

# **Trajektorienoptimierung für Roboterarme in dynamischen Umgebungen**

**Lokale Planung mit fortlaufendem Horizont unter zeitlichen Beschränkungen**

von der

Fakultät für Elektrotechnik und Informationstechnik  
der  
Technischen Universität Dortmund

genehmigte

**DISSERTATION**

zur Erlangung des akademischen Grades  
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

vorgelegt von

Maximilian Krämer, M.Sc.

Referent: Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram,  
Technische Universität Dortmund  
Korreferent: Univ.-Prof. Dr.-Ing. Dr. h.c. Burkhard Corves,  
Rheinisch-Westfälische Technische Hochschule Aachen

Tag der mündlichen Prüfung: 30.10.2024



# Vorwort

Die vorliegende Dissertation ist im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Regelungssystemtechnik der Fakultät für Elektro- und Informationstechnik der Technischen Universität Dortmund entstanden. Sie wäre ohne die Unterstützung vieler Menschen nicht möglich gewesen, denen ich an dieser Stelle meinen Dank aussprechen möchte.

Besonderer Dank gilt Herrn Univ.-Prof. Dr.-Ing. Prof. h.c. Dr. h.c. Torsten Bertram, der mich durch seine Betreuung während der gesamten Promotionszeit begleitet hat. Sein Vertrauen in meine Fähigkeiten und die Freiheiten, die er mir sowohl in fachlicher als auch persönlicher Hinsicht gewährte, waren von unschätzbarem Wert und haben maßgeblich zu meinem Wachstum beigetragen.

Ein herzliches Dankeschön geht an Herrn Univ.-Prof. Dr.-Ing. Dr. h.c. Burkhard Corves für die Übernahme des Korreferats und seine aufgewendete Zeit sowie sein Interesse an meiner Arbeit. Für seine ansteckende Leidenschaft und Inspiration danke ich Herrn apl. Prof. Dr. rer. nat. Frank Hoffmann. Seine Begeisterung für das wissenschaftliche Arbeiten und das Teilen seines umfassenden Fachwissens waren für mich nicht nur lehrreich, sondern auch eine Quelle zahlreicher Ideen und Denkanstöße.

Ein großer Dank gilt auch Herrn Dr.-Ing. Daniel Schauten, der mir mit Rat und Tat in unterschiedlichsten Angelegenheiten zur Seite stand. Seine Fähigkeit, mich mit Perspektivwechseln aus festgefahrenen Gedanken zu holen, waren stets wertvoll.

Mein Dank geht außerdem an Nicole Czerwinski für ihre unermüdliche Unterstützung in administrativen Angelegenheiten sowie an Jürgen Limhoff, Rainer Müller-Burtscheid, Sascha Kersting und Halit Cicek für die stets schnelle und unkomplizierte Hilfe bei technischen Fragen, sodass ich mich ganz meiner Arbeit widmen konnte.

Ein großes Lob gilt schließlich den ehemaligen und aktuellen Kolleginnen und Kollegen des Lehrstuhls, die ein produktives und zugleich familiäres Arbeitsklima geschaffen haben, in dem ich mich sehr wohlfühlt habe.

Die intensive Auseinandersetzung mit der Forschung, die unzähligen Stunden im Labor oder vor dem Rechner sowie die unvermeidlichen Höhen und Tiefen haben auch mein privates Leben geprägt. Meiner Familie und meinen Freunden gilt an dieser Stelle besonderer Dank dafür, dass sie den Weg für meine persönliche und berufliche Entwicklung geebnet beziehungsweise ihn stets an meiner Seite begleitet haben.

Ein ganz besonderer Dank gebührt meiner Lebensgefährtin Valentina, die während dieser Zeit vieles zurückgestellt hat, um mir die Verwirklichung meines persönlichen Ziels zu ermöglichen. Mit viel Verständnis und Geduld demonstrierte sie bedingungslosen Rückhalt und war auch in den herausforderndsten Momenten eine wichtige Stütze.

Lünen, November 2024

Maximilian Krämer



# Kurzfassung

Zur Verringerung der Produktionskosten und Verbesserung der Arbeitsbedingungen werden die Fähigkeiten von Menschen mit denen von Roboterarmen kombiniert. Roboterhersteller leisten mit kollaborativen Varianten konstruktiv einen Grundbeitrag zur sicheren Zusammenarbeit. Die Kollisionsvermeidung muss jedoch durch spezielle Algorithmen zur Bewegungsplanung der Roboter erzielt werden. Ferner entstehen variable und potentiell konfliktäre Zielstellungen zwischen Roboter und Mensch, die ebenfalls eine ständige Neuplanung der Roboterbewegung erfordern. Klassische Methoden zur Bewegungsplanung sind meist statisch und berücksichtigen Änderungen an der Umgebung oder der Aufgabe nicht in dem Maße, wie es für geteilte Arbeitsräume notwendig ist. Die vorliegende Arbeit verfolgt das Grundprinzip der wiederkehrenden Planung mit fortlaufendem Horizont über die Optimierung von lokalen Teilbewegungen. Die Ziele sind zum einen das Erreichen einer möglichst hohen Planungsfrequenz zur Steigerung der Reaktivität sowie die Bewertung der Effektivität in dynamischen Umgebungen und bei variablen Zielstellungen. Zum anderen stehen die Auswirkungen der Lokalitäten des Lösungsverfahrens und Planungsproblems sowie Ansätze zur Abmilderung im Fokus.

Die Arbeit beginnt im ersten Teil mit der Lokalisierung von relevantem Rechenaufwand, der Identifikation von Einsparpotential und Maßnahmen zur Reduzierung der Laufzeit. Darunter fällt die Auswahl der optimalen Kombination von Distanzfunktionen, der Repräsentation von Optimierungsproblem und Hindernissen sowie die Ausnutzung spezieller Strukturen im Optimierungsproblem. Im zweiten Teil steht die Durchführung und Bewertung von Versuchen zur Kollisionsvermeidung mit Personen sowie variablen und konfliktären Zielstellungen im Fokus. Der letzte Teil stellt zunächst mit der erweiterten Initialisierung über eine parallele Explorationsinstanz eine Methode gegen die Lokalität gradientenbasierter Lösungsverfahren vor. Abschließend wird zur Vermeidung von strukturellen lokalen Minima eine globale Punktfolge eingeführt.

Die Ergebnisse empfehlen die Repräsentation des vollständig diskretisierten Optimierungsproblems über einen Hypergraphen zur Bewahrung dünnbesetzter Strukturen in Kombination mit einem Innere-Punkte-Verfahren und der Modellierung von Hindernissen über sphärische Volumenkörper. Damit sind bei einer Horizontlänge von 11 Zuständen und einer Planungsfrequenz von 10 Hz Szenarien mit bis zu 220 kugelförmigen Hindernissen möglich. Ferner ist bei derselben Planungsfrequenz und einer Horizontlänge von 31 Zuständen die Kollisionsvermeidung zu einer Person möglich. Auch für die betrachteten dynamischen und konfliktären Zielstellungen in Form von beweglichen Zielzuständen und sekundären Planungskriterien sind Planungsfrequenz und -methode geeignet. Die erweiterte Initialisierung mittels paralleler Explorationsinstanz stellt eine neue Form der Initialisierung dar, die zu besseren Lösungen führt und mit anderen Initialisierungen kombiniert werden kann. Die globale Punktfolge vermeidet unter bestimmten Voraussetzungen strukturelle lokale Minima und bietet der Bewegungsplanung gleichzeitig Flexibilität zur Vermeidung von Kollisionen mit dynamischen Hindernissen.

In weiterführenden Arbeiten bietet sich die Ausnutzung weiterer aufgezeigter Einsparpotentiale sowie die Ausdehnung der Versuche auf konkrete Anwendungsszenarien an. Die Vorteile einer Kombination mit neuartigen Prädiktionsmethoden sind vielversprechend. Strukturelle lokale Minima sind weiterhin ein Thema und es liegt die Ausweitung der globalen Punktfolge auf dynamische Hindernisse sowie die Relaxation der Annahmen dieser Methode nahe.



# Inhaltsverzeichnis

<b>Nomenklatur</b>	<b>iii</b>
<b>1. Einleitung</b>	<b>1</b>
<b>2. Stand der Forschung</b>	<b>5</b>
2.1. Von der einfachen zur wiederkehrenden Trajektorienplanung . . . . .	5
2.2. Wiederkehrende Planung mit fortlaufendem Horizont . . . . .	13
<b>3. Die allgemeine Planung mit fortlaufendem Horizont ohne Kollisionsvermeidung</b>	<b>17</b>
3.1. Definition des Planungsproblems . . . . .	17
3.2. Transkriptionen des nichtlinearen Parameteroptimierungsproblems . . . . .	19
3.3. Verfahren zur Lösung des Optimierungsproblems . . . . .	23
<b>4. Kollisionsvermeidung für statische Hindernisse</b>	<b>27</b>
4.1. Modellierung der Umgebung über sphärische Volumenkörper . . . . .	27
4.2. Distanzfunktionen für sphärische Volumenkörper . . . . .	29
4.3. Nebenbedingungen und Annäherungskosten zur Kollisionsvermeidung . . . . .	32
<b>5. Lösung des Planungsproblems unter zeitlichen Beschränkungen</b>	<b>39</b>
5.1. Spezialisierung auf die Trajektorienoptimierung . . . . .	39
5.2. Praktischer Umgang mit der Optimierungszeit . . . . .	46
<b>6. Dynamische Szenen bei der Planung mit fortlaufendem Horizont</b>	<b>51</b>
6.1. Kollisionsvermeidung bei dynamischen Hindernissen . . . . .	51
6.2. Annähern und Folgen dynamischer Zielzustände . . . . .	58
<b>7. Dynamische Priorisierung von sekundären Kostentermen</b>	<b>63</b>
7.1. Einbettung konkurrierender Kostenterme in die Planung . . . . .	63
7.2. Elastizitätsbewusste Planung für Roboter mit flexiblen Armkörpern . . . . .	65
<b>8. Lokalität der Lösungsverfahren</b>	<b>73</b>
8.1. Initialisierung des Optimierungsproblems . . . . .	73
8.2. Erweiterte Initialisierung über parallele Explorationsplanung . . . . .	77
<b>9. Lokalität des Planungsproblems</b>	<b>85</b>
9.1. Strukturelle lokale Minima bei beschränkten Horizonten . . . . .	86
9.2. Temporärer Zielzustand über globale Punktführung . . . . .	89
<b>10. Zusammenfassung und Ausblick</b>	<b>99</b>
<b>A. Daten zu den verwendeten Robotersystemen</b>	<b>103</b>
A.1. Universal Robot 10 (UR10) . . . . .	103
A.2. TU Dortmund Omni-elastic Robot Adapted (TUDORA) . . . . .	105

A.3. Roboterarm mit zwei Gelenken . . . . .	107
<b>B. Begleitinformationen zu den Versuchen</b>	<b>109</b>
B.1. Allgemeine Angaben . . . . .	109
B.2. Kapitel 4 . . . . .	110
B.2.1. Spezialfall bei sich schneidenden Primitiven . . . . .	110
B.2.2. Vergleich der Berechnungszeit verschiedener Implementierungen . . . . .	111
B.2.3. Einfluss der Gewichte . . . . .	111
B.2.4. Berechnungszeiten von Kostenfunktion und Nebenbedingung . . . . .	112
B.3. Kapitel 5 . . . . .	113
B.3.1. Spezialisierung auf die Trajektorienoptimierung . . . . .	113
B.3.2. Repräsentation von Abhängigkeiten mittels Hypergraphen . . . . .	114
B.3.3. Vergleich mit anderen Frameworks zur Trajektorienoptimierung . . . . .	114
B.4. Kapitel 6 . . . . .	116
B.4.1. Auswertung der Schnappschussmethode . . . . .	116
B.4.2. Prädiktive Kollisionsvermeidung mittels Zustandsextrapolation . . . . .	117
B.4.3. Annähern und Folgen dynamischer Zielkonfigurationen . . . . .	117
B.5. Kapitel 7 . . . . .	118
B.5.1. Parametrierung der Aktivierungsfunktion . . . . .	118
B.5.2. Bewertung der dynamischen Priorisierung . . . . .	118
B.6. Kapitel 8 . . . . .	119
B.6.1. Lineare Interpolation . . . . .	119
B.6.2. Warmstart . . . . .	119
B.6.3. Auswertung für ein dynamisches Hindernis . . . . .	121
B.7. Kapitel 9 . . . . .	122
B.7.1. Strukturelle lokale Minima bei beschränkten Horizonten . . . . .	122
B.7.2. Auswahl eines Verfahrens zur Ermittlung des globalen Pfads . . . . .	123
B.7.3. Parametrisierung von $\beta$ und $K_e^-$ . . . . .	124
B.7.4. Neuplanung bei dynamischem Zielzustand . . . . .	125
B.7.5. Qualitativer Vergleich zu anderen Methoden . . . . .	126
<b>Literatur</b>	<b>127</b>

# Nomenklatur

Abkürzungen (Initialwörter) werden gemäß der Empfehlung des Dudens nur für den Plural dekliniert und dafür einheitlich mit dem Buchstaben „s“ anstelle der Endung des zugrunde liegenden Wortes ergänzt.

## Abkürzungen

AABB	<i>Axis-Aligned Bounding Box</i>
ACADO	<i>Automatic Control and Dynamic Optimization</i>
ADMM	<i>Alternating Direction Method of Multipliers</i>
AL	<i>Augmented Lagrangian</i>
ALTRO	<i>Augmented Lagrangian Trajectory Optimization</i>
APF	<i>Artificial Potential Field</i>
ARIMA	<i>Auto-Regressive Integrated Moving Average</i>
AS	<i>Active Set</i>
BLAS	<i>Basic Linear Algebra Subroutines</i>
CasADi	<i>Computer Algebra System for Algorithmic Differentiation</i>
CHOMP	<i>Covariant Hamiltonian Optimization for Motion Planning</i>
EI	Erweiterte Initialisierung
EP	<i>Expanding Polytopes</i>
ES	Einfachschießverfahren
FCL	<i>Flexible Collision Library</i>
GJK	Gilbert–Johnson–Keerthi
GMM	<i>Gaussian Mixture Model</i>
GPU	<i>Graphics Processing Unit</i>
GRAMPC	<i>Gradient-Based Nonlinear Model Predictive Control</i>
GTE	<i>Geometric Tools Engine</i>
GTO	Globale Trajektorienoptimierung
HPP	<i>Hybrid Path Planning</i>
HQP	<i>Huge Quadratic Programming</i>
HSL	<i>Harwell Subroutine Library</i>
IKA	Inverse Kinematik Algorithmus
IP	Innere-Punkte
IPOpt	<i>Interior Point Optimizer</i>
IPTOP	<i>Interior Point Trajectory Optimization</i>
ITOMP	<i>Incremental Trajectory Optimization</i>
KKT	Karush-Kuhn-Tucker
LPRM	<i>Lazy Probabilistic Roadmap Method</i>
LSTM	<i>Long Short-Term Memory</i>
MA	<i>Multifrontal Approach</i>
MAE	<i>Mean Absolute Error</i>
MHP	<i>Moving Horizon Planning</i>
MKL	<i>Math Kernel Library</i>

MPC	<i>Model Predictive Controller</i>
MRK	Mensch-Roboter Kollaboration
MS	Mehrfachschießverfahren
MUMPS	<i>Multifrontal Massively Parallel Sparse Direct Solver</i>
MVC	<i>Maximum Velocity Curve</i>
NLOpt	Sammlung von Lösungsverfahren für <i>Nonlinear Optimization</i>
NLP	Nichtlineares Parameteroptimierungsproblem
ORB	<i>Oriented Bounding Box</i>
OMPL	<i>Open Motion Planning Library</i>
OSP	Optimalsteuerungsproblem
OSQP	<i>Operator Splitting (Solver) for Quadratic Programs</i>
PARDISO	<i>Parallel Direct Solver</i>
PI	Proportional-Integral
PMP	<i>Partial Motion Planning</i>
PRM	<i>Probabilistic Roadmap Method</i>
ProDMP	<i>Probabilistic Dynamic Motion Primitives</i>
pRRT	<i>Parallel Rapidly-Exploring Random Tree</i>
qpOases	Lösungsverfahren für quadratische Optimierungsprobleme über eine <i>Online Active Set Strategy</i>
ROS	<i>Robot Operating System</i>
RRT	<i>Rapidly-Exploring Random Tree</i>
RRTC	<i>Rapidly-Exploring Random Tree Connect</i>
SCP	<i>Sequential Convex Programming</i>
SLSQP	<i>Sequential Least-Squares Quadratic Programming</i>
SpBKP	<i>Sparse Bunch-Kaufman-Parlett</i>
SQP	<i>Sequential Quadratic Programming</i>
SSV	<i>Swept Sphere Volume</i>
STOMP	<i>Stochastic Trajectory Optimization for Motion Planning</i>
T-CHOMP	<i>Time Covariant Hamiltonian Optimization for Motion Planning</i>
TrajOpt	<i>Toolkit für Trajectory Optimization</i>
TUDORA	<i>TU Dortmund Omni-Elastic Robot Adapted</i>
UR10	<i>Universal Robot 10</i>
VD	Vollständige Diskretisierung

### Griechische Symbole

$\alpha(t)$	Zeitabhängige Bedarfsgröße (z. B. Schwingungsenergie) für sekundäre Kostenterme
$\alpha_n$	Bedarfsgröße (z. B. Schwingungsenergie) für sekundäre Kostenterme zum Planungszeitpunkt $t_n$
$\bar{\alpha}$	Schwellwert zur halben Aktivierung der Aktivierungsfunktion
$\beta$	Faktor bei der Berechnung von $\Delta\rho_n$
$\Gamma(\mathbf{x})$	Steuerbarkeitsmatrix eines Roboters
$\gamma_{j,i}(\mathbf{x})$	Steuerbarkeit der $i$ -ten Schwingungsebene aus Sicht des $j$ -ten Gelenks
$\delta(t)$	Dehnungssignal aus allen Schwingungsebenen
$\delta_i(t)$	Dehnungssignal in der $i$ -ten Schwingungsebene
$\epsilon$	Radius bei der Berechnung von $K_e$
$\zeta_f(t)$	Zeitabhängiges Tupel der Zielpose des Endeffektors

$\dot{\zeta}_f(t)$	Zeitabhängiges Tupel der Geschwindigkeit der Zielpose des Endeffektors
$\boldsymbol{\eta}$	Schlupfvariablen
$\eta_i$	$i$ -te Schlupfvariable
$\Theta$	Rotationsmatrix eines Hüllkörpers im Arbeitsraum
$\Theta(\mathbf{q})$	Gelenkwinkelabhängige Rotationsmatrix eines Hüllkörpers im Arbeitsraum
$\Theta(t)$	Zeitabhängige Rotationsmatrix eines Hüllkörpers im Arbeitsraum
$\Theta_{EE,f}(t)$	Zeitabhängige Zielrotationsmatrix des Endeffektors
$\Theta_{EE,f}$	Zielrotationsmatrix des Endeffektors
$l$	Parameter der linearen Interpolation zwischen zwei Punkten
$\kappa$	Steigung der Aktivierungsfunktion bei halber Aktivierung
$\Lambda(\cdot)$	Allgemeine nichtlineare Funktion
$\lambda_{n-1}^*$	Allgemeine optimale Lagrangemultiplikatoren nach Lösung des NLP zum Zeitpunkt $t_{n-1}$
$\lambda_G$	Lagrangemultiplikatoren der Gleichheitsnebenbedingungen
$\lambda_G^*$	Optimale Lagrangemultiplikatoren der Gleichheitsnebenbedingungen
$\lambda_{G,i}$	$i$ -ter Lagrangemultiplikator der Gleichheitsnebenbedingungen
$\lambda_U$	Lagrangemultiplikatoren der Ungleichheitsnebenbedingungen
$\lambda_U^*$	Optimale Lagrangemultiplikatoren der Ungleichheitsnebenbedingungen
$\lambda_{U,i}$	$i$ -ter Lagrangemultiplikator der Ungleichheitsnebenbedingungen
$\lambda'_U$	Lagrangemultiplikatoren der konvertierten Ungleichheitsnebenbedingungen beim Innere-Punkte-Verfahren
$\lambda'_{U,i}$	$i$ -ter Lagrangemultiplikator der konvertierten Ungleichheitsnebenbedingungen beim Innere-Punkte-Verfahren
$\boldsymbol{\mu}$	Stellgröße am Roboter
$\boldsymbol{\mu}(t)$	Zeitkontinuierliche Folge von Stellgrößen am Roboter
$\boldsymbol{\mu}'(t)$	Stellgröße der adaptiven Schwingungsdämpfung am Roboter
$\boldsymbol{\mu}(t)_D$	Zeitkontinuierliche Folge von Drehmomenten an den Gelenken des Roboters
$\mu_i$	Stellgröße des $i$ -ten Gelenks am Roboter
$v_i$	Parameter eines geometrischen Primitivs
$\Xi$	Diskrete Hahn-Fensterfunktion
$\xi$	Gewicht der Barrierefunktionen beim Innere-Punkte-Verfahren
$\rho$	Räumliche Parametrierung entlang einer globalen Lösung
$\rho_n$	Räumliche Parametrierung entlang einer globalen Lösung zum Zeitpunkt $t_n$
$\rho^+$	Maximale Skalierung auf der globalen Lösung, sodass eine lokale Lösung noch gültig ist
$\rho_{\max}$	Skalierung auf einer globalen Lösung, für welche die Kosten einer lokalen Lösung ein lokales Maximum annehmen
$\Delta\rho_n$	Verschiebung der Skalierung auf $\mathcal{P}$ zum Zeitpunkt $t_n$
$\Delta\rho_{BS}$	Verschiebung der Skalierung auf $\mathcal{P}$ beim <i>Backstepping</i>
$\sigma(\cdot)$	Aktivierungsfunktion des sekundären Kostenterms
$\tau$	Zeit innerhalb der Planung
$\tau_k$	Zeitpunkte auf dem diskreten Gitter des Planungsproblems
$\mathbf{v}_{0:I-1}$	Folge von Intervallen von zeitdiskreten Teilsteuerfolgen
$\phi_{j,i}(\mathbf{x})$	Winkel zwischen Rotationsachse des $j$ -ten Gelenks und Normalenvektor der $i$ -ten Schwingungsebene
$\chi^{(i)}$	<i>Shooting node</i> zu Beginn des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\chi_{0:I-1}$	Folge von zeitdiskreten <i>shooting nodes</i> beim Mehrfachschießverfahren

$\boldsymbol{\psi}$	Parametervektor des Schätzers
$\boldsymbol{\psi}^*$	Optimaler Parametervektor nach der Schätzung
$\psi_j$	$j$ -ter Polynomparameter
$\boldsymbol{\omega}_j(\mathbf{x})$	Zustandsabhängige Rotationsachse des $j$ -ten Gelenks
$\boldsymbol{\Omega}_{i,n}$	Frequenzspektrum des $i$ -ten Dehnungssignals zum Planungszeitpunkt $t_n$
$\Omega_{i,n,l}$	$l$ -te Frequenz aus dem Frequenzspektrum des $i$ -ten Dehnungssignals zum Planungszeitpunkt $t_n$
$\Omega^-$	Untere Frequenz des Bandpassfilters
$\Omega^+$	Obere Frequenz des Bandpassfilters

### Lateinische Symbole

$\mathbf{a}^+$	Obere Schranke der Gelenkwinkelbeschleunigung innerhalb der Planung
$\mathbf{a}^-$	Untere Schranke der Gelenkwinkelbeschleunigung innerhalb der Planung
$a^+$	Skalare obere Schranke der Gelenkwinkelbeschleunigung innerhalb der Planung
$\mathbf{A}_{i,n}$	Amplitudenspektrum des $i$ -ten Dehnungssignals zum Planungszeitpunkt $t_n$
$A_{i,n,l}$	$l$ -te Amplitude aus dem Amplitudenspektrum des $i$ -ten Dehnungssignals zum Planungszeitpunkt $t_n$
$\mathbf{b}_{\text{FT}}$	Messungen innerhalb des Fensters der diskreten Fouriertransformation
$\mathbf{b}_{\text{S}}$	Messungen innerhalb des Schätzfensters
$\mathcal{C}$	Menge an Hüllkörpern von statischen Hindernissen
$c_1(\mathbf{x})$	Primärer zustandsabhängiger Kostenterm
$c_2(\mathbf{x})$	Sekundärer zustandsabhängiger Kostenterm
$c_A(\mathbf{x})$	Kostenterm mit bedarfsgesteuerter Priorisierung des sekundären Kostenterms
$c_B(\mathbf{x})$	Kostenterm mit dynamischer Priorisierung des sekundären Kostenterms
$c_H(\cdot)$	Kostenterm der Potentiale für Hindernisse und Selbstkollisionen
$c_U(\cdot)$	Kostenterm zur Bewertung der Stellgröße
$c_X(\cdot)$	Kostenterm zur Bewertung der Abweichungen zum Zielzustand der Planung
$c_\Sigma(\mathbf{x})$	Kostenterm mit summierten Untertermen
$D$	Anzahl der Optimierungsvariablen
$\mathcal{D}$	Menge an Hüllkörpern von dynamischen Hindernissen
$d(\cdot)$	Vorzeichenbehaftete Distanzfunktion zwischen zwei Hüllkörpern
$d_S(\cdot)$	Separationsdistanz zwischen zwei Hüllkörpern
$d_P(\cdot)$	Penetrationsdistanz zwischen zwei Hüllkörpern
$\tilde{d}_P(\cdot)$	Approximierte Penetrationsdistanz
$\tilde{d}_P^+$	Maximale approximierten Penetrationsdistanz
$d_0$	Aktivierungsschwellwert von Potentialfunktionen
$d_R$	Aktivierungsschwellwert der Potentialfunktion bei Selbstkollisionen
$d_C$	Aktivierungsschwellwert der Potentialfunktionen bei statischen Hindernissen
$d_D$	Aktivierungsschwellwert der Potentialfunktionen bei dynamischen Hindernissen
$d_R^-$	Minimal erlaubte Distanz zwischen Armkörpern
$d_C^-$	Minimal erlaubte Distanz zwischen Armkörpern und statischen Hindernissen
$d_D^-$	Minimal erlaubte Distanz zwischen Armkörpern und dynamischen Hindernissen bei der Planung
$d^-$	Mindestabstand externer Sicherheitsfunktionen
$d_\infty$	Singularität von Potentialfunktionen
$d_{\text{eff}}$	Effektiver Abstand zwischen Drehachse des Gelenks und Kollisionspunkt
$\mathbf{e}(t)$	Regelabweichung im Inverse Kinematik Algorithmus

$E_i$	$i$ -te Schwingungsebene
$F_S$	Fensterbreite des Schätzers
$F_{FT}$	Fensterbreite der diskreten Fouriertransformation
$\mathbf{f}(\cdot)$	Zeitkontinuierliche Systemdynamik
$\hat{\mathbf{f}}(\cdot)$	Zeitdiskrete Systemdynamik
$\mathbf{g}(\cdot)$	Zeitdiskrete Gleichheitsnebenbedingungen des NLP
$\mathbf{g}_{MS}(\cdot)$	Zeitdiskrete Gleichheitsnebenbedingungen des NLP nach dem Mehrfachschießverfahren
$\mathbf{g}_{MS}^{(i)}(\cdot)$	Zeitdiskrete Gleichheitsnebenbedingungen des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\mathbf{g}_{VD}(\cdot)$	Zeitdiskrete Gleichheitsnebenbedingungen des NLP nach der vollständigen Diskretisierung
$g_i(\cdot)$	$i$ -te Gleichheitsnebenbedingung in $\mathbf{q}$
$g_k(\cdot)$	Gleichheitsnebenbedingung für den $k$ -ten Zeitschritt auf dem Horizont
$\mathbf{G}(\cdot)$	Jacobimatrix der Gleichheitsnebenbedingungen $\mathbf{g}(\cdot)$
$\mathbf{h}(\cdot)$	Zeitdiskrete Ungleichheitsnebenbedingungen des NLP
$\mathbf{h}_{ES}(\cdot)$	Zeitdiskrete Ungleichheitsnebenbedingungen des NLP nach dem Einfachschießverfahren
$\mathbf{h}_{MS}(\cdot)$	Zeitdiskrete Ungleichheitsnebenbedingungen des NLP nach dem Mehrfachschießverfahren
$\mathbf{h}_{MS}^{(i)}(\cdot)$	Zeitdiskrete Ungleichheitsnebenbedingungen des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\mathbf{h}_{VD}(\cdot)$	Zeitdiskrete Ungleichheitsnebenbedingungen des NLP nach der vollständigen Diskretisierung
$h_i(\cdot)$	$i$ -te Ungleichheitsnebenbedingung in $\mathbf{h}$
$h'_i(\cdot)$	$i$ -te über Schlupfvariablen konvertierte Gleichheitsnebenbedingung in $\mathbf{h}$
$h_k(\cdot)$	Ungleichheitsnebenbedingung für den $k$ -ten Zeitschritt auf dem Horizont
$\mathbf{H}(\cdot)$	Jacobimatrix der Ungleichheitsnebenbedingungen $\mathbf{h}(\cdot)$
$\mathbf{I}$	Regressormatrix des Schätzers
$I$	Anzahl der Intervalle beim Mehrfachschießverfahren
$\mathbf{J}(\mathbf{q})$	Geometrische Jacobimatrix eines Roboters
$J(\cdot)$	Zeitkontinuierliche Kostenfunktion eines Optimalsteuerungsproblems
$\hat{J}(\cdot)$	Zeitdiskrete Kostenfunktion eines NLP
$\hat{J}_{IP}(\cdot)$	Mit Barrierefunktionen erweiterte Kostenfunktion beim Innere-Punkte-Verfahren
$\hat{J}_f(\cdot)$	Kostenfunktion zur Bewertung von Zielkonfigurationen
$\mathbf{K}$	Verstärkungsmatrix des Inverse Kinematik Algorithmus
$K$	Anzahl der Zeitschritte auf dem Gitter des Planungsproblems
$K'$	Anzahl der Zeitschritte auf dem Gitter eines Intervalls beim Mehrfachschießverfahren
$K_D$	Anzahl der fortlaufenden Zustände des Horizonts mit Nebenbedingungen zur Vermeidung von Kollisionen mit dynamischen Hindernissen
$K_G$	Anzahl der Zustände einer globalen Zustandstrajektorie
$K_V$	Anzahl der Zustände bis zum Zwischenzustand $\mathbf{x}_V$
$K_\epsilon$	Anzahl der Zustände einer lokalen Lösung innerhalb einer $\epsilon$ -Umgebung um $\mathbf{x}'_{f,n-1}$
$K_P$	Anzahl der Zustände des Pfades $\mathcal{P}$
$K^-$	Untere Schranke der Anzahl der Zeitschritte auf dem Gitter des Planungsproblems

