitsabstand berücksichtigt Ungenauigkeiten der Roboterlokalisierung und der Hinderniserfassung. Unter Verwendung der in Abschnitt 6.2 eingeführten Approximation durch Kugeln bedeutet das, dass für den Abstand zwischen den Mittelpunkten der Roboterkugel i und der Hinderniskugel j

$$d_{\text{stat},i,j}(\mathbf{q}(k)) := \|\mathbf{p}_{R,i}(\mathbf{q}(k)) - \mathbf{p}_{O,\text{stat},j}\| \quad (6.4)$$

gelten muss:

$$d_{\text{stat},i,j}(\mathbf{q}(k)) > r_{R,i} + r_{O,\text{stat},j} + d_{\text{safety}} . \quad (6.5)$$

Diese Nebenbedingung muss für alle Roboterkugeln $i = 1, \dots, N_R$, alle Hinderniskugeln $j = 1, \dots, N_{O,\text{stat}}$ und alle Prädiktionsschritte $k = 1, \dots, N_p$ erfüllt sein.

6.6 Vermeidung von Kollisionen mit dynamischen Hindernissen

Wie bei statischen Hindernissen kann die Aufgabe der Kollisionsvermeidung formuliert werden durch die Forderung, dass der minimale Abstand zwischen Roboter und Hindernis einen gewissen Sicherheitsabstand nicht unterschreitet. In Gegenwart von Hindernissen mit schwer prädizierbaren Bewegungen, wie beispielsweise Menschen, ist es jedoch trotz der Prädiktion des Roboterhaltens durch den Regler nicht ausreichend, mit Hilfe der Nebenbedingungen einen Sicherheitsabstand zu fordern. Zusätzlich ist ein Roboterverhalten erstrebenswert, das einen möglichst großen Abstand zwischen Roboter und dynamischem Hindernis einhält. Der größere Abstand zwischen Roboter und Hindernis verschafft dem Roboter einen größeren Freiraum für Ausweichbewegungen auch bei unvorhergesehenen Bewegungen des Hindernisses und reduziert somit die Anzahl der Fälle, in denen ein Anhalten der Roboterbewegung zur Kollisionsvermeidung notwendig

ist. Speziell im Fall der Mensch-Roboter-Kooperation erhöht ein größerer Abstand zum Roboter auch das subjektive Sicherheitsempfinden der kooperierenden Person.

Analog zu den statischen Hindernissen werden auch dynamische Hindernisse mit Hilfe von Kugeln approximiert. Unter Verwendung der in Abschnitt 6.2 eingeführten Approximation durch Kugeln gilt für den Abstand zwischen den Mittelpunkten der Roboterkugel i und der Hinderniskugel j

$$d_{\text{dyn},i,j}(\mathbf{q}(k)) := \|\mathbf{p}_{R,i}(\mathbf{q}(k)) - \mathbf{p}_{O,\text{dyn},j}(k)\|. \quad (6.6)$$

Für alle Abstände ($i = 1, \dots, N_R, j = 1, \dots, N_{O,\text{dyn}}$ und $k = 1, \dots, N_p$) muss zur Kollisionsvermeidung die Nebenbedingung

$$d_{\text{dyn},i,j}(\mathbf{q}(k)) > r_{R,i} + r_{O,\text{dyn},j} + d_{\text{safety}} =: d_{\text{dyn},i,j,\text{min}} \quad (6.7)$$

erfüllt sein. Im Gegensatz zu statischen Hindernissen ist bei dynamischen Hindernissen die Position des Mittelpunkts der Hinderniskugel jedoch zeitabhängig. Liegt eine Prädiktion der Hindernisbewegung vor, wird für jeden Abtastschritt k der prädizierte Mittelpunkt der Hinderniskugel verwendet. Ist keine Prädiktion verfügbar, werden in jedem Reglerschritt die Hindernispositionen als konstant über den gesamten Prädiktionshorizont angenommen. Diese Annahme führt dazu, dass der prädizierte Roboter-Hindernis-Abstand mit zunehmendem Prädiktionshorizont unsicherer wird. Da die Hindernisinformationen jedoch in jedem Reglerschritt aktualisiert werden und das Reglerergebnis nicht für den gesamten Prädiktionshorizont, sondern nur den ersten Abtastschritt angewendet wird, ist diese Unsicherheit vertretbar. Zur Erhöhung der Sicherheit kann in diesem Fall ein größerer Sicherheitsabstand d_{safety} gewählt werden. Alternativ kann der Sicherheitsabstand mit der Anzahl der prädizierten Schritte k vergrößert werden. Dies kann jedoch auch im Fall mit Prädiktion der Hindernisbewegung notwendig sein, abhängig davon wie zuverlässig die Prädiktion erfolgt.

Um einen möglichst großen Abstand zwischen Roboter und Hindernis zu erzielen, ohne die Ausführung der kartesischen Aufgabe zu unterbrechen,

wird die Zielfunktion der Optimierung des Reglers additiv um den Term

$$\sum_{k=0}^{N_p-1} F_O(k+1) + E_O(N_p) \quad (6.8)$$

erweitert. Der Erweiterungsterme der Kostenfunktion F_O und der Endzustandsbewertung E_O müssen so gewählt werden, dass die Kosten mit kleiner werdendem Roboter-Hindernis-Abstand steigen und die Terme zweimal stetig nach den Optimierungsvariablen differenzierbar sind. Letzteres dient der schnellen Lösung des Optimierungsproblems.

Die beste Wahl der Terme ist abhängig von dem zu realisierenden Roboterverhalten und kann nicht allgemeingültig ermittelt werden. Deshalb wird im Folgenden beispielhaft eine geeignete Wahl für den in dieser Arbeit zur Analyse verwendeten Demonstrator vorgestellt:

$$F_O(k) := \sum_{i=1}^{N_R} \sum_{j=1}^{N_O} \frac{\lambda_{O,i}}{d_{\text{dyn},i,j}^2(\mathbf{q}(k)) - d_{\text{dyn},i,j,\text{min}}^2} \quad (6.9)$$

$$E_O(N_p) := f_O F_O(N_p) \quad (6.10)$$

mit den wählbaren Parametern $\lambda_{O,i} \geq 0$ ($i = 1, \dots, N_R$) und $f_O \geq 0$. Damit geht F_O gegen unendlich, wenn sich mindestens ein Roboter-Hindernis-Abstand $d_{\text{dyn},i,j}(\mathbf{q})$ dem Mindestabstand $d_{\text{dyn},i,j,\text{min}}$ nähert. Die Kostenfunktion bewirkt eine Abstoßung des Roboters vom Hindernis. Aufgrund der Nebenbedingung ist $d_{\text{dyn},i,j}(\mathbf{q}(k)) > d_{\text{dyn},i,j,\text{min}}$ gewährleistet. Mit Hilfe der Parameter $\lambda_{O,i}$ ist eine intuitive Einstellung der Kollisionsvermeidungsterme und damit eine stärkere Abstoßung des Roboters von den Hindernissen. Mit $\lambda_{O,i} = 0$ kann eine Abstoßung für einzelne Roboterkuugeln unterbunden werden.

6.7 Reduzierung der Anzahl der Nebenbedingungen zur Kollisionsvermeidung

Werden alle Roboter-Hindernis-Abstände einzeln in den Nebenbedingungen geprüft, werden für die Kollisionsvermeidung $N_R \cdot N_O \cdot N_p$ Nebenbedingungen benötigt ($N_O := N_{O,stat} + N_{O,dyn}$). Bei einer größeren Anzahl von Hindernissen bzw. bei Hindernissen, die durch eine Vielzahl von Kugeln beschrieben werden, kann die Anzahl der Nebenbedingungen auf diese Art sehr groß werden. Die Folge ist ein sehr hoher Rechenaufwand für die Optimierung, sodass die Berechnungszeit für die Optimierung in einem Reglerschritt die zulässige Zeit eines Abtastschritts übersteigen kann.

Im Allgemeinen ist es jedoch nicht notwendig, alle Roboter- und Hinderniskugeln gegeneinander zu prüfen. Stattdessen kann die Anzahl der Nebenbedingungen auf eine geeignete Auswahl reduziert werden, sodass die Kollisionsvermeidung mit einem vertretbaren Rechenaufwand zu erzielen ist. Dies beruht auf der Annahme, dass Hindernisse mit einem größerem Abstand zum Roboter als weniger relevant einzustufen sind als Hindernisse mit kleinem Abstand zum Roboter, sodass die berücksichtigten Roboter-Hindernis-Paare eingegrenzt werden können. Zur Auswahl der zu berücksichtigenden Hindernisse sind verschiedene Ansätze möglich. Welche Auswahl sowohl die Anforderungen an die Sicherheit als auch an den Rechenaufwand der Optimierung erfüllt, ist jedoch abhängig vom Roboter und von dessen Einsatzzweck. Deshalb werden im Folgenden einige Ansätze beispielhaft in einer Simulation verglichen, um die Anwendbarkeit auf einem mobilen Manipulator zu demonstrieren.

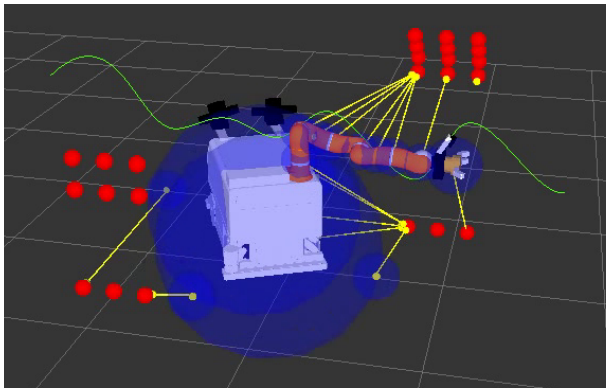


Abbildung 6.2: Simulationsszenario mit mobilem Manipulator, Roboterkugeln (blau), Solltrajektorie (grüne Linie), Hindernissen (rot) und minimalen Abständen zwischen Roboterpunkten und Hindernissen (gelbe Linien).

6.7.1 Reduzierung der Anzahl der Nebenbedingungen am Beispiel eines mobilen Manipulators mit zehn Freiheitsgraden

Simuliert wird die Kollisionsvermeidung für einen mobilen Manipulator mit zehn Freiheitsgraden (Details siehe Abschnitt 8.1). Die Robotergeometrie wird mit $N_R = 15$ Kugeln approximiert (siehe Abbildung 6.1). Die Regelung zur Verfolgung einer Solltrajektorie erfolgt mit einer Abtastzeit von $T_s = 0,1$ s und $N_p = 10$ Prädiktionsschritten. Das Simulationsszenario ist in Abbildung 6.2 dargestellt. Auf beiden Seiten der grünen Solltrajektorie befinden sich Hinderniskugeln (rot). Die Hindernisse werden als dynamische Hindernisse behandelt und sowohl in den Nebenbedingungen als auch in der Kostenfunktion berücksichtigt. Während des Abfahrens der Solltrajektorie (von links nach rechts) muss der Roboter eine enge Passage zwischen den Hindernissen durchqueren. Die Approximation der Robotergeometrie ist mit blauen Roboterkugeln dargestellt. Zusätzlich sind die minimalen Abstände zwischen den Mittelpunkten der Roboterkugeln und den Hindernissen durch gelbe Linien markiert.

Zur Reduzierung der Anzahl der Nebenbedingungen für die Kollisionsvermeidung werden folgende Ansätze verglichen:

1. Zur Ermittlung der relevanten Hinderniskugeln wird für jedes der N_L Roboterglieder die am nächsten liegende Hinderniskugel bestimmt. Diese Hinderniskugeln werden bei der Kollisionsvermeidung für alle Roboterkerne berücksichtigt. Somit sind für die Kollisionsvermeidung maximal $N_L \cdot N_R \cdot N_p$ Nebenbedingungen zu erfüllen. Die maximale Anzahl der Nebenbedingungen ist unabhängig von der Anzahl der Hindernisse.
2. Zur weiteren Reduzierung der Anzahl der Nebenbedingungen wird für alle ausgewählten Hinderniskugeln berechnet, welche Zeit mindestens benötigt wird, bis der Roboter die Hinderniskugel mit einem seiner Glieder erreichen kann (siehe Abschnitt 3.2). In jedem Prädiktionsschritt werden nur die Hinderniskugeln aus Punkt 1 berücksichtigt, die innerhalb des entsprechenden Zeitraums erreichbar sind.
3. Für jede Roboterkerne wird nur die Hinderniskugel mit dem jeweils kleinsten Abstand zu dem dazugehörigen Roboterglied berücksichtigt. Hierdurch wird im Vergleich zu Punkt 1 die Anzahl der zu prüfenden Roboter-Hindernis-Paare weiter reduziert und es müssen für die Kollisionsvermeidung lediglich maximal $N_R \cdot N_p$ Nebenbedingungen erfüllt werden.
4. Für jede Roboterkerne wird die Hinderniskugel mit dem jeweils kleinsten Abstand zur Roboterkerne berücksichtigt. Wie in Ansatz 3 müssen für die Kollisionsvermeidung maximal $N_R \cdot N_p$ Nebenbedingungen erfüllt werden.
5. Wie in Ansatz 4 wird für jede Roboterkerne die Hinderniskugel mit dem jeweils kleinsten Abstand bestimmt. Es werden in jedem Prädiktionsschritt jedoch nur die Hinderniskugeln berücksichtigt, die innerhalb des entsprechenden Prädiktionshorizonts erreichbar sind.

Derartige Einschränkungen der betrachteten Hindernisse sind möglich, da die zu berücksichtigenden Hindernisse in jedem Reglerschritt aktualisiert werden.

Wird zur Auswahl der berücksichtigten Hindernisse der Abstand zwischen Hinderniskugel und Roboterglied zugrunde gelegt, kann die Abstandsberechnung zur Auswahl der zu berücksichtigenden Hindernisse mit einem detaillierteren geometrischen Robotermodell, wie beispielsweise einem CAD-Modell, erfolgen. Erfolgt die Auswahl basierend auf dem Abstand zwischen Hinderniskugel und Roboterku­gel hat dies wiederum den Vorteil, dass große Glieder wie beispielsweise die Plattform mit mehreren Kugeln approximiert werden können und somit auch mehrere nahegelegene Hinderniskugeln berücksichtigt werden können.

Die Simulation der verschiedenen Ansätze zur Reduzierung der Anzahl der Nebenbedingungen ergibt folgende Ergebnisse (siehe Abbildung 6.3): Unter Verwendung von Ansatz 1 ist die Anzahl der Nebenbedingungen zu hoch, sodass die Rechenzeit für die Optimierung eines Reglerschritts die Reglerabtastzeit von $T_s = 0,1$ s überschreitet und ein Abfahren der Solltrajektorie nicht möglich ist. Die Reduzierung der Anzahl der Nebenbedingungen durch Berücksichtigung der Erreichbarkeit in Ansatz 2 bewirkt nur eine geringe Verkürzung der Rechenzeit der Optimierung. Sie liegt deutlich über den zulässigen 0,1 s. Sowohl mit Ansatz 3 (Berücksichtigung des Hindernisses mit kleinstem Abstand zum Roboterglied für die jeweilige Roboterku­gel) und Ansatz 4 (Berücksichtigung des Hindernisses mit kleinstem Abstand zur Roboterku­gel) kann der Roboter der Solltrajektorie erfolgreich folgen und trotz der engen Passage Kollisionen mit den Hindernissen vermeiden. Die benötigte Rechenzeit für die Optimierung eines Reglerschritts ist in beiden Fällen ähnlich und liegt bis auf einzelne Ausreißer unter der Abtastzeit von 0,1 s. Positions- und Orientierungsfehler der Endeffektorpose und der minimale Abstand zwischen dem Roboter und den Hindernissen unterscheiden sich bei beiden Ansätzen meist nur geringfügig. Zwischenzeitlich nimmt der Positionsfehler mit Ansatz 3 jedoch wesentlich größere Werte an als mit Ansatz 4. Auch entspricht die resultierende Roboterbewegung mit Ansatz 4 der von einem menschlichen Beobachter intuitiv erwarteten

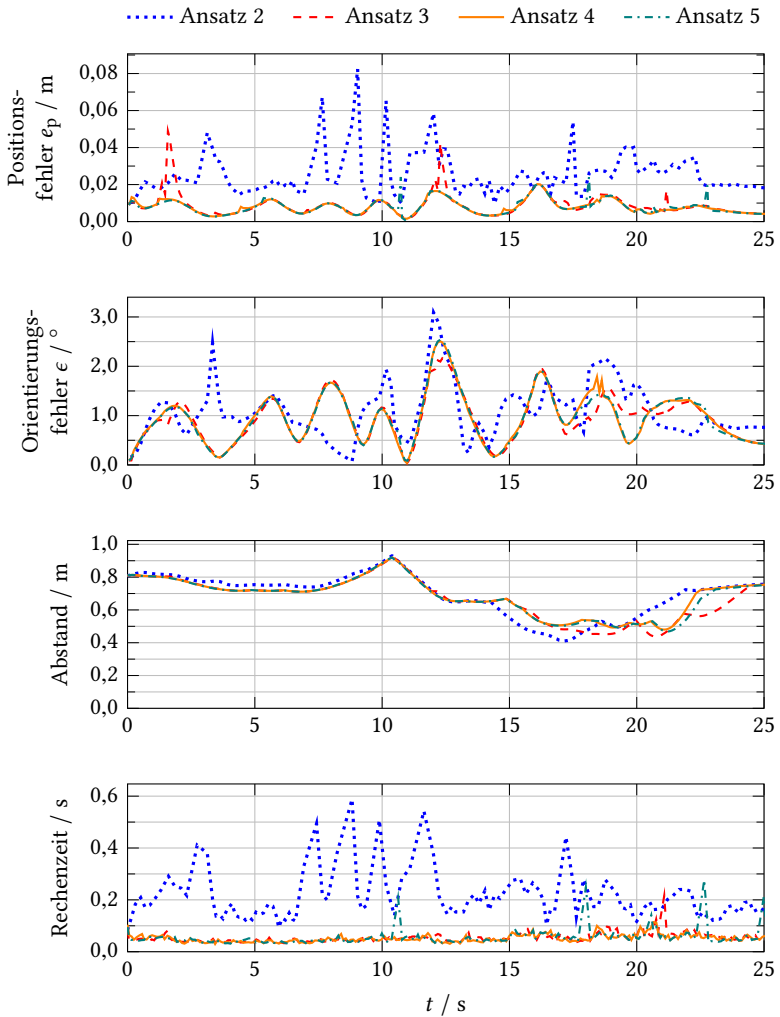


Abbildung 6.3: Simulationsergebnisse mit verschiedenen Ansätzen zur Reduzierung der Anzahl der Nebenbedingungen.

Bewegung, während die Bewegung mit Ansatz 3 weniger glatt wirkt. Dies ist darauf zurückzuführen, dass mit dem Ansatz 4 verschiedene Hinderniskugeln für die einzelnen Kugeln zur Approximation der Plattform berücksichtigt werden können. Dies ist besonders in der engen Passage von Vorteil, wenn unterschiedliche Hindernisse den vorderen und hinteren Roboterkugeln nahe kommen. Die zusätzliche Reduzierung der Nebenbedingungen in Ansatz 5 bewirkt keine weitere Verbesserung, weder in Hinblick auf die Rechenzeit noch auf die resultierende Roboterbewegung. Ursache hierfür ist vermutlich, dass lediglich Nebenbedingungen wegfallen, die während der Optimierung im Allgemeinen bereits erfüllt sind.

Die Simulationsergebnisse zeigen, dass durch die Reduzierung der Anzahl der Nebenbedingungen das Ziel der Kollisionsvermeidung simultan zum Folgen der Solltrajektorie auch bei Anwesenheit mehrerer Hindernisse erfüllt werden kann. Für den im Beispiel betrachteten mobilen Manipulator liefert der Ansatz 4 die besten Ergebnisse, weshalb dieser Ansatz auf dem in Kapitel 8 eingesetzten Demonstrator Anwendung findet.

6.8 Kombination von Arbeitsraumüberwachung und Kollisionsvermeidung

Die Auswahl der in einem Reglerschritt zu berücksichtigenden Hindernisse und ihre Approximation mit Kugeln bietet den Vorteil, dass das Konzept der Kollisionsvermeidung unabhängig von der ursprünglichen Repräsentation der Hindernisse (z. B. Octree, CAD, Punktwolke oder Voxelgitter) ist. Lediglich der Algorithmus zur Abstandsberechnung sowie die Wahl des Kugelradius muss an die Datenstruktur der Hindernisdarstellung angepasst werden.

Hier wird zur Gewinnung der für die Kollisionsvermeidung notwendigen Hindernisinformationen die in Kapitel 3 vorgeschlagene Arbeitsraumüberwachung eingesetzt. Sie liefert einen Octree, der die unbekanntenen, möglicherweise dynamischen Hindernisse in der nahen Roboterumgebung ent-

hält. Jeder Hindernisknoten j des Octree wird in eine Hinderniskugel überführt, wobei der Kugelmittelpunkt dem Mittelpunkt des Octree-Zellen entspricht. Der Kugelradius beträgt die halbe Raumdiagonale der Octree-Zelle mit Kantenlänge $r_{V,j}$:

$$r_{O,dyn,j} = \frac{\sqrt{3}}{2} r_{V,j} . \quad (6.11)$$

Werden hierfür die Hindernisknoten mit der feinsten Auflösung verwendet, besitzen alle Hinderniskugeln den gleichen Radius. Werden die Hindernisknoten mit der jeweils größtmöglichen Auflösung verwendet, kann die Anzahl der Hinderniskugeln zur Approximation der 3D-Hindernisdarstellung reduziert werden. Allerdings resultieren hieraus auch größere Kugelradien und größere Approximationsfehler. Da die Anzahl der Hindernisknoten bei Verwendung der kleinsten Auflösung keine zu großen Verzögerungen bei der Abstandsberechnung und der in Abschnitt 6.7 beschriebenen Auswahl der berücksichtigten Hinderniskugeln bewirkt, wird deshalb in den in Kapitel 8 dargestellten Versuchen die Kugelapproximation basierend auf der kleinsten Auflösung mit geringerem Approximationsfehler verwendet.

Die aus der Arbeitsraumüberwachung gewonnenen Hindernisse werden entsprechend der Worst-Case-Annahme als dynamisch angenommen. Liegt zusätzlich eine Darstellung bekannter statischer Hindernisse vor (siehe Abschnitt 3.4), die in der Arbeitsraumüberwachung berücksichtigt werden, werden auch von ihnen die Hindernisse mit den kleinsten Abständen zum Roboter bestimmt und mit Kugeln approximiert.

6.9 Zusammenfassung

In diesem Kapitel wurden Erweiterungen der kartesischen modellprädiktiven Regelung vorgestellt, die die Vermeidung von Kollisionen des Roboters mit sich selbst, mit statischen und mit dynamischen Hindernissen parallel zur Verfolgung von Solltrajektorien oder Sollpfaden erlauben. Basierend auf der Approximation der Robotergeometrie und der Hindernisse durch Kugeln wurden Nebenbedingungen formuliert, die Kollisionen verbieten. Dynamische Hindernisse werden zusätzlich auch in der Kostenfunktion

berücksichtigt, sodass eine Abstoßung des Roboters von den dynamischen Hindernissen bewirkt wird.