$K_{LM}^-$	Größte untere Schranke der Horizontlänge, für welche strukturelle lokale Minima auftreten
$K_e^-$	Horizontreserve bei der Berechnung von $\Delta\rho_n$
$k$	Laufindex auf dem Horizont
$\mathcal{K}_r$	Kugel mit Radius $r$
$L(\cdot)$	Lagrangefunktion
$L_{IP}(\cdot)$	Lagrangefunktion beim Innere-Punkte-Verfahren
$\ell$	Ähnlichkeitsmaß zwischen den Pfaden lokaler und globaler Zustandstrajektorien
$l^-$	Index der unteren Frequenz des Bandpassfilters
$l^+$	Index der oberen Frequenz des Bandpassfilters
$\mathbf{M}_{1,i}$	Homogene Transformationsmatrix für den ersten Schritt bei der Spiegelung von Zustandstrajektorien
$\mathbf{M}_{2,i}$	Homogene Transformationsmatrix für den zweiten Schritt bei der Spiegelung von Zustandstrajektorien
$M$	Dimension der Stellgröße
$N$	Dimension des Zustandsvektors
$N_C$	Dimensionen der Ungleichheitsnebenbedingungen zur Kollisionsvermeidung mit statischen Hindernissen
$N_D$	Dimensionen der Ungleichheitsnebenbedingungen zur Kollisionsvermeidung mit dynamischen Hindernissen
$N_E$	Anzahl der Schwingungsebenen
$N_G$	Anzahl der Gleichheitsnebenbedingungen
$N_P$	Polynomgrad
$N_{RC}$	Anzahl der berücksichtigten Kombinationen von Armkörpern und statischen Hindernissen zur Kollisionsvermeidung
$N_{RD}$	Anzahl der berücksichtigten Kombinationen von Armkörpern und dynamischen Hindernissen zur Kollisionsvermeidung
$N_{RR}$	Anzahl der berücksichtigten Kombinationen von Armkörpern zur Selbstkollisionsvermeidung
$N_R$	Dimensionen der Ungleichheitsnebenbedingungen zur Selbstkollisionsvermeidung
$N_U$	Anzahl der Ungleichheitsnebenbedingungen
$\mathbf{n}$	Normalenvektor
$n$	Laufindex der Planungsschritte
$\mathcal{O}$	Hüllkörper
$\mathcal{O}'$	Transformierter Hüllkörper
$\mathcal{O}(\mathbf{q})$	Gelenkwinkelabhängiger Hüllkörper
$\mathcal{O}(t)$	Zeitabhängiger Hüllkörper
$P$	Prädiktionslänge der Extrapolation
$\mathcal{P}(s)$	Geometrischer Pfad im Zustandsraum des Roboters parametrisiert über $s$
$\mathbf{p}$	Dreidimensionaler Punkt im Arbeitsraum
$\mathbf{p}'$	Transformierter Punkt im Arbeitsraum
$p(\cdot)$	Potentialfunktion
$p_{[1]}(\cdot)$	Potentialfunktion aus der Literatur
$\mathbf{Q}$	Positiv definite Gewichtungsmatrix für $c_X(\cdot)$
$Q$	Diagonalelemente von $\mathbf{Q}$
$Q_k$	Diagonalelemente von $\mathbf{Q}$ zum Zeitpunkt $t_k$ innerhalb der Planung

$\mathbf{q}$	Gelenkwinkel des Roboters
$\mathbf{q}(t)$	Ist-Gelenkwinkel des Roboters in Abhängigkeit der Zeit $t$
$q_i$	$i$ -ter Gelenkwinkel eines Roboters
$\dot{\mathbf{q}}(t)$	Ist-Gelenkwinkelgeschwindigkeiten des Roboters in Abhängigkeit der Zeit $t$
$\mathbf{R}$	Positiv definite Gewichtungsmatrix für $c_U(\cdot)$
$R$	Diagonalelemente von $\mathbf{R}$
$R_k$	Diagonalelemente von $\mathbf{R}$ zum Zeitpunkt $t_k$ innerhalb der Planung
$\mathcal{R}$	Menge an Hüllkörpern eines Roboters
$r$	Kugelradius
$r_H$	Kugelradius des Hüllkörpers eines Hindernisses
$r_R$	Kugelradius eines Hüllkörpers des Roboters
$r_{\text{MHP}}$	Radius der hindernisfreien Zielregion $\mathcal{R}(\cdot)$
$\mathbf{S}$	Diagonalmatrix mit binären Einträgen
$S$	Geometrisches Primitiv
$S_E$	Unendliche, dreidimensionale Fläche
$S_F$	Endliches, dreidimensionales Flächensegment als geometrisches Primitiv
$S_L$	Endliches, dreidimensionales Liniensegment als geometrisches Primitiv
$S_P$	Dreidimensionaler Punkt als geometrisches Primitiv
$\mathbf{s}$	Translation eines Hüllkörpers im Arbeitsraum
$\mathbf{s}(t)$	Zeitabhängige Translation eines Hüllkörpers im Arbeitsraum
$\mathbf{s}(\mathbf{x})$	Zustandsabhängige Translation eines Hüllkörpers im Arbeitsraum
$\mathbf{s}_{EE}(\mathbf{x})$	Zustandsabhängige Translation des Hüllkörpers des Endeffektors im Arbeitsraum
$\mathbf{s}_{EE,f}(t)$	Zeitabhängige Zieltranslation des Endeffektors
$\mathbf{s}_{EE,f}$	Zieltranslation des Endeffektors
$\mathbf{s}_H(t)$	Trajektorie der Translation des Hüllkörpers der Hüfte
$\tilde{\mathbf{s}}_H(t_n)$	Geschätzte lineare Geschwindigkeit des Hüllkörpers der Hüfte zum aktuellen Planungszeitpunkt
$\tilde{\ddot{\mathbf{s}}}_H(t_n)$	Geschätzte lineare Beschleunigung des Hüllkörpers der Hüfte zum aktuellen Planungszeitpunkt
$s_x$	x-Komponente der Translation eines Hüllkörpers im Arbeitsraum
$s_{x,H}(t)$	Trajektorie der x-Komponente der Translation des Hüllkörpers der Hüfte
$s_{x,H,i}$	$i$ -ter Messwert der x-Komponente der Translation des Hüllkörpers der Hüfte im Schätzfenster
$\tilde{s}_{x,H}(t_n)$	Geschätzte x-Komponente der linearen Geschwindigkeit des Hüllkörpers der Hüfte zum aktuellen Planungszeitpunkt
$\tilde{\ddot{s}}_{x,H}(t_n)$	Geschätzte x-Komponente der linearen Beschleunigung des Hüllkörpers der Hüfte zum aktuellen Planungszeitpunkt
$\tilde{s}_{x,H}(t)$	Geschätzte Trajektorie der x-Komponente der Translation des Hüllkörpers der Hüfte
$\mathbf{T}(\cdot)$	Vorwärtskinematik eines Roboters als homogene Transformationsmatrix
$\mathbf{T}_i(\cdot)$	Vorwärtskinematik des $i$ -ten Gelenks eines Roboters als homogene Transformationsmatrix
$\mathbf{T}_B$	Homogene Transformationsmatrix zwischen Weltkoordinatensystem und Roboterbasis
$\mathbf{T}_E$	Homogene Transformationsmatrix zwischen letztem per und Endeffektor
$\mathcal{T}_A$	Menge aller gültigen Zustandstrajektorien für die Aufgabe A
$T_B$	Bremszeit des Roboters bei maximaler Beschleunigung

$T_{FT}$	Abtastzeit der Dehnungsmessungen
$T_O$	Dauer der Optimierung
$T_P$	Zeitliche Schrittweite zwischen zwei Planungen
$T_S$	Abtastzeit der Messungen von Hüllkörperposen beim Schätzer
$T_T$	Zeitliche Schrittweite zwischen Gitterpunkten des Planungsproblems
$t$	Zeit außerhalb der Planung
$t_n$	Zeitpunkte der Planung
$t(n, k)$	Abbildung von Planungsschritt $n$ und Zustand $k$ auf die Zeit außerhalb der Planung
$t_f$	Zeitliche Länge der optimierten Trajektorie und Steuerfolge
$\mathcal{U}_A(t)$	Zeitabhängige Menge aller gültigen Stellgrößen im Sinne einer Planungsaufgabe A
$\mathbf{u}$	Stellgröße innerhalb der Planung
$\mathbf{u}(t)$	Zeitkontinuierliche Steuerfolge
$\mathbf{u}(\tau)$	Zeitkontinuierliche Steuerfolge innerhalb der Planung
$\mathbf{u}^*(t)$	Optimale zeitkontinuierliche Steuerfolge
$\mathbf{u}^*(\tau)$	Optimale zeitkontinuierliche Steuerfolge innerhalb der Planung
$\mathbf{u}_n^*(\tau)$	Optimale zeitkontinuierliche Steuerfolge innerhalb der Planung zum Zeitpunkt $t_n$
$\mathbf{u}^{(i)}(\tau)$	Zeitkontinuierliche Steuerfolge des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\mathbf{u}_{0:K'-1}^{(i)}$	Zeitdiskrete Teilsteuerfolge des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\mathbf{u}_k^{(i)}$	$k$ -tes Element auf der zeitdiskreten Teilsteuerfolge des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\mathbf{u}_{0:K-1}$	Zeitdiskrete Folge von Stellgrößen
$\mathbf{u}_k$	$k$ -tes Element der zeitdiskreten Folge von Stellgrößen
$\dot{\mathbf{u}}$	Zeitliche Ableitung der Stellgröße
$\mathbf{u}^+$	Obere Schranke der Stellgröße
$\mathbf{u}^-$	Untere Schranke der Stellgröße
$u^+$	Maximale Stellgröße eines Gelenks
$u$	Skalare Stellgröße
$V(\cdot)$	Allgemeine Kostenfunktion der optimierungsbasierten Trajektorienplanung
$\mathbf{v}_i$	Einheitsrichtungsvektor als Spiegelachse für die Zustände der $i$ -ten Komponente des Zustandsvektors
$\mathbf{v}_i^\perp$	Einheitsrichtungsvektor orthogonal zu $\mathbf{v}_i$
$v$	Betrag der Geschwindigkeit eines Hindernisses
$W_C$	Gewicht der Potentiale bei statischen Hindernissen
$W_D$	Gewicht der Potentiale zu dynamischen Hindernissen
$W_R$	Gewicht der Potentiale bei Selbstkollisionen
$W_S$	Gewicht der Kosten zur Steuerbarkeit von Schwingungsebenen
$\mathbf{X}_i$	In einer Matrix arrangierte homogene Punkte des zeitdiskreten Verlaufs der Werte der $i$ -ten Zustandskomponente
$\mathbf{X}'_i$	In einer Matrix arrangierte homogene Punkte des zeitdiskreten Verlaufs der gespiegelten Werte der $i$ -ten Zustandskomponente
$\mathcal{X}_A(t)$	Zeitabhängige Menge aller gültigen Zustände im Sinne einer Planungsaufgabe A
$\mathcal{X}_{AIK}$	Menge aller gültigen Zielkonfigurationen für eine Zielpose des Endeffektors
$\mathcal{X}_F(\mathbf{x}_f)$	Hindernisfreie Zielregion um $\mathbf{x}_f$

$\mathbf{x}$	Zustandsvektor
$\mathbf{x}^*$	Optimaler Zustand
$\mathbf{x}^G$	Zustand einer globalen Zustandstrajektorie
$\mathbf{x}_0$	Startzustand der Planung
$\mathbf{x}_f$	Zielzustand der Planung
$\mathbf{x}_f^*$	Optimaler Zielzustand der Planung
$\mathbf{x}'_{f,A}$	Vorübergehende Zielkonfiguration A
$\mathbf{x}'_{f,B}$	Vorübergehende Zielkonfiguration B
$\mathbf{x}'_{f,n}$	Vorübergehende Zielkonfiguration zum Zeitpunkt $t_n$
$\mathbf{x}_V$	Zwischenzustand bei der Initialisierung $\mathbf{z}^{LI+}$
$\mathbf{x}_j^P$	$j$ -ter Punkt auf $\mathcal{P}$
$\mathbf{x}_k$	$k$ -tes Element der zeitdiskreten Folge von Zuständen
$\mathbf{x}_k^{(i)}$	$k$ -tes Element auf der zeitdiskreten Teiltrajektorie der Zustände des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\tilde{\mathbf{x}}_0$	Geschätzter Anfangszustand
$\tilde{\mathbf{x}}_{H,k}$	Geschätzter Zustandsvektor des Hüllkörpers der Hüfte auf dem Horizont
$\tilde{\mathbf{x}}_{K,k}$	Geschätzter Zustandsvektor des Hüllkörpers des Kopfes auf dem Horizont
$\Delta\mathbf{x}_{H,k}$	Relativer Zustand des Hüllkörpers der Hüfte bezogen auf den aktuellsten Messwert
$\mathbf{x}(t)$	Zeitkontinuierliche Zustandstrajektorie
$\mathbf{x}^*(t)$	Optimale Zustandstrajektorie
$\mathbf{x}_f(t)$	Zeitabhängiger Zielzustand der Planung
$\mathbf{x}_H(t)$	Zustandstrajektorie des Hüllkörpers der Hüfte
$\mathbf{x}_K(t)$	Zustandstrajektorie des Hüllkörpers des Kopfes
$\dot{\mathbf{x}}(t)$	Zeitliche Ableitung der Zustandstrajektorie
$\mathbf{x}(\tau)$	Zeitkontinuierliche Zustandstrajektorie innerhalb der Planung
$\mathbf{x}^{(i)}(\tau)$	Zeitkontinuierliche Teiltrajektorie der Zustände des $i$ -ten Intervalls beim Mehrfachschießverfahren
$\mathbf{x}_{0:K-1}$	Zeitdiskrete Folge von Zuständen
$\mathbf{x}_{0:K-1,n}^*$	Optimale Zustandsfolge der Planung zum Zeitpunkt $t_n$
$\mathbf{x}_{0:K'-1}^{(i)}$	Zeitdiskrete Teiltrajektorie der Zustände des $i$ -ten Intervalls beim Mehrfachschießverfahren
$x_{0:K-1,i}$	Zeitdiskrete Folge der $i$ -ten Komponente von Zuständen
$x'_{0:K-1,i}$	Zeitdiskrete Folge der $i$ -ten Komponente von gespiegelten Zuständen
$\mathbf{x}^+$	Obere Schranke des Zustandsvektors
$\mathbf{x}^-$	Untere Schranke des Zustandsvektors
$x^+$	Skalare obere Schranke des Zustandsvektors
$x_i$	$i$ -te Komponente des Zustandsvektors
$x_{f,i}$	$i$ -te Komponente des Zielzustandsvektors
$x_{k,i}$	$i$ -te Komponente des $k$ -ten Zustandsvektors
$x$	Skalarer Zustand
$\tilde{x}_{x,H}$	$x$ -Komponente der extrapolierten Translation des Hüllkörpers der Hüfte
$\tilde{x}_{y,H}$	$y$ -Komponente der extrapolierten Translation des Hüllkörpers der Hüfte
$\tilde{x}_{z,H}$	$z$ -Komponente der extrapolierten Translation des Hüllkörpers der Hüfte
$\mathbf{y}$	Unabhängige Variable von $\Lambda(\cdot)$
$\mathbf{y}^*$	Nullstelle von $\Lambda(\cdot)$
$\mathbf{y}_j$	Annäherung an $\mathbf{y}^*$ in der $j$ -ten Iteration

$\mathcal{Z}_A$	Menge aller gültigen Optimierungsvektoren für Aufgabe A
$\mathcal{Z}_A(\mathbf{x}_0, t_n)$	Menge aller zum Zeitpunkt $t_n$ gültigen Optimierungsvektoren für Aufgabe A die im Zustand $\mathbf{x}_0$ beginnen
$\mathbf{z}$	Vektor der Optimierungsvariablen eines NLP
$\mathbf{z}_{ES}$	Vektor der Optimierungsvariablen eines NLP nach dem Einfachschießverfahren
$\mathbf{z}_{MS}$	Vektor der Optimierungsvariablen eines NLP nach dem Mehrfachschießverfahren
$\mathbf{z}_{VD}$	Vektor der Optimierungsvariablen eines NLP nach der vollständigen Diskretisierung
$\mathbf{z}^*$	Optimierungsvektor nach Lösung des NLP
$\mathbf{z}_n^*$	Optimierungsvektor nach Lösung des NLP zum Zeitpunkt $t_n$
$\mathbf{z}_n'^*$	Alternativer Optimierungsvektor nach Lösung des NLP zum Zeitpunkt $t_n$
$\mathbf{z}^I$	Initialer Optimierungsvektor
$\mathbf{z}_n^I$	Initialer Optimierungsvektor zum Planungszeitpunkt $t_n$
$\mathbf{z}^A$	Initialisierung über die aktuellen Gelenkwinkel des Roboters
$\mathbf{z}_n^A$	Initialisierung über die aktuellen Gelenkwinkel $\mathbf{q}(t_n)$ des Roboters zum Planungszeitpunkt $t_n$
$\mathbf{z}^{LI}$	Initialisierung über eine lineare Interpolation zwischen Start- und Zielzustand
$\mathbf{z}^{LI+}$	Initialisierung über eine lineare Interpolation zwischen Start- und Zielzustand mit Zwischenzuständen
$\mathbf{z}^M$	Initialisierung über eine gespiegelte alte Lösung
$\Delta \mathbf{z}$	Schrittweite des Differenzenquotienten bei der numerischen Ableitung von Größen des NLP
$\mathbf{z}^{WS}$	Initialisierung über eine alte Lösung (Warmstart)

# 1

## Einleitung

Roboter jeglicher Art sind aus sämtlichen Bereichen der Automatisierung nicht mehr wegzudenken. Insbesondere Roboterarme – im Folgenden vereinfacht Roboter genannt – zeichnen sich als universelle Handhabungsmaschinen mit Kraft, Präzision und Schnelligkeit aus. Damit diese Stärken optimal für den jeweiligen Einsatzzweck abgerufen werden, ist die Planung von Roboterbewegungen von besonderer Bedeutung.

Historisch wird die Art der Planung vornehmlich durch Anwendungen im industriellen Umfeld geprägt, die überwiegend statisch und repetitiv sind. Zum Schutz des Menschen erhalten Roboter oft zusätzlich durch Käfige einen exklusiven und deterministisch strukturierten Arbeitsraum. Damit genügt meist bereits eine einmalige Bewegungsplanung. Neuplanungen müssen nur selten durchgeführt werden, wenn sich Aufgabe oder Arbeitsraum verändern. In Abbildung 1.1 werden optimale Bereiche von Automatisierungslösungen hinsichtlich der Stückkosten in Abhängigkeit vom Produktionsvolumen skizziert. Der klassische Industrieroboter wird ab einem gesteigerten Produktionsvolumen von dedizierten Fertigungsanlagen hinsichtlich der Kosten unterboten, da diese exakt auf die Aufgabe abgestimmt sind und die Universalität von Robotern zulasten ihrer Effektivität geht. Im Übergangsbereich existieren häufig Mischformen beider Technologien. Die manuelle Fertigung wird im Zuge der Automatisierung von den traditionellen Industrierobotern in Bereiche geringster Stückzahlen verdrängt. Mischformen waren dabei bisher eher selten anzutreffen, da die Bewegungsfreiheit der Roboter gepaart mit ihren hohen Geschwindigkeiten und Massen lebensgefährliche Risiken birgt. Durch die Markteinführung sogenannter kollaborativer Roboter für Nutzlasten im Bereich einiger Kilogramm werden die Aspekte Geschwindigkeit

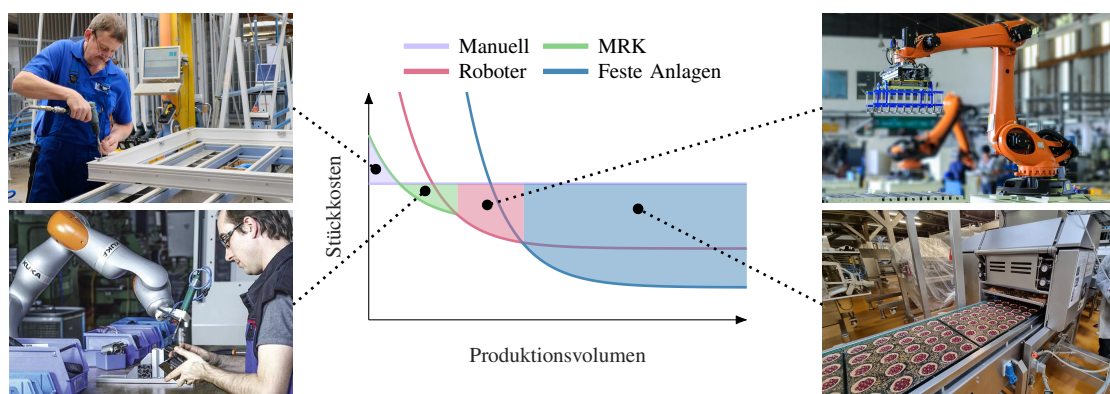


Abbildung 1.1.: Optimale Einsatzbereiche von Produktionslösungen. Symbolbilder: Manuelle Fertigung [AW 18], Mensch-Roboter Kollaboration (MRK) [Bél16], klassische Industrieroboter [Roc23] und feste Produktionsanlagen [Bak23]. Darstellung in Anlehnung an [Mat14].

und Kraft konstruktiv entschärft und damit die Benutzersicherheit und Bedienerfreundlichkeit erhöht, sodass im Rahmen der MRK<sup>1</sup> ein vierter Bereich entsteht [Mat+19]. Darin werden die Geschicklichkeit und Lernfähigkeit des Menschen mit der Ausdauer und Belastbarkeit des Roboters kombiniert. Dieser neue Anwendungsbereich in dynamischen Umgebungen, mit zum Teil variierenden oder konfliktären Zielen, stellt bisherige Algorithmen zur Bewegungsplanung vor neue Herausforderungen; unter anderem hinsichtlich Planungsfrequenz, Kollisionsvermeidung, Umgebungserfassung und Zielpriorisierung. Entsprechend leistungsfähige Algorithmen zur Bewegungsplanung stellen dabei einen signifikanten Nutzen für Mensch und Produktivität dar und unterstreichen ihren Bedarf als logischen nächsten Schritt nach den konstruktiven Maßnahmen der Roboterhersteller [Las+15; Dra+15]. In der Literatur werden im Zusammenhang mit der Bewegungsplanung häufig Begriffe wie Echtzeit oder *online* verwendet, ohne dabei einer konsistenten und einheitlichen Definition zu folgen. Diese Arbeit verwendet daher den übergeordneten Begriff der wiederkehrenden Planung.

Weitere Anwendungsfelder einer wiederkehrenden Bewegungsplanung für Roboter außerhalb der MRK sind z. B. der Zusammenschluss mehrerer verschiedenartiger Roboter für die eine holistische Bewegungsplanung nicht möglich ist [Gaf+22; Bos+11] sowie zur Laufzeit wechselnde Bewegungskriterien durch externe Einflüsse [Krä+21b]. Auch in der Weltraumforschung ist die wiederkehrende Bewegungsplanung beim Ausweichen vor Weltraumschrott oder dem Einsammeln von selbigem von Bedeutung [Wan+16].

Die vorliegende Arbeit behandelt mit der *Planung mit fortlaufendem Horizont* – im Folgenden MHP<sup>2</sup> genannt – eine Form der wiederkehrenden und optimierungsbasierten Bewegungsplanung, die bereits vielversprechende Ergebnisse erzielt hat. Das übergeordnete Ziel ist, einen Beitrag zur Bewertung und Verbesserung ihrer Anwendbarkeit zu leisten. Dies umfasst konkret die folgenden drei Teilziele:

1. Komposition eines Verfahrens zur effizienten Lösung des Planungsproblems,
2. Bewertung des Potentials und der Grenzen der erzielten Planungsfrequenzen in dynamischen Szenarien und
3. Methoden zur Begegnung der Herausforderungen der Lokalität des Lösungsverfahrens und Planungsproblems.

Nicht Teil dieser Arbeit ist die Auseinandersetzung mit dem Thema Sicherheit in der MRK vor dem Hintergrund von Garantien und industriellen Standards. Für eine ausführliche Aufstellung mit Arbeiten in diesem Bereich wird auf Robla-Gomez u. a. [Rob+17] verwiesen. Die Definition von Behrens u. a. [Beh+15] aufgreifend verfolgt diese Arbeit einen Ansatz zur kollisionsfreien Koexistenz zwischen Mensch und Roboter im geteilten Arbeitsraum und interpretiert diesen als Instrument für einen reibungslosen Betrieb und nicht als primäre Sicherheitsfunktion.

Im weiteren Verlauf wird in Kapitel 2 mit der Einordnung und Abgrenzung der Arbeit in den Stand der Forschung, inklusive der Einführung von Definitionen des allgemeinen globalen und lokalen Planungsproblems fortgefahren. Der inhaltliche Kern der Arbeit gliedert sich anschließend in drei wesentliche Teile, die in Abbildung 1.2 veranschaulicht sind.

Der erste Teil befasst sich in den Kapiteln 3 bis 5 mit technischen Herausforderungen bezüglich einer effizienten Trajektorienoptimierung am Beispiel eines kollaborativen Roboters. Die Neuartigkeit liegt dort überwiegend in Bewertung und Vergleich der Einzelschritte, die in Summe zu einer gesteigerten Planungsfrequenz führen:

---

<sup>1</sup>Mensch-Roboter Kollaboration

<sup>2</sup>*Moving Horizon Planning*

Teil 1 - Technische Herausforderungen			Teil 2 - Dynamische Szenen		Teil 3 - Lokalitäten	
<b>Kapitel 3:</b> <ul style="list-style-type: none"> <li>• Nichtlineares Parameteroptimierungsproblem</li> <li>• Transkription</li> <li>• Größenordnung</li> </ul>	<b>Kapitel 4:</b> <ul style="list-style-type: none"> <li>• Distanzfunktion</li> <li>• Kollisionen (statisch)</li> <li>• Laufzeitanalyse</li> </ul>	<b>Kapitel 5:</b> <ul style="list-style-type: none"> <li>• Strukturausnutzung</li> <li>• Lösungsverfahren</li> <li>• Laufzeitanalyse</li> <li>• Online-Dilemma</li> </ul>	<b>Kapitel 6:</b> <ul style="list-style-type: none"> <li>• Kollisionen (dynamisch)</li> <li>• Prädiktion</li> <li>• Dynamische Ziele</li> </ul>	<b>Kapitel 7:</b> <ul style="list-style-type: none"> <li>• Konkurrierende Kriterien</li> <li>• Dynamische Kriterien</li> </ul>	<b>Kapitel 8:</b> <ul style="list-style-type: none"> <li>• Parallele Exploration</li> <li>• Erweiterte Initialisierung</li> </ul>	<b>Kapitel 9:</b> <ul style="list-style-type: none"> <li>• Strukturelle lokale Minima</li> <li>• Globale Punktführung</li> </ul>
Ziel: Effiziente Lösung des Planungsproblems			Ziel: Anwendungsorientierte Bewertung		Ziel: Vermeidung lokaler Minima	

Abbildung 1.2.: Struktureller Aufbau des inhaltlichen Kerns der Arbeit gegliedert in drei Teile mit Teilkapiteln und -zielen.

**Kapitel 3** Das Kapitel verfolgt primär die Vorbereitung des ersten Teilziels in Form der Festlegung zentraler Größen, Kriterien und Bedingungen der Planung mit drei NLPs<sup>3</sup> als Resultat. Begleitet wird dies von der strukturellen Einordnung der MHP in die Regelkreistopologien von Robotern und der Skizzierung relevanter Lösungsverfahren. Eine Bewertung von Kostengewichten leistet einen ersten Beitrag zur Parametrisierung der MHP, die sich übergreifend bis in das Kapitel 6 fortsetzt.

**Kapitel 4** Das Kapitel geht mit einer Analyse der Berechnungszeit von Kostenfunktion und Nebenbedingungen der NLPs sowie dem Aufzeigen des theoretischen Einsparpotentials verstärkt auf den Kern des ersten Teilziels ein. Hierfür werden die NLPs zunächst um Annäherungskosten und Nebenbedingungen zur Kollisionsvermeidung statischer Hindernisse erweitert. Vor dem Hintergrund möglichst hoher Planungsfrequenzen erfolgt anschließend die Bewertung der Berechnungszeit unterschiedlicher Implementierungen von Distanzfunktionen und der Beitrag einer Methode zur Approximation der Penetrationsdistanz. Eine Abschätzung von Gewichten trägt derweil weiter zur Parametrisierung der MHP bei.

**Kapitel 5** Nach Reduzierung und Einordnung des Aufwands zur Berechnung einzelner Komponenten im vorherigen Kapitel wird das Teilziel in diesem Kapitel komplettiert. Für eine effiziente Lösung des NLP wird die optimale Kombination aus Transkriptionsmethode, Strukturausnutzung und Lösungsverfahren ermittelt. Der Beitrag ist neben einer neuen Kombination der drei Komponenten in erster Linie auswertungsorientiert, da Ergebnisse aus unterschiedlichen Quellen der Literatur oft nur vage miteinander vergleichbar oder widersprüchlich sind. An der Auswertung sind verschiedene Lösungsverfahren und spezialisierte Ansätze zur optimierungsbasierten Bewegungsplanung beteiligt. Insbesondere bei Laufzeiten ist eine einheitliche Implementierung und Auswertung auf derselben Hardware essentiell für belastbare Ergebnisse. Abschließend adressiert das Kapitel den praktischen Umgang mit der verbleibenden Berechnungszeit in Form von Zeitbudgets sowie die Behandlung des Online-Dilemmas und Tunneleffekts.

Der zweite Teil verlagert in den Kapiteln 6 und 7 die Ergebnisse des ersten Teils in eine anwendungsorientierte Bewertung der Berechnungszeiten unter dynamischen Gegebenheiten:

**Kapitel 6** Das Kapitel bewertet zunächst die Qualität der MHP bei dynamischen Hindernissen in Form einer Person in einem Szenario der MRK. Dies erfordert die Ausweitung des NLP auf die Kollisionsvermeidung für dynamische Hindernisse. Im Rahmen der

<sup>3</sup>Nichtlineares Parameteroptimierungsproblem

Auswertung wird neben dem Einfluss von Planungsfrequenz und Sicherheitsabständen zusätzlich der Nutzen prädizierter Hindernisbewegungen betrachtet, die über eine traditionelle Methode parallel zur MHP ermittelt werden. Die kapitelübergreifende Parametrisierung der MHP schließt mit der Auswertung zur Kollisionsvermeidung ab. Das Kapitel endet mit einem Vergleich des Übergangs- und Folgeverhaltens bei variablen Zielzuständen mit einem Ansatz aus der Literatur.

**Kapitel 7** Gegenstand dieses Kapitels ist die Priorisierung von konfliktären Kriterien während der Planung. Am Beispiel eines strukturelastischen Roboters werden zwei Methoden zur dynamischen Priorisierung der Effektivität von Schwingungsreglern vorgestellt und evaluiert. Strukturelastische Roboter bieten eine erhöhte Sicherheit durch ihre bauartbedingte Möglichkeit Kollisionsenergien abzubauen, müssen jedoch speziell zur Dämpfung von Schwingungen geregelt werden. Die erste Methode verfolgt einen proaktiven Ansatz, während die Zweite reaktiv agiert.

Der dritte und letzte Teil umfasst mit den Kapiteln 8 und 9 die Behandlung der Herausforderungen von Lokalitäten im Lösungsverfahren und Planungsproblem:

**Kapitel 8** Gradientenbasierte Lösungsverfahren erlauben eine zielgerichtete Optimierung, die allerdings nur auf lokalen Informationen der Kostenfunktion und Nebenbedingungen beruht. Zusammen mit der nicht-Konvexität des NLP ergibt diese Lokalität eine Abhängigkeit von der Initialisierung. Neben einer kurzen Bewertung der gängigsten Initialisierungen hinsichtlich ihrer Vor- und Nachteile im Kontext der MHP, wird eine neue Methode zur parallelen Explorativplanung vorgestellt, die lokale Informationen effektiver nutzt. In der abschließenden Auswertung wird die Methode mit etablierten globalen Planungsverfahren verglichen.

**Kapitel 9** Das vorletzte Kapitel überträgt die Betrachtung von Lokalitäten auf das Planungsproblem, wo sie in Form von strukturellen lokalen Minima aufgrund der Beschränkung des Planungshorizonts auftreten. Basierend auf einer Analyse zum Einfluss der Horizontlänge und Initialisierung auf diese Form der Lokalität wird eine Methode vorgestellt, welche die Auswirkungen bei der MHP reduziert. Die Methode wird hinsichtlich ihrer Effektivität bei statischen und dynamischen Hindernissen sowie variablen Zielzuständen und gegenüber etablierten globalen Verfahren bewertet.

Die Arbeit schließt in Kapitel 10 mit einer Zusammenfassung und einem Ausblick ab.

# 2

## Stand der Forschung

Dieses Kapitel beginnt im ersten Abschnitt zunächst mit einer kurzen Einführung in allgemeine Methoden der Trajektorienplanung mit dem Fokus auf optimierungsbasierte Ansätze. Der weitere Verlauf führt auf die wiederkehrende Planung und die Methoden zur Erhöhung der Planungsfrequenz sowie der Kollisionsvermeidung. Im zweiten Abschnitt wird speziell auf die MHP eingegangen und eine Einordnung der vorliegenden Arbeit in den Stand der Forschung vorgenommen. Es gelten die folgenden erweiterten Definitionen der natürlichen und reellen Zahlen:

$$\mathbb{N}_0 := \{\mathbb{N} \cup \{0\}\}, \mathbb{R}^+ := \{r \in \mathbb{R} \mid r > 0\} \text{ und } \mathbb{R}_0^+ := \{r \in \mathbb{R} \mid r \geq 0\}. \quad (2.1)$$

### 2.1. Von der einfachen zur wiederkehrenden Trajektorienplanung

Als Trajektorie  $\mathbf{x}(t)$  wird der zeitliche Verlauf der Bewegungszustände  $\mathbf{x} \in \mathbb{R}^N$  eines Roboters mit  $N \in \mathbb{N}$  Freiheitsgraden bezeichnet:

$$\mathbf{x}(t) : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R}^N \\ t & \mapsto \mathbf{x} \end{cases}. \quad (2.2)$$

Sie ordnet der streng monoton steigenden Zeit  $t$  einen Zustand  $\mathbf{x}$  zu. Ohne Beschränkung der Allgemeinheit stammen die Zustände im Folgenden aus dem Gelenkraum des Roboters. Sei  $\mathcal{T}_A \subseteq \mathbb{R} \times \mathbb{R}^N$  die Menge aller Trajektorien, welche die Bedingungen einer Aufgabe  $A$  erfüllen, dann wird die Bestimmung einer Trajektorie  $\mathbf{x}(t) \in \mathcal{T}_A$  als Trajektorienplanung bezeichnet. Die Bedingungen reichen z. B. von Start, Ziel und Verlauf über Beschränkungen der Dynamik bis hin zur Kollisionsfreiheit. Im Anschluss an die Trajektorienplanung erfolgt für gewöhnlich das Folgen der Trajektorie über einen Folgeregler, der passende Stellgrößen  $\boldsymbol{\mu} \in \mathbb{R}^M$  mit  $M \in \mathbb{N}$  bestimmt und an den Roboter überträgt:

$$\boldsymbol{\mu}(t) : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R}^M \\ t & \mapsto \boldsymbol{\mu} \end{cases}. \quad (2.3)$$

Auf diese Weise werden Planung und Ausführung getrennt, wodurch bei der Planung nahezu keine zeitlichen Beschränkungen vorliegen.

Die Menge  $\mathcal{T}_A$  liegt in der Regel nur implizit in Form der Bedingungen vor, sodass Verfahren auf unterschiedliche Weise eine Trajektorie konstruieren. Eine der einfachsten Methoden ermittelt

zunächst Wegpunkte über einen Pfadplaner, wie z. B. RRT<sup>1</sup> [LaV98] oder PRM<sup>2</sup> [Kav+96] und berechnet anschließend das Zeitgesetz zwischen den Wegpunkten entweder über interpolierende Polynome oder konstruierte Profile [Sic+09; Kha+02; Fan+16]. Die Möglichkeit, die resultierende Trajektorie in Richtung bestimmter Kriterien zu optimieren, ist dabei nur eingeschränkt über die geschickte Wahl der Profile möglich.

Die Trajektorienoptimierung verfolgt das Ziel, die geeignetste Trajektorie aus  $\mathcal{T}_A$  zu ermitteln:

$$\mathbf{x}^*(t) = \arg \min_{\mathbf{x}(t) \in \mathcal{T}_A} V(\mathbf{x}(t)). \quad (2.4)$$

Die Güte einer Trajektorie wird von einer Kostenfunktion  $V : \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}_0^+$  bewertet. Dabei bedarf es nicht zwingend einer expliziten Kostenfunktion, wie McCarthy u. a. [McC+92] für zeitoptimale Trajektorien gezeigt haben, die über den MVC<sup>3</sup>-Ansatz hergeleitet werden [Bob+85; Pfe+87; Bob88]. Beim MVC-Ansatz wird die Trajektorie über die Bestimmung von Schaltpunkten in der sogenannten Geschwindigkeitsebene zusammengesetzt, in der die MVC kinodynamische Grenzen repräsentiert. Von Kyriakopoulos u. a. [Kyr+88] wurde dieser Ansatz auf Trajektorien mit minimalem Ruck ausgeweitet. Für andere Kriterien oder zusätzliche Bedingungen ist es allerdings meist erforderlich, Trajektorien in einem iterativen Prozess mit expliziter Kostenfunktion zu planen. So z. B. der Ansatz von Cohen u. a. [Coh+10], der mit vordefinierten atomaren Bewegungsmustern über eine systematische Suche plant. Oder STOMP<sup>4</sup>, der einem evolutionären Algorithmus ähnelt und speziell auf die Berechnung von glatten und kollisionsfreien Trajektorien ausgelegt ist [Kal+11]. Nach der Idee von glatten Störungen generiert STOMP in einer Region um den aktuellen Kandidaten temporäre Trajektorien als Stichproben und kombiniert beides auf Basis der Kostenfunktion zu einem neuen glatten Kandidaten. Auf diese Weise wird die Menge  $\mathcal{T}_A$  abgetastet. Die Trajektorien werden dabei über diskrete Punkte repräsentiert. Das Pendant CHOMP<sup>5</sup> [Zuc+13] kommt ohne temporäre Stichproben aus und erzeugt glatte Verbesserungen des Kandidaten über die Gradienten der Kostenfunktion. Eine Herausforderung beider Ansätze ist jedoch die nichttriviale Parametrierung der Trajektorie, deren zeitliche Länge vor der Planung bekannt sein muss und in direktem Konflikt zu etwaigen Beschränkungen an die Geschwindigkeit und Beschleunigung steht. Pavlichenko u. a. [Pav+17] haben aus diesem Grund STOMP um die Berücksichtigung der Trajektoriendauer erweitert. Ebenfalls schwierig ist die Berücksichtigung von zeitlich variablen Kriterien oder Bedingungen, was Byravan u. a. [Byr+14] in der Erweiterung T-CHOMP<sup>6</sup> adressieren.

Die Systemtheorie besitzt im Bereich der optimalen Regelung eine weitere Methode zur Planung von optimalen Trajektorien. Ursprünglich zur optimalen Regelung von dynamischen Systemen gedacht, lässt sich die Theorie der optimalen Steuerung aufgrund der konzeptionellen Nähe zwischen den Anforderungen an eine Trajektorie und der Definition eines OSP<sup>7</sup> auch auf die Trajektorienplanung für Roboter übertragen:

$$\min_{\mathbf{x}(t), \mathbf{u}(t), t_f} J(\mathbf{x}(t), \mathbf{u}(t), t_f), \quad (2.5a)$$

---

<sup>1</sup>Rapidly-Exploring Random Tree

<sup>2</sup>Probabilistic Roadmap Method

<sup>3</sup>Maximum Velocity Curve

<sup>4</sup>Stochastic Trajectory Optimization for Motion Planning

<sup>5</sup>Covariant Hamiltonian Optimization for Motion Planning

<sup>6</sup>Time-CHOMP

<sup>7</sup>Optimalsteuerungsproblem

unter den Nebenbedingungen:

$$\mathbf{x}(0) = \mathbf{x}_0, \quad (2.5b)$$

$$\mathbf{x}(t_f) = \mathbf{x}_f, \quad (2.5c)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad \forall t \in [0, t_f], \quad (2.5d)$$

$$\mathbf{x}(t) \in \mathcal{X}_A(t) \quad \forall t \in [0, t_f], \quad (2.5e)$$

$$\mathbf{u}(t) \in \mathcal{U}_A(t) \quad \forall t \in [0, t_f]. \quad (2.5f)$$

Dieser Ansatz hat gleich mehrere Vorteile:

- Die Bedingungen an eine gültige Trajektorie für eine Aufgabe A können häufig direkt als Nebenbedingungen über  $\mathcal{X}_A \subseteq \mathbb{R}^N$  definiert werden.
- Es wird eine optimale Steuerfolge  $\mathbf{u}^*(t)$  für die zugrunde gelegten Zustandsdifferentialgleichungen  $\mathbf{f} : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R} \rightarrow \mathbb{R}^N$  ermittelt, worin das Übertragungsverhalten des Folgereglers berücksichtigt werden kann. Damit lassen sich Kriterien und Bedingungen an die Stellgröße mittels  $J : \mathbb{R}^N \times \mathbb{R}^M \times \mathbb{R} \rightarrow \mathbb{R}_0^+$  bzw.  $\mathcal{U}_A \subseteq \mathbb{R}^M$  realisieren.
- Die Dauer  $t_f$  muss nicht a priori bekannt sein und die Kriterien sowie Nebenbedingungen dürfen sich mit der Zeit verändern.

Da die gesamte Trajektorie vom Startzustand  $\mathbf{x}_0$  bis zum Zielzustand  $\mathbf{x}_f$  optimiert wird, erhält das Lösen des OSP zur besseren Unterscheidung zu späteren Methoden die Bezeichnung GTO<sup>8</sup>. Von Krämer u. a. [Krä+18b] wird auf diesem Weg eine zeitoptimale Trajektorie für den Ausfaltvorgang des Masts einer mobilen Autobetonpumpe geplant. Stryk u. a. [Str+94] planen Trajektorien eines Industrieroboters mit minimaler Zeit und Energie. Stuhlenmiller u. a. [Stu+21] demonstrieren die Flexibilität der Methode, indem sie den Kreis der Optimierung neben der Energie auf die zu erwartende Lebensdauer des Roboters erweitern. Auch wenn eine optimale Steuerfolge vorliegt und eine Steuerung möglich ist, führt diese aufgrund der Modellungenauigkeiten von  $\mathbf{f}(\cdot)$  zu akkumulierenden Abweichungen. Die Nutzung der optimalen Trajektorie  $\mathbf{x}^*(t)$  mit Folgeregler ist aufgrund der Rückkopplung für gewöhnlich robuster.

Zur Lösung des OSP haben sich die direkten Methoden etabliert, die zunächst eine Transkription mittels Einfachschießverfahren, Mehrfachschießverfahren oder Kollokation in ein NLP vollziehen und dieses dann lösen [Bet98]. Hierbei sind Frameworks wie HQP<sup>9</sup> [Fra98], CasADi<sup>10</sup> [And+18], GRAMPC<sup>11</sup> [Eng+19], ACADO<sup>12</sup> [Hou+11] und VIATOC [Kal+15] zu nennen. Da sie in erster Linie die Lösung vorhandener OSPs verfolgen und das Aufstellen eines auf die Trajektorienplanung angepassten OSP dem Anwender obliegt, werden sie nicht zu den Planungsansätzen gezählt. Sie treten in einigen der im nächsten Abschnitt aufgeführten Ansätze hintergründig in Erscheinung. Ansätze wie ALTRO<sup>13</sup> [T A+19] oder TrajOpt<sup>14</sup> [Sch+13] sind hingegen auf die Trajektorienplanung spezialisiert und werden als Planer berücksichtigt. Die genannten Verfahren nutzen lokale Gradienten zur Lösung des NLP, worauf in Kapitel 3 näher eingegangen wird. Neben den direkten Methoden existieren noch die indirekte Methode über das Maximumprinzip von Pontryagin u. a. [Pon+86] und die Dynamische Programmierung [Bel03].

<sup>8</sup>Globale Trajektorienoptimierung

<sup>9</sup>Huge Quadratic Programming

<sup>10</sup>Computer Algebra System for Algorithmic Differentiation

<sup>11</sup>Gradient-Based Nonlinear Model Predictive Control

<sup>12</sup>Automatic Control And Dynamic Optimization

<sup>13</sup>Augmented Lagrangian Trajectory Optimization

<sup>14</sup>Trajectory Optimization

## Methoden der wiederkehrenden Planung

Bei der wiederkehrenden Planung gilt es vornehmlich, den Aufwand soweit zu reduzieren, sodass ausreichende Wiederhol frequenzen erreicht werden. Die zuvor genannten Verfahren STOMP und CHOMP sind nicht auf niedrige Berechnungszeiten ausgelegt und erzielen entsprechend geringe Wiederhol frequenzen [Zha+18; Gaf+22]. Der Aufwand der Planung skaliert mit der Länge und Auflösung der Trajektorie, weshalb Steffens u. a. [Ste+16] eine adaptive Auflösung für STOMP vorschlagen. C. Park u. a. [Par+13b] beschleunigen STOMP durch die Parallelisierung auf einer GPU<sup>15</sup>. Im Folgenden werden spezialisierte Ansätze aus der Literatur vorgestellt und anhand der Art der Planung sowie der Strategie zur Aufwandsreduktion unterschieden.

Eine Möglichkeit den Aufwand der Planung zu reduzieren, ist die Modifikation einer im Vorfeld geplanten Trajektorie. Verbreitet ist dabei die Modifikation des Zeitprofils, das in der einfachsten Form angehalten wird [Cor+12]. Ansätze wie die von Flacco u. a. [Fla+12], Beckert u. a. [Bec+17] und Katzschmann u. a. [Kat+13] berücksichtigen zusätzlich Zustandsbeschränkungen und die Dynamik des Roboters. Das angepasste Zeitprofil der zugrundeliegenden Polynome wird darin über ein nichtlineares Gleichungssystem mit dem Ansatz von Kröger [Krö10] ermittelt. Daneben existieren Verfahren auf Basis der Lösung von Optimierungsproblemen [Lo+11; Rag+15; Wan+16] oder speziellen Suchalgorithmen [Lan+16; Xia+12]. Die Zeitskalierung erreicht Wiederhol frequenzen im hohen dreistelligen Bereich. Ansätze wie von Corrales u. a. [Cor+12], Beckert u. a. [Bec+17] und Ragaglia u. a. [Rag+15] bieten Sicherheitsgarantien bei der Kollisionsvermeidung. Allerdings sind sie durch das potentielle Anhalten des Roboters restriktiv und weniger effektiv wie Ansätze, die zum Preis geringerer Wiederhol frequenzen auch den räumlichen Teil der Trajektorie modifizieren. Häufige Fehlauflösungen des Sicherheitssystems, die zum Stillstand führen, reduzieren die Akzeptanz ebenso wie Kollisionen. Werden die Folgen einer Kollision abgeschätzt, relativiert sich der Vorteil von restriktiven Ansätzen. Spezielle Entwicklungskriterien und Kollisionsmodelle wie von S.-D. Lee u. a. [Lee+13] helfen dabei, die Folgen von Kollisionen zu verringern.

Abseits der Ansätze auf Basis der Zeitskalierung ist bei der wiederkehrenden Planung die strikte Trennung von Planung und Ausführung oftmals nicht sinnvoll, da der Wechsel zwischen Trajektorien im Folgeregler nicht trivial ist. Es stellt sich unter anderem die Frage, wie der Wechsel auf eine neue Trajektorie vollzogen wird, wenn diese nicht stetig aus der bisherigen hervorgeht [Llo+93; Rog+01]. Folgeregler sind oftmals weniger auf das Übergangsverhalten ausgelegt und je nach räumlicher Distanz ist dies die Aufgabe der Planung. Aus diesem Grund geben Ansätze der wiederkehrenden Planung häufig direkt die notwendige Stellgröße aus. Das Verschwimmen beider Aufgabenbereiche führt zu unscharfen Verwendungen der Begriffe Planung und Regelung in der Literatur. Deshalb werden an dieser Stelle nur solche Ansätze als Planer bezeichnet, die neben dem Soll- und Istzustand sowie der Stellgröße auch die Folgen dieser Stellgröße (z. B. Kollisionen) bewerten. Dies kann z. B. bereits über die Berechnung von mindestens einem Folgezustand realisiert werden.

Die Modifikation einer bestehenden Trajektorie in Zeit und Raum, wie z. B. von Brock u. a. [Bro+02], erfordert mehr Aufwand. Begegnet wird diesem mit der APF<sup>16</sup>-Methode, in dem der Roboter künstlichen Potentialen in Form von anziehenden Kräften von Zielposen und abstoßenden Kräften von Hindernissen ausgesetzt wird [Kha85]. Die Ableitung im Potentialfeld ergibt einen Richtungsvektor, der von Hindernissen und Startpose in Richtung Zielpose zeigt und als Kraft oder Sollgeschwindigkeit fungiert, womit sich der Suchraum von einer vollständigen Tra-

---

<sup>15</sup>Graphics Processing Unit

<sup>16</sup>Artificial Potential Field

jektorie auf einen Richtungsvektor reduziert. Auf der APF-Methode basieren weitere Ansätze wie die von Kulić u. a. [Kul+06], Ataka u. a. [Ata+16], Byrne u. a. [Byr+12] und Flacco u. a. [Fla+12]. Andere Möglichkeiten den Suchraum zu verkleinern sind die Einschränkung eines PRM auf eine Region in der Nähe eines zuvor geplanten Pfades [Li+19], die Einschränkung auf Positionen [Wan+18; Mar+23], Ebenen [Wan+17; Byr+12] oder Geschwindigkeitsvektoren [Bos+11] im Arbeitsraum sowie die Vorgabe einer endlichen Lösungsmenge [Bal+06]. Eng verwandt mit der Reduzierung des Suchraums ist die Reduzierung von Freiheitsgraden des Roboters auf zwei [Bie+18; Kan+19; Cas+09] oder drei [Byr+12].

Die genannten Vorteile der direkten Optimierung von Trajektorien spiegeln sich in der Anzahl von Ansätzen wider, die auf diese Weise Trajektorien planen. Um den Aufwand zu reduzieren, linearisieren Buizza Avanzini u. a. [Bui+18], M. Wang u. a. [Wan+16] und Jardine u. a. [Jar+17] die Nebenbedingungen des Planungsproblems und erhalten ein quadratisches Optimierungsproblem, während Schulman u. a. [Sch+13] eine konvexe Approximation aufstellen. Alternativ nutzen Magyar u. a. [Mag+19] und Fishman u. a. [Fis+20] anstelle harter Nebenbedingungen quadratische Strafterme (weiche Nebenbedingungen), welche die Nutzung einfacherer Lösungsverfahren für die nichtlineare Optimierung im Sinne der kleinsten Quadrate erlaubt. Auch lernbasierte Ansätze, wie der von Sangiovanni u. a. [San+18] über tiefes bestärkendes Lernen oder *RoboCat* [Bou+23], zählen zu den Verfahren mit weichen Nebenbedingungen und verlagern den Aufwand in die Lernphase.

Durch die breite Verfügbarkeit von Mehrkernprozessoren und GPUs ist auch Parallelisierung eine Methode, um die Wiederholfrequenz zu steigern. Beispielsweise zerlegen C. Wang u. a. [Wan+21] die Trajektorie in mehrere Teile und lösen jedes Teilproblem über die verteilte Optimierung mit dem ADMM<sup>17</sup>-Ansatz.

Bei der wiederkehrenden Planung aufgrund dynamischer Hindernisse wie Personen, ist deren Bewegung in der Regel a priori nicht bekannt und besitzt eine steigende Unsicherheit, wodurch das Ende einer vollständigen Trajektorie tendenziell unbrauchbar wird. Petti u. a. [Pet+05] motivieren vor diesem Hintergrund mit der Planung von lokal begrenzten Trajektorien das PMP<sup>18</sup>, das gleichzeitig den Berechnungsaufwand reduziert und eine weitere Variante zur Reduzierung des Suchraums darstellt. Das PMP beschreibt das Planen einer zukünftigen Teiltrajektorie innerhalb eines festen Zeitbudgets, während dem vorherige Teilstück per Folgeregler gefolgt wird. Auf diese Weise ist bekannt, wo sich der Roboter zu Beginn des nächsten Teilstücks befindet, was einen stetigen Übergang ermöglicht. Ein Nachteil ist, dass die Annahme über Veränderungen in der Umgebung während der Planung eines Teilstücks effektiv für die Dauer zweier Teilstücke gültig bleiben muss. Dies führt zu einer Halbierung der effektiven Wiederholfrequenz und erfordert entweder kleinere Teilstücke, die das Zeitbudget verkürzen oder Vorhersagen zur Umgebung, deren Genauigkeit allerdings mit der Länge abnehmen. C. Park u. a. [Par+13a] übertragen die Idee in Form des Ansatzes ITOMP<sup>19</sup> auf die Planung von Robotertrajektorien und ermöglichen langsameren Planern wie STOMP zumindest eine eingeschränkt wiederkehrende Nutzung.

### Kollisionsvermeidung bei wiederkehrender Planung

Während die wiederholte Planung häufig bereits die einzige nennenswerte Voraussetzung für variable nichtdeterministische Bewegungsziele und konfliktäre Planungskriterien ist, bildet sie für eine Kollisionsvermeidung in dynamischen Umgebungen nur die Grundlage. Entsprechend

---

<sup>17</sup>*Alternating Direction Method of Multipliers*

<sup>18</sup>*Partial Motion Planning*

<sup>19</sup>*Incremental Trajectory Optimization*

viele Möglichkeiten sie zu realisieren und zu unterscheiden sind in der Literatur anzutreffen, von denen im Folgenden einige vorgestellt werden.

Um die Umgebung innerhalb der Planung zu berücksichtigen, ist ein geeignetes Abbild von dieser erforderlich. Eine bewährte Methode ist die Dekomposition der Umgebung in Teile, die sich räumlich effizient über Hüllkörper repräsentieren lassen. Dabei hat die Wahl der Hüllkörper einen entscheidenden Einfluss auf den Berechnungsaufwand während der Planung. Bei distanzbasierter Kollisionsvermeidung ist der wohl beliebteste Hüllkörper die Kugel [Fis+20; Zub15; Li+19; Wan+17; Ata+16; Par+13a; Pav+17; Zuc+13]. Sie besitzt eine triviale und stetige Distanzfunktion. Die Modellierung von länglichen Objekten erfordert hingegen die Verwendung mehrerer Kugeln, was den Aufwand erhöht und daher stattdessen z. B. Ellipsoiden eingesetzt werden [Zha+17; Gaf+22]. Mit unendlich ausgedehnte Ebenen lassen sich Areale großräumig und effizient absperren [Jar+17]. Werden endliche Ebenen betrachtet und in konvexer Weise um ein Hindernis modelliert, ergeben sich konvexe Polyeder, die jedoch den höchsten Berechnungsaufwand bedeuten [Lan+12; Rag+15; Zha+17; Wan+21; Wan+16; Sch+13; Krä+18b]. Sie erlauben hingegen die Anpassung an eine Vielzahl von Objekten, weshalb häufig bereits wenige ausreichen, um eine Szene zu modellieren. Einen Spezialfall bilden die AABBs<sup>20</sup>, die durch ihre Ausrichtung effizienter sind als OBBs<sup>21</sup> [Mag+19; Cor+12]. Aufgrund ihrer einfachen Distanzfunktion und flexiblen Modellierungsmöglichkeiten sind SSVs<sup>22</sup> nach Larsen u. a. [Lar+99] in der wiederkehrenden Planung beliebt [Min+20; Wan+18; Cas+09; Bec+17; Par+19; Bos+11; Cor+12; Krä+20b]. Ihre Formen basieren auf Kugeln, die entlang verschiedener zweidimensionaler Primitive wie Linien- oder Flächensegmente geführt werden. Die resultierenden Formen eignen sich ebenfalls zur Modellierung der Gliedmaßen von Personen [Bec+17; Cor+12; Krä+20b].

Da die Modellierung der Umgebung über Hüllkörper je nach Komplexität der Szene aufwändig ist und insbesondere bei dynamischen Szenen potentiell häufiger durchgeführt werden muss, existieren auch modellfreie Ansätze. Sie erfassen nahende Hindernisse über Sensoren am Roboter oder in der Umgebung und ermöglichen so auf bevorstehende Kollisionen zu reagieren. Modellfreie Ansätze umgehen externe Sensorlösungen, die häufig einen nennenswerten konstruktiven und wirtschaftlichen Aufwand bedeuten. Beispiele roboterzentrischer Sensoren sind Tiefensensoren [Luc+12; Fla+12; Bui+18], Farbkameras [Gec+05; Kat+13], Infrarotsensoren [Byr+12] oder kapazitive/taktile Sensoren [Hof+16; Sto+13].

Um den Berechnungsaufwand bei vielen Hüllkörpern zu reduzieren, betrachten Flacco u. a. [Fla+12] und Minelli u. a. [Min+20] nur den Endeffektor. Auch in anderen Verfahren werden zur Beschleunigung nur einzelne Punkte am Roboter betrachtet [Zub15; Bui+18; Ata+16; Kan+19]. Gleiches gilt für Personen, bei denen oftmals nur Hand [Fis+20; Li+19; Wan+17], Arm [Wan+18] oder Oberkörper [Bec+17] Teil der Modellierung sind.

Traditionell erfolgt die Distanzberechnung für statische Szenen über die globale Bereitstellung der Abstände über sogenannte Distanzfelder [Rat+09; Coh+10; Zuc+13; Pav+17; Fis+20]. Sie unterteilen die Umgebung in Bereiche, in denen der jeweils kürzeste Abstand zu einem Hindernis gespeichert wird. Darüber lassen sich fortan die Abstände zwischen Roboter und nächstgelegtem Hindernis über die vom Roboter belegten Bereiche abrufen und bei der Planung berücksichtigen. Während das Abrufen schnell erfolgt, ist die Aktualisierung des Distanzfeldes in der Regel zu aufwändig für eine wiederkehrende Planung. Adaptive Methoden wie von Leven u. a. [Lev+02] und Pomarlan u. a. [Pom+13] dämmen diesen Aufwand etwas ein. Im Hinblick auf

---

<sup>20</sup>Axis-Aligned Bounding Box

<sup>21</sup>Oriented Bounding Box

<sup>22</sup>Swept-Sphere Volume

dynamische Hindernisse ist der Weg über lokale Distanzberechnungen während der Planung meist vielversprechender [Zub15; Wan+21; Min+20; San+18; Wan+18; Wan+17; Ata+16; Krä+20b].

Die Bandbreite zur Berechnung von Distanzen reicht in der Literatur von einzelnen Distanzfunktionen für bestimmte Hüllkörper über Sammlungen von Distanzfunktionen für eine Vielzahl von Hüllkörpern bis hin zu Ansätzen mit hierarchischen Distanzwarteschlangen. Bei den einzelnen Distanzfunktionen ist der GJK<sup>23</sup>-Algorithmus [Gil+88] zur Berechnung der Separationsdistanz zwischen konvexen Polyedern verbreitet, jedoch häufig zu langsam. Der EP<sup>24</sup>-Algorithmus [Gin01] erweitert den GJK-Algorithmus auf die Penetrationsdistanz. Eine Sammlung von speziellen Distanzfunktionen für verschiedene Arten von Hüllkörpern ist GTE<sup>25</sup> [Ebe20]. Sie bietet in erster Linie den Vorteil einer einheitlichen Schnittstelle für eine Vielzahl von Hüllkörpern und wählt die passende Distanzfunktion je nach Hüllkörperpaar automatisch. Frameworks wie FCL<sup>26</sup> [Pan+12] unterstützen darüber hinaus hierarchische Repräsentationen mit unterschiedlichen Detailstufen wie in Larsen u. a. [Lar+99] und Warteschlangen, um den Aufwand vieler Berechnungen zu reduzieren. Für SSVs berechnet sich die Separationsdistanz effizient über primitivbasierte Distanzfunktionen wie dem *Primitive Shape Algorithm* von Lumelsky [Lum85] und Larsen u. a. [Lar+99]. Sie sind nach Larsen u. a. [Lar+99] dabei viermal schneller als der GJK-Algorithmus. Bei einer Vielzahl von SSVs bieten sich Distanzfunktionen auf Basis von Zerlegungsverfahren an [Saf+19]. Die Kehrseite der distanzbasierten Kollisionsvermeidung sind Funktionen für die Penetrationsdistanz, die je nach Hüllkörper oft seltener oder deutlich komplexer sind.