Durch die Eingrenzung der in einem Reglerschritt zu berücksichtigenden Hindernisse kann die Anzahl der für die Kollisionsvermeidung notwendigen Nebenbedingungen auf einen von der Anzahl der Hindernisse unabhängigen Wert beschränkt werden. So kann das Optimierungsproblem ausreichend schnell gelöst werden, auch wenn sich eine große Anzahl von Hindernissen in der Roboterumgebung befindet.

Kapitel 7

Modellprädiktive Kontaktregelung

In diesem Kapitel wird die in Kapitel 5 entwickelte kartesische modellprädiktive Regelung erweitert, sodass der Roboter parallel zur Verfolgung von Solltrajektorien oder -pfaden nachgiebig auf stattfindende Kontakte reagieren kann.

7.1 Anforderungen

Kontakte des Roboters mit seiner Umgebung können in unterschiedlichen Situationen auftreten. Um die erforderliche Reaktion des Roboters auf einen mit seiner Umgebung stattfindenden Kontakt spezifizieren zu können, werden die Kontakte zunächst nach ihrer Intention klassifiziert. Dabei wird in beabsichtigte oder unerwünschte Kontakte unterteilt. Des Weiteren wird unterschieden, ob der Kontakt am Endeffektor oder am Roboterarm stattfindet.

Bei unerwünschten Kontakten (Kollisionen) ist im Allgemeinen ein nachgiebiges Verhalten des Roboters erwünscht, um die Kontaktkraft und Kontaktdauer zu reduzieren und damit das Verletzungsrisiko sowohl für den

Roboter als auch für den Menschen oder Gegenstand, mit dem die Kollision stattfindet, zu verkleinern. Findet die Kollision mit dem Endeffektor statt, muss der Roboter seine Solltrajektorie bzw. seinen Sollpfad verlassen, um nachgiebig reagieren zu können. Findet die Kollision dagegen mit dem Roboterarm statt, soll der Roboter, trotz der nachgiebigen Reaktion, der Sollpose des Endeffektors so gut wie möglich folgen. Hierdurch werden unerwartete Bewegungen des Endeffektors vermieden, die wiederum zu weiteren Kollisionen führen könnten. Des Weiteren kann so die eigentliche Roboter-aufgabe mit möglichst wenigen Unterbrechungen fortgesetzt werden.

Im Fall von beabsichtigten Kontakten sind unterschiedliche erwünschte Reaktionen möglich. Soll eine Roboterpose eingelernt werden oder behindert der Roboter die Arbeit einer interagierenden Person und sie will den Roboter zur Seite schieben, ist wie im Fall der Kollisionen eine nachgiebige Reaktion erwünscht. Zur Montage kann aber auch ein steifes Verhalten notwendig sein. Die Notwendigkeit einer unnachgiebigen Reaktion auf Kontakte ist damit abhängig von der Aufgabe des Roboters bzw. vom aktuellen Ausführungszustand der Aufgabe, tritt jedoch im Allgemeinen nur bei Kontakten am Endeffektor, nicht am Roboterarm auf. In dieser Arbeit wird davon ausgegangen, dass im Fall von beabsichtigten Kontakten am Endeffektor die gewünschte Reaktion bekannt ist. Liegt keine andere Information vor, wird der Kontakt als unbeabsichtigt interpretiert und es erfolgt eine nachgiebige Reaktion. Auf Kontakte am Roboterarm erfolgt generell eine nachgiebige Reaktion. Die vorgestellten Verfahren sind jedoch auch auf andere Anforderungen übertragbar.

7.2 Kontaktschätzung

Voraussetzung für die Reaktion auf Kontakte ist deren Schätzung basierend auf externen oder propriozeptiven Sensordaten. Kontakte des Roboters mit seiner Umgebung können entweder beschrieben werden durch die Drehmomente, die der Kontakt in den Gelenken des Roboters erzeugt, oder durch die einwirkende Kontaktkraft sowie den Kontaktpunkt, an dem die Kraft wirkt. Methoden zur Schätzung der Kontaktmomente in den Robotergelenken (die

sogenannten externen Momente τ_{ext}) und zur Schätzung von Kontaktkraft F_{ext} und Kontaktposition wurden im Abschnitt 2.4.1 dargestellt.

Mit dem in [DeL06] beschriebenen residuenbasierten Ansatz existiert eine gängige Methode zur Schätzung der externen Momente von Manipulatoren.

Auch für die Schätzung der Kontaktkraft und -position existiert mit der Methode von [Lik14] ein Verfahren, das keine weitere externe Sensorik benötigt. Auf Grund der Roboterkinematik kann es zu Mehrdeutigkeiten bezüglich des Kontaktpunktes und der Kontaktkraft kommen. Die gleichen externen Gelenkmomente können von Kraftvektoren mit unterschiedlichen Kontaktpositionen, Orientierungen und Beträgen verursacht werden. Da Schätzungen, die auf Grund der Mehrdeutigkeiten fehlerbehaftet sind, jedoch zu gleichen (oder ähnlichen) gewünschten Reaktionen wie die tatsächlichen Kontakte führen, eignet sich dieser Ansatz dennoch zur Kontaktschätzung für die Kontaktregelung.

Somit können für die Kontaktregelung sowohl die externen Drehmomente als auch Kontaktkraft und -position als bekannt vorausgesetzt werden.

7.3 Regelung von Kontakten am Roboterarm

Wie in Abschnitt 7.1 dargestellt, ist es für eine intuitive und sichere Mensch-Roboter-Interaktion im Allgemeinen notwendig, dass der Roboter nachgiebig auf Kontakte des Roboterarms mit seiner Umgebung reagiert. Gleichzeitig soll der Endeffektor seine Sollpose möglichst gut beibehalten.

Im Folgenden werden zwei Ansätze zur Erweiterung der kartesischen modellprädiktiven Regelung aus Kapitel 5 vorgestellt, die ein solches nachgiebiges Verhalten bei Kontakten ermöglichen. Der erste Ansatz legt hierzu die externen Gelenkmomente zugrunde, der zweite Ansatz basiert auf dem geschätzten Kontaktpunkt und auf der geschätzten Kontaktkraft.

7.3.1 Momentenbasierte Kontaktregelung

Die in dieser Arbeit vorgeschlagene kartesische modellprädiktive Regelung ist eine kinematische Roboterregelung, die auf den Gelenkpositionen und -geschwindigkeiten beruht. Gelenkmomente werden nicht explizit geregelt. Dies hat den Vorteil, dass die Regelung kein exaktes dynamisches Robotermodell benötigt und allgemein einsetzbar ist (z. B. auch für mobile Plattformen). Aus diesem Grund soll auch die Kontaktregelung nicht auf der direkten Regelung der Gelenkmomente beruhen. Stattdessen wird ein nachgiebiges Roboterverhalten erzielt, indem die einzelnen Gelenke in Richtung der einwirkenden externen Drehmomente bewegt werden. Die externen Drehmomente werden hierzu als bekannt vorausgesetzt.

Die externen Gelenkmomente werden in Sollgeschwindigkeiten der einzelnen Gelenke überführt

$$\mathbf{w}_{c,d} := \lambda_\tau \boldsymbol{\tau}_{\text{ext}}, \quad (7.1)$$

wobei $\lambda_\tau > 0$ einen Verstärkungsfaktor darstellt. Da im Allgemeinen keine Prädiktion der externen Momente verfügbar ist, wird die Sollgeschwindigkeit der Gelenke über den gesamten Prädiktionshorizont als konstant angenommen. Das Gütekriterium der kartesischen modellprädiktiven Regelung (5.56) bzw. (5.84) wird additiv um den Term

$$\sum_{k=0}^{N_p-1} F_C(k+1) + E_C(N_p) \quad (7.2)$$

erweitert. Die Erweiterung der Kostenfunktion zur Kontaktregelung wird zu

$$F_C(k) := (\mathbf{w}(k) - \mathbf{w}_{c,d}(k))^T \mathbf{Q}_C (\mathbf{w}(k) - \mathbf{w}_{c,d}(k)) \quad (7.3)$$

gewählt, wobei \mathbf{Q}_C eine positiv-definite $N \times N$ Diagonalmatrix ist. Da die Erweiterung der Kostenfunktion nur von den Stellgrößen und nicht von Zustandsgrößen des zu regelnden Systems abhängt, wird die Erweiterung

der Endzustandsbewertung zu Null gewählt:

$$E_C(N_p) := 0. \quad (7.4)$$

Mit dieser Erweiterung bevorzugt die Regelung Gelenkbewegungen in Richtung der einwirkenden Drehmomente, sodass ein nachgiebiges Verhalten entsteht. Gleichzeitig sorgt der Anteil der kartesischen Regelung im Gütekriterium dafür, dass der Endeffektor dem Sollpfad bzw. der Solltrajektorie möglichst gut folgt. Die Gewichtung, welches der beiden Ziele eher zu verfolgen ist, wenn nicht beide Ziele gleichzeitig erfüllt werden können, erfolgt über die Wahl der Gewichtungsmatrix Q_C .

7.3.2 Kraftbasierte Kontaktregelung

Wie die momentenbasierte Kontaktregelung soll auch die kraftbasierte Kontaktregelung eine rein kinematische Erweiterung der modellprädiktiven Regelung darstellen. Es wird davon ausgegangen, dass sowohl die einwirkende Kontaktkraft (Amplitude und Richtung der Kraft) als auch der Kontaktpunkt (d. h. der Punkt, an dem die Kontaktkraft wirkt) gemessen oder geschätzt werden (siehe Abschnitt 2.4.1). Ein nachgiebiges Roboterverhalten wird erreicht, indem der Kontaktpunkt in Richtung der einwirkenden Kraft bewegt wird. Hierzu wird die Kontaktkraft F_{ext} in die Sollgeschwindigkeit des Kontaktpunktes

$$v_{C,d} := \lambda_F F_{\text{ext}} \quad (7.5)$$

überführt. Dabei ist $\lambda_F > 0$ ein Verstärkungsfaktor.

Da im Allgemeinen keine Prädiktion der Kontaktkraft verfügbar ist, wird die Sollgeschwindigkeit des Kontaktpunktes über den gesamten Prädiktionshorizont als konstant angenommen. Damit betragen die Sollpositionen des Kontaktpunktes für alle Prädiktionsschritte ($k = 1, \dots, N_p$)

$$p_{C,d}(k) = p_C(0) + k \cdot T_s \cdot v_{C,d}, \quad (7.6)$$

wobei $\mathbf{p}_C(0)$ die aktuelle Position des Kontaktpunktes ist. Die tatsächliche Position des Kontaktpunktes \mathbf{p}_C ist abhängig von den Gelenkstellungen und ergibt sich aus der direkten Kinematik des Kontaktpunktes.

Um den Kontaktpunkt mit der gewünschten Geschwindigkeit in Richtung der einwirkenden Kraft zu bewegen, wird die Zielfunktion der modellprädiktiven Regelung (5.56) bzw. (5.84) um den Term

$$\sum_{k=0}^{N_p-1} F_C(k+1) + E_C(N_p) \quad (7.7)$$

erweitert. Die Erweiterungen der Kostenfunktion und der Endzustandsbewertung werden zu

$$F_C(k) := (\mathbf{p}_C(\mathbf{q}(k)) - \mathbf{p}_{C,d}(k))^T \mathbf{Q}_C (\mathbf{p}_C(\mathbf{q}(k)) - \mathbf{p}_{C,d}(k)) \quad (7.8)$$

$$E_C(N_p) := f_C F_C(N_p) \quad (7.9)$$

gewählt, wobei die positiv-definite 3×3 Diagonalmatrix \mathbf{Q}_C und $f_C \geq 0$ wählbare Parameter sind. Diese Terme bestrafen die Abweichung der resultierenden Position des Kontaktpunktes von der gewünschten Position und führen dazu, dass der Kontaktpunkt in Richtung der einwirkenden Kraft bewegt wird. Somit wird eine nachgiebige Bewegung erzielt. Gleichzeitig sorgen die kartesischen Terme der Regelung unter Ausnutzung der Redundanz des Roboters für das Beibehalten der Solltrajektorie bzw. des Sollpfades des Endeffektors soweit möglich. Durch die Wahl der Gewichtungsparemeter \mathbf{Q}_C und f_C wird beeinflusst, ob die Nachgiebigkeit während eines Kontaktes oder die Genauigkeit der Endeffektorpose einen stärkeren Einfluss auf die Bewegung haben.

Sollen mehrere Kontaktkräfte gleichzeitig berücksichtigt werden, wird für jede Kontaktkraft ein Term der Form (7.7) hinzugefügt. Durch die Wahl der Gewichtungsmatrizen können dabei alle Kontakte gleich stark in die Regelung eingehen oder es kann eine Priorisierung erfolgen.

7.4 Regelung von Kontakten am Endeffektor

Tritt der Kontakt am Endeffektor statt am Roboterarm auf, können wie in Abschnitt 7.1 diskutiert unterschiedliche Reaktionen auf den Kontakt notwendig sein: steifes, geringfügig nachgiebiges oder nachgiebiges Verhalten des Endeffektors. Es wird hier davon ausgegangen, dass auf Grund von Kontextinformationen das gewünschte Roboterverhalten bekannt ist. Alle drei erforderlichen Reaktionen lassen sich durch geeignete Parameterwahl mit der kartesischen modellprädiktiven Regelung und den Erweiterungen zur Kontaktregelung (siehe Abschnitt 7.3) realisieren:

- Ein steifes Verhalten des Endeffektors wird durch Ignorieren der Kontakte am Endeffektor erreicht. In diesem Fall werden Q_C und f_C zu Null gewählt.
- Ein geringfügig nachgiebiges Verhalten des Endeffektors wird durch die Erweiterungen in Abschnitt 7.3 und mit denselben Parametern, die zur Reaktion auf Kontakte am Roboterarm verwendet werden, erzielt. Die Erweiterung der Kostenfunktion F_C bewirkt im Falle eines Kontakts zunächst ein nachgiebiges Verhalten. Dieses führt jedoch auch zu einem Anstieg des Pfadverfolgungs- bzw. Trajektorienverfolgungsfehlers. Mit steigendem kartesischem Fehler, haben die Terme des Gütekriteriums zur Pfad- bzw. Trajektorienverfolgung den größeren Einfluss. Die Nachgiebigkeit des Endeffektors nimmt mit zunehmendem Pfadverfolgungs- bzw. Trajektorienverfolgungsfehler ab.
- Ein nachgiebiges Verhalten des Endeffektors wird durch das Aufgeben des kartesischen Sollpfades bzw. der Solltrajektorie erreicht. In diesem Fall werden Q_e und R_e zu Null gewählt.

Somit können die unterschiedlichen Anforderungen an die Reaktion auf Endeffektorkontakte durch einfaches Umschalten der Reglerparameter erreicht werden.

7.5 Kombination von Kollisionsvermeidung und Kontaktregelung

In vielen Anwendungen ist es wünschenswert, die Strategie zur Reaktion auf stattfindende Kontakte mit einer Kollisionsvermeidungsmethode zu kombinieren, um so die Anzahl der unerwünschten Kontakte zu reduzieren. Dies stellt bei der vorgeschlagenen kartesischen modellprädiktiven Regelung keine Schwierigkeit dar, da die Erweiterungen zur Kollisionsvermeidung und zur Kontaktregelung gleichzeitig zum Einsatz kommen können. Es werden die Zielfunktion

$$J = \sum_{k=0}^{N_p-1} (F_K(k) + F_O(k+1) + F_C(k+1)) + E_K(N_p) + E_O(N_p) + E_C(N_p), \quad (7.10)$$

die neben dem Gütekriterium für die kartesische Aufgabe auch die Terme zur Kollisionsvermeidung und Kontaktregelung enthält, sowie die in Kapitel 5 und 6 eingeführten Nebenbedingungen zur Erfüllung der kartesischen Aufgabe und zur Kollisionsvermeidung eingesetzt. Lediglich eine Modifikation der Nebenbedingungen zur Kollisionsvermeidung (6.5) und (6.7) ist notwendig: Verletzt ein Hindernis diese Nebenbedingungen und wird gleichzeitig ein Kontakt detektiert, wird dieses Hindernis bei der Kollisionsvermeidung ignoriert, sodass eine geeignete Reaktion auf den Kontakt ermittelt werden kann. Andernfalls könnte keine Lösung des Optimierungsproblems gefunden werden, da bei einem Kontakt die Nebenbedingung zur Kollisionsvermeidung immer verletzt ist. So wird die Kollision mit Hindernissen soweit möglich vermieden. Ist dies jedoch nicht möglich, z. B. weil sich das Hindernis zu schnell bewegt oder nicht von der Arbeitsraumüberwachung erfasst wird, findet eine nachgiebige Reaktion auf die Kollision statt.

7.6 Zusammenfassung

In diesem Kapitel wurden Erweiterungen der kartesischen modellprädiktiven Regelung vorgestellt, die eine nachgiebige Reaktion auf stattfindende

Kontakte ermöglichen. Die Erweiterungen setzen eine Messung bzw. Schätzung der externen Gelenkmomente oder eine Messung bzw. Schätzung der Kontaktkraft und des Kontaktpunktes voraus. Nachgiebigkeit wird erreicht, indem der Roboter in Richtung der einwirkenden Momente oder der einwirkenden Kraft bewegt wird, während gleichzeitig durch Ausnutzung der Redundanz die Solltrajektorie bzw. der Sollpfad der Endeffektorpose weiter verfolgt werden kann. Die Kontaktregelung kann mit der Erweiterung zur Kollisionsvermeidung kombiniert werden, sodass unerwünschte Kontakte soweit möglich vermieden werden und gleichzeitig eine nachgiebige Reaktion auf dennoch stattfindende Kontakte erreicht werden kann.

Kapitel 8

Ergebnisse

Die kartesische modellprädiktive Roboterregelung sowie ihre Erweiterungen zur Kollisionsvermeidung mit Arbeitsraumüberwachung und zur Kontaktregelung werden in mehreren Experimenten auf einem mobilen Manipulator mit zehn Freiheitsgraden und auf einen stationären Manipulator mit sieben Freiheitsgraden eingesetzt. In diesem Kapitel wird zunächst der Aufbau des Demonstrators vorgestellt. Es folgen Experimente zur Verfolgung kartesischer Sollpfade, zur Kollisionsvermeidung, zur Kontaktregelung und zur Kombination von Kollisionsvermeidung und Kontaktregelung sowie die Analyse der Ergebnisse.

8.1 Demonstrator

In den folgenden Abschnitten wird die Implementierung der entwickelten Verfahren auf einer Versuchsplattform vorgestellt. Diese dient der Evaluation der Regelung und Arbeitsraumüberwachung.

8.1.1 Versuchsplattform

Die Versuchsplattform ist ein mobiler Manipulator bestehend aus der mobilen Plattform KUKA OmniRob und dem Leichtbauarm KUKA LWR 4 (siehe Abbildung 8.1a).

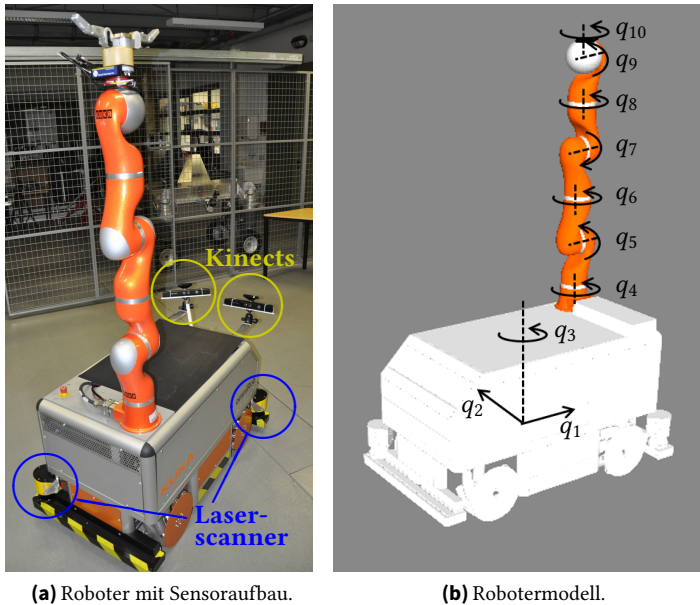


Abbildung 8.1: Versuchsplattform *IOSB.amp O2*.

Die mobile Plattform ist omnidirektional und besitzt drei Freiheitsgrade: translatorische Bewegungen vorwärts/rückwärts und rechts/links sowie die Rotation um die Hochachse. Die Plattformposition und -orientierung werden mit Hilfe der Plattformodometrie und einer kartenbasierten Lokalisierung ermittelt. Die Stellgrößen der mobilen Plattform sind die translatorischen Geschwindigkeiten v_x (vorwärts) und v_y (nach links) sowie die rotatorische Geschwindigkeit $\dot{\psi}$ um die Hochachse. Sie beziehen

sich auf das lokale Koordinatensystem der Plattform, d. h. auf die aktuelle Plattformposition und -orientierung.

Der Roboterarm besitzt sieben Rotationsgelenke. Als Stellgrößen dienen die Gelenkgeschwindigkeiten. Die aktuellen Gelenkpositionen und Gelenkgeschwindigkeiten liegen als Messgrößen vor. Zusätzlich werden in jedem Gelenk die auftretenden Drehmomente gemessen [Bis10]. Unter Verwendung des dynamischen Robotermodells wird auch eine Schätzung der externen Drehmomente bereitgestellt.

In dieser Arbeit wird der Roboterarm sowohl als Teil des mobilen Manipulators mit zehn Freiheitsgraden (*IOSB.amp O2*) als auch als eigenständiger, feststehender Manipulator mit sieben Freiheitsgraden verwendet, um die entwickelten Verfahren mit unterschiedlichen Kinematiken validieren zu können. Beide Kinematiken werden wie in Abschnitt 4.2 dargestellt als eine Kette von Translations- und Rotationsgelenken modelliert. Das Modell des mobilen Manipulators ist in Abbildung 8.1b beispielhaft dargestellt. Die Gelenke 1 und 2 beschreiben die translatorische Bewegung des Roboters in einem globalen Koordinatensystem. Gelenk 3 rotiert die Plattform um ihre Hochachse. Die Gelenke 4 bis 10 sind die Drehgelenke des Manipulators. Die Beschränkungen der einzelnen Gelenke sowie die in den Experimenten maximal zugelassenen Gelenkgeschwindigkeiten und -beschleunigungen sind in Tabelle 8.1 zu finden. Dabei ist zu beachten, dass die Plattformposition und die Drehung der Plattform unbeschränkt sind. Die Denavit-Hartenberg-Parameter der gesamten kinematischen Kette sind in Anhang B angegeben.

Die Modellierung der translatorischen Plattformbewegung in einem ortsfesten Koordinatensystem erfordert die Umrechnung der Gelenkgeschwindigkeiten w_1 , w_2 und w_3 in die Stellgrößen der Plattform v_x , v_y und $\dot{\psi}$, die sich auf das lokale Roboterkoordinatensystem beziehen:

$$\begin{bmatrix} v_x \\ v_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(q_3) & \sin(q_3) & 0 \\ -\sin(q_3) & \cos(q_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}. \quad (8.1)$$

Die Gelenkposition q_3 gibt die aktuelle Plattformorientierung an.

Tabelle 8.1: Gelenkbeschränkungen der Versuchsplattform *IOSB.amp O2*.

Gelenk i	$q_{\min,i}$	$q_{\max,i}$	$w_{\max,i}$	$a_{\max,i}$
1	$-\infty$	∞	0,7 m/s	0,4 m/s ²
2	$-\infty$	∞	0,7 m/s	0,4 m/s ²
3	$-\infty$	∞	40 °/s	23 °/s ²
4	-160 °	160 °	94,5 °/s	320 °/s ²
5	-120 °	120 °	94,5 °/s	320 °/s ²
6	-160 °	160 °	94,5 °/s	320 °/s ²
7	-120 °	120 °	94,5 °/s	320 °/s ²
8	-160 °	160 °	94,5 °/s	320 °/s ²
9	-120 °	120 °	94,5 °/s	320 °/s ²
10	-160 °	160 °	94,5 °/s	320 °/s ²

Als Werkzeug am Roboterflansch wird in den Versuchen entweder eine drei-fingrige Roboterhand (BarrettHand, siehe Abbildung 8.1a) oder eine Halterung mit Stift (siehe Abbildung 8.11) verwendet. Die Roboterhand wird eingesetzt, um Objekte zwischen verschiedenen Tischen transportieren zu können. Mit dem Stift kann der Roboter entsprechend den vorgegebenen Sollpfaden geometrische Formen zeichnen, sodass eine anschauliche Überprüfung der Genauigkeit der resultierenden Endeffektorbewegung möglich ist.

Zur Umgebungserfassung ist der Roboter mit mehreren Abstandssensoren ausgestattet (siehe Abbildung 8.1a). Die Sensoren sind auf der Plattform angebracht, um den großen Arbeitsraum des mobilen Manipulators mit wenigen Sensoren abdecken zu können. Ein weiterer Vorteil ist, dass Lokalisierungsfehler sich nicht auf den geschätzten Roboter-Hindernis-Abstand auswirken. Zwei 2D-Laserscanner (SICK S300) mit einem Gesichtsfeld von 270 ° sind an zwei gegenüberliegenden Ecken der Plattform montiert und überwachen eine Ebene rund um die Plattform. Sie erfassen unter anderem die Beine von Personen neben der Plattform und auf dem Boden stehende Gegenstände. Zwei Tiefenkameras (Xbox 360 Kinect) mit einem Gesichtsfeld von jeweils 57 ° × 43 ° beobachten die Umgebung des Roboterarms. Sie erfassen beispielsweise Körper und Arme von Personen, die mit dem Robo-

ter interagieren. Ihre Platzierung ist in Hinblick auf die Abdeckung eines möglichst großen Raums und die Reduzierung von Verdeckungen durch den Roboterarm gewählt. Für ihre extrinsische Kalibrierung wurde ein Verfahren entwickelt, das den Manipulator als Kalibrierobjekt verwendet und die 3D-Punktwolken der Sensoren mit dem 3D-Modell des Roboters vergleicht [Fre14].

8.1.2 Implementierung

Die entwickelten Verfahren wurden zur Analyse und Validierung auf dem Roboter in C++ implementiert und in das Robot Operating System (ROS) [ROS17] eingebunden. Dieses unterstützt einen modularen Aufbau der Software. Es stellt standardisierte Schnittstellen für den Datenaustausch zwischen den einzelnen Modulen zur Verfügung, beinhaltet eine Beschreibungssprache für Robotermodelle (Unified Robot Description Format (URDF)) und bietet die Möglichkeit, Ergebnisse in einer 3D-Visualisierung darzustellen.

Abbildung 8.2 zeigt eine Übersicht über das zur Evaluation verwendete Gesamtsystem. Die Lokalisierung der mobilen Plattform basiert auf der Plattformodometrie und nutzt die Messungen der Laserscanner zur globalen Referenzierung in einer Karte. Sie wird vom Roboterhersteller zur Verfügung gestellt.

Es liegt ein geometrisches 3D-Modell des Roboters inklusive Werkzeug in Form einer Beschreibung in URDF vor. A-priori bekannte statische Hindernisse liegen im Umgebungsmodell vor und werden ebenfalls in URDF beschrieben.

Unbekannte Hindernisse in der Roboterumgebung werden mit der in Kapitel 3 eingeführten Arbeitsraumüberwachung detektiert. Die Implementierung der Arbeitsraumüberwachung nutzt die Softwarebibliothek *OctoMap* [Hor13] für die Datenstruktur des Octree und zur Strahlverfolgung. Die Arbeitsraumüberwachung ist parametrierbar bezüglich der Anzahl der Sensoren, des jeweiligen Sensortyps und der Sensoreigenschaften sowie der Auflösung und Begrenzung des Octree zur 3D-Darstellung der Hindernisse. Auf dem Demonstrator werden die Daten der zwei Laserscanner und der zwei

Kinect-Sensoren verwendet (siehe Abbildung 8.1a). Die Ausfilterung des Roboters bzw. der bekannten Umgebung aus den Kinect-Daten erfolgt mit dem *Realtime URDF Filter* [Blo17] basierend auf den URDF-Beschreibungen von Roboter und Umgebung.

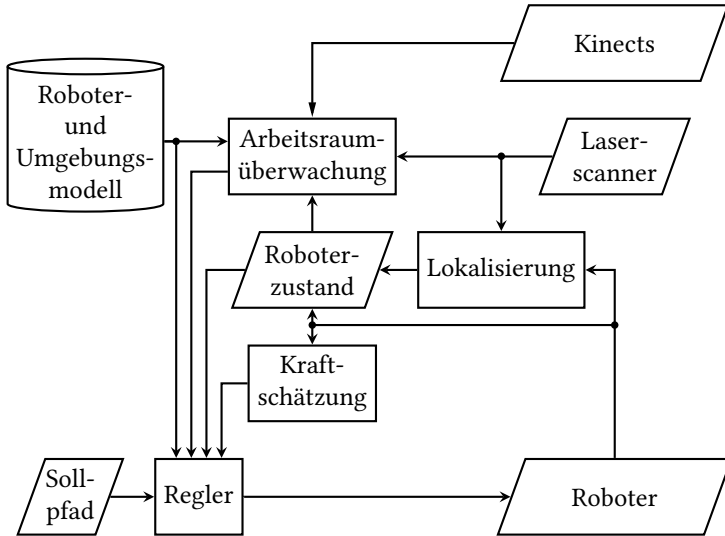


Abbildung 8.2: Gesamtsystem zur Evaluation der entwickelten Verfahren.

In den Experimenten werden dem Regler Sollpfade der Endeffektorpose in Form einer Wegpunktliste vorgegeben. Die Wegpunkte werden mit kubischen Splines interpoliert (siehe Anhang A), um einen zweimal stetig differenzierbaren Sollpfad zu erhalten. Die Regelung der Endeffektorpose entsprechend des gegebenen Sollpfads erfolgt mit der in Kapitel 5 vorgeschlagenen kartesischen modellprädiktiven Regelung. Das Optimierungsproblem der Regelung wird mit dem Verfahren der inneren Punkte mit Liniensuche (engl. *line search*) der Softwarebibliothek *IPOpt* [Wäc06] gelöst. Es ist auf die Lösung von Optimierungsproblemen mit vielen Optimierungsvariablen ausgelegt und nutzt die spezielle Struktur des Optimierungsproblems modellprädiktiver Regelungen. Das für die Prädiktion benötigte kinematische Robotermodell wird durch die Denavit-Hartenberg-Parameter spezifi-

ziert. So ist die Implementierung der Regelung durch Parametrierung auf unterschiedliche Roboterkinematiken mit beliebig vielen Gelenken anwendbar. Die für die Optimierung benötigten Jacobi- und Hessematrizen der Roboterkinematik werden basierend auf den Denavit-Hartenberg-Parametern explizit berechnet, um die Rechenzeit der Optimierung zu reduzieren.

Die für die Vermeidung von Eigen- und Fremdkollisionen (siehe Kapitel 6) benötigte Approximation der Robotergeometrie durch Kugeln wird automatisiert aus dem URDF-Modell des Roboters gewonnen. Für die Vermeidung von Kollisionen mit Hindernissen werden sowohl die modellierte statische Umgebung als auch der aus der Arbeitsraumüberwachung gewonnene Hindernis-Octree berücksichtigt. In jedem Reglerschritt wird für jede Roboterkugel der nächstliegende Punkt des Umgebungsmodells und der nächstliegende Hindernisknoten des Octree berechnet. Diese werden in der Regelung als statisches und dynamisches Hindernis für die jeweilige Roboterkugel berücksichtigt.

Für die Reaktion auf stattfindende Kontakte des Roboterarms mit seiner Umgebung (siehe Kapitel 7) sind sowohl die momentenbasierte als auch die kraftbasierte Erweiterung der Regelung umgesetzt. Die für die kraftbasierte Kontaktregelung benötigte Schätzung der Kontaktkraft erfolgt wie in [Lik14] vorgeschlagen unter Verwendung der externen Gelenkmomente.

Sämtliche Softwaremodule laufen auf einem auf dem Roboter verbauten Mini-PC mit einem 2,8 GHz Intel Core i7 Prozessor und einer Nvidia Quadro 410 Grafikkarte.

8.1.3 Simulation

Neben der Analyse der Verfahren auf dem realen Roboter wurden umfangreiche Simulationen der Roboterregelung durchgeführt. Die Simulation beruht auf dem in Abschnitt 4.2.6 eingeführten einfachen, diskreten Robotermodell des mobilen Manipulators ohne Berücksichtigung von Messungenauigkeiten, Beschleunigungsbeschränkungen oder Zeitverzögerungen. Auf Grund der modularen Softwarestruktur lässt sich die Ansteuerung des realen Roboters direkt durch das Simulationsmodell ersetzen, sodass die Simulation und

die Regelung des realen Roboters mit derselben Software erfolgen. Trotz der erwähnten Vereinfachungen entspricht das geregelte Roboterverhalten in der Simulation dem tatsächlichen Verhalten sehr gut.

8.2 Kartesische Roboterregelung

Im Folgenden werden mehrere Experimente der kartesischen modellprädiktiven Roboterregelung des in Abschnitt 8.1 beschriebenen stationären und mobilen Manipulators dargestellt und analysiert. Wenn nicht anders angegeben, werden für die Regelung eine Abtastzeit von 0,1 s und $N_p = 10$ Prädiktionsschritte verwendet, sodass sich ein Prädiktionshorizont von 1 s ergibt. Die Erweiterung zur Vermeidung von Eigenkollisionen ist während aller Experimente aktiv. Die Parameter des Gütekriteriums zur Pfad- bzw. Trajektorienverfolgung in den Gleichungen (5.61) und (5.62) bzw. (5.90) und (5.91) werden für den mobilen Manipulator *IOSB.amp O2* wie folgt gewählt:

$$\mathbf{Q}_e = \text{diag}(\eta_p, \eta_p, \eta_p, \eta_o, \eta_o, \eta_o, \eta_o) \quad (8.2)$$

$$\mathbf{Q}_q = \text{diag}(0, 0, 0, \eta_q, \eta_q, \eta_q, \eta_q, \eta_q, \eta_q) \quad (8.3)$$

$$\mathbf{Q}_w = \text{diag}(\eta_{w1}, \eta_{w1}, \eta_{w2}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}) \quad (8.4)$$

$$\mathbf{R}_e = 100 \mathbf{Q}_e \quad (8.5)$$

$$\mathbf{R}_q = \mathbf{0} \quad (8.6)$$

mit $\eta_p = 200 \text{ }^1/\text{m}^2$, $\eta_o = 150$, $\eta_q = 0,1 \text{ }^1/\text{rad}^2$, $\eta_{w1} = 7 \text{ }^2/\text{m}^2$, $\eta_{w2} = 7 \text{ }^2/\text{rad}^2$ und $\eta_{w3} = 5 \text{ }^2/\text{rad}^2$. \mathbf{Q}_e und \mathbf{R}_e gewichten die einzelnen Elemente des Positionsfehlers gleich stark. Ebenso werden die einzelnen Elemente des Orientierungsfehlers gleich stark gewichtet. Die Diagonalelemente in \mathbf{Q}_w werden ausreichend groß gewählt, um Schwingungen der Gelenkgeschwindigkeiten zu vermeiden. Rotationen der Armgelenke werden gegenüber Rotationen der Plattform leicht bevorzugt. Die den Plattformgelenken zugeordneten Elemente in \mathbf{Q}_q werden zu Null gewählt, da die Gelenke keine Positionsbeschränkung aufweisen und keine Bevorzugung von Gelenkstellungen in der Mitte des zulässigen Bereichs notwendig ist. Für die weiteren Elemente in \mathbf{Q}_q

werden kleine Gewichtungen gewählt, da Gelenkstellungen in der Mitte des zulässigen Bereichs nur zu bevorzugen sind, wenn hierdurch Positions- und Orientierungsfehler nicht wesentlich ansteigen. Gegenüber kleinen Parameteränderungen zeigt sich das Regelungsverhalten robust. Generell ist die Wahl geeigneter Reglerparameter jedoch abhängig vom gewünschten Roboterverhalten. Die aufgelisteten Parameter eignen sich zur Validierung des Regelungskonzepts in den im Folgenden dargestellten Demonstrationsszenarien. Die Wahl optimaler Parameter liegt jedoch nicht im Fokus dieser Arbeit.

8.2.1 Fehlermaße

Die Ergebnisanalyse betrachtet unter anderem den Positions- und Orientierungsfehler des Endeffektors. Im Fall einer Solltrajektorie gibt der Positionsfehler des Endeffektors den euklidischen Abstand zwischen der tatsächlich resultierenden Endeffektorposition $\mathbf{p}_{EE}(k)$ und der Sollposition des Endeffektors $\mathbf{p}_{traj}(k)$ in einem Zeitschritt k an:

$$e_p(k) := \|\mathbf{p}_{EE}(k) - \mathbf{p}_{traj}(k)\| , \quad (8.7)$$

wobei die resultierende Endeffektorposition mit Hilfe der direkten Roboterkinematik aus den gemessenen Gelenkwinkeln geschätzt wird, da keine Möglichkeit der direkten Messung der Endeffektorpose besteht. Dieser Positionsfehler entspricht dem Positionsanteil des in die Kostenfunktion eingehenden Trajektorienverfolgungsfehlers (siehe Gleichung (5.1)).

Aus Gründen der Anschaulichkeit wird für die Orientierung nicht ebenfalls der Orientierungsanteil des Trajektorienverfolgungsfehlers verwendet, sondern der Winkel zwischen der resultierenden Endeffektororientierung $\mathbf{o}_{EE}(k)$ und der Sollorientierung des Endeffektors $\mathbf{o}_{traj}(k)$. Der Orientierungsfehler zum Zeitpunkt k ist definiert als:

$$\epsilon(k) := \arccos \left(2 \langle \mathbf{o}_{EE}(k), \mathbf{o}_{traj}(k) \rangle^2 - 1 \right) , \quad (8.8)$$

wobei $\langle \cdot, \cdot \rangle$ das Skalarprodukt zweier Quaternionen darstellt.

Im Fall eines geometrischen Sollpfads werden analog der Positionsfehler

$$e_p(k) := \|\mathbf{p}_{EE}(k) - \mathbf{p}_{\text{pfad}}(\theta(k))\| \quad (8.9)$$

und der Orientierungsfehler

$$\epsilon(k) := \arccos \left(2 \langle \mathbf{o}_{EE}(k), \mathbf{o}_{\text{pfad}}(\theta(k)) \rangle^2 - 1 \right) \quad (8.10)$$

betrachtet. Diese geben den euklidischen Abstand bzw. den Winkel zwischen der aktuellen Endeffektorpose und der auf Grund des ermittelten Pfadparameters $\theta(k)$ berechneten Sollpose an.

Zusätzlich ist bei Pfaden ein weiteres Fehlermaß üblich. Dieses Fehlermaß betrachtet den Abstand der Endeffektorposition zum Sollpfad

$$d_{\text{pfad}}(k) := \min_{\theta} (\|\mathbf{p}_{EE}(k) - \mathbf{p}_{\text{pfad}}(\theta)\|) . \quad (8.11)$$

Dieses Abstandsmaß berücksichtigt die Aufgabe der Pfadverfolgung besser, dass der Endeffektor sich auf dem vorgegebenen Pfad befinden soll, jedoch keine zeitliche Vorgabe erfolgt, wann sich der Endeffektor an einer bestimmten Stelle des Pfads befinden muss. Besteht der Sollpfad jedoch nicht nur aus der Sollposition, sondern enthält auch die Sollorientierung, würde ein Abstandsmaß benötigt, das sowohl die Position als auch die Orientierung beinhaltet. Werden Positions- und Orientierungsfehler getrennt betrachtet mit dem Orientierungsfehler

$$\alpha_{\text{pfad}}(k) := \min_{\theta} (\|\mathbf{o}_{EE}(k) - \mathbf{o}_{\text{pfad}}(\theta)\|) , \quad (8.12)$$

wird die notwendige Synchronisierung von Position und Orientierung außer Acht gelassen. Deshalb wird dieses Fehlermaß in dieser Arbeit nicht betrachtet, zumal es nie größere Werte als den Positionsfehler e_p und den Orientierungsfehler ϵ liefert.

8.2.2 Verfolgen von Sollpfaden

Die beiden in Abschnitt 5.4 vorgestellten Ansätze zur Folgeregelung eines Sollpfads der Endeffektorpose werden anhand eines Simulationsbeispiels des mobilen Manipulators *IOSB.amp O2* miteinander verglichen. Der gegebene geometrische Sollpfad ist in Abbildung 6.2 auf Seite 94 dargestellt. Die Orientierung ist über den gesamten Pfad konstant. Hindernisse werden zunächst nicht betrachtet.

Der erste Ansatz beruht auf Verwendung der vorgegebenen kartesischen Geschwindigkeiten (siehe Abschnitt 5.4.1). Die Pfadparametergeschwindigkeit wird so gewählt, dass die translatorische Pfadgeschwindigkeit $v_{\text{pfad}} = 0,16 \text{ m/s}$ erreicht wird. Die Pfadgeschwindigkeit ist so gewählt, dass zum Abfahren des Pfads ungefähr die gleiche Zeit wie beim zweiten Ansatz benötigt wird. Die rotatorische Pfadgeschwindigkeit hat auf Grund der konstanten Orientierung keinen Einfluss. Zusätzlich findet ein Abbremsvorgang statt, wenn der Abstand zum Pfadende den Schwellwert $d_a = 0,2 \text{ m}$ unterschreitet.

Im zweiten Ansatz wird in jedem Reglerschritt die optimale Pfadparametergeschwindigkeit berechnet. Die Gewichtungparameter des Gütekriteriums in den Gleichungen (5.89)–(5.92) sind $Q_{\dot{\theta}} = 0,2 \text{ s}^2$, $Q_{\theta} = 0,1$ und $R_{\theta} = 10$. Die maximale Pfadparametergeschwindigkeit wird so gewählt, dass sie der maximalen translatorischen Geschwindigkeit $v_{\text{pfad,max}} = 0,25 \text{ m/s}$ entspricht.

Abbildung 8.3 zeigt für beide Fälle die translatorische Endeffektorgeschwindigkeit v_{EE} , den Positionsfehler e_p , den Orientierungsfehler ϵ sowie die für einen Reglerschritt benötigte Rechenzeit. Die Endeffektorgeschwindigkeit ist bei vorgegebener Geschwindigkeit wie erwartet nahezu konstant mit Ausnahme des Abbremsens am Pfadende. Im Fall der Pfadoptimierung variiert die Endeffektorgeschwindigkeit deutlich und nimmt teils größere und teils kleinere Werte an. Das Abbremsen am Pfadende ist in beiden Fällen sehr ähnlich. Durch die Optimierung der Pfadparametergeschwindigkeit bleiben Positions- und Orientierungsfehler kleiner als bei vorgegebener konstanter Geschwindigkeit. Der Ansatz der Pfadoptimierung erzielt also die besseren Ergebnisse. Der Nachteil ist jedoch die deutlich längere Rechenzeit der Optimierung in einem Reglerschritt. Diese ist so gravierend, dass das

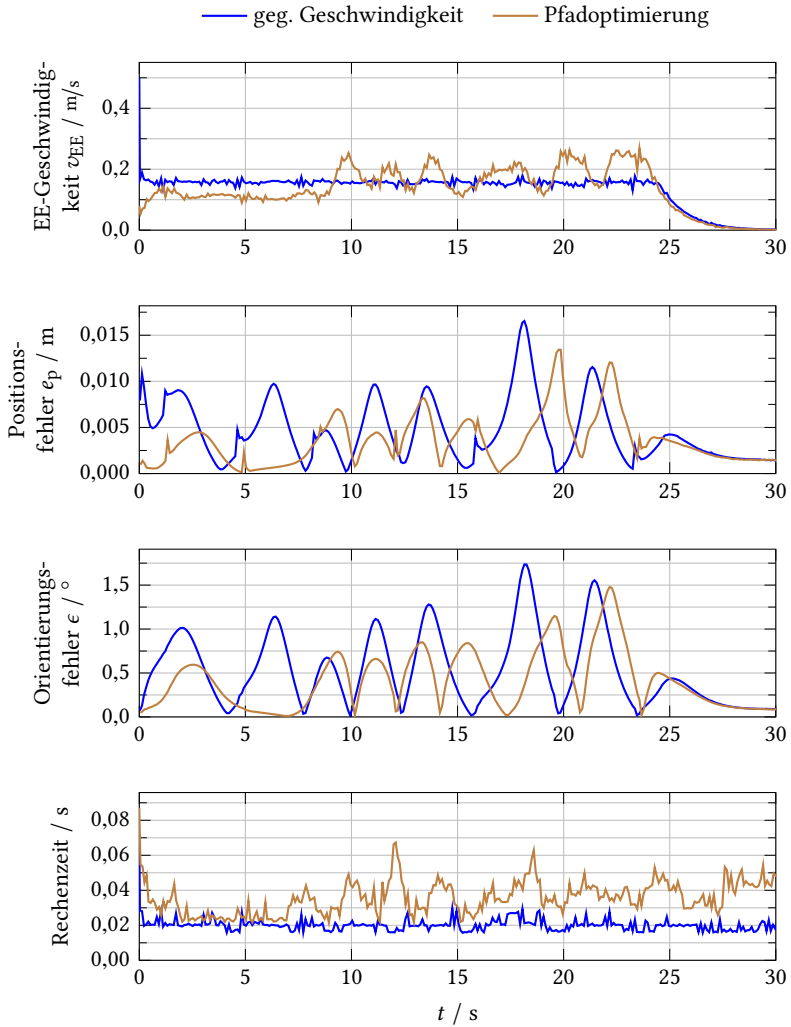


Abbildung 8.3: Simulationsergebnisse der kartesischen modellprädiktiven Regelung mit verschiedenen Ansätzen zur Pfadverfolgung.

Optimierungsproblem nicht mehr innerhalb der Abtastzeit $T_s = 0,1$ s gelöst werden kann, wenn zusätzliche Beschränkungen zur Vermeidung von Kollisionen mit Hindernissen berücksichtigt werden müssen. Deshalb werden im Folgenden die Sollpfade mit den vorgegebenen maximalen kartesischen Geschwindigkeiten abgefahren.