Bei optimierungsbasierter Planung gibt es zwei Möglichkeiten die Kollisionsvermeidung zu realisieren. Die Formulierung entsprechender Planungskriterien ist die erste Möglichkeit, die in Form von Potentialen auch bei der APF-Methode zum Einsatz kommt [Ata+16; Fla+12; Bro+02; Byr+12; Moh+95; Kim+92]. Die Planungskriterien reichen von abstands-basierten Straftermen [Rat+09; Zuc+13; Pav+17; Zub15; Bal+06; San+18; Par+13a; Krä+20b] über direkte Abstandsmaximierung [Coh+10; Fis+20] bis hin zur distanzfreien Minimierung von Schnittvolumina [Mag+19]. Hierzu zählen auch Mischformen wie der Gefahrenindex von Kulić u. a. [Kul+05], der neben dem Abstand auch effektive Masse und Geschwindigkeiten verrechnet. Skalare Planungskriterien besitzen den Vorteil, dass sie das NLP nicht vergrößern und Lösungsverfahren erlauben, die nur weiche Nebenbedingungen unterstützen und oftmals schneller sind als solche, die harte Nebenbedingungen unterstützen [Mag+19; Fis+20]. Ein Nachteil ist allerdings der Kompromiss zwischen anderen Planungskriterien, womit Kollisionen in den Lösungen nicht ausgeschlossen werden können. Dies lässt sich mit der Formulierung von harten Nebenbedingungen als zweiten Möglichkeit zur Kollisionsvermeidung beheben. Bei der Verwendung von Hüllkörpern in Form von Ebenen oder konvexen Polyedern stellen Landry u. a. [Lan+12] die Nebenbedingung zur Separation über das Lemma von Farkas [Far02] auf und kommen ohne explizite Berechnung des Abstands aus. Eine verwandte Methode zur Umgehung einer expliziten Abstandsberechnung nutzt die Ähnlichkeit der Abstandsdefinition mit der Lösung eines gesonderten Optimierungsproblems. Für sphärische Volumenkörper [Cas+09], Ebenen [Zha+17] oder Ellipsoiden [Gaf+22] wird darin die Abstandsberechnung inklusive der Forderung eines Mindestabstands in das NLP der Planung integriert. Der Nachteil all dieser Varianten ist die Vergrößerung des NLP durch die Einführung von Hilfsvariablen und die einhergehende Erhöhung des Aufwands. Daneben stehen

---

<sup>23</sup>Gilbert-Johnson-Keerthi

<sup>24</sup>*Expanding Polytopes*

<sup>25</sup>*Geometric Tools Engine*

<sup>26</sup>*Flexible Collision Library*

Methoden, die den Abstand oder eine verwandte Größe explizit berücksichtigen und das NLP nur moderat vergrößern. Genutzt werden die Annäherungsgeschwindigkeit [Bos+11; Wan+16], der Bremsweg [Bui+18; Bec+17] oder der Abstand [Wan+21; Min+20; Wan+18; Wan+17; Zub15; Sch+13; Krä+20b].

Während in statischen Umgebungen nur die Zukunft des Roboters relevant ist, kommt bei dynamischen Umgebungen jene von beweglichen Hindernissen hinzu. Neben der grundsätzlichen Art eine Bewegung zu präzisieren, unterscheiden sich Ansätze auch in der Modellierung der Prädiktionsunsicherheit. Die einfachste Art nimmt die Umgebung als statischen Schnappschuss an, der sich nur zwischen den Planungszyklen verändert und verzichtet damit auf eine Prädiktion [Zub15; Pav+17; Bui+18; Bie+18; Bro+02; Krä+20b]. Sie ist praktisch mit jedem Ansatz, der statische Hindernisse unterstützt, nutzbar und wird als Schnappschussmethode bezeichnet. Ko u. a. [Ko+93] haben diese Methode dahingehend erweitert, dass die Unsicherheit durch übergeordnete statische Hüllkörper approximiert wird, in denen sich die dynamischen Hindernisse für die Dauer des Planungshorizonts bewegen. Diese Idee ist über eine Abschätzung der möglichen Bewegung von Personen auf Szenarien der MRK erweiterbar [Per+17; Rag+15; Par+13a]. Casalino u. a. [Cas+19] modellieren die Unsicherheit über Wahrscheinlichkeiten und bestimmen den übergeordneten Hüllkörper über einen Bereich mit kritischer Kollisionswahrscheinlichkeit. Eine weitere Möglichkeit stellt die direkte Minimierung der Kollisionswahrscheinlichkeit dar. Dies gestaltet sich formal ähnlich wie abstands-basierte Strafterme bei präzisierter Hindernispose [And+16; Par+19; Kan+19].

Andere Methoden präzisieren die Bewegung des Hindernisses, um sie mit der des Roboters bei der Kollisionsvermeidung aktiv zu berücksichtigen [Li+19; Pet+05]. Hierfür optimieren Rösmann u. a. [Rös+17a] und Fishman u. a. [Fis+20] Multiagentensysteme, in denen die Bewegungsabsichten und -eigenschaften jedes Agenten gesondert modelliert werden. Andere Ansätze verfolgen die Minimierung von Kenngrößen wie die Krümmung [Oya+04] oder Drehmomentvariation [Zan+17] des menschlichen Skeletts. Durch das Aufleben tiefer neuronaler Netze sind auch die datenbasierten Methoden zur Bewegungsprädiktion immer beliebter [Xue+17; Wan+17; Her+19]. Insbesondere die explizite Berücksichtigung von kinematischen Zusammenhängen und übergeordnetem Kontextwissen wie Aktionserkennung verbessern die Prädiktionsgüte bei menschlichen Bewegungen [Aks+19; Las+17]. Der Bedarf an Trainingsdaten, der Trainingsaufwand und die Wiederholffrequenz im einstelligen Bereich stehen einem großflächigen Einsatz bei der wiederkehrenden Planung jedoch aktuell noch im Weg [Ren+23a]. Bei repetitiven Aufgaben bewährt sich vor diesem Hintergrund der Einsatz weniger komplexer Modelle wie GMM<sup>27</sup> [Per+15; Kan+19; Las+17], ProDMP<sup>28</sup> [Luo+19] oder ARIMA<sup>29</sup> [Wan+18].

Eine Form der Bewegungsprädiktion, die kein vorheriges Training benötigt und parallel zur Planung arbeitet, ist die Kombination eines Zustandsschätzers mit einer Extrapolation [Min+20; San+18; Las+17; Ren+23a]. Als Zustandsschätzer kommen klassische Beobachter [Wei+18], Kalman-Filter [Cor+12; Ata+16], gleitende Mittelwerte [Bal+06] oder Savitzky-Golay-Filter [Las+17] zur Anwendung. Ferner verbessern sich damit Ansätze wie jene, die auf der Annahme einer konstanten Hindernisgeschwindigkeit basieren [Fla+12; Bos+11; Kul+06].

---

<sup>27</sup>Gaussian Mixture Model

<sup>28</sup>Probabilistic Dynamic Motion Primitives

<sup>29</sup>Auto-Regressive Integrated Moving Average

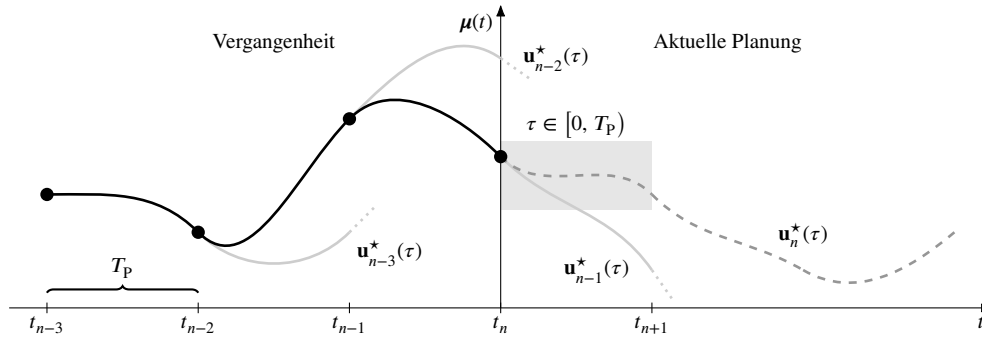


Abbildung 2.1.: Zeitlicher Ablauf bei der wiederkehrenden Planung. Die optimierte Steuerfolge  $\mathbf{u}_n^*(\tau)$  wird nur für die Dauer  $T_p$  eines Planungszyklus verwendet, ehe sie durch die aus dem darauffolgenden Zyklus ersetzt wird.

## 2.2. Wiederkehrende Planung mit fortlaufendem Horizont

Die Anwendung des PMP auf die Lösung eines OSP zu äquidistanten Zeitpunkten  $t_n = nT_p$  mit  $n = 0, 1, \dots$  und  $T_p \in \mathbb{R}^+$  führt in Richtung MHP. Der Unterschied zur PMP beruht auf einer nochmals höheren Wiederholfrequenz, wodurch die Länge der effektiv zu folgenden Teilstücke reduziert und der Folgeregler ersetzt werden. Das Stellgesetz lautet demnach:

$$\mu(t) := \mathbf{u}_n^*(\tau) \text{ mit } \tau \in [0, T_p) \text{ und } t \in [t_n, t_{n+1}). \quad (2.6)$$

Das Subskript  $n$  signalisiert die Zugehörigkeit zum  $n$ -ten Planungszyklus. Da das OSP wiederkehrend gelöst wird, bezeichnet zur besseren Unterscheidung mit der realen Zeit  $t$  von jetzt an  $\tau$  die Zeit der Größen innerhalb der Optimierung. Abbildung 2.1 verdeutlicht den zeitlichen Ablauf der MHP. Aus regelungstechnischer Sicht entspricht dieses Prinzip einem MPC<sup>30</sup> [Grü+17; Die+06]. Im Vergleich zum OSP der GTO in Gleichung (2.5), agiert die MHP lokal, weshalb die Nebenbedingung (2.5c) entfernt wird und  $t_f$  konstant ist. Die Annäherung an den Zielzustand  $\mathbf{x}_f$  obliegt damit einer geeigneten Kostenfunktion  $J(\cdot)$ . Der Anfangszustand  $\mathbf{x}_0$  wird in jedem Planungszyklus durch Messungen auf den Istwert aktualisiert. Die Länge der bei der Planung betrachteten Teilstücke, im Folgenden (Planungs-)Horizont genannt, wird in einem Kompromiss aus Aufwand und Anforderungen problemspezifisch variiert.

Zur Lösung sind grundsätzlich alle Verfahren anwendbar, die auch das globale OSP sowie Teiltrajektorien unterstützen und dabei effizient genug für den wiederkehrenden Betrieb sind. Insbesondere die zuvor erwähnten Frameworks HQP, CasADi, ACADO, VIATOC und GRAMPC sind speziell für MPCs ausgelegt. Im Gegensatz zu den Trajektorienplanern ALTRO und TajOpt fokussieren sie sich verstärkt auf die Lösung komplexerer Dynamiken  $\mathbf{f}(\cdot)$  der Regelstrecke und weniger auf Nebenbedingungen wie z. B. die Kollisionsvermeidung. Mit den Fortschritten bei der Lösung von NLPs erfährt die MHP zunehmende Beliebtheit. Sie greift die eingangs genannten Vorteile der optimierungsbasierten Planung auf und kombiniert sie mit einer skalierbaren Wiederholfrequenz, womit sie für den Anwendungsbereich in dynamischen Umgebungen mit zum Teil variierenden und konfliktären Zielen geeignet ist. Im Folgenden werden einige Varianten aus der Literatur vorgestellt und die vorliegende Arbeit davon abgegrenzt.

Bezüglich der Transkription von OSP zu NLP sind mit dem Einfeldschießverfahren [Bui+18; T A+19], dem Mehrfeldschießverfahren [Die+06] und der Kollokation [Gaf+22; Fis+20; Min+20; Kan+19; Zub15] alle in dieser Arbeit betrachteten Möglichkeiten vertreten. Letztere wird dabei

<sup>30</sup>Model Predictive Controller

meist implizit gewählt, wenn kein vorheriges OSP aufgestellt wird, da die vollständige Diskretisierung von Trajektorie und Steuerfolge generell eine plausible Vorgehensweise ist. Die Ausnutzung von Strukturen im NLP, die auf dieser Transkription beruhen, wird allerdings nur von Gafur u. a. [Gaf+22] behandelt. Dies birgt angesichts der höchsten Größenordnung des resultierenden NLP Verbesserungspotential, das in der vorliegenden Arbeit verstärkt genutzt wird. Unter der Größenordnung wird die Dimension des Optimierungsvektors sowie die Anzahl der Nebenbedingungen zusammengefasst. Das Einzelschießverfahren liefert die kleinste Größenordnung, jedoch keine ausnutzbaren Strukturen und die höchste Sensitivität. Als Sensitivität wird die Stärke der Veränderung in Kostenfunktion und Nebenbedingungen bezeichnet, die durch Variation der Optimierungsvariablen entsteht. Eine hohe Sensitivität beeinflusst zusammen mit der Größenordnung die Effektivität von Lösungsverfahren negativ. Das Mehrschießverfahren erlaubt eine Abwägung zwischen beiden Extremen und ist vor allem bei MPCs beliebt [San+01; Ter+04]. Die vorliegende Arbeit grenzt sich insofern von den genannten Verfahren ab, dass eine neue Kombination aus Transkription, Strukturausnutzung und Lösungsverfahren vorgestellt und mit HQP sowie ALTRO als Stellvertreter von Ansätzen aus der optimalen Regelung und Trajektorienoptimierung verglichen wird. Eine Möglichkeit zur Strukturausnutzung ist die Angabe vollständiger analytischer Ableitungen, was jedoch häufig nicht möglich ist. Eine alternative Methode ist die automatische Differentiation (z. B. CasADi), die eine Funktion so weit in Unterfunktionen zerlegt, bis analytische Ableitungen oder strukturelle Konstanten ersichtlich werden. Diese Methode ist bei statischen NLPs aufgrund ihrer Schnelligkeit oft zu bevorzugen, eignet sich hingegen weniger für dynamische [Rös+18]. Zur Strukturausnutzung wird in der vorliegenden Arbeit daher der Hypergraph von Rösmann u. a. [Rös+18] eingesetzt, der seine Stärken in der Unterstützung von variablen Größenordnungen ausspielt. Variable Größenordnungen entstehen z. B. durch dynamische Hindernisse, die den Arbeitsraum betreten oder verlassen.

Bei der Modellierung mit Hüllkörpern verfolgen Gafur u. a. [Gaf+22], Fishman u. a. [Fis+20] und Balan u. a. [Bal+06] eine vollständige Abdeckung des Roboters, wovon die ersten beiden jedoch Vereinfachungen, wie einen Ellipsoiden oder eine Kugel, bei der Modellierung von weiteren Akteuren verwenden. Balan u. a. [Bal+06] modellieren präziser, verwenden jedoch ausschließlich Kugeln (46 für einen Menschen und 41 für einen Roboter) und skalieren ungünstig in strukturierten Umgebungen. Die vorliegende Arbeit verfolgt eine vollständige Modellierung von Roboter, Mensch und Objekten des Arbeitsraums mittels SSVs sowie unendlichen Ebenen und ermöglicht damit präzise und flexible Abbildungen der Umgebung bei gleichzeitig hoher Effizienz. Die Realisierung der Kollisionsvermeidung auf der Ebene des NLP erfolgt bei Fishman u. a. [Fis+20] und Kanazawa u. a. [Kan+19] ausschließlich über Kostenfunktionen und birgt die Gefahr von Kompromisslösungen mit Kollisionen. Die Verwendung von harten Nebenbedingungen ist zwar eine systematische Methode dies zu verhindern, eignet sich jedoch als ausschließliches Mittel ebenso nur bedingt, da Lösungen bis an den Rand einer Kollision optimiert werden. Angesichts der Ungenauigkeiten der Umgebungserfassung führt dies zu potentiellen Überschreitungen und extremen Reaktionen im Lösungsverfahren [Gaf+22; Min+20; Bui+18; Wan+18; Wan+17; Bal+06]. Für die MHP wird in der vorliegenden Arbeit daher die Kombination beider Maßnahmen verfolgt, wie auch von Zube [Zub15]. Bei der Behandlung von dynamischen Hindernissen verwenden Buizza Avanzini u. a. [Bui+18], Zube [Zub15] und Gafur u. a. [Gaf+22] keine Bewegungsprädiktion. Fishman u. a. [Fis+20] und Y. Wang u. a. [Wan+17] hingegen nutzen ein Multiagentensystem sowie ein LSTM<sup>31</sup>-Netz und betreiben damit einen erhöhten Aufwand. Auch Kanazawa u. a. [Kan+19] und Y. Wang u. a. [Wan+18] benötigen für die Erstellung des GMM eine Trainingsphase bzw. eine laufende Parameteridentifikation für den

---

<sup>31</sup>Long Short-Term Memory

autoregressiven Einschrittprädiktor. Während diese Methoden zu besseren Prädiktionen führen, verwendet die vorliegende Arbeit mit der Extrapolation des geschätzten Bewegungszustands einen einfacheren aber dennoch effektiven Ansatz [Min+20; Bal+06].

Die vorliegende Arbeit sieht die Kollisionsvermeidung als Beitrag zu einer unterbrechungsfreien Koexistenz zwischen Mensch und Roboter im geteilten Arbeitsraum und verfolgt nicht das Ziel von Sicherheitsgarantien. Daher werden industrielle Sicherheitsstandards nicht thematisiert und die konstruktive Begrenzung von Kraft und Geschwindigkeit durch kollaborative Roboter als ausreichend angesehen. Planungsverfahren, die industrielle Standards berücksichtigen, sind unter anderem die von Ragaglia u. a. [Rag+15] und Beckert u. a. [Bec+17].

In Bezug auf dynamische Planungskriterien nutzt keiner der genannten Ansätze die Möglichkeiten der MHP. Eine Methode um sekundäre Kriterien umzusetzen führt über den Nullraum des Roboters. Dieser beschreibt die Menge an Gelenkwinkelgeschwindigkeiten, die sich nicht auf den Endeffektor auswirken. Über die Projektion einer gewünschten Sekundärbewegung in den Nullraum, lässt sich eine anteilige Bewegung berechnen, welche die Primärbewegung des Endeffektors nicht beeinflusst. Sekundärbewegungen ergeben sich z. B. aus sekundären Kriterien wie Kollisionsvermeidung [Li+19; Kiv+17], Abstand von Gelenkgrenzen [Bro+02] oder Nähe zu Singularitäten [Sic+09]. Der Nachteil der Projektion ist jedoch, dass der Nullraum keine Garantie über das Ausmaß der Erfüllung der sekundären Bewegung gibt. Im schlimmsten Fall sind die Sekundärbewegung und der Nullraum disjunkt, womit nur die Primärbewegung vollzogen wird. Pavlichenko u. a. [Pav+17] erweitern STOMP auf mehrere normierte Kriterien, die allerdings eine händische Priorisierung vor der Planung erfordern. Insbesondere in der Trajektorienplanung ist häufig nur der Zielzustand vorgegeben und die genaue Bewegung wird zugunsten weiterer Ziele, wie z. B. der Kollisionsvermeidung, frei gewählt. Diese temporäre Abwägung von Zielen ist über den Nullraum nur bedingt möglich, weshalb die vorliegende Arbeit die Vorteile der wiederkehrenden Optimierung nutzt und für selbige zwei Varianten vorstellt, zeitlich und räumlich konkurrierende Kostenterme in einer Kostenfunktion zu vereinen.

Gradientenbasierte Verfahren zur Lösung eines nicht-konvexen NLP sind abhängig von einer geeigneten Initialisierung. In der mobilen Robotik wird zur Exploration des Suchraums auf die parallele Planung in verschiedenen Homotopieklassen zurückgegriffen, sodass möglichst viele topologisch unterschiedliche Pfade berücksichtigt werden [Rös+17a; Rös+17b]. Der Aufwand zur Betrachtung aller Homotopieklassen skaliert allerdings schlecht mit der Komplexität der Umgebung und ist daher primär für einen zweidimensionalen Arbeitsraum eines mobilen Roboters geeignet. Darüber hinaus ist die Idee der Homotopieklassen nicht ohne Weiteres sinnvoll auf die einseitig offene kinematische Kette von stationären Robotern übertragbar. Es erfordert z. B. eine allgemein aufwändige Abbildung der Hindernisse in den Gelenkraum samt dortiger Definition von Topologien, wie es Kuntz u. a. [Kun+16] unter der Annahme konvexer Hindernisregionen für eine nicht wiederkehrende Planung umsetzen. Brock u. a. [Bro+02] definieren Homotopieklassen über die Menge aller Gelenkwinkeltrajektorien, die im Arbeitsraum innerhalb eines vom Roboter überstrichenen und anschließend systematisch vergrößerten Volumens liegen. Hierfür ist es allerdings erforderlich, Pfadkandidaten im Vorfeld zu bestimmen. Von den genannten Verfahren der MHP gehen nur Balan u. a. [Bal+06] indirekt auf die Initialisierung ein, indem die Lösungsmenge auf eine endliche Zahl von Kandidaten eingeschränkt und daraus in jedem Schritt die Beste ermittelt wird. Die vorliegende Arbeit greift die Idee der Nutzung mehrerer Initialisierungen auf und bettet sie in eine für den Warmstart geeignete parallele Abfolge ein. Darüber hinaus wird die Effektivität der Initialisierungen hinsichtlich der Exploration des Suchraums durch Ausnutzung älterer Informationen gesteigert.

Während die Planung von Teiltrajektorien der MHP ihren Laufzeitvorteil verleiht, gilt es zu

beachten, dass dabei ein lokales Planungsproblem gelöst wird und nicht das Globale aus Gleichung (2.5). Damit wird die MHP, wie auch die APF-Methode, zu einer lokalen Planung und ist anfällig für strukturelle lokale Minima [Kim+92; Bro+02; Byr+12]. Lösungen dieser Art veranlassen den Roboter vor Hindernissen stehenzubleiben, wenn innerhalb des Planungshorizonts jeder<sup>32</sup> Versuch das Hindernis zu umfahren zu höheren Kosten führt. Um die lokale Planung mit globalen Informationen anzureichern, werden einzelne Wegpunkte eines globalen Pfades nacheinander abgefahren [Imr+16; Mac+03] oder in Gänze verfolgt [Zub15; Fau+17]. Nachteilig ist, dass qualitativ hochwertige Pfade erforderlich sowie aufwändig sind und die Flexibilität beim Ausweichen vor dynamischen Hindernissen einschränkt. Von den genannten Methoden der MHP adressieren Gafur u. a. [Gaf+22], Minelli u. a. [Min+20] und Buizza Avanzini u. a. [Bui+18] lokale Minima und verwenden Zwischenpunkte. Da diese jedoch reaktiv erzeugt werden, ist zunächst das Eintreten eines lokalen Minimums erforderlich. Die vorliegende Arbeit begegnet der Herausforderung der lokalen Planung mit einer globalen Zielführung, welche den Zielzustand temporär aber laufend in die Nähe des Roboters verlegt und dabei das Wissen über die gesamte Szene nutzt. Sie erweitert damit die Idee von Wegpunkten wie von Imran u. a. [Imr+16] und kombiniert sie mit einer proaktiven und globalen Komponente, die gleichzeitig das Ausweichen vor dynamischen Hindernissen begünstigt.

---

<sup>32</sup>In Abhängigkeit der Initialisierung kann dieser Fall auch bereits eintreten, obwohl es noch mindestens eine gültige Lösung zum Umfahren des Hindernisses gibt.

# 3

## Die allgemeine Planung mit fortlaufendem Horizont ohne Kollisionsvermeidung

Dieses Kapitel bildet die Grundlage für den ersten Teil der Arbeit und stellt das formale Planungsproblem der MHP ohne Kollisionsvermeidung in Form von drei unterschiedlich transkribierten NLPs auf. Hierfür werden in Abschnitt 3.1 zunächst Zustand und Stellgröße sowie weitere Komponenten des OSP wie Systembeschreibung, Kostenfunktion und einfache Beschränkungen aufgabenorientiert definiert. Die Transkriptionen des OSP in verschiedene NLPs erfolgt in Abschnitt 3.2 über das Einzelschießverfahren, Mehrfachschießverfahren und die vollständige Diskretisierung zusammen mit der Angabe der resultierenden Größenordnungen. Abschnitt 3.3 stellt die Vorgehensweise zweier bekannter Lösungsverfahren für NLPs vor, mit dem Fokus auf die Auswirkungen der Größenordnung auf die unterlagerten Gleichungssysteme und schafft damit die Voraussetzungen zur Auswertung einer effizienten Vorgehensweise zur Lösung des Planungsproblems in Kapitel 5.

### 3.1. Definition des Planungsproblems

Die zu planenden Zustände  $\mathbf{x}$  einer Trajektorie richten sich nach den Vorgaben der übergeordneten Aufgabe. Zentrale Größen sind die Gelenkwinkel, durch welche Nebenbedingungen für Gelenkgrenzen oder die Kollisionsvermeidung realisiert werden. Auch die notwendige Bewertung des Zielzustands über eine Kostenfunktion ist damit möglich.

**Definition 3.1.1:** *Der Zustandsvektor  $\mathbf{x} \in \mathbb{R}^N$  entspricht den  $N$  Gelenkwinkeln des Roboters.*

Die MHP bietet die Möglichkeit, Planung und Folgeregler zu vereinen und direkt eine zur Trajektorie führende Stellgröße auszugeben. Von entscheidender Bedeutung ist dabei den Zusammenhang zwischen  $\mathbf{x}(\tau)$  und  $\mathbf{u}(\tau)$  in Form einer Systemdynamik  $\mathbf{f}(\cdot)$  zu modellieren. Die Rückkopplung der MHP ist dabei in der Lage, Modellungenauigkeiten und Störungen zu kompensieren. Direkten Einfluss auf die Modellierung von  $\mathbf{f}(\cdot)$  hat die Wahl der Stellgröße  $\mathbf{u}$ . Sie ist abhängig von den Möglichkeiten mit dem Robotersystem zu kommunizieren und von der Komplexität dessen Dynamik. So gibt es Fälle, in denen der Roboter über Drehmomente an den Gelenken angesteuert wird [Die+06]. Damit sind zwar Kostenfunktionen und Nebenbedingungen bezüglich der Drehmomente realisierbar, allerdings ist auch die nichtlineare Differentialgleichung des Mehrgrößensystems während der Optimierung zu lösen. Bei Roboterarmen haben sich dedizierte Regelungskonzepte etabliert, welche das nichtlineare Mehrgrößensystem in verschiedenen Kaskaden regeln und nach außen vereinfachen. Sie bieten mit Sollwerten für Gelenkwinkel, Gelenk-

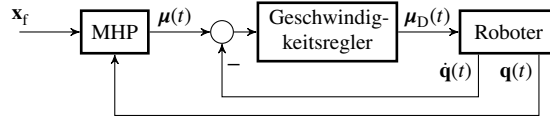


Abbildung 3.1.: Strukturelle Einordnung der MHP in die vorhandenen Schnittstellen des Roboters.

winkelgeschwindigkeit und -beschleunigung eine universelle Schnittstelle für die übergeordnete MHP. Die Stellgröße der MHP ist entsprechend die Sollgeschwindigkeit [Krä+20b; Zub15], -beschleunigung [Wan+16] oder Geschwindigkeitsinkremente [Bui+18; Jar+17], jeweils unter der Annahme diese idealerweise verzögerungsfrei und ohne Überschwingen umzusetzen.

**Definition 3.1.2:** Die Stellgröße  $\mathbf{u} \in \mathbb{R}^M$  (bzw.  $\boldsymbol{\mu} \in \mathbb{R}^M$ ) entspricht den Gelenkwinkelsollgeschwindigkeiten der  $M = N$  Gelenke des Roboters.

Über  $\mathbf{u}$  lassen sich damit auch kinodynamische Vorgaben der Aktorik oder Glätte abbilden. Den Definitionen 3.1.1 und 3.1.2 folgend, wird  $\mathbf{f}(\cdot)$  schließlich über einen  $N$ -fach parallelen Integrator modelliert:

$$\mathbf{f}(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) := \mathbf{u}(\tau). \quad (3.1)$$

Nähert sich die Zeitkonstante des Geschwindigkeitsreglers der Dauer des Planungszyklus  $T_p$  an, bietet sich die Modellierung über verzögerte Integratoren oder Totzeiten an [Krä+20b]. Abbildung 3.1 verdeutlicht den resultierenden strukturellen Aufbau der MHP mit unterlagerter Geschwindigkeitsregelung. Am Roboter werden die aktuellen Gelenkwinkel  $\mathbf{q}(t)$  und -geschwindigkeiten  $\dot{\mathbf{q}}(t)$  gemessen und zur MHP bzw. zum Geschwindigkeitsregler zurückgeführt. Die MHP empfängt den Zielzustand  $\mathbf{x}_f$  und gibt die Gelenkwinkelsollgeschwindigkeit als Stellgröße  $\boldsymbol{\mu}(t)$  gemäß Gleichung (2.6) aus. Der Geschwindigkeitsregler inklusive aller nicht eingezeichneten weiteren Kaskaden erwirkt schließlich ein Drehmoment  $\boldsymbol{\mu}_D \in \mathbb{R}^N$  an den Gelenken.