Die erzielte Positionsgenauigkeit des Reglers liegt sowohl mit als auch ohne Pfadoptimierung im Bereich der Positionsgenauigkeit anderer Regelansätze, soweit deren Ergebnisse in der Literatur dargestellt werden, obwohl dabei Roboter mit weniger Freiheitsgraden betrachtet werden und lediglich die Endeffektorposition ohne Berücksichtigung der Orientierung vorgegeben wird. So wird in [Pap00] die 2D-Endeffektorposition eines mobilen Manipulators mit fünf Freiheitsgraden entlang einer Trajektorie geregelt. In der Simulation liegen die Positionsfehler der x - und y -Achse im Bereich von $-0,6$ cm bis $0,2$ cm bzw. $-0,1$ cm bis $0,8$ cm. In [And12a] wird die 3D-Endeffektorposition eines realen mobilen Manipulators mit sechs Freiheitsgraden geregelt. Der Pfadverfolgungsfehler beträgt dabei $0,5$ cm bis 5 cm. Im Vergleich dazu liegt der Positionsfehler in der hier durchgeführten Simulation im Bereich von $0,1$ mm bis $1,6$ cm. Ein exakter Vergleich ist jedoch nicht möglich, da die zu befahrenden Pfade bzw. Trajektorien und die Endeffektorgeschwindigkeiten variieren.

Wird eine höhere Genauigkeit der Endeffektorpose benötigt, lässt sich diese mit der kartesischen modellprädiktiven Regelung erzielen, indem die Gewichtung des Pfadverfolgungsfehlers Q_e vergrößert wird. Die höhere Positions- und Orientierungsgenauigkeit wird dabei mit größeren Stellgrößen und größeren Stellgrößenänderungen erkauft.

8.3 Kollisionsvermeidung

Die Ergebnisse der Kollisionsvermeidung werden anhand verschiedener Szenarien auf dem realen mobilen Manipulator analysiert. Die Hindernisse werden mit Hilfe der in Kapitel 3 vorgestellten Arbeitsraumüberwachung detektiert. Die Kollisionsvermeidung erfolgt mit der in Kapitel 6 eingeführten Erweiterung der kartesischen modellprädiktiven Regelung.

8.3.1 Arbeitsraumüberwachung

Voraussetzung für die Vermeidung von Kollisionen mit dynamischen Hindernissen ist die Arbeitsraumüberwachung zur Detektion dieser Hindernisse. Die Arbeitsraumüberwachung für den mobilen Manipulator *IOSB.amp O2* erfolgt mit den in Kapitel 3 vorgestellten Verfahren. Sie beruht auf den Sensordaten der zwei Laserscanner und der zwei Tiefenkameras (Kinect-Sensoren), die auf der Plattform verbaut sind. Die Parameter der Arbeitsraumüberwachung sind in Tabelle 8.2 angegeben.

Tabelle 8.2: Parameter der Arbeitsraumüberwachung

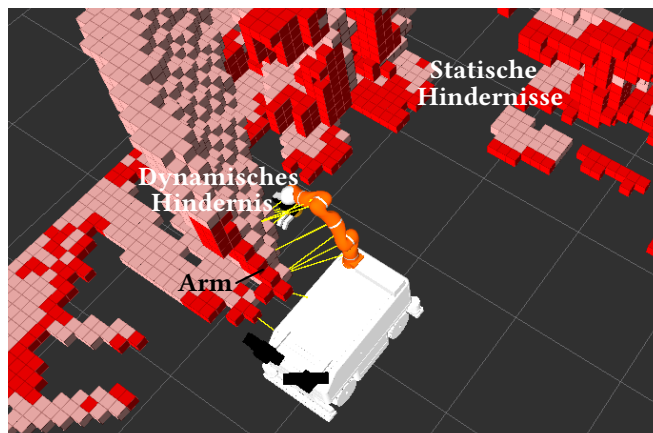
Parameter	Wert	Einheit
<i>2D-Laserscanner</i>		
Horizontaler Öffnungswinkel	270	°
Horizontale Auflösung	0,5	°
Maximale Reichweite	29	m
<i>Tiefenkamera (Kinect-Sensor)</i>		
Horizontaler Öffnungswinkel	57	°
Vertikaler Öffnungswinkel	43	°
Horizontale Auflösung	640	Pixel
Vertikale Auflösung	480	Pixel
Maximale Reichweite	7	m
<i>3D-Hindernisdarstellung</i>		
Auflösung in x-, y- und z-Richtung	0,1	m

In Abbildung 8.4 sind zwei Schnappschüsse der Visualisierung der Arbeitsraumüberwachung dargestellt. Sie zeigen neben dem Modell des Roboters den aktuellen Hindernis-Octree, wobei die durch Hindernisse belegten Zellen rot und die durch Hindernisse verdeckten Zellen rosa dargestellt sind. Zusätzlich sind die minimalen Abstände zwischen der Hindernisdarstellung und den Mittelpunkten der Roboterkugeln, die in der Kollisionsvermeidung der Regelung verwendet werden, abgebildet. Während in Abbildung 8.4a kein Vorwissen über statische Hindernisse in der Roboterumgebung vorliegt, wird in Abbildung 8.4b das dargestellte Umgebungsmodell berücksichtigt. Es enthält Wände und Tische der Roboterum-

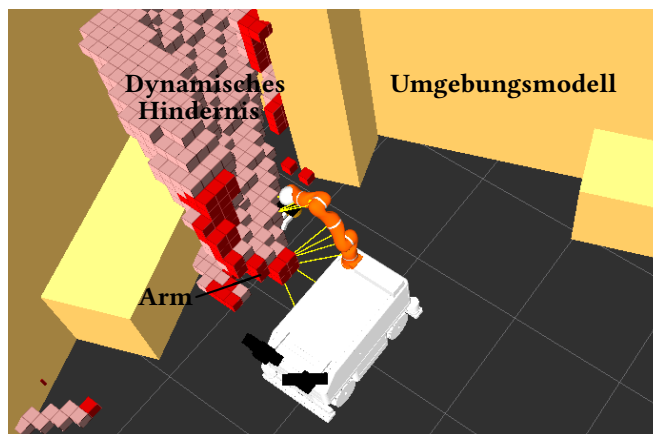
gebung. Dementsprechend ist in der Octree-Repräsentation der Hindernisse in Abbildung 8.4b eine wesentlich geringere Anzahl von Hindernissen enthalten.

In den beiden dargestellten Schnappschüssen befindet sich neben dem Roboter ein Mensch, der seinen Arm nach dem Roboter ausstreckt. Da sich der Arm zwischen dem Roboter und den die Umgebung des Roboterarms überwachenden Tiefenkameras befindet, wird die Notwendigkeit der Berücksichtigung von Verdeckungen deutlich: Die Sensoren erfassen nur den Teil des Arms, der den Sensoren zugewandt ist. Bereiche, die näher am Roboter liegen, können nicht erfasst werden. Würden nur die erfassten Hindernisse berücksichtigt, würde ein zu großer Roboter-Hindernis-Abstand ermittelt werden. Mit Hilfe der verdeckten (rosa dargestellten) Hinderniszellen wird der Roboter-Hindernis-Abstand dagegen konservativ abgeschätzt. So kann die Sicherheit des Menschen während der Interaktion gewährleistet werden.

Die benötigte Rechenzeit zur Ermittlung der Hindernisrepräsentation ist abhängig von der Anzahl und Größe der Hindernisse im zu überwachenden Bereich sowie von der Größe des zu überwachenden Bereichs. Es wird hier ein Beispielszenario betrachtet, in dem sich ein Mensch mehrfach dem Roboter nähert und den Arm auch zwischen den Roboterarm und die Tiefenkameras bewegt, wodurch große Verdeckungen entstehen. Zudem befinden sich weitere Hindernisse wie Tische oder Wände im Nahbereich des Roboters. Die Arbeitsraumüberwachung wird sowohl für den Fall eines bekannten Umgebungsmodells, das Wände und Tische in der Roboterumgebung bereits enthält (siehe Abbildung 8.4b), als auch ohne Vorwissen durchgeführt. Dabei wird zur Bestimmung des durch den Roboter erreichbaren Bereichs ein Zeithorizont von 1 s und die maximal möglichen Robotergereschwindigkeiten verwendet. Der Zeithorizont von 1 s entspricht dem für die Regelung verwendeten Prädiktionshorizont. Die maximale Ausdehnung des Octree beträgt in diesem Fall $7,6 \text{ m} \times 7,4 \text{ m} \times 2 \text{ m}$. Zum Vergleich wird das gleiche Szenario auch mit einem Zeithorizont von 0,5 s überwacht. Das entspricht einer maximalen Octree-Ausdehnung von $3,8 \text{ m} \times 3,7 \text{ m} \times 2 \text{ m}$. Die Ergebnisse der benötigten Rechenzeiten sind in den Tabellen 8.3 bis 8.6 ange-



(a) Ohne Umgebungsmodell.



(b) Mit Umgebungsmodell.

Abbildung 8.4: Visualisierung der 3D-Hindernisdarstellung: belegte Hinderniszellen (rot), verdeckte Hinderniszellen (rosa), Abstände zwischen Hindernisdarstellung und Mittelpunkten der Roboterkugeln (gelbe Linien).

geben. Dabei erfolgt die Filterung der Daten der einzelnen Kinect-Sensoren, die Verdeckungsanalyse der Hindernispunkte der einzelnen Sensoren, die Verdeckungsanalyse der Roboterpunkte der einzelnen Sensoren, die Datenfusion und die Abstandsberechnung in parallelen Prozessen. Die Datenfusion verwendet die jeweils aktuellsten Ergebnisse der Verdeckungsanalysen der einzelnen Sensoren. Das *Realtime URDF Filter* zur Filterung der Roboterpunkte und des Umgebungsmodells benötigt in allen Fällen zur Verarbeitung eines Tiefenbildes im Mittel 39 ms (mindestens 31 ms, maximal 48 ms).

Tabelle 8.3: Rechenzeiten der Arbeitsraumüberwachung mit Zeithorizont 1 s ohne Umgebungsmodell.

Prozess	Mittlere Rechenzeit	Minimale Rechenzeit	Maximale Rechenzeit
<i>Vorverarbeitung und Verdeckungsanalyse der Hindernispunkte</i>			
Laserscanner 1	2,4 ms	0,6 ms	6,3 ms
Laserscanner 2	2,5 ms	0,7 ms	7,3 ms
Kinect 1	170,7 ms	91,0 ms	244,2 ms
Kinect 2	193,5 ms	141,7 ms	269,1 ms
<i>Vorverarbeitung und Verdeckungsanalyse der Roboterpunkte</i>			
Kinect 1	6,8 ms	4,1 ms	13,3 ms
Kinect 2	8,2 ms	5,5 ms	15,5 ms
Datenfusion	5,6 ms	2,8 ms	30,5 ms
Abstandsberechnung	0,5 ms	0,2 ms	4,7 ms

In allen Fällen sind die mittleren Zeiten für die Verarbeitung der Laserscannerdaten mit 0,3 ms bis 2,5 ms, für die Datenfusion mit 1,0 ms bis 5,6 ms und die Abstandsberechnung mit 0,1 ms bis 0,5 ms sehr gering. Somit werden die durch die Laserscanner erfassten Hindernisse sehr schnell detektiert, im Mittel nach 1,3 ms bis 8,1 ms nach Erfassung durch den Sensor. Diese Zeit umfasst die Vorverarbeitung und Verdeckungsanalyse des Laserscanners und die Datenfusion. Längere Rechenzeiten entstehen bei der Verarbeitung der Kinect-Daten. Dabei fällt zunächst auf, dass die Rechenzeit zur Verarbeitung der Daten der Kinect 2 länger ist als die der Kinect 1. Die Ursache hierfür ist die größere Anzahl bzw. der größere Umfang der Hindernisse im Gesichtsfeld der Kinect 2. Des Weiteren hat das Vorhandensein eines Umge-

Tabelle 8.4: Rechenzeiten der Arbeitsraumüberwachung mit Zeithorizont 1 s mit Umgebungsmodell.

Prozess	Mittlere Rechenzeit	Minimale Rechenzeit	Maximale Rechenzeit
<i>Vorverarbeitung und Verdeckungsanalyse der Hindernispunkte</i>			
Laserscanner 1	0,7 ms	0,2 ms	3,8 ms
Laserscanner 2	0,7 ms	0,2 ms	3,2 ms
Kinect 1	17,0 ms	10,8 ms	32,0 ms
Kinect 2	17,4 ms	9,4 ms	35,1 ms
<i>Vorverarbeitung und Verdeckungsanalyse der Roboterpunkte</i>			
Kinect 1	21,1 ms	13,6 ms	31,9 ms
Kinect 2	25,3 ms	18,9 ms	38,5 ms
Datenfusion	2,7 ms	0,6 ms	24,2 ms
Abstandsberechnung	0,3 ms	0,1 ms	2,2 ms

Tabelle 8.5: Rechenzeiten der Arbeitsraumüberwachung mit Zeithorizont 0,5 s ohne Umgebungsmodell.

Prozess	Mittlere Rechenzeit	Minimale Rechenzeit	Maximale Rechenzeit
<i>Vorverarbeitung und Verdeckungsanalyse der Hindernispunkte</i>			
Laserscanner 1	0,4 ms	0,2 ms	1,5 ms
Laserscanner 2	0,5 ms	0,2 ms	1,8 ms
Kinect 1	36,3 ms	27,3 ms	67,1 ms
Kinect 2	65,0 ms	52,3 ms	92,9 ms
<i>Vorverarbeitung und Verdeckungsanalyse der Roboterpunkte</i>			
Kinect 1	4,4 ms	3,4 ms	7,7 ms
Kinect 2	5,0 ms	3,9 ms	9,0 ms
Datenfusion	1,1 ms	0,4 ms	9,7 ms
Abstandsberechnung	0,2 ms	0,1 ms	2,8 ms

Tabelle 8.6: Rechenzeiten der Arbeitsraumüberwachung mit Zeithorizont 0,5 s mit Umgebungsmodell.

Prozess	Mittlere Rechenzeit	Minimale Rechenzeit	Maximale Rechenzeit
<i>Vorverarbeitung und Verdeckungsanalyse der Hindernispunkte</i>			
Laserscanner 1	0,3 ms	0,1 ms	1,9 ms
Laserscanner 2	0,4 ms	0,1 ms	2,5 ms
Kinect 1	3,1 ms	1,2 ms	14,5 ms
Kinect 2	13,9 ms	8,0 ms	24,8 ms
<i>Vorverarbeitung und Verdeckungsanalyse der Roboterpunkte</i>			
Kinect 1	8,4 ms	4,1 ms	14,1 ms
Kinect 2	10,5 ms	6,8 ms	16,5 ms
Datenfusion	1,0 ms	0,1 ms	20,1 ms
Abstandsberechnung	0,1 ms	0,02 ms	1,0 ms

bungsmodells einen großen Einfluss auf die Rechenzeit, da sich dadurch die Anzahl der Hindernispunkte deutlich reduziert und stattdessen die Anzahl der Roboterpunkte (die auch das Umgebungsmodell beinhalten) vergrößert. So liegen die mittleren Rechenzeiten zur Verarbeitung der Kinect-Daten bei 2,9 ms bis 17,4 ms mit vorhandenem Umgebungsmodell und 36,4 ms bis 193,5 ms ohne Umgebungsmodell. Der berücksichtigte Zeithorizont bzw. die Ausmaße des Hindernis-Octree haben ebenfalls einen großen Einfluss auf die benötigte Rechenzeit, insbesondere wenn viele Hindernispunkte vorhanden sind. Die gesamte Verzögerung zwischen Erfassen eines Hindernisses durch die Kinect-Sensoren und Berechnung des Abstands zwischen diesem Hindernis und dem Roboter beträgt damit im Mittel 43 ms bis 239 ms. Diese Zeit umfasst die Filterung des Roboters und gegebenenfalls des Umgebungsmodells, die Vorverarbeitung und Verdeckungsanalyse des Kinect-Sensors, die Datenfusion und die Abstandsberechnung. Insbesondere im Fall des längeren Zeithorizonts und ohne Umgebungsmodell ist die benötigte Zeit deutlich länger als die zur Detektion durch die Laserscanner. In der Anwendung zeigt sich, dass die Rechenzeiten jedoch klein genug sind, um eine ausreichend schnelle Reaktion des Roboters zu ermöglichen. Auch zeigt sich, dass die Verwendung des kürzeren Zeithorizonts in vielen Fällen ausreichend ist.

8.3.2 Kollisionsvermeidung mit dynamischem Hindernis

Die Kollisionsvermeidung wird zunächst in einem Beispielszenario betrachtet, in dem der mobile Manipulator *IOSB.amp O2* dem bereits in Abschnitt 6.7 zum Vergleich der Auswahl der relevanten Hindernisse betrachteten Pfad folgt. Dieser Pfad ist in Abbildung 8.5 als grüne Linie dargestellt. Die Roboterumgebung wird mit der Arbeitsraumüberwachung erfasst. Es liegt kein Umgebungsmodell vor. Nahe des Roboters befinden sich Wände und Tische. Des Weiteren hält sich ein Mensch im Arbeitsbereich des Roboters auf, der sich während des Versuchs dem Roboter nähert und einen Arm nach dem Roboter ausstreckt.

Die kartesische modellprädiktive Regelung wird mit der Erweiterung zur Kollisionsvermeidung mit dynamischen Hindernissen eingesetzt. Die Parameter der Kostenfunktion zur Kollisionsvermeidung sind $\lambda_{O,i} = 0,1 \text{ m}^2$ für alle Roboterkugeln $i = 1, \dots, N_R$ und $f_O = 20$.

Die Schnappschussesequenz in Abbildung 8.5 zeigt für mehrere Zeitpunkte das Ergebnis der Arbeitsraumüberwachung, die Abstände zwischen der Hindernisdarstellung und den Mittelpunkten der Roboterkugeln sowie die resultierende Roboterbewegung. Die den Mensch repräsentierenden Hinderniszellen befinden sich in Abbildung 8.5b-f links vom Roboter. Auf der anderen Seite des Roboters sind Repräsentationen der durch die Laser-scanner beobachteten Wand zu sehen.

Trotz des sich bewegenden Hindernisses, das sich dem Roboter aktiv nähert, ist der Roboter in der Lage, dem Sollpfad ohne Kollision zu folgen. Dies wird durch die Erweiterung des kartesischen modellprädiktiven Reglers möglich, die eine Ausweichbewegung des Roboters bewirkt. Diese Ausweichbewegung ist in Abbildung 8.6 zu erkennen: Die Abbildung zeigt die Position des Plattformmittelpunkts auf der horizontalen xy -Ebene während des Abfahrens des Sollpfads sowohl mit als auch ohne Einfluss des dynamischen Hindernisses. Wenn sich der Mensch dem Roboter nähert, bewegt sich die Plattform rückwärts (in positive x -Richtung), um den Abstand zum Menschen möglichst groß zu halten. Aufgrund der Ausweichbewegung unterscheiden sich auch die Plattformpositionen am Ende des Pfades. Dabei nimmt jedoch der Endeffektor in beiden Fällen die gleiche Pose ein. Lediglich

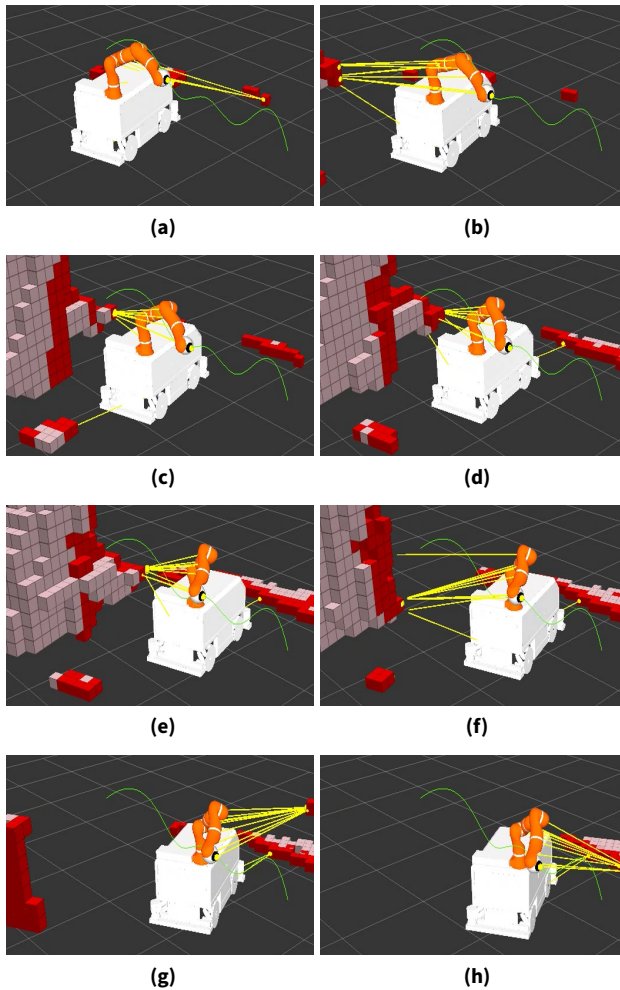


Abbildung 8.5: Schnappschusssequenz der Kollisionsvermeidung des mobilen Manipulators mit einem dynamischen Hindernis mit kartesischem Sollpfad (grüne Linie), Hindernis-Octree mit belegten (rot) und verdeckten Zellen (rosa) und Abständen zwischen Hindernisdarstellung und Mittelpunkten der Roboterkugeln (gelbe Linien) (aus [Zub15b]).

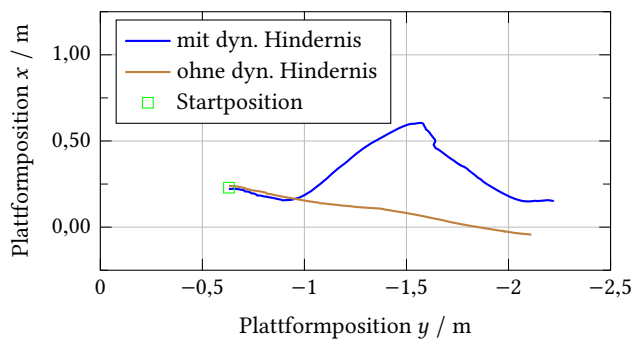
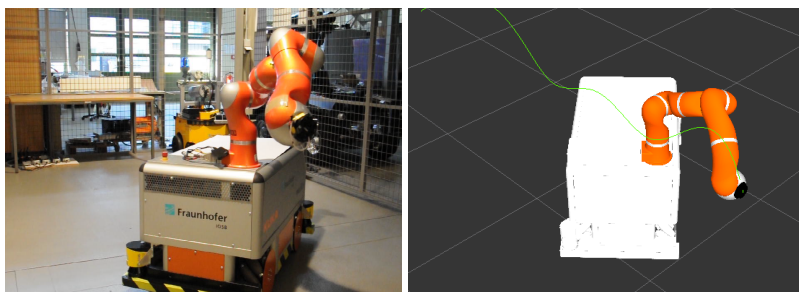
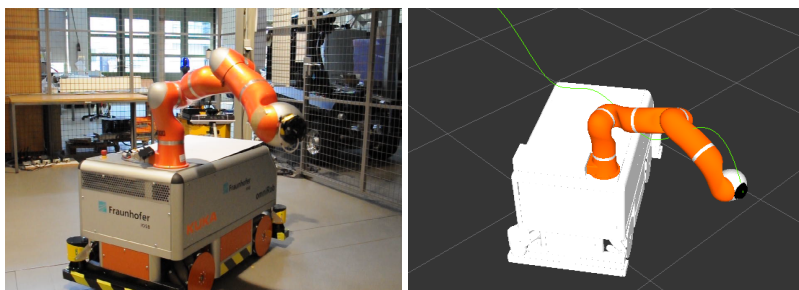


Abbildung 8.6: Plattformposition des mobilen Manipulators mit und ohne Vermeidung eines dynamischen Hindernisses.