Da bei der MHP die geplanten Trajektorien von begrenzter Länge sind, ist nicht garantiert, dass sie  $\mathbf{x}_f$  bei der ersten Iteration innerhalb des Horizonts erreichen können. Aus diesem Grund darf Nebenbedingung (2.5c) nicht verwendet werden. Stattdessen wird über den Kostenterm  $c_x : \mathbb{R}^N \rightarrow \mathbb{R}_0^+$  die Abweichung eines Zustands  $\mathbf{x}$  zum Zielzustand  $\mathbf{x}_f$  quadratisch bewertet, um so eine generelle Tendenz der Trajektorie in Richtung Zielzustand zu erwirken:

$$c_x(\mathbf{x}) := (\mathbf{x} - \mathbf{x}_f)^T \mathbf{Q} (\mathbf{x} - \mathbf{x}_f) \text{ mit } \mathbf{Q} \in \mathbb{R}^{N \times N}. \quad (3.2)$$

Befindet sich die MHP im Zielzustand, verliert  $c_x(\cdot)$  den Einfluss auf die Optimierung. Damit die Optimierung im Zielzustand die Tendenz besitzt Bewegungen zu vermeiden, was schließlich auch die Konvergenz begünstigt, werden über den Kostenterm  $c_U : \mathbb{R}^N \rightarrow \mathbb{R}_0^+$  die Stellgrößen bewertet:

$$c_U(\mathbf{u}) := \mathbf{u}^T \mathbf{R} \mathbf{u} \text{ mit } \mathbf{R} \in \mathbb{R}^{N \times N}. \quad (3.3)$$

Für die positiv definiten Gewichtsmatrizen  $\mathbf{Q}$  und  $\mathbf{R}$  werden ohne Beschränkung der Allgemeinheit mit  $\mathbf{Q}, \mathbf{R} \in \mathbb{R}^+$  skalierte Einheitsmatrizen angenommen:

$$\mathbf{Q} := \text{diag}(\dots \mathbf{Q} \dots), \quad (3.4)$$

$$\mathbf{R} := \text{diag}(\dots \mathbf{R} \dots). \quad (3.5)$$

Der Kostenterm  $c_U(\cdot)$  besitzt einen weiteren Effekt, der zu einer Verlangsamung der Bewegung führt, je näher der Roboter dem Zielzustand kommt und die Neigung zum Überschwingen reduziert. Bei der Annäherung an den Zielzustand sinken die Kosten in Gleichung (3.2) auf den Wert

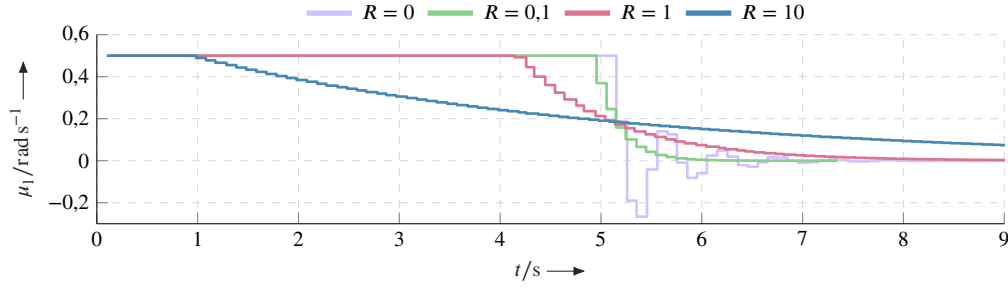


Abbildung 3.2.: Stellgrößen der MHP am ersten Gelenk für unterschiedliche Gewichtsverhältnisse  $\frac{Q=1}{R}$ .

von Gleichung (3.3) wodurch sich in der Folge beide Terme gleichermaßen reduzieren und ein sanftes Anhalten auf dem Zielzustand bewirken. Über das Verhältnis zwischen  $Q$  und  $R$  lässt sich der Effekt in seiner Intensität verändern, wie Abbildung 3.2 für  $Q = 1$  und unterschiedliche Werte für  $R$  verdeutlicht. Für  $R > Q$  wird die Bewegung zu stark verzögert, sodass im Folgenden  $0 < R < Q$  gilt.

Weitere Kostenterme zur Erzielung einer zeitoptimalen Bewegung [Xia+12; Kat+13; Bie+18] oder Bewertung der Glätte der Steuerfolge sind ebenfalls denkbar. Letzteres erfolgt z. B. über die Bewertung der Differenzen aufeinanderfolgender Stellgrößen [Krä+20b] oder durch Bewertung der Beschleunigung [Bui+18].

Nebenbedingungen werden genutzt, um Bedingungen der Planung abzubilden. In erster Linie sind damit zunächst Beschränkungen der Kinematik und Aktorik gemeint. Den Definitionen 3.1.1 und 3.1.2 folgend repräsentieren sie Gelenkwinkelgrenzen  $\mathbf{x}^-$  und  $\mathbf{x}^+$  sowie Geschwindigkeitsbeschränkungen  $\mathbf{u}^-$  und  $\mathbf{u}^+$ :

$$\mathcal{X}_A := \{ \mathbf{x} \in \mathbb{R}^N \mid \mathbf{x}^- \leq \mathbf{x} \leq \mathbf{x}^+ \}, \quad (3.6)$$

$$\mathcal{U}_A := \{ \mathbf{u} \in \mathbb{R}^N \mid \mathbf{u}^- \leq \mathbf{u} \leq \mathbf{u}^+, \mathbf{a}^- \leq \dot{\mathbf{u}} \leq \mathbf{a}^+ \}. \quad (3.7)$$

Es wird außerdem die Gelenkwinkelbeschleunigung  $\dot{\mathbf{u}}$  als Nebenbedingung bezüglich  $\mathcal{U}_A$  aufgenommen und über einen Differenzenquotienten bestimmt. Sie dient damit der Bewertung der Glätte der Steuerfolge. Beschränkungen an den Ruck wie bei Lange u. a. [Lan+16] sind auf dieselbe Weise möglich, werden an dieser Stelle aber nicht verfolgt. Zur Vereinfachung der Notation werden im Folgenden symmetrische Beschränkungen  $\mathbf{x}^+ = -\mathbf{x}^-$ ,  $\mathbf{u}^+ = -\mathbf{u}^-$  und  $\mathbf{a}^+ = -\mathbf{a}^-$  angenommen. Wenn nicht anders angegeben, besitzen die Beschränkungen jeweils identische Werte, z. B.:

$$\mathbf{x}^+ := [x^+ \quad \dots \quad x^+]^T \text{ mit } x^+ \in \mathbb{R}^+. \quad (3.8)$$

Analog dazu sind  $u^+$  und  $a^+$  definiert.

## 3.2. Transkriptionen des nichtlinearen Parameteroptimierungsproblems

Die Trajektorie  $\mathbf{x}(\tau)$  und Steuerfolge  $\mathbf{u}(\tau)$  liegen bisher in kontinuierlicher Form für ein OSP vor. Sie werden im Folgenden über drei direkte Methoden im Zuge der Transkription des OSP in ein NLP mit diskreten Größen überführt [Bet98]. Ein NLP in minimaler Ausführung ist wie folgt definiert:

$$\min_{\mathbf{z}} \hat{J}(\mathbf{z}), \quad (3.9a)$$

unter den Nebenbedingungen:

$$\mathbf{h}(\mathbf{z}) \leq \mathbf{0}, \quad (3.9b)$$

$$\mathbf{g}(\mathbf{z}) = \mathbf{0}. \quad (3.9c)$$

Darin beschreibt  $\mathbf{z} \in \mathbb{R}^D$  den Vektor der  $D \in \mathbb{N}$  Optimierungsvariablen, von denen jene Werte gesucht sind, welche die diskrete Kostenfunktion  $\hat{J} : \mathbb{R}^D \rightarrow \mathbb{R}^+$  unter Einhaltung der  $N_U \in \mathbb{N}_0$  Ungleichheits- und  $N_G \in \mathbb{N}_0$  Gleichheitsnebenbedingungen  $\mathbf{h} : \mathbb{R}^D \rightarrow \mathbb{R}^{N_U}$  bzw.  $\mathbf{g} : \mathbb{R}^D \rightarrow \mathbb{R}^{N_G}$  minimieren. Die genaue Diskretisierung von  $\mathbf{x}(\tau)$  und  $\mathbf{u}(\tau)$  und damit auch die Anzahl der Optimierungsvariablen  $D$  hängt von der konkreten Transkriptionsmethode ab. Allen Varianten gemein ist ein zugrundeliegendes zeitlich äquidistantes Gitter:

$$\tau_k = kT_T \text{ mit } k = 0, 1, \dots, K-1 \text{ und } T_T \in \mathbb{R}^+, \quad (3.10)$$

mit der Horizontlänge  $K > 1$ . Die Einführung der Nebenbedingungen (3.6) und (3.7) impliziert eine minimale Horizontlänge  $K \geq K^-$ . Der Horizont erfordert eine Mindestlänge  $K^-$ , welche das Stoppen der Roboterbewegung im Einklang mit den Nebenbedingungen ermöglicht:

$$K^- = \left\lceil \frac{u^+}{-T_T a^+} \right\rceil. \quad (3.11)$$

## Einfachschießverfahren

Beim Einfachschießverfahren wird die Steuerfolge als diskrete Folge repräsentiert:

$$\mathbf{u}_{0:K-2} := \mathbf{u}_0 \quad \mathbf{u}_1 \quad \dots \quad \mathbf{u}_k \quad \dots \quad \mathbf{u}_{K-2}. \quad (3.12)$$

Daraus ergibt sich bei Bedarf die zeitkontinuierliche Steuerfolge als stückweise konstant:

$$\mathbf{u}(\tau) := \mathbf{u}_k = \text{const. für } \tau \in [\tau_k, \tau_{k+1}). \quad (3.13)$$

Die Optimierungsvariablen sind:

$$\mathbf{z}_{\text{ES}}^T := [\mathbf{u}_{0:K-2}^T]. \quad (3.14)$$

Da  $\mathbf{z}_{\text{ES}}$  nur die diskrete Steuerfolge umfasst, wird die Zustandstrajektorie  $\mathbf{x}(\tau)$  aus  $\mathbf{u}(\tau)$  während der Optimierung durch Integration der Systemdynamik (3.1) ausgehend vom gemessenen Anfangszustand  $\mathbf{x}_0 \equiv \mathbf{q}(t_n)$  laufend berechnet und an den Zeitpunkten  $\tau_k$  abgetastet:

$$\mathbf{x}_{0:K-1} := \mathbf{x}_0 \quad \mathbf{x}_1 \quad \dots \quad \mathbf{x}_k \quad \dots \quad \mathbf{x}_{K-1} \text{ mit } \mathbf{x}_k := \mathbf{x}(\tau_k). \quad (3.15)$$

In diesem Fall verwendet  $\mathbf{x}_{0:K-1}$  dasselbe zeitliche Gitter wie  $\mathbf{u}_{0:K-2}$ , es kann jedoch auch individuell abgetastet werden. Zu beachten ist, dass  $\mathbf{u}_{0:K-2}$  ein Element weniger besitzt als  $\mathbf{x}_{0:K-1}$ , da eine Stellgröße  $\mathbf{u}_k$  jeweils auf das nachfolgende Zeitintervall  $[\tau_k, \tau_{k+1})$  der Trajektorie einwirkt. Die Kostenterme  $c_X(\cdot)$  und  $c_U(\cdot)$  werden auf jedem diskreten Zustand und jeder diskreten Stellgröße ausgewertet und in der Kostenfunktion  $\hat{J}(\cdot)$  zusammengefasst:

$$\hat{J}(\mathbf{z}_{\text{ES}}) := \sum_{k=1}^{K-1} c_X(\mathbf{x}_k, k) + \sum_{k=0}^{K-2} c_U(\mathbf{u}_k, k). \quad (3.16)$$

Die Kostenterme werden um den Gitterindex  $k$  erweitert, womit zeitabhängige Gewichte  $Q_k$  und  $R_k$  möglich sind. Später wird die Zeitabhängigkeit bei der Kollisionsvermeidung für die Prädiktion der Bewegung von dynamischen Hindernissen genutzt.

Das resultierende NLP lautet wie folgt:

$$\min_{\mathbf{z}_{\text{ES}}} \hat{J}(\mathbf{z}_{\text{ES}}), \quad (3.17a)$$

unter den Nebenbedingungen:

$$\mathbf{h}_{\text{ES}}(\mathbf{z}_{\text{MS}}) := \begin{cases} \mathbf{x}_k - \mathbf{x}^+ & k = 1, 2, \dots, K-1 \\ \mathbf{x}^- - \mathbf{x}_k & k = 1, 2, \dots, K-1 \\ \mathbf{u}_k - \mathbf{u}^+ & k = 0, 1, \dots, K-2 \\ \mathbf{u}^- - \mathbf{u}_k & k = 0, 1, \dots, K-2 \\ \mathbf{u}_k - \mathbf{u}_{k-1} - \mathbf{a}^+ T_{\text{T}} & k = 0, 1, \dots, K-2 \\ \mathbf{u}_{k-1} - \mathbf{u}_k - \mathbf{a}^+ T_{\text{T}} & k = 0, 1, \dots, K-2 \end{cases} \leq \mathbf{0}. \quad (3.17b)$$

Das NLP besitzt damit  $D = (K-1)N$  Optimierungsvariablen und  $N_{\text{U}} = 6N(K-1)$  Ungleichheitsnebenbedingungen. Gleichheitsnebenbedingungen gibt es keine ( $N_{\text{G}} = 0$ ). Die Stellgröße  $\mathbf{u}_{-1}$  entspricht  $\mathbf{u}_{n-1}^*(0)$  oder  $\mathbf{0}$  bei der ersten Planung. Aufgrund der Integration der Systemdynamik reagiert diese Transkription in Kostenfunktion und Nebenbedingungen sensitiv auf Änderungen in der Steuerfolge.

## Mehrfachschießverfahren

Das Mehrfachschießverfahren wendet das Einfachschießverfahren auf  $I > 1$  Intervalle an:

$$\mathbf{v}_{1:I} := \mathbf{u}_{0:K'-2}^{(1)} \quad \mathbf{u}_{0:K'-2}^{(2)} \quad \dots \quad \mathbf{u}_{0:K'-2}^{(i)} \quad \dots \quad \mathbf{u}_{0:K'-2}^{(I)}, \quad (3.18)$$

mit der Intervalllänge  $K' > 1$  (bezogen auf die Teilzustandsfolgen) und den diskreten (sowie kontinuierlichen) Teilsteuerfolgen:

$$\mathbf{u}_{0:K'-2}^{(i)} := \mathbf{u}_0^{(i)} \quad \mathbf{u}_1^{(i)} \quad \dots \quad \mathbf{u}_k^{(i)} \quad \dots \quad \mathbf{u}_{K'-2}^{(i)}, \quad (3.19)$$

$$\mathbf{u}^{(i)}(\tau) := \mathbf{u}_k^{(i)} = \text{const. für } t \in [\tau_k, \tau_{k+1}). \quad (3.20)$$

Es zählen neben der Steuerfolge auch die Zustände (engl. *shooting nodes*)  $\chi^{(i)} \in \mathbb{R}^N$  am Beginn der Intervalle zu den Optimierungsvariablen:

$$\chi_{1:I} := \chi^{(1)} \quad \chi^{(2)} \quad \dots \quad \chi^{(i)} \quad \dots \quad \chi^{(I)}. \quad (3.21)$$

Damit ergibt sich der Vektor  $\mathbf{z}_{\text{MS}}$  zu:

$$\mathbf{z}_{\text{MS}}^{\text{T}} := [\mathbf{v}_{1:I}^{\text{T}} \quad \chi_{2:I}^{\text{T}}]. \quad (3.22)$$

Hierbei entspricht  $\chi^{(1)} \equiv \mathbf{q}(t_n)$  dem gegenwärtigen Zustand des Roboters und ist daher unveränderlich. Die Zustandstrajektorie  $\mathbf{x}^{(i)}(\tau)$  des  $i$ -ten Intervalls wird wie beim Einfachschießverfahren aus  $\mathbf{u}^{(i)}(\tau)$  durch Integration der Systemdynamik (3.1) ausgehend von  $\chi^{(i)}$  berechnet und an den Zeitpunkten  $\tau_k$  abgetastet, um sie in Kostenfunktion und Nebenbedingungen zu bewerten:

$$\mathbf{x}_{0:K'-1}^{(i)} := \underbrace{\mathbf{x}_0^{(i)} \quad \mathbf{x}_1^{(i)} \quad \dots \quad \mathbf{x}_k^{(i)} \quad \dots \quad \mathbf{x}_{K'-1}^{(i)}}_{\equiv \chi^{(i)}} \quad \text{mit } \mathbf{x}_k^{(i)} := \mathbf{x}^{(i)}(\tau_k). \quad (3.23)$$

Die Zustandstrajektorien verwenden in diesem Fall dasselbe Gitter wie die erzeugende Teilsteuerfolgen, sie können jedoch auch individuell abgetastet werden. Die Intervalllänge  $K'$  ist der Einfachheit halber für alle Intervalle gleich, kann jedoch auch variabel gewählt werden. Es ergibt sich die gesamte Gitterlänge  $K = I(K' - 1) + 1$ . Der stetige Übergang zwischen den Intervallen wird über die Nebenbedingung  $\mathbf{x}_{K'-1}^{(i)} = \chi^{(i+1)}$  erzwungen.

Das resultierende NLP lautet wie folgt:

$$\min_{\mathbf{z}_{\text{MS}}} \hat{J}(\mathbf{z}_{\text{MS}}), \quad (3.24a)$$

unter den Nebenbedingungen:

$$\mathbf{h}_{\text{MS}}^{(i)}(\mathbf{z}_{\text{MS}}) := \begin{cases} \mathbf{x}_k^{(i)} - \mathbf{x}^+ & k = 1, 2, \dots, K' - 1 \\ \mathbf{x}^- - \mathbf{x}_k^{(i)} & k = 1, 2, \dots, K' - 1 \\ \mathbf{u}_k^{(i)} - \mathbf{u}^+ & k = 0, 1, \dots, K' - 2 \\ \mathbf{u}^- - \mathbf{u}_k^{(i)} & k = 0, 1, \dots, K' - 2 \\ \mathbf{u}_k^{(i)} - \mathbf{u}_{k-1}^{(i)} - \mathbf{a}^+ T_T & k = 0, 1, \dots, K' - 2 \\ \mathbf{u}_{k-1}^{(i)} - \mathbf{u}_k^{(i)} - \mathbf{a}^+ T_T & k = 0, 1, \dots, K' - 2 \end{cases}, \quad (3.24b)$$

$$\mathbf{g}_{\text{MS}}^{(i)}(\mathbf{z}_{\text{MS}}) := \{ \mathbf{x}_{K'-1}^{(i)} - \chi^{(i+1)} \}, \quad (3.24c)$$

$$\mathbf{h}_{\text{MS}}^{\text{T}}(\mathbf{z}_{\text{MS}}) := (\mathbf{h}_{\text{MS}}^{(1)}(\mathbf{z}_{\text{MS}}) \quad \mathbf{h}_{\text{MS}}^{(2)}(\mathbf{z}_{\text{MS}}) \quad \dots \quad \mathbf{h}_{\text{MS}}^{(I)}(\mathbf{z}_{\text{MS}})) \leq \mathbf{0}, \quad (3.24d)$$

$$\mathbf{g}_{\text{MS}}^{\text{T}}(\mathbf{z}_{\text{MS}}) := (\mathbf{g}_{\text{MS}}^{(1)}(\mathbf{z}_{\text{MS}}) \quad \mathbf{g}_{\text{MS}}^{(2)}(\mathbf{z}_{\text{MS}}) \quad \dots \quad \mathbf{g}_{\text{MS}}^{(I-1)}(\mathbf{z}_{\text{MS}})) = \mathbf{0}. \quad (3.24e)$$

Das NLP besitzt damit  $D = (K + I - 2)N$  Optimierungsvariablen,  $N_{\text{U}} = 6N(K - 1)$  Ungleichheitsnebenbedingungen und  $N_{\text{G}} = (I - 1)N$  Gleichheitsnebenbedingungen. Die Stellgröße  $\mathbf{u}_{-1}$  entspricht  $\mathbf{u}_{n-1}^*(0)$  oder  $\mathbf{0}$  bei der ersten Planung. Die Sensitivität auf Änderungen in den Optimierungsvariablen wird auf die kürzeren Schießintervalle verteilt und dadurch abgemildert. Für  $I = 1$  geht das Mehrfachschießverfahren in das Einzelschießverfahren über.

## Vollständige Diskretisierung

Die vollständige Diskretisierung ist eine Variante der Kollokation, in der neben der Steuerfolge auch die Zustandstrajektorie im Vorfeld diskret repräsentiert wird:

$$\mathbf{x}_{0:K-1} := \mathbf{x}_0 \quad \mathbf{x}_1 \quad \dots \quad \mathbf{x}_k \quad \dots \quad \mathbf{x}_{K-1}. \quad (3.25)$$

Üblicherweise verwenden Kollokationsmethoden nach einem bestimmten Muster verteilte Zwischenpunkte, um die Approximationsgenauigkeit zu erhöhen [Die+06]. Da  $\mathbf{f}(\cdot)$  aus parallelen einfachen Integratoren besteht, wird auf Zwischenpunkte verzichtet. Die Optimierungsvariablen sind damit:

$$\mathbf{z}_{\text{VD}}^{\text{T}} := [\mathbf{u}_{0:K-2}^{\text{T}} \quad \mathbf{x}_{1:K-1}^{\text{T}}]. \quad (3.26)$$

Hierbei entspricht  $\mathbf{x}_0 \equiv \mathbf{q}(t_n)$  dem gegenwärtigen Zustand des Roboters und ist daher unveränderlich. Die Berechnung von  $\mathbf{x}_{1:K-1}$  über Integration von Gleichung (3.1), wie zuvor bei den Schießverfahren, entfällt. Allerdings entfällt damit auch der implizite Zusammenhang zwischen Stellgröße und Zustand, welcher bei der vollständigen Diskretisierung jetzt explizit über Zustandsgleichungen realisiert wird:

$$\mathbf{x}_{k+1} = \hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k), \quad (3.27)$$

worin  $\hat{\mathbf{f}}(\cdot)$  die über das explizite Eulerverfahren diskretisierte Systemdynamik (3.1) darstellt:

$$\hat{\mathbf{f}}(\mathbf{x}_k, \mathbf{u}_k) := \mathbf{x}_k + \mathbf{u}_k T_T. \quad (3.28)$$

Das resultierende NLP lautet wie folgt:

$$\min_{\mathbf{z}_{\text{VD}}} \hat{J}(\mathbf{z}_{\text{VD}}), \quad (3.29a)$$

unter den Nebenbedingungen:

$$\mathbf{h}_{\text{VD}}(\mathbf{z}_{\text{VD}}) := \begin{cases} \mathbf{x}_k - \mathbf{x}^+ & k = 1, 2, \dots, K-1 \\ \mathbf{x}^- - \mathbf{x}_k & k = 1, 2, \dots, K-1 \\ \mathbf{u}_k - \mathbf{u}^+ & k = 0, 1, \dots, K-2 \\ \mathbf{u}^- - \mathbf{u}_k & k = 0, 1, \dots, K-2 \\ \mathbf{u}_k - \mathbf{u}_{k-1} - \mathbf{a}^+ T_T & k = 0, 1, \dots, K-2 \\ \mathbf{u}_{k-1} - \mathbf{u}_k - \mathbf{a}^+ T_T & k = 0, 1, \dots, K-2 \end{cases} \leq \mathbf{0}, \quad (3.29b)$$

$$\mathbf{g}_{\text{VD}}(\mathbf{z}_{\text{VD}}) := \{ \mathbf{x}_{k+1} - \mathbf{x}_k - \mathbf{u}_k T_T \quad k = 0, 1, \dots, K-2 \} = \mathbf{0}. \quad (3.29c)$$

Das NLP besitzt damit  $D = 2N(K-1)$  Optimierungsvariablen,  $N_U = 6N(K-1)$  Ungleichheitsnebenbedingungen und  $N_G = N(K-1)$  Gleichheitsnebenbedingungen. Die Stellgröße  $\mathbf{u}_{-1}$  entspricht  $\mathbf{u}_{n-1}^*(0)$  oder  $\mathbf{0}$  bei der ersten Planung. Die vollständige Diskretisierung führt auf das NLP mit der geringsten Sensitivität. Die NLPs (3.24) und (3.29) gehen für  $K' = 2$ ,  $I = K-1$  und, wenn die Integration der Systemdynamik (3.1) über das explizite Eulerverfahren erfolgt, annähernd ineinander über. Nur der  $K-1$ -te Zustand und die  $K-2$ -te Gleichheitsnebenbedingung kommen bei der vollständigen Diskretisierung explizit hinzu.

Im weiteren Verlauf wird der Begriff der Größenordnung über die Parameter  $D$ ,  $N_G$  und  $N_U$  definiert. Während die Transkriptionsmethode Einfluss auf  $D$  und  $N_G$  nimmt, ist  $N_U = 6N(K-1)$  davon unabhängig. In allen Varianten wird die Größenordnung von den Ungleichheitsnebenbedingungen in Form der Zustands- und Stellgrößenbeschränkungen dominiert. Im nächsten Kapitel kommen zum Zweck der Kollisionsvermeidung weitere Ungleichheitsnebenbedingungen hinzu.

### 3.3. Verfahren zur Lösung des Optimierungsproblems

Ziel des Lösungsverfahrens ist, einen Vektor  $\mathbf{z}^*$  zu finden, welcher die Optimalitätsbedingungen eines NLP wie (3.9) erfüllt. Im Folgenden werden zunächst kurz die Optimalitätsbedingungen und die damit verbundenen Begriffe wie Lagrangefunktion und Lagrangemultiplikatoren erläutert, ehe dann auf die Lösungsverfahren und die involvierten Gleichungssysteme als Grundlage für die Aufwandsanalysen in Kapitel 5 eingegangen wird.

Die Lagrangefunktion  $L : \mathbb{R}^D \times \mathbb{R}^{N_G} \rightarrow \mathbb{R}$  verbindet Kostenfunktion  $\hat{J}(\cdot)$  und Gleichheitsnebenbedingungen  $\mathbf{g}(\cdot)$  über eine Linearkombination:

$$L(\mathbf{z}, \lambda_G) := \hat{J}(\mathbf{z}) + \sum_{i=1}^{N_G} \lambda_{G,i} g_i(\mathbf{z}) \quad \text{mit} \quad \lambda_G^T := (\lambda_{G,1} \quad \lambda_{G,2} \quad \dots \quad \lambda_{G,N_G}) \quad (3.30)$$

und den Lagrangemultiplikatoren  $\lambda_{G,i} \in \mathbb{R}$  als Koeffizienten. Es ist  $g_i(\cdot)$  die  $i$ -te Zeile in  $\mathbf{g}(\cdot)$ . Die Lagrangefunktion bildet über  $\nabla_{\mathbf{z}} L(\mathbf{z}^*, \lambda_G^*) = \mathbf{0}$  einen Teil der notwendigen Optimalitätsbedingungen bei Gleichheitsnebenbedingungen ab, der besagt, dass der Gradient der Kostenfunktion

und die Gradienten der Nebenbedingungen bis auf ein Skalar  $\lambda_{G,i}$  parallel zueinander stehen [Noc+06]:

$$\nabla_z \hat{J}(\mathbf{z}^*) = -\mathbf{G}^T(\mathbf{z}^*)\lambda_G^*, \quad (3.31)$$

mit der Jacobimatrix  $\mathbf{G}(\cdot) \in \mathbb{R}^{N_G \times D}$  von  $\mathbf{g}(\cdot)$ . Die Nebenbedingungen  $\mathbf{g}(\mathbf{z}^*) = \mathbf{0}$  vervollständigen die notwendigen Optimalitätsbedingungen:

$$\nabla_z \hat{J}(\mathbf{z}^*) + \mathbf{G}^T(\mathbf{z}^*)\lambda_G^* = \mathbf{0}, \quad (3.32a)$$

$$\mathbf{g}(\mathbf{z}^*) = \mathbf{0}. \quad (3.32b)$$

Bei Hinzunahme von  $\mathbf{h}(\cdot)$  erweitert sich die Lagrangefunktion (3.30) zu:

$$L(\mathbf{z}, \lambda_G, \lambda_U) := \hat{J}(\mathbf{z}) + \sum_{i=1}^{N_G} \lambda_{G,i} g_i(\mathbf{z}) + \sum_{i=1}^{N_U} \lambda_{U,i} h_i(\mathbf{z}) \text{ mit } \lambda_U^T := (\lambda_{U,1} \ \lambda_{U,2} \ \dots \ \lambda_{U,N_U}). \quad (3.33)$$

Die Optimalitätsbedingungen, die nach ihren Entdeckern auch KKT<sup>1</sup>-Bedingungen genannt werden, lauten:

$$\nabla_z \hat{J}(\mathbf{z}^*) + \mathbf{G}^T(\mathbf{z}^*)\lambda_G^* + \mathbf{H}^T(\mathbf{z}^*)\lambda_U^* = \mathbf{0} \quad (\nabla_z L(\cdot) = \mathbf{0}), \quad (3.34a)$$

$$\mathbf{g}(\mathbf{z}^*) = \mathbf{0}, \quad (3.34b)$$

$$\lambda_{U,i}^* h_i(\mathbf{z}^*) = 0 \text{ für } i = 1, 2, \dots, N_U, \quad (3.34c)$$

$$\mathbf{h}(\mathbf{z}^*) \leq \mathbf{0}, \quad (3.34d)$$

$$\lambda_U^* \geq 0, \quad (3.34e)$$

mit der Jacobimatrix  $\mathbf{H}(\cdot) \in \mathbb{R}^{N_U \times D}$  von  $\mathbf{h}(\cdot)$ . Die Lagrangemultiplikatoren  $\lambda_U \in \mathbb{R}^{N_U}$  haben die Aufgabe, erfüllte Ungleichheitsnebenbedingungen über  $\lambda_{U,i} = 0$  zu deaktivieren. Dies wird über Gleichung (3.34c) in Form der Komplementaritätsbedingung abgebildet. Für hinreichende Bedingungen sowie Bedingungen zweiter Ordnung wird auf Nocedal u. a. [Noc+06] verwiesen. Entsprechende Lösungsverfahren stellen bis auf einige Ausnahmen alle eine Art der Newton-Methode dar, die Nullstellen  $\Lambda(\mathbf{y}^*) = \mathbf{0}$  einer nichtlinearen Funktion  $\Lambda(\mathbf{y})$  iterativ berechnet [Die+09; Noc+06; Bet98]:

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \Delta \mathbf{y}_j \text{ für } j = 0, 1, \dots \quad (3.35)$$

In jeder Iteration wird ein Lösungskandidat  $\mathbf{y}_{j+1}$  bestimmt, welcher sich der Nullstelle  $\mathbf{y}^*$  annähert. Der Newton-Schritt  $\Delta \mathbf{y}_j$  lässt sich über folgendes Gleichungssystem berechnen:

$$\nabla \Lambda(\mathbf{y}_j) \Delta \mathbf{y}_j = -\Lambda(\mathbf{y}_j). \quad (3.36)$$

Durch geschickte Wahl von  $\Lambda(\mathbf{y})$  und  $\mathbf{y}$  im Hinblick auf die Optimalitätsbedingungen repräsentieren die Nullstellen die Lösung des NLP (3.9). Ein Vergleich der Optimalitätsbedingungen zwischen (3.32) und (3.34) zeigt, dass dieser Fall bereits bei einem Optimierungsproblem ohne Ungleichungen vorliegt. Die Lösungsverfahren unterscheiden sich daher im Wesentlichen in der Behandlung der Ungleichheitsnebenbedingungen. Zwei der bekanntesten Arten werden im Folgenden kurz vorgestellt.

<sup>1</sup>Karush-Kuhn-Tucker

## Innere-Punkte-Methoden

IP<sup>2</sup>-Methoden wandeln die Ungleichheitsnebenbedingungen in zusätzliche Gleichheitsnebenbedingungen um:

$$h'_i(\mathbf{z}, \eta_i) := h_i(\mathbf{z}) + \eta_i = 0 \text{ für } i = 1, 2, \dots, N_U, \quad (3.37)$$

mit den Schlupfvariablen  $\eta_i \in \mathbb{R}$ , die über eine logarithmische Barrierefunktion in die Kostenfunktion aufgenommen werden und damit nicht als explizite Ungleichheitsnebenbedingungen in Erscheinung treten:

$$\hat{J}_{\text{IP}}(\mathbf{z}, \boldsymbol{\eta}) := \hat{J}(\mathbf{z}) - \xi \sum_{i=1}^{N_U} \ln(\eta_i) \text{ für } \xi \in \mathbb{R}^+ \text{ und } \boldsymbol{\eta}^T = (\eta_1 \ \eta_2 \ \dots \ \eta_{N_U}). \quad (3.38)$$

Die Lagrangefunktion (3.30) wird um die Barrierefunktion und die neuen Gleichheitsnebenbedingungen erweitert:

$$\begin{aligned} L_{\text{IP}}(\mathbf{z}, \boldsymbol{\eta}, \boldsymbol{\lambda}_G, \boldsymbol{\lambda}'_U) &:= \hat{J}_{\text{IP}}(\mathbf{z}, \boldsymbol{\eta}) + \sum_{i=1}^{N_G} \lambda_{G,i} g_i(\mathbf{z}) + \sum_{i=1}^{N_U} \lambda'_{U,i} h'_i(\mathbf{z}, \eta_i), \\ \boldsymbol{\lambda}_G^T &:= (\lambda_{G,1} \ \lambda_{G,2} \ \dots \ \lambda_{G,N_G}), \\ \boldsymbol{\lambda}'_U{}^T &:= (\lambda'_{U,1} \ \lambda'_{U,2} \ \dots \ \lambda'_{U,N_U}). \end{aligned} \quad (3.39)$$

Das resultierende Optimierungsproblem besitzt zwar mehr Optimierungsvariablen, aber nur noch Gleichheitsnebenbedingungen. Eine Lösung muss daher nur den Optimalitätsbedingungen (3.32) genügen, welche um die neuen Nebenbedingungen (3.37) und die Schlupfvariablen ergänzt werden:

$$\nabla_z \hat{J} + \mathbf{G}^T \boldsymbol{\lambda}_G + \mathbf{H}^T \boldsymbol{\lambda}'_U = \mathbf{0} \quad (\nabla_z L_{\text{IP}}(\cdot) = \mathbf{0}), \quad (3.40a)$$

$$-\xi \boldsymbol{\eta}^{-1} + \boldsymbol{\lambda}'_U = \mathbf{0} \quad (\nabla_{\boldsymbol{\eta}} L_{\text{IP}}(\cdot) = \mathbf{0}), \quad (3.40b)$$

$$\mathbf{g} = \mathbf{0}, \quad (3.40c)$$

$$\mathbf{h} + \boldsymbol{\eta} = \mathbf{0}, \quad (3.40d)$$

mit  $\boldsymbol{\eta}^{-1} := (\eta_1^{-1} \ \eta_2^{-1} \ \dots \ \eta_{N_U}^{-1})^T$ . Die Funktionsargumente werden zur besseren Lesbarkeit ausgelassen. Werden zur Lösung von (3.40) die Funktion  $\Lambda(\mathbf{y})$  und Variable  $\mathbf{y}$  wie folgt gewählt:

$$\mathbf{y}^T := (\mathbf{z} \ \boldsymbol{\eta} \ \boldsymbol{\lambda}_G \ \boldsymbol{\lambda}'_U) \text{ und } \Lambda(\mathbf{z}, \boldsymbol{\eta}, \boldsymbol{\lambda}_G, \boldsymbol{\lambda}'_U) := \begin{bmatrix} \nabla_z L_{\text{IP}} \\ \nabla_{\boldsymbol{\eta}} L_{\text{IP}} \\ \mathbf{g} \\ \mathbf{h} + \boldsymbol{\eta} \end{bmatrix}, \quad (3.41)$$

ergibt sich das Gleichungssystem für den Newton-Schritt zu:

$$\begin{bmatrix} \nabla_{zz}^2 L_{\text{IP}} & \mathbf{0} & \mathbf{G}^T & \mathbf{H}^T \\ \mathbf{0} & \xi \boldsymbol{\eta}^{-2} & \mathbf{0} & \mathbf{I}_{N_U} \\ \mathbf{G} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{H} & \mathbf{I}_{N_U} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \boldsymbol{\eta} \\ \Delta \boldsymbol{\lambda}_G \\ \Delta \boldsymbol{\lambda}'_U \end{bmatrix} = \begin{bmatrix} -\nabla_z \hat{J} - \mathbf{G}^T \boldsymbol{\lambda}_G - \mathbf{H}^T \boldsymbol{\lambda}'_U \\ \xi \boldsymbol{\eta}^{-1} - \boldsymbol{\lambda}'_U \\ -\mathbf{g} \\ -\mathbf{h} - \boldsymbol{\eta} \end{bmatrix}, \quad (3.42)$$

mit  $\boldsymbol{\eta}^{-2} := \text{diag} \left( \eta_1^{-2} \quad \eta_2^{-2} \quad \dots \quad \eta_{N_U}^{-2} \right)^T$ . Die Funktionsargumente und Indizes der Iteration werden zur besseren Lesbarkeit ausgelassen. Die Matrix  $\nabla \Lambda(\cdot)$  wird auch KKT-Matrix genannt und besitzt die Dimensionen  $(D + 2N_U + N_G) \times (D + 2N_U + N_G)$ . Wird die Hessematrix  $\nabla_{zz}^2 L_{IP}$  über Approximationsverfahren angenähert, wird von Quasi-Newton-Verfahren gesprochen. Implementierungen der IP-Methode sind z. B. IPOpt von Wächter u. a. [Wac+06].

### Active Set und Sequential Quadratic Programming

Das AS<sup>3</sup> beschreibt die Menge von aktiven Ungleichheitsnebenbedingungen. Aktiv bedeutet, dass der Lösungskandidat von der Ungleichheitsnebenbedingung beschränkt wird und sie damit effektiv zu einer Gleichheitsnebenbedingung wird. AS-Methoden versuchen diese Menge laufend zu bestimmen, um anschließend wie bei der IP-Methode nur den Optimalitätsbedingungen (3.32) genügen zu müssen, ohne dabei jedoch das Optimierungsproblem zu verändern. Werden zur Lösung von (3.32) die Funktion  $\Lambda(\mathbf{y})$  und Variable  $\mathbf{y}$  wie folgt gewählt:

$$\mathbf{y}^T := (\mathbf{z} \quad \boldsymbol{\lambda}) \quad \text{und} \quad \Lambda(\mathbf{z}, \boldsymbol{\lambda}) := \begin{bmatrix} \nabla_{\mathbf{z}} L(\mathbf{z}, \boldsymbol{\lambda}) \\ \mathbf{g}(\mathbf{z}) \end{bmatrix}, \quad (3.43)$$

ergibt sich folgendes Gleichungssystem für den Newton-Schritt:

$$\begin{bmatrix} \nabla_{zz}^2 L & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\nabla_{\mathbf{z}} \hat{J} - \mathbf{G}^T \boldsymbol{\lambda} \\ -\mathbf{g} \end{bmatrix}. \quad (3.44)$$

Bei leerem AS ist die Dimension  $(D + N_G) \times (D + N_G)$ . Die obere Schranke liegt bei  $(D + N_G + N_U) \times (D + N_G + N_U)$  mit vollständigem<sup>4</sup> AS.

Nocedal u. a. [Noc+06] zeigen, dass (3.44) den Optimalitätsbedingungen eines quadratischen Optimierungsproblems mit quadratisch approximierter Lagrangefunktion (3.30) und linearisierten Gleichheitsnebenbedingungen entspricht. Damit repräsentiert jede Newton-Iteration die Lösung dieses quadratischen Unterproblems. Diese Ähnlichkeit motiviert die SQP<sup>5</sup>-Methoden. Bei diesen werden zusätzlich die Ungleichheitsnebenbedingungen linearisiert und in das quadratische Unterproblem aufgenommen, welches dann von Verfahren dieser Problemklasse gelöst wird und einer Iteration von  $\mathbf{y}$  auf dem NLP ähnelt. Die Behandlung der Ungleichungen erfolgt auf der Ebene des Unterproblems, z. B. über AS-, IP- oder *Penalty*-Methoden. Häufig wird die Variante in (3.44) auch bereits zu den SQP-Methoden gezählt, obwohl das AS im Vordergrund steht. Eng verwandt mit den SQP-Methoden sind die SCP<sup>6</sup>-Methoden. Anstelle eines quadratischen Unterproblems wird laufend eine konvexe Approximation von Zielfunktion (nicht Lagrangefunktion) und Nebenbedingungen erstellt.

Eine Softwareimplementierung der AS-SQP-Methode ist z. B. SLSQP<sup>7</sup> [Kra88]. HQP ist eine reine SQP-Methode, da anstelle einer AS- eine IP-Methode verwendet wird, um die Ungleichheitsnebenbedingungen des quadratischen Unterproblems zu eliminieren. Zur Lösung quadratischer Optimierungsprobleme haben sich Implementierungen wie qpOases<sup>8</sup> [Fer+14] oder OSQP<sup>9</sup> [Ste+20] bewährt. Eine SCP-Methode wird in TrajOpt [Sch+13] implementiert.

<sup>3</sup>Active Set

<sup>4</sup>Bei einem sinnvollen NLP sind üblicherweise nicht alle Ungleichheitsnebenbedingungen gleichzeitig aktiv

<sup>5</sup>Sequential Quadratic Programming

<sup>6</sup>Sequential Convex Programming

<sup>7</sup>Sequential Least-Squares Quadratic Programming

<sup>8</sup>Lösungsverfahren für quadratische Optimierungsprobleme über eine *Online Active Set Strategy*

<sup>9</sup>Operator Splitting (Solver) for Quadratic Programs

# 4

## Kollisionsvermeidung für statische Hindernisse

Nachdem im vorherigen Kapitel unterschiedliche NLPs für die MHP aufgestellt wurden, thematisiert dieses Kapitel die Erweiterung auf die Kollisionsvermeidung. Dafür werden in Abschnitt 4.1 zunächst die SSVs als Hüllkörper formal eingeführt und Hindernisse in statisch, dynamisch und Armkörper klassifiziert. Abschnitt 4.2 behandelt die zugehörigen Distanzfunktionen und stellt eine effiziente Methode zur Approximation der Penetrationsdistanz vor, welche die Nutzung von gradientenbasierten Lösungsverfahren verbessert. Des Weiteren erfolgt ein Vergleich von Berechnungszeiten eigener und vorhandener Implementierungen von Distanzfunktionen mit dem Ziel einer effizienten Menge für alle Hüllkörperkombinationen. In Abschnitt 4.3 werden schließlich die distanzbasierten Nebenbedingungen und die Annäherungskosten zur Kollisionsvermeidung aufgestellt. Dabei werden Schwellwerte und Gewichte aus praktischer Sicht motiviert. Das Kapitel schließt mit einer Auswertung der resultierenden Berechnungszeiten von Kostenfunktion und Ungleichheitsnebenbedingungen sowie Potential für Einsparungen ab.

Die Ergebnisse dieses Kapitels basieren in Teilen auf den Publikationen [Krä+20b; Krä+18b].

### 4.1. Modellierung der Umgebung über sphärische Volumenkörper

Die Kollisionsvermeidung läuft im Wesentlichen auf die Bewertung und Wahrung von Abständen hinaus. Bei der MHP sind daher insbesondere solche Formen im Fokus, die sich neben einer effizienten räumlichen Passung auch effizient in die Berechnungen der Planung integrieren. Entsprechend beliebt sind SSVs  $\mathcal{O} \subset \mathbb{R}^3$ . Sie entstehen aus der Minkowskisumme  $\oplus$  zwischen einem Primitiv  $S \subset \mathbb{R}^3$  und einer Kugel  $\mathcal{K}_r \subset \mathbb{R}^3$  mit Radius  $r \in \mathbb{R}^+$ :

$$\mathcal{O} = \mathcal{K}_r \oplus S. \quad (4.1)$$

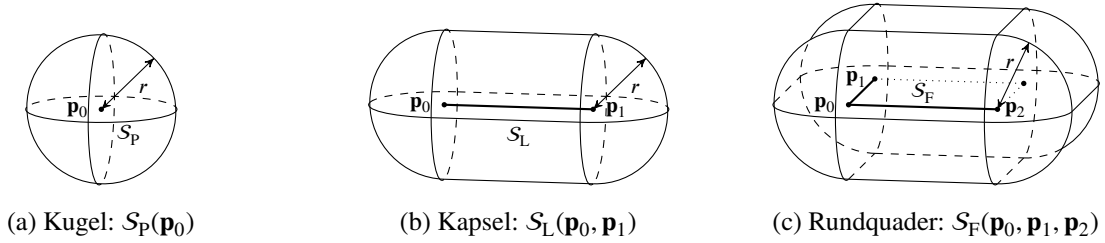
Als Primitive werden Punkte, Linien- oder Flächensegmente verwendet:

$$S_p(\mathbf{p}_0) := \mathbf{p}_0 \in \mathbb{R}^3, \quad (4.2)$$

$$S_L(\mathbf{p}_0, \mathbf{p}_1) := \{ \mathbf{p} \in \mathbb{R}^3 \mid \mathbf{p} = \mathbf{p}_0 + v_1(\mathbf{p}_1 - \mathbf{p}_0), v_1 \in [0, 1] \}, \quad (4.3)$$

$$S_F(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2) := \{ \mathbf{p} \in \mathbb{R}^3 \mid \mathbf{p} = \mathbf{p}_0 + v_1(\mathbf{p}_1 - \mathbf{p}_0) + v_2(\mathbf{p}_2 - \mathbf{p}_0), v_1, v_2 \in [0, 1] \}. \quad (4.4)$$

Sie bestehen aus einem Anfangspunkt  $\mathbf{p}_0$  und wahlweise aus weiteren End- bzw. Eckpunkten  $\mathbf{p}_1$  und  $\mathbf{p}_2$ . Bildlich gesehen ergeben sich die Hüllkörper aus dem Entlangfahren der Kugel am


 Abbildung 4.1.: Drei SSVs und ihre bildenden Primitive  $S$ .

Primitiv. Abbildung 4.1 zeigt die drei resultierenden Hüllkörper. Der Punkt ist das neutrale Element, das genau die Kugel  $\mathcal{K}_r$  ergibt. Das Liniensegment erzeugt einen Zylinder mit halbrunden Enden (Kapsel) und das Flächensegment einen Quader mit halbrunden Seitenflächen (Rundquader). Insbesondere die zweite Variante bietet sich an, um Armkörper von Robotern und Personen effizient zu umschließen. Im weiteren Verlauf ist bei Abständen zwischen Objekten immer jener zwischen den Hüllkörpern gemeint.

Neben den beschränkten SSVs eignen sich auch ausgedehnte Formen, um z. B. den Arbeitsraum zum Boden, zu Wänden oder zur Decke hin zu beschränken. Erreicht wird dies, indem die Definition der Flächensegmente in  $S_F$  auf unbeschränkte Ebenen erweitert und  $\mathcal{K}_0$  gewählt wird:

$$S_E(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2) := \{ \mathbf{p} \in \mathbb{R}^3 \mid \mathbf{p} = \mathbf{p}_0 + \nu_1(\mathbf{p}_1 - \mathbf{p}_0) + \nu_2(\mathbf{p}_2 - \mathbf{p}_0), \nu_1, \nu_2 \in \mathbb{R} \}. \quad (4.5)$$

Die folgende Notation beschreibt die Translation  $\mathbf{s} \in \mathbb{R}^3$  und Rotation  $\Theta \in \text{SO}(3)$  eines Hüllkörpers  $\mathcal{O}$  bzw. der bildenden Punkte  $\mathbf{p}_0, \mathbf{p}_1$  und  $\mathbf{p}_2$  des zugrundeliegenden Primitivs  $S$ :

$$\mathcal{O}' = \Theta \mathcal{O} + \mathbf{s} \Rightarrow \mathbf{p}'_i = \Theta \mathbf{p}_i + \mathbf{s} \text{ für } i = 0, 1, 2. \quad (4.6)$$

Mit Hilfe der Hüllkörperpose wird für den weiteren Verlauf zwischen drei Klassen von Hüllkörpern unterschieden:

$$\mathcal{C} := \{ \mathcal{O}(t) \subseteq \mathcal{O} \mid \Theta(t) = \text{const.}, \mathbf{s}(t) = \text{const.}, t \in [t_n, t_n + T_p] \}, \quad (4.7)$$

$$\mathcal{D} := \{ \mathcal{O}(t) \subseteq \mathcal{O} \mid \Theta(t), \mathbf{s}(t) \}, \quad (4.8)$$

$$\mathcal{R} := \{ \mathcal{O}(\mathbf{x}) \subseteq \mathcal{O} \mid \Theta(\mathbf{x}), \mathbf{s}(\mathbf{x}) \}. \quad (4.9)$$

In  $\mathcal{C}$  befinden sich Hüllkörper von Hindernissen, die innerhalb der Dauer  $T_p$  des  $n$ -ten Planungszyklus statisch sind. Für  $\mathcal{D}$  entfällt diese Beschränkung und die Menge umfasst Hüllkörper von Hindernissen, die sich auch während der Planung bewegen. Die Hüllkörper des Roboters sind in Abhängigkeit der Gelenkwinkel  $\mathbf{x}$  in der Menge  $\mathcal{R}$  zusammengefasst. Rotation und Translation werden darin über die Vorwärtskinematik  $\mathbf{T} : \mathbb{R}^N \rightarrow \text{SE}(3)$  berechnet:

$$\mathbf{T}(\mathbf{x}) = \begin{pmatrix} \Theta(\mathbf{x}) & \mathbf{s}(\mathbf{x}) \\ \mathbf{0} & 1 \end{pmatrix} := \mathbf{T}_B \prod_{i=1}^N [\mathbf{T}_i(x_i)] \mathbf{T}_E, \quad (4.10)$$

$$\mathbf{x}^T := (x_1 \quad x_2 \quad \dots \quad x_i \quad \dots \quad x_N),$$

mit den Teiltransformationen  $\mathbf{T}_i : \mathbb{R} \rightarrow \text{SE}(3)$  und den homogenen Matrizen  $\mathbf{T}_B$  und  $\mathbf{T}_E$  für die Lage der Roboterbasis im Raum bzw. des Endeffektors im letzten Armkörper. Die genauen Angaben zur Modellierung und relativen Positionierung der in dieser Arbeit verwendeten Robotersysteme befinden sich in Anhang A. Aus Sicht der Planung ist jeder Hüllkörper ein Hindernis, weshalb beide Begriffe synonym verwendet werden.

## 4.2. Distanzfunktionen für sphärische Volumenkörper

Für die abstands-basierte Kollisionsvermeidung wird die vorzeichen-behaftete Distanz, im Folgenden nur Distanz genannt, zwischen zwei Hüllkörpern  $\mathcal{O}_1$  und  $\mathcal{O}_2$  über eine Distanzfunktion  $d : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$  berechnet:

$$d(\mathcal{O}_1, \mathcal{O}_2) := d_S(\mathcal{O}_1, \mathcal{O}_2) - d_P(\mathcal{O}_1, \mathcal{O}_2). \quad (4.11)$$

Sie besteht aus der Separations- und Penetrationsdistanz  $d_S, d_P : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$ :

$$d_S(\mathcal{O}_1, \mathcal{O}_2) := \arg \min_{\mathbf{s}} \|\mathbf{s}\| \text{ unter der Nebenbedingung } (\mathcal{O}_1 + \mathbf{s}) \cap \mathcal{O}_2 \neq \emptyset, \quad (4.12)$$

$$d_P(\mathcal{O}_1, \mathcal{O}_2) := \arg \min_{\mathbf{s}} \|\mathbf{s}\| \text{ unter der Nebenbedingung } (\mathcal{O}_1 + \mathbf{s}) \cap \mathcal{O}_2 = \emptyset, \quad (4.13)$$

welche als der kürzeste Abstand zwischen  $\mathcal{O}_1$  und  $\mathcal{O}_2$  bzw. als die kürzeste Entfernung zur Auflösung einer Kollision definiert sind. Die Varianten für SSVs basieren im Wesentlichen auf den Distanzen zwischen den zugrundeliegenden Primitiven<sup>1</sup>:

$$d_S(\mathcal{O}_1, \mathcal{O}_2) = \begin{cases} d_S(\mathcal{S}_1, \mathcal{S}_2) - (r_1 + r_2) & \text{wenn } d_S(\mathcal{S}_1, \mathcal{S}_2) > r_1 + r_2 \\ 0 & \text{sonst} \end{cases}, \quad (4.14)$$

$$d_P(\mathcal{O}_1, \mathcal{O}_2) = \begin{cases} r_1 + r_2 - d_S(\mathcal{S}_1, \mathcal{S}_2) & \text{wenn } 0 < d_S(\mathcal{S}_1, \mathcal{S}_2) < r_1 + r_2 \\ 0 & \text{wenn } d_S(\mathcal{S}_1, \mathcal{S}_2) > r_1 + r_2 \\ d_P(\mathcal{S}_1, \mathcal{S}_2) + r_1 + r_2 & \text{wenn } d_S(\mathcal{S}_1, \mathcal{S}_2) = 0 \end{cases}. \quad (4.15)$$

Werden beide Funktionen wie in Gleichung (4.11) zusammengefasst ergibt sich:

$$d(\mathcal{O}_1, \mathcal{O}_2) = \begin{cases} d_S(\mathcal{S}_1, \mathcal{S}_2) - (r_1 + r_2) & \text{wenn } d_S(\mathcal{S}_1, \mathcal{S}_2) > 0 \\ -d_P(\mathcal{S}_1, \mathcal{S}_2) - (r_1 + r_2) & \text{wenn } d_S(\mathcal{S}_1, \mathcal{S}_2) = 0 \end{cases}. \quad (4.16)$$

Zur Berechnung der Separationsdistanz  $d_S(\cdot)$  bieten sich spezialisierte primitivbasierte Distanzfunktionen wie die von Lumelsky [Lum85] und Larsen u. a. [Lar+99] an. Die Berechnung der Penetrationsdistanz  $d_P(\cdot)$  kann hingegen nicht mit den genannten Algorithmen erfolgen und wird häufig mit 0 approximiert. Auf eine geeignetere Approximation wird im nächsten Abschnitt eingegangen.

### Spezialfall bei sich schneidenden Primitiven

Es wird folgendes eindimensionales NLP betrachtet, dessen Ziel die Maximierung des Abstands zwischen zwei Hüllkörpern der Form  $\mathcal{S}_L$  ist:

$$\min_{s_x} -d(\mathcal{O}_1, \mathcal{O}_2(s_x)), \quad (4.17a)$$

unter den Nebenbedingungen:

$$s_x - 1 \leq 0 \text{ und } -s_x - 1 \leq 0. \quad (4.17b)$$

Abbildung 4.2 zeigt in der linken Hälfte den initialen Aufbau mit  $s_x = 0$  und in der rechten Hälfte eine der beiden optimalen Lösungen bei  $s_x^* = 1$ . Die Optimierungsvariable ist die absolute

<sup>1</sup>Mit  $\mathcal{K}_0$  lässt sich die Distanzfunktion  $d(\cdot)$  auch direkt auf Primitive  $\mathcal{S}$  anwenden.

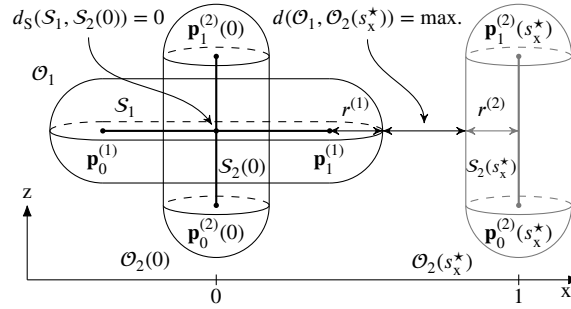


Abbildung 4.2.: Anfangszustand (links) und optimale Lösung (rechts) des Separationsproblems zweier Hüllkörper  $S_L$  im Schnittpunkt ihrer Primitive.

Translation  $s_x$  des senkrechten Hüllkörpers  $\mathcal{O}_2(s_x)$  in x-Richtung. Der Einfachheit halber wird auf weitere Kostenterme und Nebenbedingungen verzichtet. Weitere Details befinden sich in Anhang B.2.1.

Die Lösung von NLPs dieser Art umfasst, wie in Kapitel 3 erläutert, unter anderem die Ableitung der Kostenfunktion:

$$-\frac{\partial d(\mathcal{O}_1, \mathcal{O}_2(s_x))}{\partial s_x} = \begin{cases} -\frac{\partial d_S(S_1, S_2(s_x))}{\partial s_x} & \text{wenn } d_S(S_1, S_2(s_x)) > 0 \\ \frac{\partial d_P(S_1, S_2(s_x))}{\partial s_x} & \text{wenn } d_S(S_1, S_2(s_x)) = 0 \end{cases} \quad (4.18)$$

Die fehlende Penetrationsdistanz  $d_P(\cdot)$  sorgt dafür, dass im Schnittpunkt  $d_S(S_1, S_2(s_x)) = 0$  zweier Primitive die Distanz in Gleichung (4.16) konstant und Gleichung (4.18) Null ist. Dies hat zur Folge, dass das Lösungsverfahren keine Suchrichtung und damit keinen Lösungskandidaten ermitteln kann, der den Wert der Zielfunktion verbessert. Das Lösungsverfahren scheitert aufgrund mangelnder Verbesserungen in der Zielfunktion nach einigen Iterationen und erzielt keine Veränderung. Hüllkörper  $\mathcal{O}_2(s_x)$  verbleibt bei  $s_x = 0$  und damit in einer Kollision. Wird dies auf die Trajektorienoptimierung übertragen, bedeutet das den Verlust der Fähigkeit eine Kollision bei  $d_S(S_1, S_2) = 0$  in der Trajektorie aufzulösen. Die Relevanz dieses Problems wird stärker, je größer die potentielle Menge an Schnittpunkten der beteiligten Primitive ist. Das Problem tritt im Falle von  $(S_P, S_L)$  seltener auf als bei  $(S_L, S_L)$  und für  $(S_L, S_F)$  am häufigsten. Im Folgenden wird daher eine approximierten Penetrationsdistanz  $\tilde{d}_P(\cdot)$  eingeführt.

Die grundsätzliche Idee hinter den Algorithmen von Lumelsky [Lum85] und Larsen u. a. [Lar+99] zur Berechnung von  $d_S(\cdot)$  ist, die Endpunkte der kürzesten Verbindung zu bestimmen, mit denen die Distanzberechnung trivial ist. Hierfür werden zunächst die Segmentparameter  $v_i$  für  $i = 1, 2, \dots$  (maximal  $i = 4$  bei  $(S_F, S_F)$ ) beider Hüllkörper berechnet. Die approximierten Penetrationsdistanz nutzt diese Parameter, um im Falle von  $d_S(S_1, S_2) = 0$  eine Pseudodistanz zu bestimmen, welche einen negativen Gradienten in Richtung der Segmentenden erzeugt:

$$\tilde{d}_P(S_1, S_2) := \sum_{\forall v_i} 4(v_i - v_i^2) \tilde{d}_P^+ \text{ mit } \tilde{d}_P^+ \in \mathbb{R}^+. \quad (4.19)$$

Die Pseudodistanzen approximieren die Penetrationsdistanz und nehmen zur Mitte des Primitives ( $v_i = 0,5$ ) den Maximalwert  $\tilde{d}_P^+$  an, während sie zu den Rändern ( $v_i \in \{0, 1\}$ ) zu Null werden. Abbildung 4.3 zeigt den Vergleich zwischen der unveränderten und der modifizierten Distanzfunktion in Abhängigkeit der x-Position  $s_x$  des zweiten Hüllkörpers aus dem Beispiel in Abbildung 4.2. Wie zu erkennen ist, wird das konstante Plateau während des Schnitts beider Primitive vermieden. Mit  $\tilde{d}_P(\cdot)$  anstelle von  $d_P(\cdot)$  in Gleichung (4.16) ergibt sich in Gleichung (4.18) durch die implizite Abhängigkeit der Segmentparameter  $v_i$  von der Lage der Hüllkörper auch bei  $d_S(S_1, S_2) = 0$  ein Gradienten zur Auflösung der Kollision.

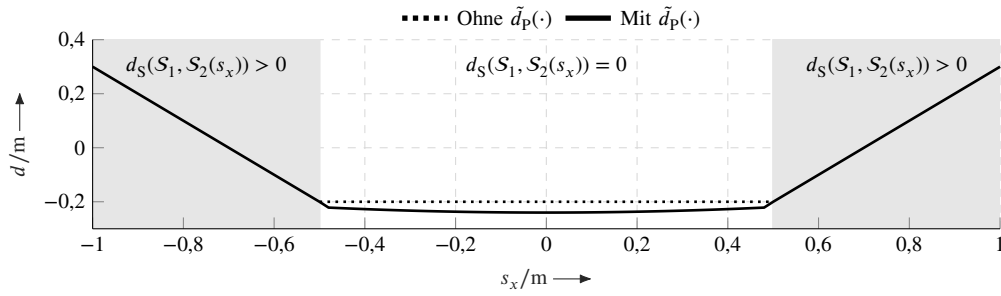


Abbildung 4.3.: Distanz zwischen beiden Hüllkörpern in Abhängigkeit der Position  $s_x$  für  $\tilde{d}_p^+ = 0,02$  mit und ohne approximierter Penetrationsdistanz.

## Vergleich der Berechnungszeit verschiedener Implementierungen

Da die Distanzberechnung einen wesentlichen Anteil der Berechnungszeit bei der Optimierung einnimmt, liegt ein besonderer Fokus auf die effiziente Implementierung der primitivbasierten Distanzfunktionen von Lumelsky [Lum85] und Larsen u. a. [Lar+99]. Darüber hinaus gilt es alle relevanten Fälle abzudecken, um vielseitige Szenen zu ermöglichen. Dies sind Nachteile von existierenden Sammlungen wie GTE oder FCL, die oftmals Aufwand für die Umwandlung der Datentypen in ein unterstütztes Format erfordern oder die benötigten Fälle unvollständig unterstützen. Aus diesem Grund ergänzt eine eigene Implementierung der Algorithmen von Lumelsky [Lum85] und Larsen u. a. [Lar+99] die Möglichkeiten von GTE und FCL, welche im Folgenden bewertet werden. Alle in dieser Arbeit erfolgten Implementierungen wurden in C++ umgesetzt.