(a) Mit dynamischem Hindernis.



(b) Ohne dynamisches Hindernis.

Abbildung 8.7: Vergleich der Gelenkkonfiguration des mobilen Manipulators am Pfadende mit und ohne Vermeidung eines dynamischen Hindernisses.

die Gelenkkonfiguration unterscheidet sich auf Grund der Ausweichbewegung. Dies ist in Abbildung 8.7 zu sehen, die die Gelenkkonfiguration des mobilen Manipulators am Pfadende sowohl für das Experiment mit dynamischem Hindernis als auch für das gleiche Experiment ohne dynamisches Hindernis zeigt.

Abbildung 8.8 zeigt den Positions- und Orientierungsfehler, den minimalen Abstand zwischen der Hindernisdarstellung und den Kugeln der Roboterapproximation sowie die benötigte Rechenzeit für einen Reglerschritt. Während der Annäherung des Menschen an den Roboter (im Zeitabschnitt zwischen 15 s und 20 s) treten der größte Positionsfehler mit 3,2 cm und der größte Orientierungsfehler mit $3,1^\circ$ auf. Ohne störenden Einfluss durch das dynamische Hindernis liegen Positions- und Orientierungsfehler meist in einer Größenordnung von 3 mm bzw. $0,4^\circ$. Auf Grund der Ausweichbewegung sinkt der minimale Abstand zwischen Hindernisdarstellung und Roboterapproximation nicht unter 27 cm. Die Rechenzeit für einen Reglerschritt beträgt während des ganzen Experiments maximal 60 ms und ist damit für eine Abtastzeit von 100 ms ausreichend klein.

Damit liegt der Positionsfehler im Bereich des in [And12a] mit dem Ansatz der priorisierten Aufgaben erzielten Pfadverfolgungsfehlers. Im Gegensatz zu [And12a] berücksichtigt die kartesische modellprädiktive Regelung jedoch auch die Orientierung des Endeffektors. Es werden Kollisionen mit dem gesamten Roboter, nicht nur mit der Plattform, vermieden. Zusätzlich berücksichtigt die kartesische modellprädiktive Regelung harte Nebenbedingungen zur Fremd- und Eigenkollisionsvermeidung.

8.3.3 Kollisionsvermeidung mit statischen und dynamischen Hindernissen

In einem zweiten Anwendungsbeispiel wird die Kollisionsvermeidung auf dem mobilen Manipulator *IOSB.amp O2* unter Verwendung eines bekannten Umgebungsmodells und in Gegenwart eines dynamischen Hindernisses betrachtet. Das Umgebungsmodell enthält als feststehende Einrichtungsgegenstände die Wände und Tische der Roboterzelle (siehe Abbildung 8.9).

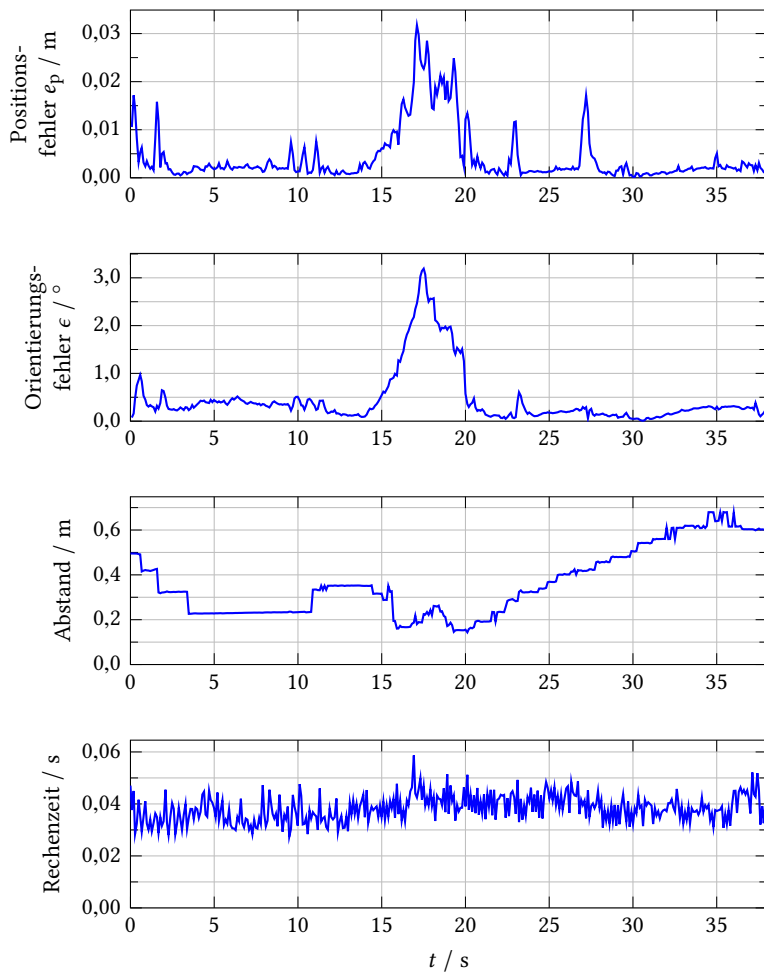
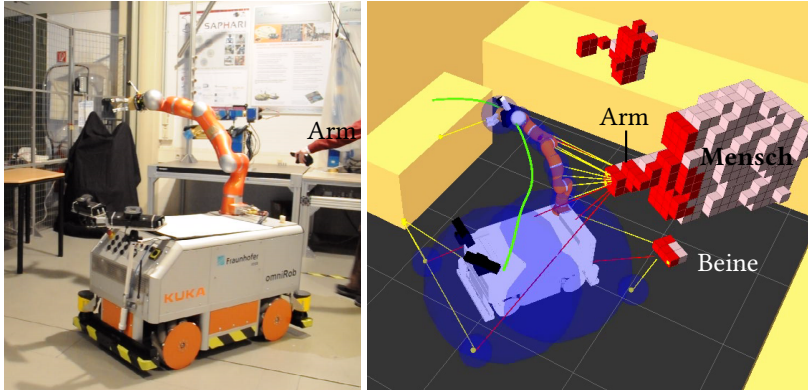


Abbildung 8.8: Ergebnisse der Kollisionsvermeidung des mobilen Manipulators mit einem dynamischen Hindernis.

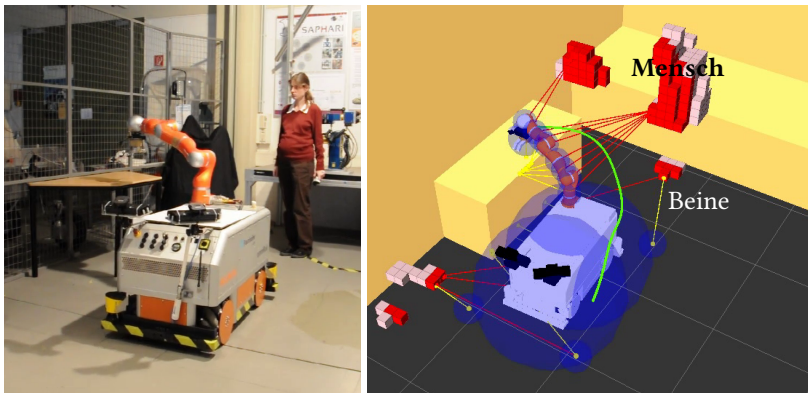
Diese statischen Hindernisse werden lediglich in den Beschränkungen der Kollisionsvermeidung berücksichtigt, da eine Annäherung des Roboters an diese Gegenstände erlaubt ist. Alle anderen Objekte in der Roboterumgebung wie Stühle, Gegenstände auf den Tischen oder Menschen werden durch die 3D-Arbeitsraumüberwachung erfasst und als mögliche dynamische Hindernisse interpretiert. Sie werden sowohl in den Beschränkungen als auch in der Kostenfunktion der Kollisionsvermeidung berücksichtigt. Der Sollpfad der Endeffektorpose führt ausgehend von der Startposition des Roboters über einen der Tische in der Roboterzelle (siehe Abbildung 8.9). Dies entspricht einer häufig auszuführenden Bewegung beim Greifen oder Absetzen von Objekten auf einem Tisch. Während des Experiments nähert sich ein Mensch dem Roboter und zwingt ihn dadurch eine Ausweichbewegung durchzuführen, während er dem vorgegeben Pfad folgt.

Abbildung 8.9 zeigt zwei Schnappschüsse des Experiments. Neben dem Sollpfad (grün), dem Umgebungsmodell (gelb) und der aktuellen 3D-Hindernisdarstellung zeigen sie die Approximation der Robotergeometrie durch Kugeln inklusive des Sicherheitsabstands (blau). Die Robotergeometrie wird mit $N_R = 15$ Kugeln approximiert. Die Kugeln inklusive Sicherheitsabstand der mobilen Plattform sind deutlich größer als die Plattform selbst, da auf Grund der großen Masse und der möglichen hohen Geschwindigkeiten ein großer Sicherheitsabstand eingehalten werden muss. Des Weiteren sind die minimalen Abstände zwischen Roboter und Hindernissen, die in den Beschränkungen (gelbe Linien) bzw. in der Kostenfunktion (rote Linien) berücksichtigt werden, dargestellt. Der Schnappschuss zum Zeitpunkt $t = 19$ s in Abbildung 8.9a zeigt, wie sich ein Mensch dem Roboter von rechts nähert. Der Mensch wird dabei durch die Kinect-Sensoren und die Laserscanner erfasst. Der zweite Schnappschuss in Abbildung 8.9b zeigt den Roboter am Pfadende. Der Endeffektor des Roboters befindet sich über dem im Umgebungsmodell enthaltenen Tisch.

Die resultierenden Positions- und Orientierungsfehler, minimalen Abstände zwischen Roboter und Hindernissen sowie die benötigte Rechenzeit für einen Reglerschritt während des Experiments sind in Abbildung 8.10 zu sehen. Während sich der Roboter auf den Tisch zubewegt, liegen Positions-



(a) Annäherung eines dynamischen Hindernisses ($t = 19$ s).



(b) Endpose ($t = 34$ s).

Abbildung 8.9: Schnappschüsse der Kollisionsvermeidung des mobilen Manipulators mit statischen und dynamischen Hindernissen mit kartesischem Sollpfad (grüne Linie), Umgebungsmodell (gelb), Hindernis-Octree mit belegten (rot) und verdeckten Zellen (rosa), Roboterapproximation (blau) und Abständen des Roboters zu statischen und dynamischen Hindernissen (für Kostenfunktion rote Linien, für Beschränkungen gelbe Linien).

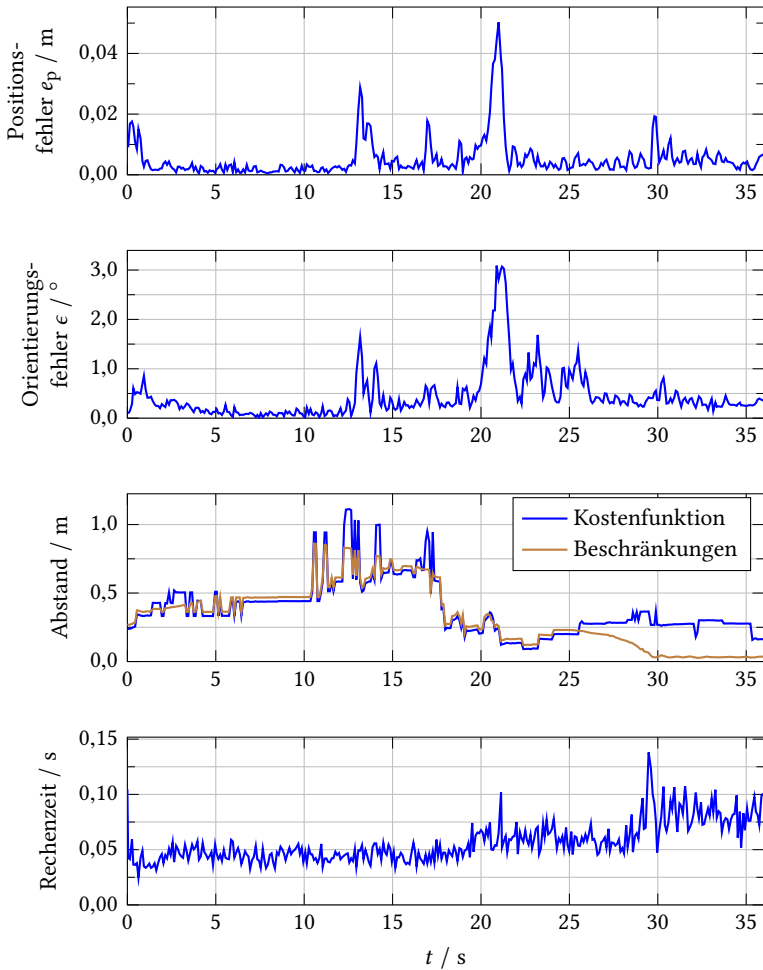


Abbildung 8.10: Ergebnisse der Kollisionsvermeidung mit statischen und dynamischen Hindernissen.

und Orientierungsfehler hauptsächlich in einem Bereich von bis zu 1 cm bzw. $0,3^\circ$ und sind damit für den Anwendungsfall ausreichend klein. Die zwei Bereiche mit größeren Fehlern von bis zu 3 cm bzw. 2° sind auf starke Bodenunebenheiten zurückzuführen. Ab dem Zeitpunkt $t = 18$ s nähert sich ein Mensch dem Roboter. Auf Grund der aus der Kollisionsvermeidung resultierenden Ausweichbewegung kann der Roboter einen Kontakt mit dem Mensch vermeiden. In Folge der Ausweichbewegung steigen Positions- und Orientierungsfehler kurzzeitig auf 5 cm bzw. 3° . Gegen Ende des Experiments nähert sich der Roboter dem als statisches Hindernis bekannten Tisch und der in den Beschränkungen berücksichtigte minimale Abstand (hier der Abstand zum Tisch) sinkt auf 2 cm, während der minimale Abstand zu dynamischen Hindernissen im Bereich von 25 cm liegt.

Dieses Experiment zeigt die Notwendigkeit der Unterscheidung zwischen statischen und dynamischen Hindernissen: Die Annäherung des Roboters an den Tisch bis auf einen sehr kleinen Abstand ist notwendig, um die Zielpose des Endeffektors erreichen zu können und kann ohne Sicherheitseinschränkung erlaubt werden, da es sich um ein statisches Hindernis handelt. Nähern sich jedoch Roboter und Mensch, hält der Roboter einen möglichst großen Abstand zu dem dynamischen Hindernis und kann so eine Kollision erfolgreich verhindern.

8.4 Kontaktregelung

Ergebnisse der Erweiterung der kartesischen modellprädiktiven Regelung für die Reaktion auf Kontakte des Roboters mit seiner Umgebung werden für den stationären Roboterarm mit sieben Freiheitsgraden (siehe Abschnitt 8.1) vorgestellt, da für den Manipulator Schätzungen der externen Drehmomente verfügbar sind. Es wird ein Szenario betrachtet, in dem der Roboterarm mit einem am Endeffektor montierten Stift durch das Folgen eines a-priori definierten Sollpfades einen Kreis mit einem Durchmesser von 20 cm auf ein auf einem Tisch liegendes Papier zeichnet (siehe Abbildung 8.11). Bezüglich der Orientierung des Endeffektors wird lediglich die senkrechte Ausrichtung

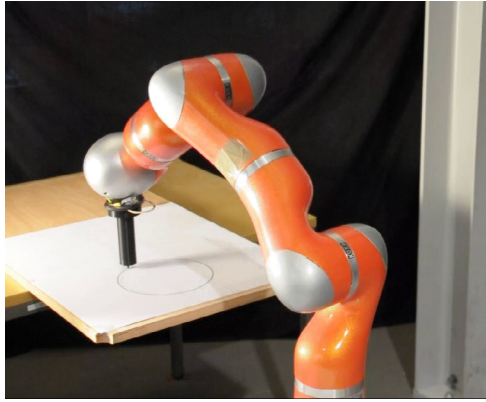


Abbildung 8.11: Versuchsaufbau des stationären Manipulators mit Stift.

des Stifts mit der Spitze nach unten vorgegeben. Der geometrische Pfad soll dabei mit konstanter Geschwindigkeit abgefahren werden.

Die Parameter der kartesischen Regelung werden hierzu wie folgt gewählt:

$$\mathbf{Q}_e = \text{diag}(\eta_p, \eta_p, \eta_p, 0, 0, \eta_o, \eta_o) \quad (8.13)$$

$$\mathbf{Q}_q = \text{diag}(\eta_q, \eta_q, \eta_q, \eta_q, \eta_q, \eta_q, \eta_q) \quad (8.14)$$

$$\mathbf{Q}_w = \text{diag}(\eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}) \quad (8.15)$$

$$\mathbf{R}_e = 100 \mathbf{Q}_e \quad (8.16)$$

$$\mathbf{R}_q = \mathbf{0} \quad (8.17)$$

mit $\eta_p = 200 \text{ } 1/\text{m}^2$, $\eta_o = 50$, $\eta_q = 0,01 \text{ } 1/\text{rad}^2$ und $\eta_{w3} = 15 \text{ } \text{s}^2/\text{rad}^2$. Im Vergleich zur Parameterwahl in Gleichung (8.2) für die Experimente in den Abschnitten 8.2 und 8.3 werden Abweichungen von der Sollorientierung schwächer gewichtet, da die exakte Orientierung der Stiftspitze während des Zeichnens keine hohe Priorität hat. Die Wahl des vierten und fünften Diagonalelements der Gewichtungsmatrix \mathbf{Q}_e zu Null erlaubt dem Endeffektor eine freie Rotation um die vertikale z -Achse. So wird für den Stift lediglich die Ausrichtung nach unten vorgegeben.

Während der Durchführung des Experiments berührt ein Mensch den Roboterarm und schiebt den Ellenbogen des Roboters zur Seite. Dies wird durch die mit Hilfe der Erweiterung zur Kontaktregelung erzielten Nachgiebigkeit des Roboterarms möglich.

Diese Arbeit legt keinen Fokus auf die Schätzung der durch den Kontakt entstehenden Kontaktkräfte bzw. der externen Drehmomente in den Roboter gelenken. Beides wird als gegeben vorausgesetzt. Zur Analyse auf dem Demonstrator werden die von der Robotersteuerung bereitgestellten externen Drehmomente direkt verwendet – lediglich eine einfache Schwellwertentscheidung erfolgt zur Unterscheidung, ob ein externer Kontakt vorliegt oder nicht. Zur Schätzung von Kontaktkraft und -position wird das in [Lik14] vorgeschlagene Verfahren verwendet. Mit diesem Verfahren können Betrag, Richtung und Position einer senkrecht auf den Roboterarm wirkenden Kraft geschätzt werden. Es beruht lediglich auf den externen Drehmomenten und dem kinematischen Modell des Roboters und benötigt keine zusätzlichen Sensoren.

8.4.1 Momentenbasierte Kontaktregelung mit Kontakt am Roboterarm

Zunächst werden die Ergebnisse des momentenbasierten Ansatzes zur Kontaktregelung vorgestellt. Dieser beruht auf den durch den Kontakt entstehenden externen Drehmomenten der einzelnen Roboter gelenke. Die externen Momente werden von der Robotersteuerung bereitgestellt, sind jedoch fehlerbehaftet und stark verrauscht. Deshalb werden nur Momente berücksichtigt, deren Beträge die Schwellwerte von $\tau_{\text{ext,min},i} = 1,8\text{Nm}$ für die Gelenke $i = 1, \dots, 4$ bzw. $\tau_{\text{ext,min},i} = 1,2\text{Nm}$ für die Gelenke $i = 5, \dots, 7$ überschreiten. Die Parameter der Regelung aus den Gleichungen (7.1) und (7.3) betragen $\lambda_\tau = 0,07 \text{ rad/Nms}$ und $\mathbf{Q}_C = \eta_c \mathbf{I}$ mit $\eta_c = 50 \text{ s}^2/\text{rad}$. Die Matrix \mathbf{I} ist eine 7×7 Einheitsmatrix.

Der Ablauf des Experiments ist in der Bildfolge in Abbildung 8.12 zu sehen. Abbildung 8.13 zeigt die mit dem Schwellwert gefilterten Beträge externen Drehmomente der beiden untersten Roboter gelenke, den resul-

tierenden Positionsfehler der Stiftspitze sowie die für einen Reglerschritt benötigte Rechenzeit. Ohne Kontakt liegt der Positionsfehler der Stiftspitze bei maximal 2 mm. Während der zwei Kontakte mit dem Menschen in den Zeitintervallen von 17 s bis 23 s und 29 s bis 31 s führt der Roboter eine nachgiebige Bewegung in Richtung der externen Drehmomente aus, sodass sich der Ellenbogen des Roboters mit einem leichten Druck nach links bewegen lässt. Auf Grund der Ausweichbewegung steigt der Positionsfehler auf bis zu 8,2 mm. Die für einen Reglerschritt benötigte Rechenzeit weist während des Kontakts keinen erkennbaren Anstieg auf.

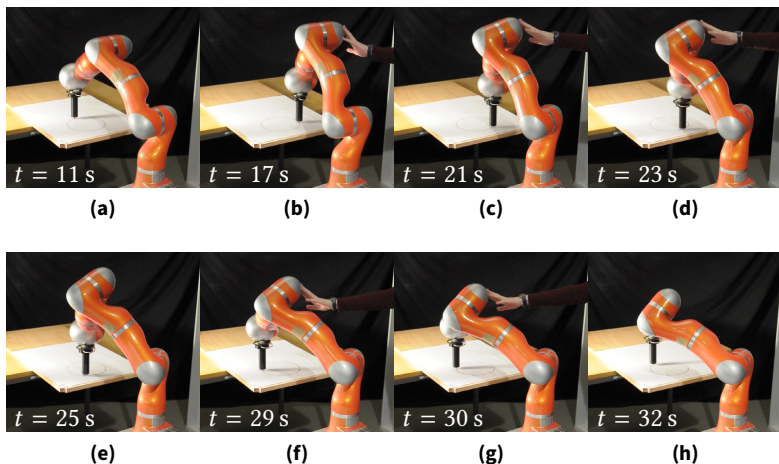


Abbildung 8.12: Bildfolge der momentenbasierten Kontaktregelung.