Tabelle 4.1 zeigt die Berechnungszeiten jeweils auf die mittlere Dauer der eigenen Implementierung normiert und mit Angabe des Mittelwerts in Klammern. Es werden für jeden Fall und Algorithmus 100 000 Wiederholungen ausgeführt, wobei ein Hüllkörper der Paare jeweils über alle Wiederholungen fixiert ist, während der andere über gleichverteilte Posen zufällig bestimmt wird. Weitere Details befinden sich in Anhang B.2.2. Da die Roboter in dieser Arbeit (siehe Anhang A) ausschließlich über  $S_p$  und  $S_L$  approximiert werden, werden nur Paare berücksichtigt, in denen mindestens eins dieser Primitive vorkommt. Die Berechnungszeiten von GTE verdeutlichen den zusätzlichen Aufwand bei der Umwandlung der Datenstrukturen zwischen 7 % und 31 % zur eigenen Implementierung. FCL verwendet mit der Softwarebibliothek *Eigen* [Gue+21] bereits dieselben Datenstrukturen wie die Implementierungen dieser Arbeit und nutzt in den Fällen  $(S_p, S_L)$  und  $(S_L, S_L)$  die Stärken einer über Jahre optimierten Implementierung aus. Sie benötigt nur 71 % bzw. 85 % der Zeit der eigenen Implementierungen. Für den Fall  $(S_p, S_F)$  überführt FCL das Rechteck in zwei Dreiecke, was mehr als viermal so lange dauert wie mit dem Algorithmus von Lumelsky [Lum85] und den Vorteil spezialisierter Distanzfunktionen unter-

Tabelle 4.1.: Normierte Berechnungszeiten der Separationsdistanz verschiedener Implementierungen. Die schnellsten Varianten sind fettgedruckt.

	$(S_p, S_p)$	$(S_p, S_L)$	$(S_p, S_F)$	$(S_p, S_E)$	$(S_L, S_L)$	$(S_L, S_F)$	$(S_L, S_E)$
Eigene	<b>1,0</b> (16 ns)	1,0 (28 ns)	<b>1,0</b> (39 ns)	<b>1,0</b> (21 ns)	1,0 (39 ns)	• <sup>b</sup>	<b>1,0</b> (25 ns)
GTE	• <sup>a</sup>	1,07	1,21	• <sup>a</sup>	1,31	<b>198 ns</b>	• <sup>a</sup>
FCL	• <sup>a</sup>	<b>0,71</b>	4,15	• <sup>a</sup>	<b>0,85</b>	• <sup>b</sup>	• <sup>a</sup>

<sup>a</sup> Der Fall ist trivial und erfordert keine Frameworks.

<sup>b</sup> Nicht unterstützter Fall

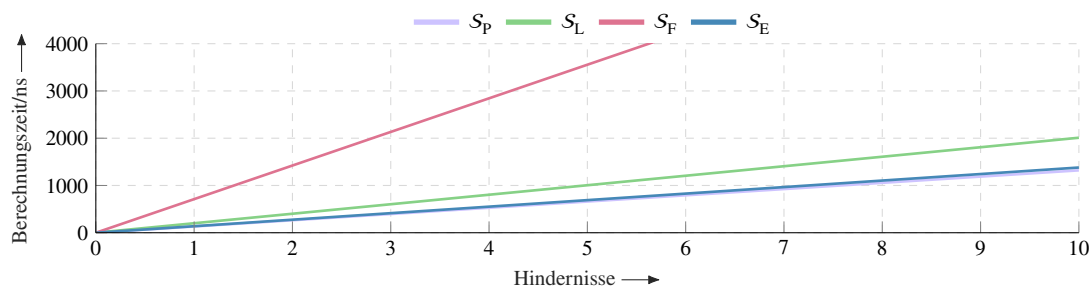


Abbildung 4.4.: Aufwand der Distanzberechnungen aller Kombinationen zwischen UR10 und verschiedenen Formen von Hüllkörpern von Hindernissen.

streicht. Anzumerken ist die verhältnismäßig hohe Berechnungszeit für den Fall  $(S_L, S_F)$ , der damit das komplexeste Hüllkörperpaar darstellt. Für die MHP gelten im weiteren Verlauf die Fälle gemäß der jeweils schnellsten Implementierung. Da die Pseudodistanz aus dem vorherigen Abschnitt nur auf den Segmentparametern  $v_i$  basiert, lässt sie sich in alle primitivbasierten Ansätze integrieren.

Die trivialen Fälle  $(S_P, S_P)$  und  $(S_P, S_E)$  erfordern keine Frameworks, da sie in wenigen Zeilen implementiert sind und keine unterschiedlichen Herangehensweisen begünstigen. Dies gilt auch für den Fall  $(S_L, S_E)$ . Obwohl sich  $S_E$  formal kaum von  $S_F$  unterscheidet, vereinfacht die Tatsache, dass  $S_E$  unendlich ausgedehnt ist und der Roboter sich damit immer nur auf einer Halbebene befinden darf, die Berechnung.

Betrachtet man  $\mathcal{R}$  eines bestimmten Roboters, so lässt sich unter Verwendung von Tabelle 4.1 für jede mögliche Hüllkörperform eines Hindernisses ein theoretischer Gesamtaufwand aus allen Kombinationen  $\mathcal{R} \times \mathcal{C}$  berechnen. Wird die Anzahl der Hindernisse skaliert ergibt sich eine Übersicht, mit dessen Hilfe Substitutionen von Hüllkörperformen bewertet werden können. Abbildung 4.4 demonstriert das für den Roboter UR10<sup>2</sup> (siehe Anhang A.1). Die Formen  $S_P$  und  $S_E$  skalieren annähernd gleich, wobei ab zwei  $S_P$  (264 ns) die Nutzung eines  $S_L$  (201 ns) günstiger ist, sofern es die Modellierung der Hindernisse zulässt. Die Form  $S_F$  verursacht mit Abstand den höchsten Anstieg der Berechnungszeit, ist jedoch mit 711 ns ab vier  $S_L$  (804 ns) die bessere Wahl. Abbildung 4.4 dient zusätzlich zur Projektion von Versuchsergebnissen auf andere Hinderniskonstellationen.

### 4.3. Nebenbedingungen und Annäherungskosten zur Kollisionsvermeidung

Nachdem eine effiziente Berechnung der Separations- und Penetrationsdistanzen realisiert ist, werden sie jetzt über Nebenbedingungen und einen Kostenterm für die distanzbasierte Kollisionsvermeidung bei statischen Hindernissen verwendet.

#### Kollisionsvermeidung über Nebenbedingungen

Die Methoden zur Kollisionsvermeidung bei der MHP dienen dem unterbrechungsfreien Ablauf und damit der Vermeidung des Auslösens von Sicherheitsfunktionen, welche den Roboter und die Planung stoppen. Eine solche Sicherheitsfunktion überwacht permanent den Abstand zwischen

<sup>2</sup>Universal Robot 10

Hüllkörpern und löst beim Unterschreiten eines Mindestabstands  $d^- \in \mathbb{R}^+$  einen Not-Halt aus. Der Mindestabstand  $d^-$  entspricht dabei dem Bremsweg des Roboters unter Annahme einer höheren Maximalbeschleunigung, als sie für den Normalbetrieb der MHP zur Verfügung steht. Bezogen auf ein Gelenk ergibt sich z. B.:

$$d^- = \frac{u^+}{2} d_{\text{eff}} T_B, \quad (4.20)$$

mit dem effektiven Abstand  $d_{\text{eff}} \in \mathbb{R}^+$  zwischen Drehachse und Kollisionspunkt, der Bremszeit  $T_B \in \mathbb{R}^+$  sowie der maximalen Gelenkwinkelgeschwindigkeit  $u^+ \in \mathbb{R}$  des betrachteten Gelenks. Um der Planung ein Bewusstsein über Hindernisse und die Existenz der invasiven Sicherheitsfunktion zu geben, werden Nebenbedingungen aufgestellt, die einen Mindestabstand oberhalb von  $d^-$  fordern. Zusammen mit der minimalen Horizontlänge  $K^-$  ermöglicht dies der Planung statische Hindernisse rechtzeitig wahrzunehmen und im Rahmen der Planungskriterien komfortabel vor ihnen anzuhalten oder sie zu umfahren. Die Verletzung eines Schwellwerts bedeutet demnach nicht zwangsläufig direkt eine reale Kollision, wird im weiteren Verlauf der Einfachheit halber aber weiterhin so bezeichnet. Über die Nebenbedingung wird die Kollisionsfreiheit an die Gültigkeit einer Lösung geknüpft. Distanzbasierte Realisierungen sind dabei die verbreitetste Methode Nebenbedingungen zur Separation von Hüllkörpern aufzustellen und werden auch in dieser Arbeit verfolgt. Die Ungleichheitsnebenbedingungen  $\mathbf{h}(\cdot)$  der NLPs aus Kapitel 3 werden entsprechend ergänzt:

$$\mathbf{h}(\mathbf{z}) := \left\{ \begin{array}{l} \vdots \\ (3.17b), (3.24d) \text{ oder } (3.29b) \\ \vdots \\ \hline d_{\text{R}}^- - d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2(\mathbf{x}_k)) \quad k = 1, 2, \dots, K-1 \\ \vdots \\ \forall (\mathcal{O}_1(\cdot), \mathcal{O}_2(\cdot)) \in \mathcal{R} \times \mathcal{R} \\ \hline d_{\text{C}}^- - d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2) \quad k = 1, 2, \dots, K-1 \\ \vdots \\ \forall (\mathcal{O}_1(\cdot), \mathcal{O}_2) \in \mathcal{R} \times \mathcal{C} \end{array} \right. \leq \mathbf{0}. \quad (4.21)$$

Die Schwellwerte  $d_{\text{R}}^- > d^-$  und  $d_{\text{C}}^- > d^-$  erlauben eine individuelle Anpassung je nach Hindernisklasse und sind für die Versuche in den Begleitinformationen in Anhang B angegeben.

Durch die Kollisionsvermeidung erhöht sich die Anzahl an Ungleichheitsbedingungen unabhängig von der Transkriptionsmethode auf  $N_{\text{U}} = 6N(K-1) + N_{\text{R}} + N_{\text{C}}$ . Darin sind  $N_{\text{R}} = N_{\text{RR}}(K-1)$  und  $N_{\text{C}} = N_{\text{RC}}(K-1)$  sowie  $N_{\text{RR}} := |\mathcal{R} \times \mathcal{R}|$  und  $N_{\text{RC}} := |\mathcal{R} \times \mathcal{C}|$  die Anzahl der Kollisionspaare der jeweiligen Kategorie. Für  $N_{\text{RR}}$  sind üblicherweise nicht alle Kombinationen  $|\mathcal{R}| \cdot |\mathcal{R}|$  relevant, da nicht alle Armkörper miteinander kollidieren können. Bei  $N_{\text{RC}}$  wird durch Betrachtung aller Kombinationen  $|\mathcal{R}| \cdot |\mathcal{C}|$  die Anzahl an Ungleichheitsnebenbedingungen  $N_{\text{U}}$  z. B. für den UR10 ( $N = 6$ ,  $N_{\text{RR}} = 8$  und  $K = 31$ ) bereits ab 5 Hindernissen von der Kollisionsvermeidung dominiert, da für jedes Hindernis  $|\mathcal{R}| = 6$  Kombinationen hinzukommen. Abbildung 4.5 vergleicht die Anzahl der Ungleichheitsnebenbedingungen mit und ohne Kollisionsvermeidung für den genannten Fall. Die konstante Verschiebung bei 0 Hindernissen ergibt sich durch  $N_{\text{R}} = 240$ . Werden anstelle von  $|\mathcal{R}| \cdot |\mathcal{C}|$  Kombinationen nur die jeweilige Kombination mit dem kleinsten Abstand berücksichtigt, müssen nur  $N_{\text{RC}} = 1 \cdot |\mathcal{C}|$  Fälle betrachtet werden. Der Schnittpunkt verschiebt sich auf 28 Hindernisse bzw. verringert sich  $N_{\text{U}}$  von 2220 auf 1470. Auf die gleiche

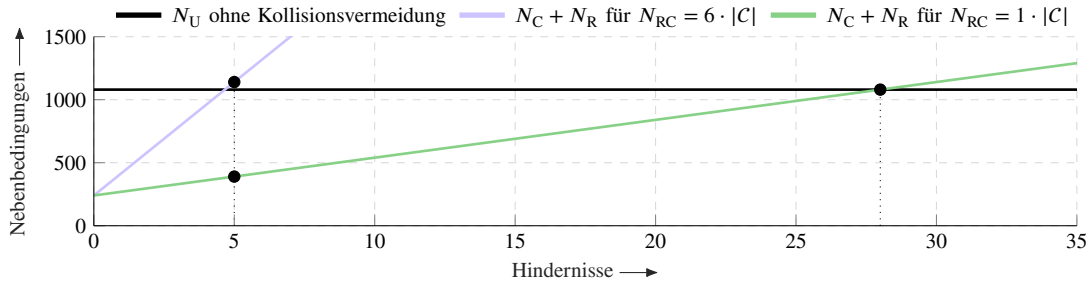


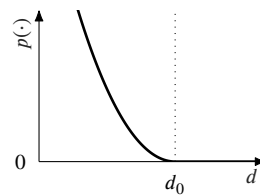
Abbildung 4.5.: Vergleich der Anzahl an Ungleichheitsnebenbedingungen ohne Kollisionsvermeidung mit denen, die durch Betrachtung aller Distanzpaare bzw. nur des kritischen Paares durch die Kollisionsvermeidung hinzukommen.

Weise lässt sich auch  $N_{RR}$  weiter auf 1 reduzieren, was aber aufgrund der bereits reduzierbaren Potenzmenge  $|\mathcal{R} \times \mathcal{R}|$  nicht verfolgt wird.

### Annäherungskosten zur Bewertung der Nähe zwischen Hüllkörpern

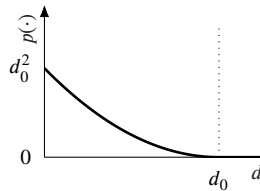
Bisher wird über die Nebenbedingungen ein Mechanismus zur Berücksichtigung von Mindestabständen realisiert. Ohne Weiteres wird die Planung diese Mindestabstände ausnutzen und Lösungen bis an deren Rand optimieren, was angesichts von Ungenauigkeiten der Umgebungserfassung zu Überschreitungen und extremen Reaktionen des Lösungsverfahrens führen kann [Krä+20b]. Um dies zu entschärfen, wird mit Annäherungskosten ein abstoßender Effekt zwischen Hüllkörpern erzeugt, sobald ein Aktivierungsschwellwert  $d_0 \in \mathbb{R}^+$  unterschritten wird. Auf diese Weise wird dem NLP die Nähe zu Hindernissen auf einem größeren Intervall bewusst. Nachfolgend sind mehrere bekannte Potentialfunktionen  $p_{[1]} : \mathbb{R} \rightarrow \mathbb{R}$  definiert, die hinsichtlich der Verwendung als Annäherungskosten diskutiert werden:

$$p_{[1]}(d) := \begin{cases} \frac{1}{2} \left( \frac{1}{d} - \frac{1}{d_0} \right)^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases}$$



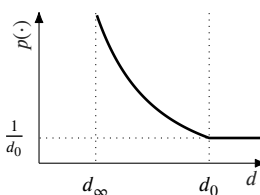
[Kha85], (4.22)

$$p_{[2]}(d) := \begin{cases} (d - d_0)^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases}$$



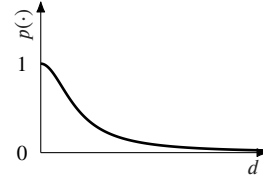
[Moh+95], (4.23)

$$p_{[3]}(d) := \begin{cases} \infty & d \leq d_\infty \\ \frac{1}{d} & d_\infty < d < d_0 \\ \frac{1}{d_0} & d > d_0 \end{cases}$$



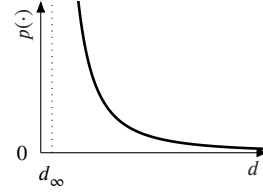
[Bal+06], (4.24)

$$p_{[4]}(d) := \frac{1}{d^2 + 1}$$



$$[\text{Tea+14}], \quad (4.25)$$

$$p_{[5]}(d) := \frac{1}{d^2 - d_\infty^2}$$



$$[\text{Zub15}]. \quad (4.26)$$

Die Distanz  $d(\cdot)$  zwischen zwei Hüllkörpern ist zur besseren Lesbarkeit mit  $d$  abgekürzt. Der Schwellwert  $d_\infty < d_0$  markiert, soweit verfügbar, den Punkt ab dem die Potentialfunktion gegen  $\infty$  strebt.

Die Potentiale  $p_{[3]}(\cdot)$ ,  $p_{[4]}(\cdot)$  und  $p_{[5]}(\cdot)$  sind als Kostenterme ungeeignet, da der Funktionswert auch für unkritische Abstände ( $d \gg 0$  und  $d \gg d_0$ ) nie den Wert 0 annimmt und es dadurch zu Kompromissen bei der Optimierung kommt. Je nach relativer Gewichtung zwischen dem Potential und dem Kostenterm (3.2) stellt sich eine Kompromisslösung ein und der Zielzustand  $\mathbf{x}_f$  wird nicht erreicht. Das Potential  $p_{[1]}(\cdot)$  ist für Abstände oberhalb des Schwellwerts  $d_0$  gleich 0, strebt allerdings wie  $p_{[3]}(\cdot)$  gegen unendlich für  $d \rightarrow 0$ . Selbiges Problem hat  $p_{[5]}(\cdot)$  für  $d \rightarrow d_\infty$ . Unendlich hohe Kosten verleihen dem Kostenterm den Charakter einer harten Nebenbedingung. Lösungsverfahren nutzen diesen Effekt zur Berücksichtigung von Ungleichheitsnebenbedingungen (vgl. Barrierefunktion in (3.38)). Wenn die Lösung eines NLP eine Bedingung, wie den nicht zu unterschreitenden Abstand zu einem Hindernis, einhalten soll, sind dafür die Schnittstellen für harte Nebenbedingungen wie im Abschnitt zuvor zu präferieren. Sie bieten den Vorteil, dass das Lösungsverfahren zielgerichtet und damit effizienter arbeitet, da es die Absichten des Problems präziser erfasst. Da Potentialfunktionen in ihrer traditionellen Verwendung üblicherweise nicht mit negativen Distanzen umgehen müssen, zeigt sich bei  $p_{[1]}(\cdot)$ ,  $p_{[3]}(\cdot)$  und  $p_{[5]}(\cdot)$  das Problem, dass der Funktionswert für  $d < 0$  wieder abnimmt, was eine Trajektorie schließlich weiter in eine Kollision hineinführt. Damit besitzt lediglich  $p_{[2]}(\cdot)$  geeignete Eigenschaften zur Verwendung als Kostenterm. Darauf aufbauend wird das folgende Potential  $p : \mathbb{R} \rightarrow \mathbb{R}^+$  definiert, welches die Abhängigkeit bei  $d = 0$  vom Schwellwert  $d_0$  löst:

$$p(d) := \begin{cases} \left(\frac{d}{d_0} - 1\right)^2 & d \leq d_0 \\ 0 & d > d_0 \end{cases}. \quad (4.27)$$

Dies ermöglicht neben einer einfacheren Gewichtung eine flexible Wahl von Schwellwerten, z. B. so, dass sich der Roboter wegen der Vermeidung von Selbstkollisionen nicht unnötig einschränkt oder Hindernisse potentielle Zielzustände in ihrer Nähe nicht verbieten.

Der Kostenterm  $c_H : \mathbb{R}^N \times \mathbb{R} \rightarrow \mathbb{R}_0^+$  realisiert die Annäherungskosten und besteht aus den über  $W_R \in \mathbb{R}^+$  und  $W_C \in \mathbb{R}^+$  gewichteten Potentialsummen der verschiedenen Hüllkörperklassen:

$$c_H(\mathbf{x}_k, k) := W_R \sum_{(\mathcal{O}_1, \mathcal{O}_2) \in \mathcal{R} \times \mathcal{R}} p(d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2(\mathbf{x}_k))) + W_C \sum_{(\mathcal{O}_1, \mathcal{O}_2) \in \mathcal{R} \times \mathcal{C}} p(d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2)). \quad (4.28)$$

Die Potenzmenge  $\mathcal{R} \times \mathcal{R}$  umfasst dabei, wie schon bei den Nebenbedingungen, nur relevante Paare, die tatsächlich kollidieren können. Es werden sowohl Selbstkollisionen des Roboters, als auch solche mit statischen Hindernissen betrachtet. Zur besseren Parametrierung wird der

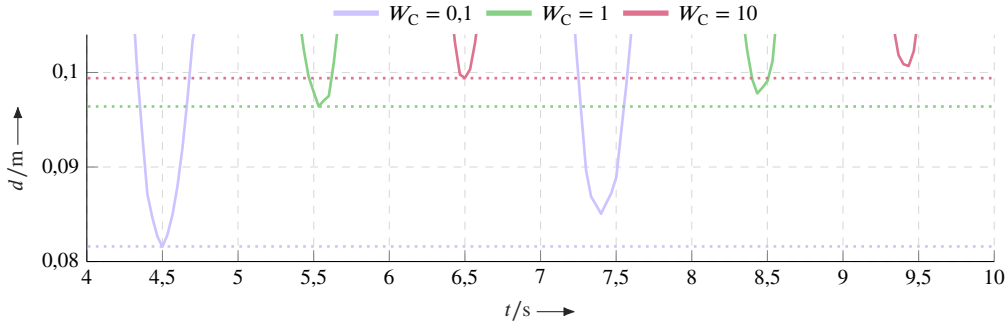


Abbildung 4.6.: Zeitlicher Verlauf des kleinsten Abstands zwischen Roboter und Hindernissen für unterschiedliche Gewichte  $W_C$  und konstantes  $Q = 1$  sowie  $R = 0$ . Die jeweils erreichte minimale Distanz für eine Bewegung ist gestrichelt eingezeichnet.

Aktivierungsschwellwert  $d_0$  für beide Varianten mit  $d_R > d_R^-$  und  $d_C > d_C^-$  gesondert definiert. Kostenterm (4.28) wird als weiterer Summand zu den laufenden Kosten  $\hat{J}(\cdot)$  aus Gleichung (3.16) aller NLPs addiert.

## Einfluss der Gewichte

Stören Hindernisse die Bewegung, konkurrieren die Annäherungskosten (4.28) mit den Zielkosten (3.2) sowie den Bewegungskosten (3.3) und es ergibt sich ein gewichteter Kompromiss zwischen Zielnähe, Bewegung und Hindernisnähe. Gewicht  $W_C$  und Schwellwert  $d_C$  haben demnach einen Einfluss auf das Ausweichverhalten. Die MHP plant zur Untersuchung die Bewegung des UR10, die durch ein Hindernis der Form  $S_p$  in der Nähe des Startzustands und eines in der Nähe des Zielzustands gestört wird. Der Aktivierungsschwellwert liegt bei  $d_C = 0,1$  m und die Nebenbedingungen zur Kollisionsvermeidung sind inaktiv. Weitere Details zum Versuchsaufbau befinden sich in Anhang B.2.3.

Abbildung 4.6 stellt den Verlauf des kleinsten Abstands zwischen Roboter und Hindernissen für verschiedene Gewichte  $W_C$  dar. Die übrigen Gewichte sind  $Q = 1$  und  $R = 0$ . Zur besseren Lesbarkeit sind die Verläufe jeweils um eine Sekunde versetzt. Anhand der gestrichelten Hilfslinien, die den jeweils kleinsten Abstand der Bewegung angeben, ist erkennbar, wie die Abstände beim ersten Hindernis konsistent kleiner ausfallen als beim zweiten Hindernis. Der Grund dafür ist, dass zu Beginn der Planung die Zielkosten (3.2) am größten sind und die Annäherung an den Zielzustand den Wert der Kostenfunktion stärker senkt als ihn die Potentiale durch einen geringeren Abstand erhöhen. Dieser Effekt relativiert sich zum einen durch sinkende Zielkosten, wenn der Roboter dem Zielzustand näher kommt und zum anderen durch das Gewichtsverhältnis  $\frac{W_C}{Q}$ . Je größer das Verhältnis, desto geringer die Überschreitungen des Schwellwerts  $d_C$ . Da die Kostenterme (3.2) und (3.3) beschränkt sind, wird ein  $W_C$  für welches die Annäherungskosten bei  $d = \frac{d_C}{2}$  stets größer sind, wie folgt abgeschätzt:

$$p \left( \frac{d_C}{2} \right) = \frac{W_C}{4} > 4Q\mathbf{x}^+ \cdot \mathbf{x}^+ + R\mathbf{u}^+ \cdot \mathbf{u}^+. \quad (4.29)$$

Da die Potentiale nur lokal wirken, sind hohe Werte für  $W_C$  unter Beachtung der Konditionierung<sup>3</sup> des NLP unkritisch. Die Gewichte der Annäherungskosten sind für die weiteren Versuche in

<sup>3</sup>Werte für Kosten und Nebenbedingungen sollten in einer Größenordnung liegen, um negative Effekte aufgrund der begrenzten Genauigkeit von Dezimalzahlen zu vermeiden.

den Begleitinformationen in Anhang B angegeben. Alternativ zur Erhöhung von  $W_C$ , lassen sich sämtliche Kosten vor der Gewichtung über Sättigungsfunktionen wie bei Fishman u. a. [Fis+20] in das Einheitsintervall abbilden.

### Berechnungszeiten von Kostenfunktion und Nebenbedingung

Zur Lösung des NLP werden Kostenfunktion und Nebenbedingungen mehrfach ausgewertet, womit die Dauer der Optimierung wesentlich vom Aufwand der Funktionsaufrufe abhängt. Abbildung 4.7 vergleicht die jeweils über 10 000 Wiederholungen gemittelten Berechnungszeiten für Kostenfunktion  $\hat{J}(\cdot)$  und Nebenbedingungen  $\mathbf{h}(\cdot)$  in zwei Szenarien für den UR10. Im ersten Szenario wird für eine feste Anzahl von  $|C| = 10$  Hüllkörpern ( $S_p$ ) der Einfluss der Horizontlänge  $K$  evaluiert. Das zweite Szenario bewertet für eine feste Horizontlänge  $K = 30$  den Einfluss der Anzahl  $|C|$ . Die Selbstkollisionsvermeidung ist in beiden Szenarien aktiv. Aus den Messwerten wird jeweils ein lineares Modell angenähert, um einen Trend abzuleiten. Weitere Informationen zum Aufbau und den Modellparametern befinden sich in Anhang B.2.4.

Abbildung 4.7a zeigt, wie der Aufwand für jeden Schritt auf dem Horizont in der Kostenfunktion mit  $4,77 \mu\text{s}$  annähernd genauso stark steigt, wie bei den Nebenbedingungen mit  $4,60 \mu\text{s}$ . Durch die Distanzfunktionen skalieren Kostenfunktion und Nebenbedingungen aus Sicht der Hinderniszahl  $|C|$  mit  $11,41 \mu\text{s}$  bzw.  $11,73 \mu\text{s}$  ebenfalls annähernd gleich pro Hindernis, wie Abbildung 4.7b verdeutlicht. Dabei verursacht ein weiteres Hindernis für Kostenfunktion und Nebenbedingungen soviel Mehraufwand wie 2,39 bzw. 2,55 Horizontschritte. Eine Verringerung der Hindernisse zahlt sich demnach mehr aus als eine Verkürzung des Horizonts.

Die annähernd gleichen Trends zwischen Kostenfunktion und Nebenbedingungen deuten darauf hin, dass der maßgebliche Aufwand von identischen Komponenten herrührt. Abbildung 4.8 veranschaulicht die Aufteilung der über 10 000 Wiederholungen gemittelten Berechnungszeiten von Kostenfunktion  $\hat{J}(\cdot)$  und Nebenbedingungen  $\mathbf{h}(\cdot)$  bei reiner Selbstkollisionsvermeidung. Der wesentliche Aufwand der Kostenfunktion in Abbildung 4.8a setzt sich aus der Vorwärtskinematik (4.10) und den Distanzfunktionen (4.16) mit 38,7 % bzw. 45,1 % zusammen. Die quadratischen Kostenterme (3.2) und (3.3) sowie die Potentialfunktion zur Berechnung der Annäherungskosten rücken mit je 6,7 % bzw. 2,9 % in den Hintergrund. Der Aufwand der Ungleichheitsnebenbedingungen in Abbildung 4.8b besteht ebenfalls im Wesentlichen aus der Vorwärtskinematik (47,6 %) und den Distanzfunktionen (52,4 %), womit dies die maßgeblichen Komponenten sind.

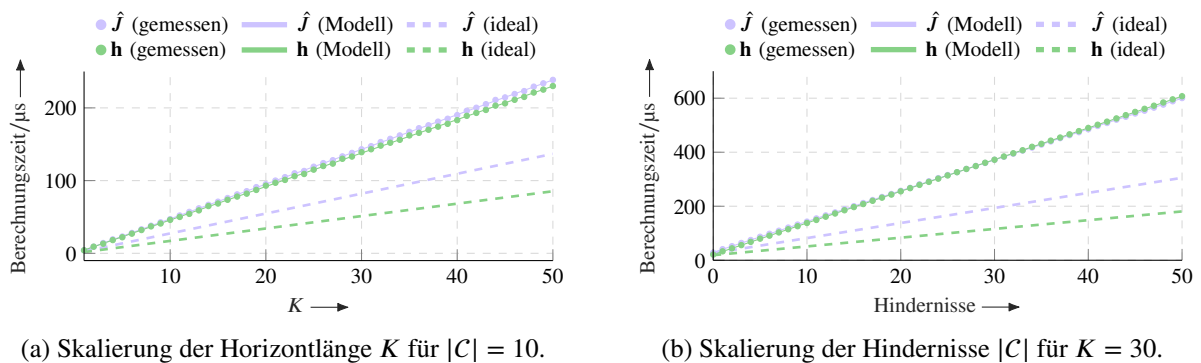
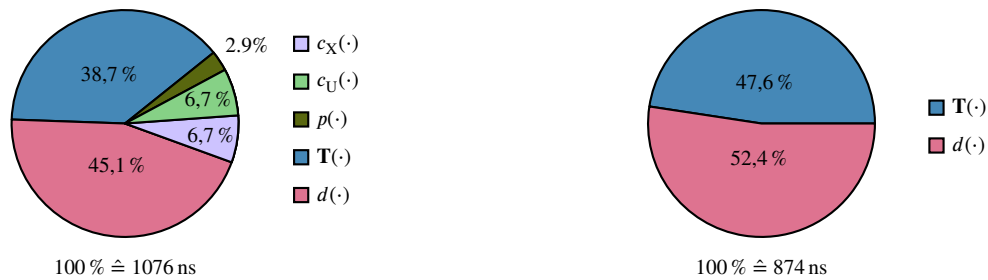


Abbildung 4.7.: Mittlere Berechnungszeiten von Kostenfunktion  $\hat{J}(\cdot)$  und Nebenbedingungen  $\mathbf{h}(\cdot)$  für den UR10 mit Selbstkollisionsvermeidung. Die idealisierten Geraden betrachten nur den Aufwand dominanter Funktionen.