8.4.2 Kraftbasierte Kontaktregelung mit Kontakt am Roboterarm

Mit der kraftbasierten Kontaktregelung wird ein ähnliches Experiment durchgeführt wie zuvor mit der momentenbasierten Kontaktregelung. Die Kontaktkräfte werden dabei mit dem in [Lik14] vorgeschlagenen Verfahren geschätzt. Um eine zuverlässige Kontaktdetektion zu erzielen, werden nur

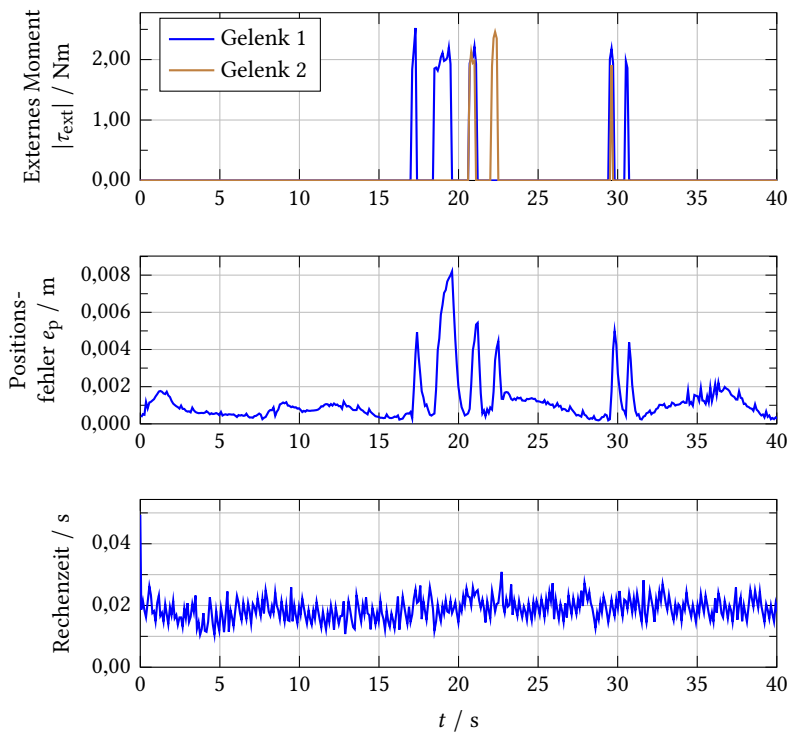


Abbildung 8.13: Ergebnisse der momentenbasierten Kontaktregelung.

Kontaktkräfte berücksichtigt, deren Amplituden einen Schwellwert von 10 N überschreiten und deren Schätzung mit ausreichender Güte erfolgt. Die Parameter der Kontaktregelung aus den Gleichungen (7.5), (7.8) und (7.9) werden zu $\lambda_F = 0,003 \text{ m/Ns}$, $\mathbf{Q}_C = \eta_c \mathbf{I}$ mit $\eta_c = 1000 \text{ 1/m}^2$ und $f_c = 20$ gewählt. \mathbf{I} ist eine 3×3 Einheitsmatrix.

Abbildung 8.14 zeigt eine Bildfolge des Experiments. In Abbildung 8.15 sind die Beträge der auftretenden externen Drehmomente der beiden untersten Gelenke, die Amplitude der geschätzten und mit dem Schwellwert gefilterten Kontaktkraft, der Positionsfehler der Stiftspitze sowie die für einen Reglerschritt benötigte Rechenzeit dargestellt. Ohne Kontakt beträgt der Positionsfehler maximal 2 mm. Während der zwei Kontakte mit dem Menschen in den Zeitintervallen von 16,5 s bis 18,5 s und 29 s bis 30 s führt der Roboter eine nachgiebige Bewegung in Richtung der einwirkenden Kontaktkraft aus, sodass der Ellenbogen des Roboters eine Ausweichbewegung nach links durchführt. Auf Grund der Ausweichbewegung steigt der Positionsfehler auf bis zu 1,5 cm. Die für einen Reglerschritt benötigte Rechenzeit steigt während des Kontakts nur geringfügig an.

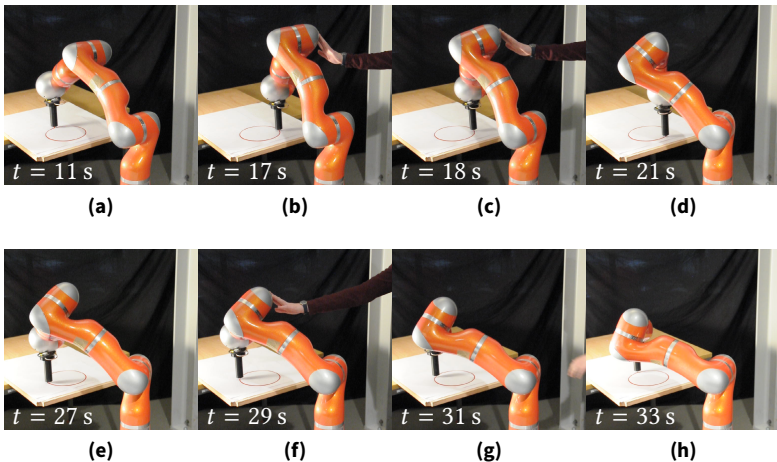


Abbildung 8.14: Bildfolge der kraftbasierten Kontaktregelung.

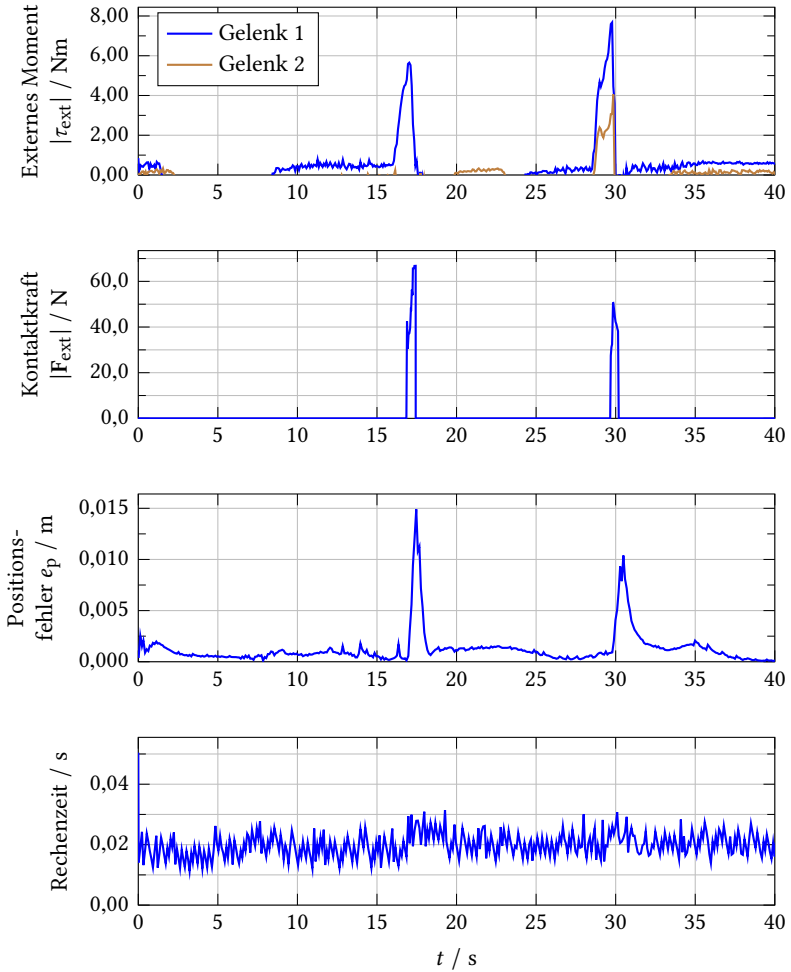


Abbildung 8.15: Ergebnisse der kraftbasierten Kontaktregelung.

Beim Vergleich der beiden Ansätze zur Kontaktregelung ist festzustellen, dass in beiden Fällen das Ziel eines nachgiebigen Verhaltens bei gleichzeitiger Einbehaltung der Sollpose erreicht werden kann. Das Experiment mit der kraftbasierten Kontaktregelung wurde mit höheren Kontaktkräften bzw. größeren externen Drehmomenten durchgeführt. Die größere Kraft ist hier notwendig, um mit dem verwendeten Verfahren zur Kraftschätzung Kontakte zuverlässig detektieren zu können. Bezüglich der Regelung besteht jedoch keine Einschränkung, sie könnte mit einer genaueren Kraftschätzung auch bereits auf niedrigere Kräfte reagieren. Im Experiment der kraftbasierten Kontaktregelung erfolgt die Ausweichbewegung entsprechend der größeren Krafteinwirkung mit höherer Geschwindigkeit. Hierdurch erklärt sich, dass in diesem Fall der Positionsfehler größer ist als im Experiment der momentenbasierten Kontaktregelung.

Weitere Ergebnisse der Kontaktregelung sind in [Zub16] zu finden. Sie beinhalten die Reaktion auf Kontakte mit sich nicht bewegenden Objekten und Kontakte am Endeffektor des Roboters.

8.5 Kombinierte Kollisionsvermeidung und Kontaktregelung

Abschließend wird ein Experiment durchgeführt, bei dem sowohl die Erweiterungen zur Kollisionsvermeidung (Kapitel 6) als auch die Erweiterung zur Kontaktregelung (Kapitel 7) parallel eingesetzt werden. Das Demonstrationsszenario ist ähnlich wie das in Abschnitt 8.4 betrachtete Szenario. Es wird jedoch statt des stationären Manipulators der mobile Manipulator eingesetzt. Dieser zeichnet ebenfalls mit Hilfe eines am Endeffektor montierten Stifts einen Kreis auf ein auf dem Tisch liegendes Papier. Kontakte des Roboterarms werden basierend auf den externen Drehmomenten mit Hilfe der Kraftschätzung aus [Lik14] detektiert. Der Arbeitsraum des Roboters wird, wie in Abschnitt 8.3 beschrieben, mit je zwei auf dem Roboter montierten Laserscannern und Tiefenkameras überwacht. Es existiert ein Umgebungsmodell, das die Wände der Roboterumgebung und den Tisch, auf dem der

Roboter arbeitet, enthält. Die im Umgebungsmodell enthaltenen Objekte werden als statisch betrachtet, alle anderen durch die Arbeitsraumüberwachung erfassten Objekte als dynamisch.

Folgende Reglerparameter weichen von den in den Abschnitten 8.2 und 8.3 angegebenen Werten ab:

$$\mathbf{Q}_e = \text{diag}(\eta_p, \eta_p, \eta_p, 0, 0, \eta_o, \eta_o) \quad (8.18)$$

$$\mathbf{Q}_w = \text{diag}(\eta_{w1}, \eta_{w1}, \eta_{w2}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}, \eta_{w3}) \quad (8.19)$$

$$\mathbf{R}_e = 100 \mathbf{Q}_e \quad (8.20)$$

mit $\eta_p = 200 \text{ 1/m}^2$, $\eta_o = 50$, $\eta_{w1} = 20 \text{ s}^2/\text{m}^2$, $\eta_{w2} = 20 \text{ s}^2/\text{rad}^2$ und $\eta_{w3} = 5 \text{ s}^2/\text{rad}^2$. Wie für den Roboterarm werden im Vergleich zur Parameterwahl in Gleichung (8.2) Abweichungen von der Sollorientierung schwächer gewichtet, da die exakte Orientierung der Stiftspitze in diesem Fall keine hohe Priorität hat. Die Wahl des vierten und fünften Diagonalelements von \mathbf{Q}_e zu Null erlaubt die freie Rotation des Endeffektors um die Hochachse. Die Geschwindigkeiten der Plattformgelenke werden mit η_{w1} und η_{w2} stärker gewichtet als in Gleichung (8.4), da während des Zeichenexperiments Bewegungen des Arms vor Bewegungen der Plattform zu bevorzugen sind. Der Grund hierfür ist die höhere Genauigkeit der Armbewegung im Vergleich zur Plattformbewegung. Die Parameter der Kraftregelung werden wie für den Roboterarm zu $\lambda_F = 0,003 \text{ m/Ns}$, $\mathbf{Q}_C = \eta_c \mathbf{I}$ mit $\eta_c = 1000 \text{ 1/m}^2$ und $f_C = 20$ gewählt. Die Matrix \mathbf{I} ist eine 3×3 Einheitsmatrix.

Während der mobile Manipulator auf dem Tisch zeichnet, nähert sich ein Mensch dem Roboter und schiebt den Ellenbogen des Roboters zur Seite. Dies ist in der Bildfolge in Abbildung 8.16 zu sehen. Abbildung 8.17 zeigt den durch den Roboter gezeichneten Kreis. Die hierbei in der Regelung berücksichtigten minimalen Abstände, die geschätzte Kontaktkraft, der resultierende Positionsfehler sowie die x -Koordinate der Plattformposition sind in Abbildung 8.18 dargestellt.

Die Annäherung des Menschen an den Roboter wird von der Arbeitsraumüberwachung erfasst. Es werden sowohl die Beine mit Hilfe der Laserscanner als auch Oberkörper und Arme mit Hilfe der Tiefenkameras detek-

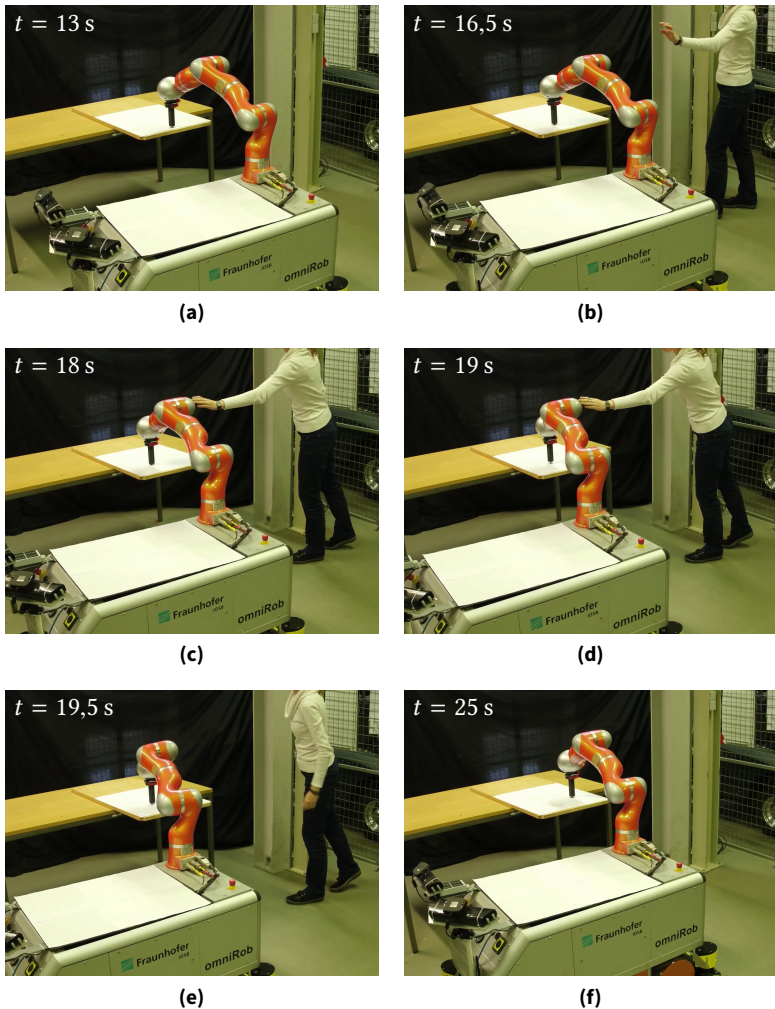
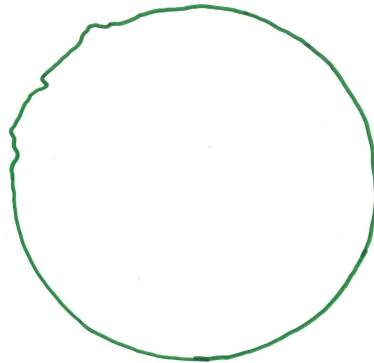
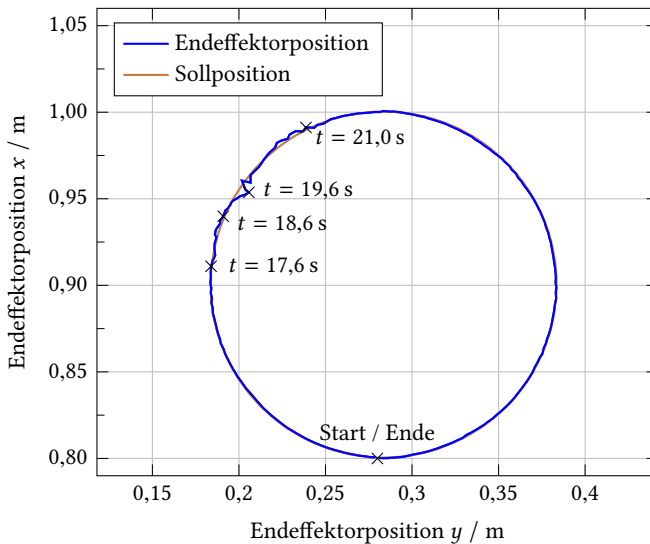


Abbildung 8.16: Bildfolge des parallelen Einsatzes von Kollisionsvermeidung und kraftbasierter Kontaktregelung.



(a) Zeichenergebnis



(b) Gemessene Endeffektorposition

Abbildung 8.17: Endeffektorposition des parallelen Einsatzes von Kollisionsvermeidung und kraftbasierter Kontaktregelung.

tiert. Deshalb beginnt der Roboter eine Ausweichbewegung bereits bevor der Kontakt zwischen Mensch und Roboter stattfindet. Diese Ausweichbewegung ist am deutlichsten in der Bewegung der Plattform parallel zum Tisch (in negative x -Richtung) zu erkennen. Dementsprechend weicht die x -Position der Plattform bereits früh von der Position während einer Versuchsdurchführung ohne dynamische Hindernisse (Referenz) ab (siehe Abbildung 8.18). Ab $t = 16,8$ s nimmt der in der Kostenfunktion berücksichtigte Abstand zu den dynamischen Hindernissen stark ab und die Geschwindigkeit der Ausweichbewegung nimmt zu. Dies liegt daran, dass der Mensch seinen Arm nach dem Roboter ausstreckt. Da sich der Mensch jedoch zu schnell bewegt, ist es dem Roboter nicht möglich, einen Kontakt zu verhindern. Im Zeitraum von $t = 18,6$ s bis $t = 19,5$ s wird eine Kontaktkraft geschätzt, die ebenfalls zu einer nachgiebigen Ausweichbewegung des Roboters führt. Es ist zu erkennen, dass dieser Kontakt lediglich mit Hilfe der Kraftschätzung detektiert werden kann. Der von der Arbeitsraumüberwachung ermittelte minimale Abstand zwischen Roboter und Hindernis unterschreitet dabei den Abstand von 15 cm nicht, da die Hand und Teile des Arms des Menschen vom Roboterarm verdeckt werden. Der Positionsfehler bleibt während der Annäherung des Hindernisses klein und nimmt Werte bis zu 3 mm an. In Folge des Kontakts entsteht der größte Positionsfehler mit 1,1 cm. Während des gesamten Experiments liegt die Rechenzeit für einen Reglerschritt unter der Abtastzeit von 0,1 s (im Mittel bei 59 ms) und steigt auch während des Kontakts nicht wesentlich an.

Während des Experiments liegt die Reglergenauigkeit in einem ähnlichen Bereich wie die Lokalisierungsgenauigkeit eines lasergestützten Lokalisierungsverfahrens für mobile Roboter [Röw12]. Die Lokalisierungsgenauigkeit stellt einen wichtigen Faktor für die absolute Positioniergenauigkeit mobiler Manipulatoren dar. Der Einfluss von Lokalisierungsfehlern ist auch in Abbildung 8.17 zu erkennen. Während die gemessene Endeffektorposition in Abbildung 8.17b abgesehen von den Auswirkungen der Ausweichbewegung dem gewünschten Kreis sehr nahe kommt, ist der gezeichnete Kreis in Abbildung 8.17a leicht gestaucht.

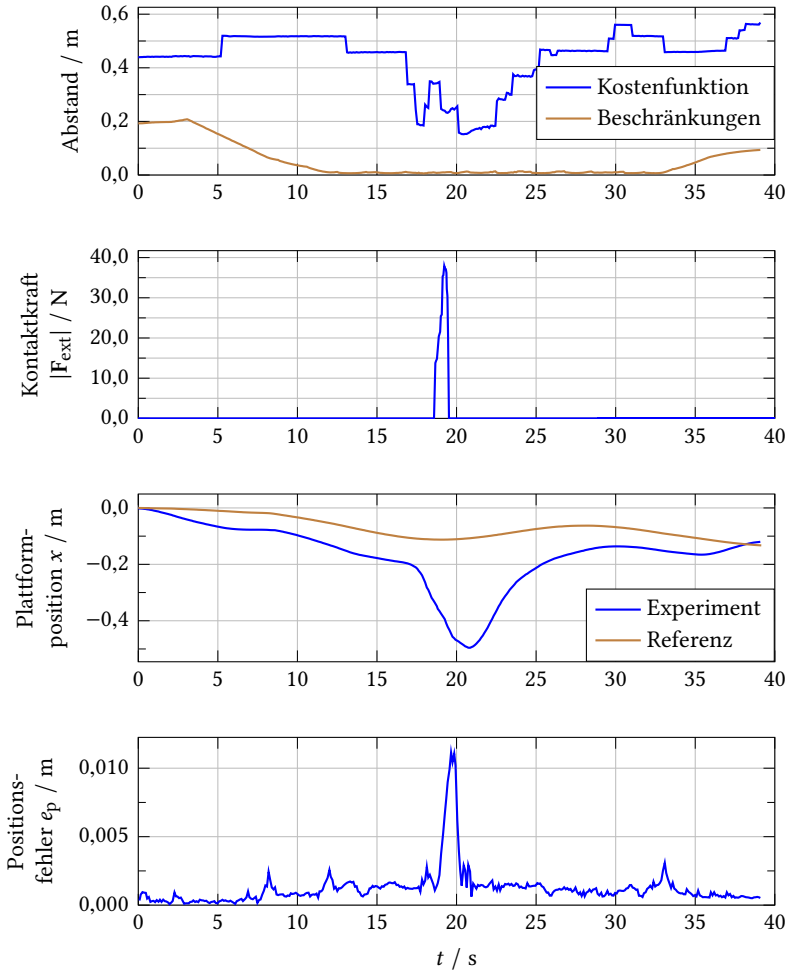


Abbildung 8.18: Ergebnisse des parallelen Einsatzes von Kollisionsvermeidung und kraftbasierter Kontaktregelung.

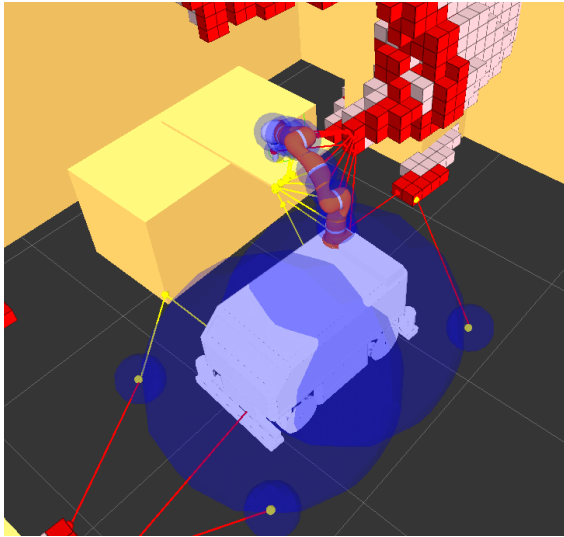


Abbildung 8.19: Visualisierung der statischen (gelb) und dynamischen (rot/rosa) Hindernisse sowie der in den Beschränkungen (gelbe Linien) und Kostenfunktion (rote Linien) berücksichtigten Abstände zum Zeitpunkt $t = 16,7$ s.

In diesem Experiment wird die Notwendigkeit der gesonderten Behandlung von statischen und dynamischen Hindernissen noch einmal deutlich. Der in den Beschränkungen zur Kollisionsvermeidung berücksichtigte minimale Abstand ist sehr klein, da der Abstand zwischen dem Roboter und dem statischen Tisch beim Zeichnen notwendigerweise sehr klein ist (siehe Abbildung 8.19). Dieser kleine Abstand wird aber von der Regelung gestattet, solange er nicht zu einer Kollision führt. Gleichzeitig sollen bereits größere Abstände zum dynamischen Hindernis Mensch zu einer Ausweichbewegung des Roboters führen. Dies wird durch das Eingehen der Abstände zu dynamischen Hindernissen in die Kostenfunktion der Regelung erreicht.

Das Experiment zeigt, wie durch den Einsatz von Kollisionsvermeidung und Kontaktregelung auf nicht zu vermeidende Kontakte reagiert wird: mit Hilfe der Kollisionsvermeidung wird der Kontakt soweit möglich reduziert. Auch wenn keine vollständige Vermeidung möglich ist, wird schon vor dem

Eintreten des Kontakts eine Ausweichbewegung eingeleitet. Mit Hilfe der Kontaktregelung wird dann angemessen auf den stattfindenden Kontakt reagiert.

Weitere Ergebnisse zum parallelen Einsatz von Kollisionsvermeidung und Kontaktregelung auf dem stationären Manipulator sind in [Zub16] zu finden.

8.6 Zusammenfassung

In diesem Kapitel wurde das Konzept der kartesischen modellprädiktiven Regelung mit den Erweiterungen zur Kollisionsvermeidung und zur Reaktion auf stattfindende Kontakte in mehreren Demonstrationsszenarien evaluiert. Als Versuchsplattformen wurden sowohl ein mobiler Manipulator mit zehn Freiheitsgraden als auch ein stationärer Manipulator mit sieben Freiheitsgraden eingesetzt. Auf beiden in Hinblick auf die Aufgabenstellung redundanten Robotern konnte die Endeffektorpose entlang gegebener Pfade mit für die Anwendungen akzeptablen Positions- und Orientierungsfehlern bewegt werden. Mit Hilfe der Erweiterung zur Kollisionsvermeidung konnten Kollisionen mit statischen und dynamischen Hindernissen erfolgreich verhindert werden. Die unterschiedliche Reaktion auf statische und dynamische Hindernisse (statische Hindernisse dienen nur als Beschränkung, zu dynamischen Hindernissen wird der Abstand maximiert) erlaubt nicht nur eine möglichst sichere Interaktion von Mensch und Roboter im freien Raum, sondern auch die Arbeit an bekannten Objekten wie beispielsweise auf einem Tisch. Die Erfassung der unbekannt (dynamischen) Umgebung erfolgte mit der in dieser Arbeit vorgeschlagenen 3D-Arbeitsraumüberwachung. Es zeigte sich, dass die Rechenzeiten der Regelung und der Arbeitsraumüberwachung für den Einsatz auf einem realen Roboter ausreichend kurz sind. Mit Hilfe der Erweiterung zur Kontaktregelung konnten basierend auf externen Gelenkmomenten oder der geschätzten Kontaktkraft nachgiebige Reaktionen auf stattfindende Kontakte demonstriert werden. Mit dem gleichzeitigen Einsatz beider Erweiterungen zur Kollisionsvermeidung und zur Reaktion auf stattfindende Kontakte konnten

die Vorteile beider Ansätze in Hinblick auf die Mensch-Roboter-Interaktion genutzt werden. Es wurde gezeigt, dass so bereits vor dem Auftreten eines Kontakts eine Ausweichbewegung eingeleitet wird. Kann der Kontakt nicht vermieden werden, erfolgt eine nachgiebige Reaktion bei Eintreten des Kontakts.

Kapitel 9

Zusammenfassung und Ausblick

9.1 Zusammenfassung der Beiträge

In dieser Arbeit wurde die kartesische Regelung der Endeffektorpose stationärer und mobiler Manipulatoren mittels einer nichtlinearen modellprädiktiven Regelung (engl. *Nonlinear Model Predictive Control*, NMPC) untersucht. Dabei wurde sowohl die Verfolgung von zeitbehafteten Trajektorien als auch von geometrischen Sollpfaden der Endeffektorpose betrachtet. Durch die NMPC erfolgt eine Minimierung des Trajektorien- bzw. Pfadverfolgungsfehlers. Roboterbeschränkungen werden direkt bei der Optimierung berücksichtigt. Da der Regelung keine Invertierung der direkten Kinematik zugrunde liegt, treten keine Probleme mit kinematischen Singularitäten auf. Es wurde gezeigt, dass sich die NMPC eignet, um unterschiedliche Kinematiken zu regeln, und dass sich die Regelung einfach für verschiedene Anforderungen erweitern lässt.

Als wichtige Aspekte der Mensch-Roboter-Interaktion wurden Erweiterungen zur Vermeidung von Eigen- und Fremdkollisionen mit statischen und dynamischen Hindernissen sowie zur Reaktion auf stattfindende Kon-

takte vorgeschlagen. Für die Kollisionsvermeidung wurden Terme für die Kostenfunktion und Beschränkungen der NMPC definiert und analysiert, die die Vermeidung von Eigen- und Fremdkollisionen unter Berücksichtigung der gesamten Robotergeometrie (nicht nur des Endeffektors) ermöglichen. Damit lassen sich unterschiedliche Reaktionen auf statische und dynamische Hindernisse erzielen. Für die Kontaktreaktion wurden Terme für die Kostenfunktion definiert und analysiert, die eine nachgiebige Reaktion auf stattfindende Kontakte ermöglichen. Dabei wurde auch die Möglichkeit betrachtet, dass von Roboterarm und Endeffektor unterschiedliche Reaktionen erwartet werden. Zur Kontaktreaktion wird für den Regelalgorithmus kein dynamisches Robotermodell vorausgesetzt. Die Erweiterungen können gleichzeitig eingesetzt werden, sodass keine Umschaltung zwischen verschiedenen Regelungen notwendig ist.

Als Voraussetzung für die Kollisionsvermeidung mit dynamischen Hindernissen wurde eine Methode zur 3D-Arbeitsraumüberwachung des Roboters vorgeschlagen. Für die Detektion von Hindernissen im nahen Umfeld des Roboters wird die Information mehrerer (heterogener) Abstandssensoren fusioniert. Die Sensoren können sowohl im Arbeitsraum als auch auf dem Roboter installiert sein. Für eine konservative Abschätzung der Hindernisausdehnung werden Verdeckungen durch die Hindernisse und durch den Roboter berücksichtigt.

Die erfolgreiche Anwendung der Verfahren wurde in zahlreichen Experimenten auf einem mobilen Manipulator mit zehn Freiheitsgraden und auf einem stationären Manipulator mit sieben Freiheitsgraden gezeigt. Die Rechenzeiten der kartesischen modellprädiktiven Regelung sind für eine Reglerabtastrzeit von 100 ms bei einem Prädiktionshorizont von 1 s ausreichend kurz und damit für den Einsatz auf einem realen mobilen Manipulator geeignet. Die resultierenden Positionsfehler des Endeffektors betragen im Mittel wenige Millimeter. Während der Vermeidung von Kollisionen mit dynamischen Hindernissen und während der Reaktion auf stattfindende Kontakte können sie auf wenige Zentimeter ansteigen. Damit liegt die Reglergenauigkeit in einem ähnlichen Bereich wie die Genauigkeit einer laser-gestützten Lokalisierung mobiler Roboter [Röw12].

9.2 Ausblick

Das in dieser Arbeit vorgestellte Regelungskonzept ermöglicht es dem Roboter, unterschiedlich auf verschiedene Hindernisarten wie statische oder dynamische Hindernisse zu reagieren. In den Versuchen konnte gezeigt werden, dass dies wichtig ist, um sowohl die Sicherheit von Menschen in der Roboterumgebung zu gewährleisten als auch die Aufgabenausführung des Roboters nicht unnötig einzuschränken. In den durchgeführten Versuchen wurden hierzu die statischen Hindernisse als bekannt vorausgesetzt und die dynamischen Hindernisse mit Hilfe der 3D-Arbeitsraumüberwachung zur Laufzeit detektiert. Für einen flexibleren Einsatz in unterschiedlichen Umgebungen wäre die Erweiterung des Arbeitsraumüberwachungskonzepts hilfreich, sodass eine Klassifizierung der Hindernisse zur Laufzeit möglich wird. Das Regelungskonzept könnte auch Prädiktionen von Hindernisbewegungen (z. B. [Fre14]) zur Kollisionsvermeidung nutzen. Durch die Anbindung an ein umfangreiches Umweltmodell könnte das Verhalten des Reglers beispielsweise auch an die Art des zu manipulierenden Objekts angepasst werden. In dieser Hinsicht bietet die leichte Erweiterbarkeit des Regelungskonzepts viele Möglichkeiten.

Die in den Versuchen erzielte Reglergenauigkeit ist für viele im Kontext der Mensch-Roboter-Interaktion auftretende Aufgaben, wie zum Beispiel den Transport von Gegenständen, die Übergabe von Gegenständen an einen Menschen oder das Halten eines Bauteils, ausreichend genau. Für andere Aufgaben, wie die Montage von Bauteilen, können aber höhere Anforderungen an die Reglergenauigkeit bestehen. Für derartige Fälle kann die Begrenzung des Positions- und Orientierungsfehlers mittels zusätzlicher Nebenbedingungen der Regelung betrachtet werden.

Zu untersuchen bleibt auch die Möglichkeit, die kartesische modellprädiktive Regelung in anderen Anwendungsfeldern wie beispielsweise auf autonomen Baggern einzusetzen. Bagger mit Rad- oder Kettenantrieb stellen ebenfalls mobile Manipulatoren dar. Im Gegensatz zu Industrierobotern sind die hydraulischen Antriebe von Baggern jedoch im Allgemeinen durch lange Totzeiten und starke Nichtlinearitäten im Übertragungsverhalten gekenn-

zeichnet. Mit Hilfe eines erweiterten Modells, das basierend auf einer Systemidentifikation die Totzeiten und Nichtlinearitäten abbildet, könnten diese direkt in der modellprädiktiven Regelung berücksichtigt werden. Es ist zu untersuchen, ob so ein besseres Reglerverhalten als mit konventionellen Reglern zu erzielen ist. Gleichzeitig wären auch hier die einheitliche Behandlung von Baggerarm und -plattform sowie die Ausnutzung der Redundanz des Baggers zur Kollisionsvermeidung durch die Regelung von Interesse. Dabei müssen, anders als auf dem in dieser Arbeit eingesetzten Demonstrator, auch Nebenbedingungen, die die nichtholonome Kinematik der Baggerplattform abbilden, berücksichtigt werden.

Anhang A

Quaternionen

Umrechnung zwischen Rotationsmatrix und Einheitsquaternionen

Die zur Ermittlung der direkten Kinematik benötigte Umrechnung der Orientierung von Rotationsmatrizen zu Quaternionen kann mit unterschiedlichen Rechenvorschriften erfolgen [Sci00; Sic08]. Jedoch sind nicht alle gängigen Vorschriften numerisch stabil. Im Folgenden wird eine numerisch stabile Konvertierung der Rotationsmatrix \mathbf{R} mit den Matrixelementen $R_{i,j}$ ($i = 1,2,3$ und $j = 1,2,3$) in die Einheitsquaternion $\mathbf{o} = [o_x, o_y, o_z, o_w]^T$ angegeben.

Gilt $R_{1,1} + R_{2,2} + R_{3,3} \geq 0$, ergibt sich

$$o_w = \frac{\sqrt{1 + R_{1,1} + R_{2,2} + R_{3,3}}}{2} \quad (\text{A.1})$$

$$o_x = \frac{R_{3,2} - R_{2,3}}{4o_w} \quad (\text{A.2})$$

$$o_y = \frac{R_{1,3} - R_{3,1}}{4o_w} \quad (\text{A.3})$$

$$o_z = \frac{R_{2,1} - R_{1,2}}{4o_w} . \quad (\text{A.4})$$

Gilt $R_{1,1} + R_{2,2} + R_{3,3} < 0$ und $R_{1,1} \geq \max(R_{2,2}, R_{3,3})$, ergibt sich

$$o_x = \frac{\sqrt{1 + R_{1,1} - R_{2,2} - R_{3,3}}}{2} \quad (\text{A.5})$$

$$o_y = \frac{R_{1,3} - R_{3,1}}{4o_x} \quad (\text{A.6})$$

$$o_z = \frac{R_{2,1} - R_{1,2}}{4o_x} \quad (\text{A.7})$$

$$o_w = \frac{R_{3,2} - R_{2,3}}{4o_x} . \quad (\text{A.8})$$

Analog gilt für den Fall $R_{1,1} + R_{2,2} + R_{3,3} < 0$ und $R_{2,2} \geq \max(R_{1,1}, R_{3,3})$

$$o_y = \frac{\sqrt{1 - R_{1,1} + R_{2,2} - R_{3,3}}}{2} \quad (\text{A.9})$$

$$o_x = \frac{R_{3,2} - R_{2,3}}{4o_y} \quad (\text{A.10})$$

$$o_z = \frac{R_{2,1} - R_{1,2}}{4o_y} \quad (\text{A.11})$$

$$o_w = \frac{R_{1,3} - R_{3,1}}{4o_y} \quad (\text{A.12})$$

und für $R_{1,1} + R_{2,2} + R_{3,3} < 0$ und $R_{3,3} \geq \max(R_{1,1}, R_{2,2})$

$$o_z = \frac{\sqrt{1 - R_{1,1} - R_{2,2} + R_{3,3}}}{2} \quad (\text{A.13})$$

$$o_x = \frac{R_{3,2} - R_{2,3}}{4o_z} \quad (\text{A.14})$$

$$o_y = \frac{R_{1,3} - R_{3,1}}{4o_z} \quad (\text{A.15})$$

$$o_w = \frac{R_{2,1} - R_{1,2}}{4o_z} . \quad (\text{A.16})$$

Interpolation mit B-Splines

Für die in Abschnitt 5.4.2 vorgestellte Regelung zur Pfadverfolgung mit Optimierung des Pfadparameters wird ein zweimal stetig differenzierbarer Sollpfad benötigt. Ist der Pfad als eine Folge von Posen gegeben, kann ein geeigneter Sollpfad durch Interpolation mit zweimal stetig differenzierbaren Splines gewonnen werden. Im Folgenden wird hierfür die Quaternioneninterpolation mit B-Splines angegeben. Die Herleitung der Interpolation und weitere Details sind in [Kim95a; Kim95b] zu finden.

Die B-Spline-Interpolation $\mathbf{o}(t)$ einer Folge von $N + 1$ Quaternionen $\{\mathbf{o}_i\}$, die jede Quaternion \mathbf{o}_i bei $t = t_i$ interpoliert, ist gegeben durch

$$\mathbf{o}(t) = \mathbf{o}_0^{\tilde{B}_{0,k}(t)} \prod_{i=1}^N \exp(\boldsymbol{\omega}_i \tilde{B}_{i,k}(t)) \quad (\text{A.17})$$

mit $\omega_i = \log(\mathbf{o}_{i-1}^{-1} \mathbf{o}_i)$ und den Basisfunktionen k -ter Ordnung $\widetilde{B}_{i,k}(t)$. Die Basisfunktionen ergeben sich zu

$$\widetilde{B}_{i,k}(t) = \sum_{j=1}^N B_{j,k}(t) \quad (\text{A.18})$$

$$= \begin{cases} \sum_{j=1}^{i+k} B_{j,k}(t) & \text{für } t_i < t < t_{i+k-1} \\ 1 & \text{für } t \geq t_{i+k-1} \\ 0 & \text{für } t \leq t_i \end{cases} \quad (\text{A.19})$$

mit

$$B_{i,1}(t) = \begin{cases} 1 & \text{für } t_i < t \leq t_{i+1} \\ 0 & \text{sonst} \end{cases} \quad (\text{A.20})$$

$$B_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} B_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(t) \quad (\text{A.21})$$

Die resultierende Quaternionenkurve ist $(k - 2)$ -mal stetig differenzierbar. Damit erfüllt die B-Spline-Interpolation mit Basisfunktionen vierter Ordnung die Anforderungen der Pfadverfolgungsregelung.

Anhang B

Denavit-Hartenberg- Parameter der Versuchsplattform

Wie in Abschnitt 4.2.2 dargelegt, wird das kinematische Robotermodell basierend auf der Denavit-Hartenberg-Konvention (DH-Konvention) ermittelt. Für die Versuchsplattform *IOSB.amp O2* sind die den einzelnen Robotergliedern zugeordneten Koordinatensysteme in Abbildung B.1 skizziert. Es handelt sich bei allen Koordinatensystemen um rechtshändige Koordinatensysteme. Aus Gründen der Übersichtlichkeit sind nur die x - und z -Achsen der Koordinatensysteme beschriftet. In der Skizze befindet sich der Roboter in der Nullstellung, d. h. alle Gelenkpositionen nehmen den Wert Null ein. Die resultierenden DH-Parameter der Versuchsplattform sind in Tabelle B.1 angegeben.

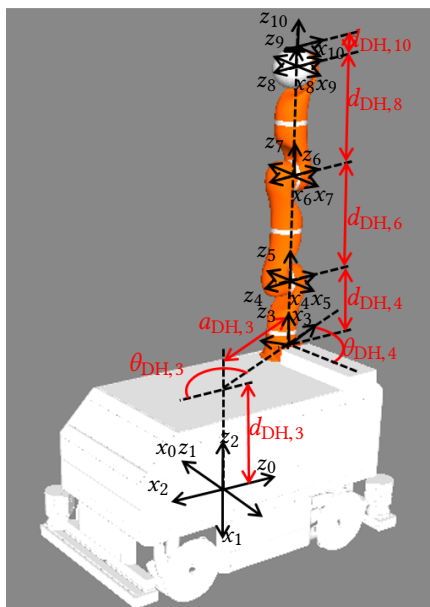


Abbildung B.1: DH-Koordinatensysteme der Versuchsplattform IOSB.amp O2.

Tabelle B.1: DH-Parameter der Versuchsplattform IOSB.amp O2.

Gelenk i	$\alpha_{DH,i}/\text{rad}$	$a_{DH,i}/\text{m}$	$d_{DH,i}/\text{m}$	$\theta_{DH,i}/\text{rad}$
1	$-\frac{\pi}{2}$	0	q_1	$-\frac{\pi}{2}$
2	$-\frac{\pi}{2}$	0	q_2	$\frac{\pi}{2}$
3	0	0,4528	0,649	$-2,7471 + q_3$
4	$\frac{\pi}{2}$	0	0,31	$-1,9653 + q_4$
5	$-\frac{\pi}{2}$	0	0	q_5
6	$-\frac{\pi}{2}$	0	0,4	q_6
7	$\frac{\pi}{2}$	0	0	q_7
8	$\frac{\pi}{2}$	0	0,39	q_8
9	$-\frac{\pi}{2}$	0	0	q_9
10	0	0	0,08	q_{10}

Literatur

- [And12a] V. Andaluz, F. Roberti, J. M. Toibero und R. Carelli. „Adaptive unified motion control of mobile manipulators“. In: *Control Engineering Practice* 20.12 (2012), S. 1337–1352.
- [And12b] P. Anderson-Sprecher und R. Simmons. „Voxel-based motion bounding and workspace estimation for robotic manipulators“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2012, S. 2141–2146.
- [Bay02] B. Bayle, J.-Y. Fourquet, F. Lamiroux und M. Renaud. „Kinematic control of wheeled mobile manipulators“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2002, S. 1572–1577.
- [Bdi12] K. Bdirina, D. Djoudi, M. Lagoun und N. Hamani. „Nonlinear Predictive Control of Rigid Link Manipulator“. In: *International Conference on Trends in Industrial and Mechanical Engineering (ICTIME)*. 2012.
- [Bec09] E. Becker, G. Guerra-Filho und F. Makedon. „Automatic sensor placement in a 3D volume“. In: *International Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*. 2009.
- [Bis10] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppel, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald und G. Hirzinger. „The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing“. In: *Joint 41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (Robotik)*. 2010, S. 741–748.

- [Blo16] R. Bloss. „Collaborative robots are rapidly providing major improvements in productivity, safety, programing ease, portability and cost while addressing many new applications“. In: *Industrial Robot: An International Journal* 43 (2016), S. 463–468.
- [Blo17] N. Blodow. *Realtime URDF Filter*. URL: https://github.com/blodow/realtime_urdf_filter (besucht am 22. Mai 2017).
- [Bog16] R. Bogue. „Europe continues to lead the way in the collaborative robot business“. In: *Industrial Robot: An International Journal* 43 (2016), S. 6–11.
- [Boh11] J. Bohren, R. B. Rusu, E. G. Jones, E. Marder-Eppstein, C. Pantofaru, M. Wise, L. Mösenlechner, W. Meeussen und S. Holzer. „Towards autonomous robotic butlers: Lessons learned with the PR2“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, S. 5568–5575.
- [Cas09] J. Cascio, M. Karpenko, Q. Gong, P. Sekhavat und I. Ross. „Smooth proximity computation for collision-free optimal control of multiple robotic manipulators“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2009, S. 2452–2457.
- [Cir12] A. Cirillo, P. Cirillo und S. Pirozzi. „A Modular and Low-cost Artificial Skin for Robotic Applications“. In: *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2012, S. 961–966.
- [DeL06] A. De Luca, A. Albu-Schaffer, S. Haddadin und G. Hirzinger. „Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2006, S. 1623–1630.
- [DeL08] A. De Luca und L. Ferrajoli. „Exploiting robot redundancy in collision detection and reaction“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008, S. 3299–3305.
- [DeL98] A. De Luca, G. Oriolo und C. Samson. „Feedback Control of a Nonholonomic Car-Like Robot“. In: *Robot Motion Planning and Control*. Hrsg. von J.-P. Laumond. Springer, 1998, S. 171–253.

- [Die02] M. Diehl, H. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy und F. Allgöwer. „Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations“. In: *Journal of Process Control* 12.4 (2002), S. 577–585.
- [Duc08] V. Duchaine, S. Bouchard und C. Gosselin. „Motion Control of a Robot Manipulator in Free Space Based on Model Predictive Control“. In: *Robot Manipulators*. Hrsg. von M. Ceccarelli. InTech, 2008.
- [Emt14] T. Emter und J. Petereit. „Integrated multi-sensor fusion for mapping and localization in outdoor environments for mobile robots“. In: *Proceedings of SPIE Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*. Bd. 9121. 2014.
- [Emt17] T. Emter, C. Frese, A. Zube und J. Petereit. „Algorithmen-Toolbox für autonome mobile Robotersysteme“. In: *ATZoffhighway* 3 (2017).
- [Fah09] F. Fahimi. *Autonomous Robots - Modeling, Planning, and Control*. Springer, 2009.
- [Fau09] T. Faulwasser und R. Findeisen. „Ein prädiktiver Ansatz zur Lösung nichtlinearer Pfadverfolgungsprobleme unter Beschränkungen“. In: *at - Automatisierungstechnik*. Oldenbourg, 2009.
- [Fau12] T. Faulwasser. „Optimization-based Solutions to Constrained Trajectory-tracking and Path-following Problems“. Diss. Otto-von-Guericke-Universität Magdeburg, 2012.
- [Fau13] T. Faulwasser, J. Matschek, J. P. Zometa und R. Findeisen. „Predictive Path-following Control: Concept and Implementation for an Industrial Robot“. In: *IEEE Multi-Conference on Systems and Control*. 2013.
- [Fet13] A. Fetzner, C. Frese und C. Frey. „Obstacle Detection and Tracking for Safe Human-Robot Interaction Based on Multi-Sensory Point Clouds“. In: *6th International Workshop on Human-Friendly Robotics (HFR)*. 2013.
- [Fet14] A. Fetzner, C. Frese und C. Frey. „A 3D Representation of Obstacles in the Robot’s Reachable Area Considering Occlusions“. In: *Joint 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik)*. 2014, S. 85–92.
- [Fin04] R. Findeisen. „Nonlinear Model Predictive Control: A Sampled-Data Feedback Perspective“. Diss. Universität Stuttgart, 2004.

- [Fis09] M. Fischer und D. Henrich. „3D Collision Detection for Industrial Robots and Unknown Obstacles Using Multiple Depth Images“. In: *Advances in Robotics Research - Theory, Implementation, Application*. Hrsg. von T. Kröger und F. M. Wahl. Springer, 2009.
- [Fla10] F. Flacco und A. De Luca. „Multiple depth/presence sensors: Integration and optimal placement for human/robot coexistence“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2010, S. 3916–3923.
- [Fla12] F. Flacco, T. Kroger, A. De Luca und O. Khatib. „A depth space approach to human-robot collision avoidance“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2012, S. 338–345.
- [Fre14] C. Frese, A. Fetzner und C. Frey. „Multi-Sensor Obstacle Tracking for Safe Human-Robot Interaction“. In: *Joint 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik)*. 2014, S. 784–791.
- [Gon09] J.-J. Gonzalez-Barbosa, T. Garcia-Ramirez, J. Salas, J.-B. Hurtado-Ramos und J.-d.-J. Rico-Jimenez. „Optimal camera placement for total coverage“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2009, S. 844–848.
- [Gu09] D. Gu und H. Hu. „A Model Predictive Controller for Robots to Follow a Virtual Leader“. In: *Robotica* 27.6 (2009), S. 905–913.
- [Had08] S. Haddadin, A. Albu-Schaeffer, A. De Luca und G. Hirzinger. „Collision detection and reaction: A contribution to safe physical Human-Robot Interaction“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008, S. 3356–3363.
- [Hed05] R. Hedjar und P. Boucher. „Nonlinear Receding-Horizon Control of Rigid Link Robot Manipulators“. In: *International Journal of Advanced Robotic Systems* 2 (2005), S. 15–24.
- [Hei03] J. Heinzmann und A. Zelinsky. „Quantitative Safety Guarantees for Physical Human-Robot Interaction“. In: *The International Journal of Robotics Research* 22 (2003), S. 479–504.
- [Hen08] D. Henrich und T. Gecks. „Multi-camera collision detection between known and unknown objects“. In: *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. 2008.

- [Her13] A. Hermann, S. Klemm, Z. Xue, A. Roennau und R. Dillmann. „GPU-based real-time collision detection for motion execution in mobile manipulation planning“. In: *International Conference on Advanced Robotics (ICAR)*. 2013.
- [Hol15] D. Holz, A. Topalidou-Kyniazopoulou, F. Rovida, M. R. Pedersen, V. Krüger und S. Behnke. „A skill-based system for object perception and manipulation for automating kitting tasks“. In: *IEEE International Conference on Emerging Technologies Factory Automation (ETFA)*. 2015.
- [Hor13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss und W. Burgard. „OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees“. In: *Autonomous Robots* (2013). Software verfügbar auf <http://octomap.github.com>.
- [Hus16] A. Hussein, P. Marín-Plaza, D. Martín, A. de la Escalera und J. M. Armingol. „Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection“. In: *IEEE Intelligent Vehicles Symposium (IV)*. 2016, S. 104–109.
- [Huy09] D. Q. Huynh. „Metrics for 3D Rotations: Comparison and Analysis“. In: *Journal of Mathematical Imaging and Vision* 35.2 (2009), S. 155–164.
- [Ide11] S. Ide, T. Takubo, K. Ohara, Y. Mae und T. Arai. „Real-time trajectory planning for mobile manipulator using model predictive control with constraints“. In: *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 2011, S. 244–249.
- [IFR16] *Executive Summary World Robotics 2016 Industrial Robots*. URL: https://ifr.org/img/uploads/Executive_Summary_WR_Industrial_Robots_20161.pdf (besucht am 19. Juni 2017).
- [Jac04] D. Jacob. *Roboter in der Automobilindustrie*. 2004. URL: http://www.muenchner-wissenschaftstage.de/mwt2004/content/e160/e707/e728/e751/filetitle/Jacob_ger.pdf (besucht am 19. Juni 2017).
- [Jac80] C. L. Jackins und S. L. Tanimoto. „Oct-trees and their use in representing three-dimensional objects“. In: *Computer Graphics and Image Processing* 14.3 (1980), S. 249–270.
- [Kan09] K. Kanjanawanishkul und A. Zell. „Path following for an omnidirectional mobile robot based on model predictive control“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2009.

- [Kha85] O. Khatib. „Real-time obstacle avoidance for manipulators and mobile robots“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. Bd. 2. 1985, S. 500–505.
- [Kim95a] M.-J. Kim, M.-S. Kim und S. Y. Shin. „A C2-continuous B-spline quaternion curve interpolating a given sequence of solid orientations“. In: *Conference on Computer Animation*. 1995, S. 72–81.
- [Kim95b] M.-J. Kim, M.-S. Kim und S. Y. Shin. „A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives“. In: *22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. 1995, S. 369–276.
- [Kit96] Y. Kitamura, F. Kishino, T. Tanaka und M. Yachida. „Real-time path planning in a dynamic 3-D environment“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Bd. 2. 1996, S. 925–931.
- [Kla05] G. Klančar, D. Matko und S. Blažič. „Mobile Robot Control on a Reference Path“. In: *Mediterranean Conference on Control and Automation*. 2005, S. 1343–1348.
- [Kla07] G. Klančar und I. Škrjanc. „Tracking-error model-based predictive control for mobile robots in real time“. In: *Robotics and Autonomous Systems* 55.6 (2007), S. 460–469.
- [Li11] Y. Li, S. Ge, C. Yang, X. Li und K. P. Tee. „Model-free impedance control for safe human-robot interaction“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, S. 6021–6026.
- [Lik14] N. Likar und L. Žlajpah. „External Joint Torque-based Estimation of Contact Information“. In: *International Journal of Advanced Robotic Systems* 11.107 (2014), S. 1–11.
- [Mag14] E. Magrini, F. Flacco und A. De Luca. „Estimation of contact forces using a virtual force sensor“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2014, S. 2126–2133.
- [Mau11] I. Maurovic, M. Baotic und I. Petrovic. „Explicit Model Predictive Control for trajectory tracking with mobile robots“. In: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2011, S. 712–717.

- [Mer08] A. Merabet und J. Gu. „Robust nonlinear predictive control based on state estimation for robot manipulator“. In: *International Journal of Applied Mathematics and Mechanics* 5.1 (2008), S. 48–64.
- [Mil12] G. Milighetti, E. Hoffmann, A. Fetzner und H.-B. Kuntze. „Visually and Force Controlled Opening and Closing of Doors by Means of a Mobile Robot Arm“. In: *Joint 43th International Symposium on Robotics and 7th German Conference on Robotics (ISR/Robotik)*. 2012, S. 123–128.
- [Moh01] A. Mohri, S. Furuno und M. Yamamoto. „Trajectory planning of mobile manipulator with end-effector’s specified path“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Bd. 4. 2001, S. 2264–2269.
- [Ngo07] M. D. Ngo, N. T. Phuong, V. H. Duy, H. K. Kim und S. B. Kim. „Control of two Wheeled Welding Mobile Manipulator“. In: *International Journal of Advanced Robotic Systems* 4 (2007), S. 293–302.
- [Noc06] J. Nocedal und S. J. Wright. *Numerical Optimization*. Hrsg. von T. V. Mikosch und S. M. Robinson. Springer Series in Operations Research. Springer, 2006.
- [Omr03] D. Omrcen, B. Nemeč und L. Zlajpah. „Torque-velocity control algorithm for on-line obstacle avoidance for mobile manipulators“. In: *IEEE International Conference on Industrial Technology (ICIT)*. Bd. 2. 2003, S. 784–789.
- [Pap00] E. Papadopoulos und J. Poulakakis. „Planning and Model-Based Control for Mobile Manipulators“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2000.
- [Pen14] J. Peng, J. Yu und J. Wang. „Robust adaptive tracking control for nonholonomic mobile manipulator with uncertainties“. In: *ISA Transactions* 53.4 (2014), S. 1035–1043.
- [Pet13] J. Petereit, T. Emter und C. W. Frey. „Mobile Robot Motion Planning in Multi-Resolution Lattices with Hybrid Dimensionality“. In: *IFAC Intelligent Autonomous Vehicles Symposium*. 2013, S. 158–163.
- [Pla11] R. Platt, M. Abdallah und C. Wampler. „Multiple-priority impedance control“. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, S. 6033–6038.

- [Poi00] P. Poignet und M. Gautier. „Nonlinear model predictive control of a robot manipulator“. In: *International Workshop on Advanced Motion Control*. 2000, S. 401–406.
- [Pol14] M. Polverini, A. Zanchettin und P. Rocco. „Real-time collision avoidance in human-robot interaction based on kinetostatic safety field“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2014, S. 4136–4141.
- [Pri17] S. Primatesta und B. Bona. „Motion control of mobile robots with Particle Filter Model Predictive Equilibrium Point Control“. In: *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2017, S. 11–16.
- [ROS17] *ROS.org | Powering the world's robots*. URL: <http://www.ros.org> (besucht am 22. Mai 2017).
- [Röw12] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff und W. Burgard. „On the position accuracy of mobile robot localization based on particle filters combined with scan matching“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, S. 3158–3164.
- [Ryb12] P. Rybski, P. Anderson-Sprecher, D. Huber, C. Niessl und R. Simmons. „Sensor fusion for human safety in industrial workcells“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, S. 3612–3619.
- [Sad12] H. Sadeghian, M. Keshmiri, L. Villani und B. Siciliano. „Null-space impedance control with disturbance observer“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, S. 2795–2800.
- [Sal12] G. H. Salazar-Silva, M. A. Moreno-Armendáriz und J. Álvarez Gallegos. „Modeling and Control in Task-Space of a Mobile Manipulator with Cancellation of Factory-Installed Proportional-Derivative Control“. In: *Computación y Sistemas* 16 (2012), S. 409–419.
- [Sci00] L. Sciavicco und B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.
- [Ser98] H. Seraji. „A Unified Approach to Motion Control of Mobile Manipulators“. In: *International Journal of Robotics Research* 17.2 (1998), S. 107–118.

- [Sey16] R. Seyboldt, C. Frese und A. Zube. „Sampling-based Path Planning to Cartesian Goal Positions for a Mobile Manipulator Exploiting Kinematic Redundancy“. In: *47st International Symposium on Robotics (ISR)*. 2016, S. 1–9.
- [Sic08] B. Siciliano und O. Khatib, Hrsg. *Springer Handbook of Robotics*. Springer, 2008.
- [Sic90] B. Siciliano. „Kinematic Control of Redundant Robot Manipulators: A Tutorial“. In: *Journal of Intelligent and Robotic Systems* 3.3 (1990), S. 201–212.
- [Suc17] I. A. Sucas und S. Chitta. *MoveIt!* 2017. URL: <http://moveit.ros.org> (besucht am 22. Mai 2017).
- [Tan03] H. Tanner, S. Loizou und K. Kyriakopoulos. „Nonholonomic navigation and control of cooperating mobile manipulators“. In: *IEEE Transactions on Robotics and Automation* 19.1 (2003), S. 53–64.
- [Tan09] J. Tan, F. Duan, Y. Zhang, R. Kato und T. Arai. „Safety design and development of human-robot collaboration in cellular manufacturing“. In: *IEEE International Conference on Automation Science and Engineering (CASE)*. 2009, S. 537–542.
- [Utk91] V. I. Utkin, S. V. Drakunov, H. Hashimoto und F. Harashima. „Robot path obstacle avoidance control via sliding mode approach“. In: *IEEE/RSJ International Workshop on Intelligence for Mechanical Systems*. 1991, S. 1287–1290.
- [Viv05] A. Vivas und V. Mosquera. „Predictive Functional Control of a PUMA Robot“. In: *ICGST Conference on Automatic Control and System Engineering (ACSE)*. 2005.
- [Wäc06] A. Wächter und L. T. Biegler. „On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming“. In: *Mathematical Programming* 106.1 (2006), S. 25–57.
- [Wan08] Y. Wang, H. Lang und C. de Silva. „Visual Servo Control and Parameter Calibration for Mobile Multi-robot Cooperative Assembly Tasks“. In: *IEEE International Conference on Automation and Logistics (ICAL)*. 2008, S. 635–639.

- [Wan12] L. Wang. „Collaborations towards adaptive manufacturing“. In: *IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 2012, S. 14–21.
- [Wu13] P.-W. Wu, Y.-C. Lin, C.-M. Wang und L.-C. Fu. „Grasping the object with collision avoidance of wheeled mobile manipulator in dynamic environments“. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, S. 5300–5305.
- [Xue17] L. Xue und G. Zhiyong. „Adaptive sliding mode tracking control for nonholonomic wheeled mobile robots with finite time convergence“. In: *Chinese Control Conference (CCC)*. 2017, S. 720–725.
- [Zha12] H. Zhang, Y. Jia, N. Xi und A. Song. „Obstacle avoidance for mobile manipulation by real-time sensor-based redundancy resolution“. In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2012, S. 2369–2374.
- [Zub14] A. Zube, A. Jung und C. Frese. „Robot Path Adaptation for Shared Human-Robot Workspaces“. In: *7th International Workshop on Human-Friendly Robotics (HFR)*. 2014.
- [Zub15a] A. Zube. „Cartesian Nonlinear Model Predictive Control of Redundant Manipulators Considering Obstacles“. In: *IEEE International Conference on Industrial Technology (ICIT)*. 2015, S. 137–142.
- [Zub15b] A. Zube. „Combined Workspace Monitoring and Collision Avoidance for Mobile Manipulators“. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2015.
- [Zub15c] A. Zube und C. Frese. „Collision Avoidance for Human-Robot Interaction Distinguishing between Static and Dynamic Obstacles“. In: *8th International Workshop on Human-Friendly Robotics (HFR)*. 2015.
- [Zub16] A. Zube, J. Hofmann und C. Frese. „Model Predictive Contact Control for Human-Robot Interaction“. In: *47th International Symposium on Robotics (ISR)*. 2016, S. 279–285.

Liste der Veröffentlichungen

- [Emt17] T. Emter, C. Frese, A. Zube und J. Petereit. „Algorithmen-Toolbox für autonome mobile Robotersysteme“. In: *ATZoffhighway 3* (2017).
- [Sey16] R. Seyboldt, C. Frese und A. Zube. „Sampling-based Path Planning to Cartesian Goal Positions for a Mobile Manipulator Exploiting Kinematic Redundancy“. In: *47st International Symposium on Robotics (ISR)*. 2016, S. 1–9.
- [Zub16] A. Zube, J. Hofmann und C. Frese. „Model Predictive Contact Control for Human-Robot Interaction“. In: *47th International Symposium on Robotics (ISR)*. 2016, S. 279–285.
- [Zub15a] A. Zube. „Cartesian Nonlinear Model Predictive Control of Redundant Manipulators Considering Obstacles“. In: *IEEE International Conference on Industrial Technology (ICIT)*. 2015, S. 137–142.
- [Zub15b] A. Zube. „Combined Workspace Monitoring and Collision Avoidance for Mobile Manipulators“. In: *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2015.
- [Zub15c] A. Zube und C. Frese. „Collision Avoidance for Human-Robot Interaction Distinguishing between Static and Dynamic Obstacles“. In: *8th International Workshop on Human-Friendly Robotics (HFR)*. 2015.

- [Fet14] A. Fetzner, C. Frese und C. Frey. „A 3D Representation of Obstacles in the Robot’s Reachable Area Considering Occlusions“. In: *Joint 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik)*. 2014, S. 85–92.
- [Fre14] C. Frese, A. Fetzner und C. Frey. „Multi-Sensor Obstacle Tracking for Safe Human-Robot Interaction“. In: *Joint 45th International Symposium on Robotics and 8th German Conference on Robotics (ISR/Robotik)*. 2014, S. 784–791.
- [Zub14] A. Zube, A. Jung und C. Frese. „Robot Path Adaptation for Shared Human-Robot Workspaces“. In: *7th International Workshop on Human-Friendly Robotics (HFR)*. 2014.
- [Fet13] A. Fetzner, C. Frese und C. Frey. „Obstacle Detection and Tracking for Safe Human-Robot Interaction Based on Multi-Sensory Point Clouds“. In: *6th International Workshop on Human-Friendly Robotics (HFR)*. 2013.
- [Mil12] G. Milighetti, E. Hoffmann, A. Fetzner und H.-B. Kuntze. „Visually and Force Controlled Opening and Closing of Doors by Means of a Mobile Robot Arm“. In: *Joint 43th International Symposium on Robotics and 7th German Conference on Robotics (ISR/Robotik)*. 2012, S. 123–128.

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

- Band 1** Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz. 2006
ISBN 3-86644-070-7
- Band 2** Elisabeth Peinsipp-Byma
Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse. 2007
ISBN 978-3-86644-149-1
- Band 3** Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme. 2010
ISBN 978-3-86644-457-7
- Band 4** Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2010
ISBN 978-3-86644-469-0
- Band 5** Thomas Usländer
Service-oriented design of environmental information systems. 2010
ISBN 978-3-86644-499-7
- Band 6** Giulio Milighetti
Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter. 2010
ISBN 978-3-86644-568-0
- Band 7** Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2011
ISBN 978-3-86644-609-0
- Band 8** Eduardo Monari
Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken. 2011
ISBN 978-3-86644-729-5

- Band 9** Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen. 2011
ISBN 3-86644-760-8
- Band 10** Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile. 2012
ISBN 978-3-86644-798-1
- Band 11** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2012
ISBN 978-3-86644-855-1
- Band 12** Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES). 2013
ISBN 978-3-86644-955-8
- Band 13** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2013
ISBN 978-3-86644-988-6
- Band 14** Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip. 2013
ISBN 978-3-7315-0041-4
- Band 15** Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8
- Band 16** Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung. 2014
ISBN 978-3-7315-0167-1
- Band 17** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0212-8

- Band 18** Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance. 2015
ISBN 978-3-7315-0320-0
- Band 19** Marco Huber
Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications. 2015
ISBN 978-3-7315-0338-5
- Band 20** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2014
ISBN 978-3-7315-0401-6
- Band 21** Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen. 2015
ISBN 978-3-7315-0426-9
- Band 22** Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration. 2016
ISBN 978-3-7315-0457-3
- Band 23** Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung. 2016
ISBN 978-3-7315-0460-3
- Band 24** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2016
ISBN 978-3-7315-0519-8
- Band 25** Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement. 2016
ISBN 978-3-7315-0538-9
- Band 26** Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonardaten für ein autonomes Unterwasserfahrzeug. 2016
ISBN 978-3-7315-0541-9

- Band 27** Janko Petereit
Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments. 2017
ISBN 978-3-7315-0580-8
- Band 28** Erik Ludwig Krempel
Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen. 2017
ISBN 978-3-7315-0598-3
- Band 29** Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017
ISBN 978-3-7315-0638-6
- Band 30** Andrey Belkin
World Modeling for Intelligent Autonomous Systems. 2017
ISBN 978-3-7315-0641-6
- Band 31** Chettapong Janya-Anurak
Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos. 2017
ISBN 978-3-7315-0642-3
- Band 32** David Münch
Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information. 2017
ISBN 978-3-7315-0644-7
- Band 33** Jürgen Beyerer, Alexey Pak (Eds.)
Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2017
ISBN 978-3-7315-0678-2
- Band 34** Jürgen Beyerer, Alexey Pak and Miro Taphanel (Eds.)
Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. 2018
ISBN 978-3-7315-0779-6
- Band 35** Michael Grinberg
Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking. 2018
ISBN 978-3-7315-0781-9

- Band 36** Christian Herrmann
Video-to-Video Face Recognition for Low-Quality Surveillance Data. 2018
ISBN 978-3-7315-0799-4
- Band 37** Chengchao Qu
Facial Texture Super-Resolution by Fitting 3D Face Models. 2018
ISBN 978-3-7315-0828-1
- Band 38** Miriam Ruf
Geometrie und Topologie von Trajektorienoptimierung für vollautomatisches Fahren. 2018
ISBN 978-3-7315-0832-8
- Band 39** Angelika Zube
Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung. 2018
ISBN 978-3-7315-0855-7

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

Gemeinsame Arbeitsräume von Mensch und Roboter sowie die physische Interaktion von Mensch und Roboter sind vielversprechende Möglichkeiten, um Produktionsabläufe flexibler zu gestalten. Grundlegende Voraussetzung hierfür ist zunächst, dass die Sicherheit der Menschen im Roboterumfeld stets gewährleistet ist. Ebenso entscheidend für die Akzeptanz eines kollaborierenden Roboters ist, dass sich Menschen in der unmittelbaren Umgebung des Roboters wohlfühlen. Um diese Ziele zu erreichen, wird in der vorliegenden Arbeit eine Methode zur Überwachung der komplexen, dynamischen Umgebung des Roboters vorgestellt. Die Roboterbewegung wird basierend auf dem Konzept der modellprädiktiven Regelung unter Berücksichtigung der detektierten Hindernisse und der beabsichtigten oder unbeabsichtigten Kontakte des Roboters mit seiner Umgebung geregelt, um Kollisionen zu vermeiden und angemessen auf Kontakte zu reagieren. Die Ansätze werden auf einem mobilen Manipulator validiert.

ISSN 1863-6489
ISBN 978-3-7315-0855-7