(a) Aufteilung der Berechnungszeit beim Aufruf der Kostenfunktion  $\hat{J}(\cdot)$ .

(b) Aufteilung der Berechnungszeit beim Aufruf der Ungleichheitsnebenbedingungen  $\mathbf{h}(\cdot)$  mit reiner Selbstkollisionsvermeidung.

Abbildung 4.8.: Aufteilung der Berechnungszeiten von Kostenfunktion und Ungleichheitsnebenbedingungen für den UR10.

Die Vorwärtskinematik  $\mathbf{T}(\cdot)$  skaliert in Abhängigkeit von  $K$  und die Distanzfunktionen in der Abhängigkeit von  $|C|$ . Die Zustands- und Stellgrößenbeschränkungen erfordern wie die Gleichheitsnebenbedingungen keine nennenswerten Berechnungen und werden ausgelassen.

Werden die isolierten Berechnungszeiten (siehe Anhang B.2.4) der am Aufwand beteiligten Komponenten betrachtet und mit  $K$  bzw.  $|C|$  skaliert, ergibt sich neben der bisher empirischen Auswertung eine theoretische untere Schranke, die jeweils gestrichelt in Abbildung 4.7 angegeben ist. Demnach lässt sich der Zuwachs pro Horizontlänge für Kostenfunktion und Nebenbedingungen im Idealfall auf  $2,74 \mu\text{s}$  bzw.  $1,71 \mu\text{s}$  senken. Für die Skalierung pro Hindernis ergeben sich  $5,58 \mu\text{s}$  und  $3,24 \mu\text{s}$ . Eine Vermutung für die gegenwärtig vergleichsweise schlecht skalierenden Nebenbedingungen ist die Dimension von  $\mathbf{h}(\cdot)$  (vgl. Abbildung 4.5). Die Repräsentation von derart großen Vektoren mit zur Laufzeit dynamischen Längen ist langsamer als in der Kostenfunktion eine Summe von Skalaren zu berechnen. Hinzu kommt, dass neben den isolierten Aufrufen der Komponenten Datenstrukturen vorbereitet, kopiert und geprüft sowie Aufrufe durch unterschiedliche Ebenen gereicht und Signale zu Analyse Zwecken nach außen geführt werden müssen. Bei der Implementierung zu Forschungszwecken ist letztlich ein Kompromiss zwischen maximaler Effizienz des Algorithmus und den Möglichkeiten zur Erweiterung und Modifikation zu finden. Vielversprechendes Einsparpotential lässt sich anhand von Abbildung 4.8b identifizieren. Sie zeigt, dass  $52,4\%$  der Berechnungszeit für die Nebenbedingungen auf die Abstandsberechnungen fällt. Da sie ebenfalls in der Kostenfunktion für jeweils dieselben Zustände auftreten, lassen sie sich durch eine Spezialisierung der Implementierung des Lösungsverfahrens einsparen. Beispielsweise ist dies der Fall bei der Berechnung von Differenzenquotienten mit der Schrittweite  $\Delta\mathbf{z}$ , wenn nach der Auswertung von  $\hat{J}(\mathbf{z} + \Delta\mathbf{z})$  mit den Ergebnissen der Abstandsberechnungen auch  $\mathbf{h}(\mathbf{z} + \Delta\mathbf{z})$  bestimmt wird. Hierfür bieten sich alternierende Teilaufrufe oder die Zwischenspeicherung von Ergebnissen an. Der weitere Vorteil ist, dass damit die Vorwärtskinematik für die Nebenbedingungen ebenfalls eingespart wird und der Aufwand, abgesehen für das Speichern von Zwischenergebnissen, vollständig entfällt.

# 5

## Lösung des Planungsproblems unter zeitlichen Beschränkungen

Nachdem in den letzten Kapiteln Transkriptionsmethoden vorgestellt und das Planungsproblem mit statischer Kollisionsvermeidung aufgebaut wurde, verfolgt dieses Kapitel die Vervollständigung des ersten Teils mit dem Ziel einer effizienten Lösung des Planungsproblems am Beispiel des Roboters UR10. Konkret umfasst dies die Festlegung auf eine Kombination aus Transkriptionsmethode, Strukturausnutzung und Lösungsverfahren unter Beachtung der Größenordnung und Sensitivität sowie dem praxisorientierten Umgang mit der verbleibenden Optimierungszeit. Hierfür verdeutlicht Abschnitt 5.1 zunächst den Einfluss der Transkriptionsmethode und behandelt mit der Strukturausnutzung über einen Hypergraphen eine Maßnahme, die Größenordnung durch Vorwissen zum Optimierungsproblem zu relativieren. Des Weiteren werden IP- und AS-Lösungsverfahren für die MHP bewertet und mit anderen spezialisierten Methoden verglichen. Abschnitt 5.2 untersucht das Potential suboptimaler Lösungen zur Reduzierung der Optimierungszeit sowie der Korrektur von durch selbiger verursachter Verzögerungen. Das Kapitel schließt mit der Festlegung von Zeitkonstanten ab, sodass die Anwendung der MHP im zweiten Teil der Arbeit möglich ist.

Als Grundlage der Realisierung der MHP und des Hypergraphen dient ein Framework zum Vergleich unterschiedlicher Regelungskonzepte, das im Rahmen der Arbeit von Rösman [Rös19] mit dem Fokus auf die zeitoptimale modellprädiktive Regelung unter Mitwirkung entstanden ist. Das Framework bietet unter anderem die Möglichkeit Parameter und Transkriptionsmethoden für Optimalsteuerungsprobleme und verknüpfte Lösungsverfahren zu variieren. Das Framework wurde im Rahmen dieser Arbeit mit den Kostenfunktionen und Nebenbedingungen zur MHP (Kapitel 3 und 4) sowie der Lösungsverfahren aus diesem Kapitel ausgebaut.

Die Ergebnisse dieses Kapitels basieren in Teilen auf den Publikationen [Krä+20b; Rös+18].

### 5.1. Spezialisierung auf die Trajektorienoptimierung

Mit den Transkriptionsmethoden aus Kapitel 3 wird das OSP in unterschiedlich sensitive NLPs mit verschiedenen Größenordnungen überführt. Bei den Schießverfahren erhöht sich die Sensitivität, je länger die Schießintervalle werden, da das Lösungsverfahren in ihnen nur über die Integration der Systemdynamik Einfluss auf die Zustände und damit Kostenfunktion und Nebenbedingungen nehmen kann. Daneben beeinflusst die Transkriptionsmethode die Größenordnung in Form von  $D$  und  $N_G$ , wie Abbildungen 5.1 und 5.2 jeweils in Abhängigkeit der Horizontlänge  $K$  und verschiedene Intervallzahlen  $I$  für den UR10 ( $N = 6$ ) verdeutlichen. Bei der vollständi-

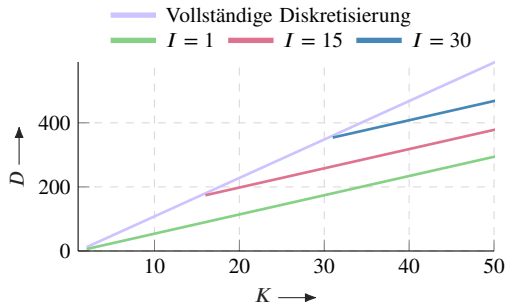


Abbildung 5.1.: Auswirkung der Horizontlänge auf die Dimension des Optimierungsvektors für unterschiedliche Transkriptionen.

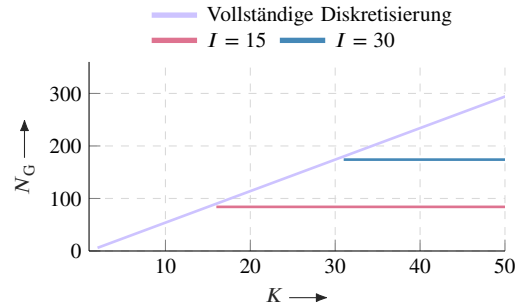


Abbildung 5.2.: Auswirkung der Horizontlänge auf die Dimension der Gleichheitsnebenbedingungen für unterschiedliche Transkriptionen.

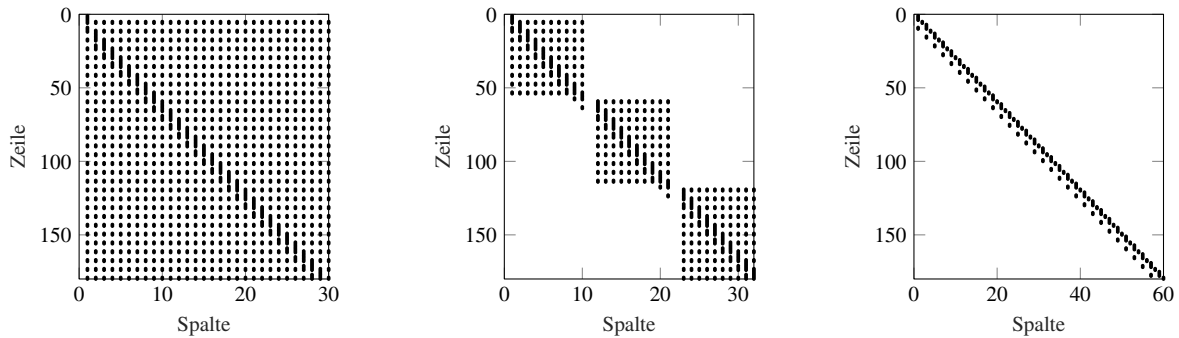
gen Diskretisierung steigen  $D$  und  $N_G$  am stärksten. Bei den Schießverfahren skaliert  $N_G$  mit der Intervallzahl und nicht der Horizontlänge. Für  $I = 1$  ergibt sich das Einfeldschießverfahren, das keine Gleichheitsnebenbedingungen besitzt. Die Schießverfahren erreichen bei kleinstmöglicher Intervalllänge  $K' = 2$  annähernd dieselben Dimensionen wie die vollständige Diskretisierung bei  $K = I + 1$ .

Für die vollständige Diskretisierung ergibt sich für  $K = 31$  eine Größenordnung von  $D = 360$  und  $N_G = 180$ . Nimmt man darüber hinaus Selbstkollisionen und  $|C| = 4$  hinzu, ist  $N_U = 1440$ . Werden diese Größenordnungen auf die KKT-Matrix in Gleichung (3.42) für die IP-Methode übertragen, ergibt sich eine Dimension von  $3420 \times 3420$ . Bei einer AS-Methode nach (3.44) sind es noch ungefähr  $1440 \times 1440$ . Die genaue Zahl hängt von den aktiven Ungleichheitsnebenbedingungen ab, die in diesem Zahlenbeispiel für die konstanten Beschränkungen nur einseitig gesättigt ( $3N(K - 1)$ ) angenommen werden. Wird dasselbe Optimierungsproblem mit dem Einfeldschießverfahren transkribiert ( $I = 1$ ), belaufen sich die Dimensionen für die IP- und AS-Methode auf  $3060 \times 3060$  bzw.  $1080 \times 1080$ .

Die zuvor genannten Größenordnungen sind in Tabelle 5.1 für  $K = 31$  und weiteren Intervallzahlen zusammengetragen. Die Intervalle sind dabei so aufgeteilt, dass sie äquidistant und in ganzzahliger Länge resultieren. Der Wert  $I = 30$  repräsentiert die vollständige Diskretisierung. Es ist erkennbar, dass die IP-Methode zwar das größte Gleichungssystem löst, die AS-Methode allerdings auf das Finden des AS angewiesen ist. In jedem Fall benötigt die Lösung derart großer Gleichungssysteme Zeit und erhöht neben dem Aufwand für Funktionsaufrufe die Optimierungszeit. Sie ist in der letzten Spalte von Tabelle 5.1 in Form der durchschnittlich benötigten Optimierungszeit bis zum Erreichen eines Kostenschwellwerts angegeben. Die MHP wird mit dem Startzustand initialisiert und führt 10 Planungszyklen durch, wobei die optimier-

Tabelle 5.1.: Gegenüberstellung der Größenordnungen mit den resultierenden Dimensionen der KKT-Matrix und Optimierungszeiten für unterschiedliche Transkriptionen bei  $K = 31$ .

$I$	$D$	$N_G$	$N_U$	Dimension KKT-Matrix		Optimierungszeit in s	
				IP	AS	IPOpt	SLSQP
1	180	0	1440	3060	1080	6,50	11,39
2	186	6	1440	3072	1092	5,23	8,22
3	192	12	1440	3084	1104	5,79	7,91
6	210	30	1440	3120	1140	6,96	6,96
30	360	180	1440	3420	1440	7,28	7,05



(a) Einfachschießverfahren mit 1978 Einträgen ungleich Null. (b) Mehrfachschießverfahren mit 778 Einträgen ungleich Null. (c) Vollständige Diskretisierung mit 238 Einträgen ungleich Null.

Abbildung 5.3.: Strukturen der Jacobimatrix  $\mathbf{H}(\cdot)$  für verschiedene Transkriptionen.

ten Stellgrößen nicht vom Roboter umgesetzt werden, um den Einfluss der teilweise hohen und variierenden Laufzeiten auf die Entwicklung der Versuche zu eliminieren. Verwendet werden die AS-SQP-Methode SLSQP und die IP-Methode IPOpt, womit zwei bekannte Klassen von Lösungsverfahren vertreten sind. Weitere Daten zum Versuch befinden sich in Anhang B.3.1. Die sinkenden Optimierungszeiten bei der IP-Methode verdeutlichen, dass es von fallender Größenordnung zunächst stärker profitiert als es von der steigenden Sensitivität nachteilig beeinflusst wird. Erst beim Einfachschießverfahren dominiert schließlich die Sensitivität und die Optimierungszeit steigt. Die AS-Methode reagiert stärker auf die Sensitivität und profitiert von der Verringerung der Größenordnung kaum. Lediglich die Aufteilung  $I = 6$  kann relativ zur vollständigen Diskretisierung leichte Verbesserungen erzielen, da sich die Größenordnung am stärksten verringert. Im Direktvergleich ist die AS-Methode nur bei vollständiger Diskretisierung schneller, ansonsten setzt sich trotz der höheren Dimension der KKT-Matrix die IP-Methode durch oder ist gleichauf. Die kürzeste Optimierungszeit von 5,23 s erzielt das IP-Verfahren für  $I = 2$ . Die erzielten Optimierungszeiten sind noch weit von einer brauchbaren<sup>1</sup> Wiederholfrequenz entfernt und benötigen weitere Verbesserungen. Der nächste Abschnitt zeigt, wie durch Vorwissen zum NLP die Dimensionen relativiert und die Lösungsverfahren weiter beschleunigt werden.

## Besonderheiten von diskreten Optimalsteuerungsproblemen

Gleichungen (3.42) und (3.44) verdeutlichen bereits, dass nicht alle Einträge der KKT-Matrizen besetzt bzw. variabel sind. Die  $3420 \times 3420$  KKT-Matrix bei der IP-Methode aus dem letzten Abschnitt besitzt z. B. mindestens 71,5 % strukturelle Konstanten. Liegt dem NLP ein OSP zugrunde, kommen weitere hinzu, die unter anderem in Form von Nullen durch Unabhängigkeit zwischen den Optimierungsvariablen und den Nebenbedingungen sowie Summanden der Kostenfunktion in den Jacobi- und Hessematrizen entstehen. Da diese Abhängigkeiten von der Transkriptionsmethode beeinflusst werden, ist die Struktur stärker oder schwächer ausgeprägt. Abbildung 5.3 zeigt die Struktur einer Jacobimatrix  $\mathbf{H}(\cdot) \in \mathbb{R}^{N_U \times D}$  der Ungleichheitsnebenbedingungen für jede Transkriptionsmethode. Das zugrundeliegende NLP verwendet aus Darstellungsgründen  $K = 31$ ,  $N = 1$  und keine Kollisionsvermeidung. Von Null verschiedene Einträge werden als Punkte repräsentiert. Die Dimension beim Einfachschießverfahren ist mit  $180 \times 30$

<sup>1</sup>Ein exakter Wert ist abhängig von der zu erwartenden Bewegungsgeschwindigkeit der Hindernisse. Die ISO 13855 nimmt z. B. bis zu  $2,5 \text{ m s}^{-1}$  bei gehenden Personen an. Im Rahmen praktischer Versuche hat sich eine Wiederholfrequenz um die 10 Hz als brauchbar erwiesen.



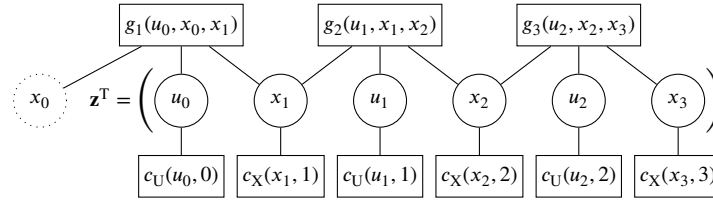


Abbildung 5.4.: Ausschnitt des Hypergraphens eines NLP nach vollständiger Diskretisierung für  $K = 4$  und  $N = 1$ . Eingetragen sind nur die quadratischen Kostenterme und Gleichheitsnebenbedingungen.

Anschließend werden die Abhängigkeiten der Teilfunktionen zu den Optimierungsvariablen in einem Hypergraphen verknüpft. Sofern vorhanden, können bei den Teilfunktionen auch analytische Ableitungen oder strukturelle Konstanten angegeben werden, ansonsten kommen zentrale Differenzen zum Einsatz.

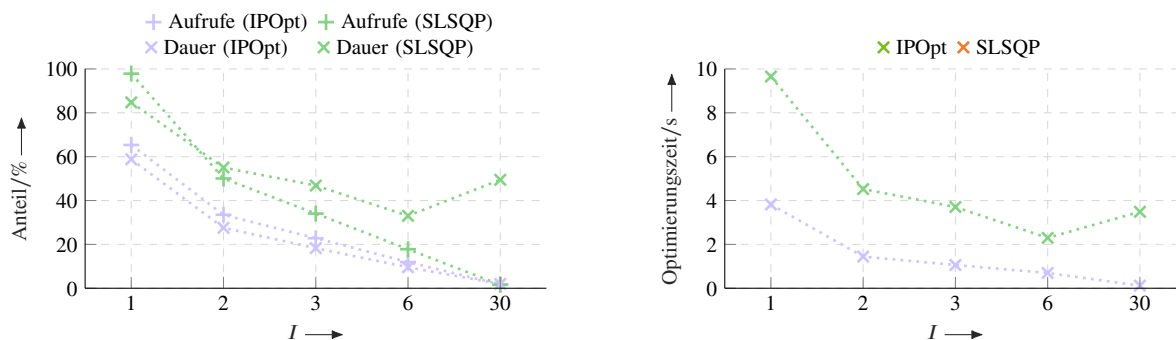
Abbildung 5.4 zeigt einen Teil des Hypergraphens eines NLP mit  $K = 4$  und  $N = 1$ , das über die vollständige Diskretisierung transkribiert ist. Die Knoten sind als Kreise dargestellt und repräsentieren die Optimierungsvariablen, wobei  $x_0$  der gegenwärtige Roboterzustand und damit konstant ist. Kanten werden über Rechtecke dargestellt und verbinden ungerichtet einen oder mehrere abhängige Knoten. Sie repräsentieren die Teilfunktionen der Kostenfunktionen und Nebenbedingungen. Dargestellt sind nur die  $N_G = 3$  Gleichheitsnebenbedingungen  $g_i(\cdot)$  mit  $i = 1, 2, 3$  sowie die Kostenfunktion  $\hat{J}(\cdot)$  mit den rein quadratischen Kostentermen  $c_X(x_k, k)$  ( $k = 1, 2, 3$ ) und  $c_U(u_k, k)$  ( $k = 0, 1, 2$ ). Für die Teilfunktionen  $g_i(\cdot)$  entsprechen die Triple  $(u_i, x_i, x_{i+1})$  jeweils der kleinsten Menge an abhängigen Optimierungsvariablen. Die Teilfunktionen der Kostenfunktion liegen bereits in rein zustandsabhängigen Termen  $c_X(x_k, k)$  sowie von den Stellgrößen abhängigen Termen  $c_U(u_k, k)$  vor. Wird der Gradient  $\nabla_z \hat{J}(\cdot)$  über zentrale Differenzen bestimmt:

$$\nabla_z \hat{J}(\cdot) \approx \frac{\hat{J}(\mathbf{z} + \Delta \mathbf{z}) - \hat{J}(\mathbf{z} - \Delta \mathbf{z})}{2\Delta \mathbf{z}}, \quad (5.4)$$

sind  $2D = 12$  Aufrufe auf  $\hat{J}(\cdot)$  und damit auf jeder Teilfunktion nötig, die überwiegend Nullen berechnen, da sie von den meisten Optimierungsvariablen unabhängig sind. Durch den Hypergraphen sind die Teilfunktionen von  $\hat{J}(\cdot)$  und ihre Abhängigkeiten bekannt, sodass nur sie gezielt aufgerufen und die abhängigen Optimierungsvariablen variiert werden. Damit wird jede Teilfunktion nur zweimal aufgerufen.

Um den Einfluss des Hypergraphens bei verschiedenen Kombinationen aus Lösungsverfahren und Transkriptionsmethode zu bewerten, werden in Abbildung 5.5a die prozentualen Anteile an den Aufrufen der Annäherungskosten (4.28) und an der insgesamt erzielten Optimierungszeit aus Tabelle 5.1 angegeben. Die neuen Optimierungszeiten werden auf dieselbe Weise erhoben wie in Tabelle 5.1. Es ist zu sehen, wie der Hypergraph bei vollständiger Diskretisierung ( $I = 30$ ) die größte Reduktion von Aufrufen ermöglicht und dieser Effekt in Richtung  $I = 1$  abnimmt, da die Abhängigkeiten zu den Optimierungsvariablen zunehmen. SLSQP kann nur bei  $I = 30$  geringfügig mehr profitieren (1,6 %) als IPOpt (1,8 %) und liegt ansonsten unterhalb der Reduktion von IPOpt. Das liegt daran, dass die Realisierung von SLSQP Ableitungen und Funktionswerte zusammen berechnet und somit eine grundsätzlich höhere Anzahl von Aufrufen neben jenen entsteht, die für die Ableitungen notwendig sind. Während dies der Hypergraph für  $I = 30$  noch abfedert, verstärkt sich der Effekt in Richtung  $I = 1$ .

Durch die verringerten Aufrufe wird die Optimierungszeit für alle Intervallzahlen gesenkt. Dabei erreicht SLSQP durch das Zusammenspiel von verringerter Größenordnung und Funktionsaufrufen für  $I = 6$  seine größte Reduktion auf 33 % (2,3 s) der vorherigen Optimierungszeit.



(a) Prozentuale Reduktion von Aufrufen der Annäherungskosten und der mittleren Optimierungszeit gegenüber Tabelle 5.1.

(b) Absolutwerte der mittleren Optimierungszeiten bei Strukturausnutzung.

Abbildung 5.5.: Einfluss der Strukturausnutzung durch den Hypergraphen auf die Aufrufe der Annäherungskosten und mittleren Optimierungszeit.

Dies ist insgesamt die Konstellation mit der geringsten Optimierungszeit für SLSQP, wie Abbildung 5.5b mit absoluten Zahlen verdeutlicht. Anders als SLSQP, nutzt IPOpt zusätzlich die diagonalähnlichen Strukturen aus und erzielt zusammen mit den geringeren Aufrufen eine Reduzierung der Optimierungszeit auf 2 % (0,12 s) bei  $I = 30$ . Während IPOpt ohne Strukturausnutzung noch von der Reduzierung der Größenordnung profitiert (Tabelle 5.1), dominiert mit Strukturausnutzung die Anfälligkeit für die Sensitivität, sodass  $I = 30$  insgesamt die Konstellation mit der geringsten Optimierungszeit für IPOpt darstellt. Auf eine Auswertung der Optimierungszeiten bei Warmstarts wird aufgrund der bereits ca. 19-mal langsameren Laufzeit von SLSQP verzichtet, auch wenn AS-Methoden Vorteile bei Warmstarts besitzen.

IPOpt besitzt die Möglichkeit verschiedene Lösungsverfahren für das lineare Gleichungssystem (3.42) des Newton-Schritts zu verwenden. Tabelle 5.2 fasst die mittleren Optimierungszeiten aus 10 Wiederholungen mit verschiedenen Verfahren für  $I = 30$  zusammen und normiert dabei auf MA<sup>27</sup>, der für die bisherigen Ergebnisse verwendet wurde. Weitere Details zu den Implementierungen befinden sich in Anhang B.3.2. Die Verfahren MA77 und MUMPS<sup>3</sup> fallen mit einer erhöhten Laufzeit auf. Ersterer ist für speicherintensive Probleme optimiert und letzterer auf Parallelität, was beides einen gewissen Mehraufwand verursacht, der sich als nicht vorteilhaft erweist [Tas+19]. MA57 ist die Weiterentwicklung von MA27 und nutzt hardwareoptimierte BLAS<sup>4</sup>-Routinen und ist speziell für dünn besetzte Matrizen geeignet, womit er sich von seinem Vorgänger und den Varianten MA97 und PARDISO<sup>5</sup> absetzt.

Da IPOpt für  $I = 30$  mit MA57 die kürzeste Optimierungszeit von ca. 0,11 s erzielt, stellt diese

Tabelle 5.2.: Mittlere und auf MA27 normierte Optimierungszeiten unter Verwendung unterschiedlicher Lösungsverfahren für das Gleichungssystem des Newton-Schritts bei IPOpt.

MA27 [Duf+83]	MA57 [Duf04]	MA77 [Rei+09]	MA97 [Hog+11]	MUMPS [Ame+01]	PARDISO [Sch+01]
1,0 (0,12 s)	0,95	1,49	1,0	2,16	1,04

<sup>2</sup>Multifrontal Approach

<sup>3</sup>Multifrontal Massively Parallel Sparse Direct Solver

<sup>4</sup>Basic Linear Algebra Subprograms

<sup>5</sup>Parallel Direct Solver

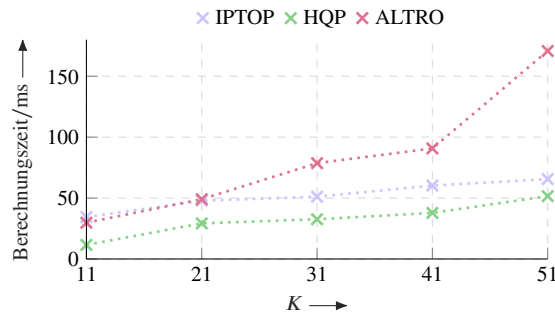


Abbildung 5.6.: Durchschnittliche Berechnungszeiten ohne Hindernisse aber mit Selbstkollisionsvermeidung für unterschiedliche Horizontlängen.

Konstellation zusammen mit dem Hypergraphen die Realisierung der MHP für den weiteren Verlauf der Arbeit dar und wird mit IPTOP<sup>6</sup> abgekürzt.

## Vergleich mit anderen Frameworks zur Trajektorienoptimierung

Nachdem die Auswertung in den vorherigen Abschnitten IPTOP mit der Kombination aus vollständiger Diskretisierung, Hypergraph und dem Lösungsverfahren IPOpt als schnellste Variante bestimmt hat, bleibt abschließend zu bewerten, wie diese Kombination gegenüber weiteren spezialisierten Ansätzen aus der Literatur einzuordnen ist. Für die anschließende Auswertung wird mit ALTRO und HQP jeweils ein auf die Trajektorienoptimierung bzw. auf die optimale Regelung spezialisiertes Verfahren ausgewählt. Ersteres basiert auf dem Einzelschießverfahren und letzteres auf der vollständigen Diskretisierung, womit eine Variation der Intervallzahl entfällt. Beide Verfahren nutzen das Vorwissen, dass das NLP aus einem OSP stammt. ALTRO löst das NLP mit einer AL<sup>7</sup>-Methode und HQP über den SQP-Ansatz mit einer IP-Methode für das quadratische Unterproblem. Die Implementierungen von Kostenfunktion und Nebenbedingungen sowie die Initialisierung über den aktuellen Zustand sind für alle Verfahren gleich. Die Auswertung variiert die Horizontlänge und Hüllkörperzahl. Als Hüllkörper werden Kugeln ( $S_p$ ) verwendet, da es primär um die Dimensionsänderungen der Ungleichheitsnebenbedingungen geht. Weitere Details zum Aufbau befinden sich in Anhang B.3.3.

Abbildung 5.6 vergleicht die mittleren Laufzeiten von je 10 Wiederholungen der drei Verfahren für verschiedene Horizontlängen ohne Hindernisse mit Selbstkollisionen. Dieser Fall entspricht der Planung bei dynamischen Zielzuständen im hindernisfreien Raum. Ohne Hindernisse werden nur die Distanzen für die Selbstkollisionsvermeidung berechnet und es überwiegt die Größenordnung des NLP. HQP geht mit den niedrigen Größenordnungen am besten um und benötigt die geringste Optimierungszeit. Zusätzlich besitzt das NLP bei reiner Selbstkollisionsvermeidung vergleichsweise wenig nichtlineare Nebenbedingungen, sodass das quadratische Unterproblem bei HQP nah am NLP liegt und die Zielstrebigkeit der Newton-Schritte höher ist. IPTOP benötigt eine um ca. 19,3 ms höhere Berechnungszeit als HQP, der sich jedoch für größere  $K$  angesichts des Zuwachses pro 10 Horizontschritte als linearen Trend von 7,47 ms für IPTOP und 8,86 ms für HQP relativiert. Bei AL-Methoden werden die Nebenbedingungen in Form von gewichteten Straftermen in die Kostenfunktion integriert. Zu Beginn sind die Gewichte klein und die Optimierung fokussiert sich primär auf die Minimierung der ursprünglichen Kostenfunktion. Mit fortschreitenden Iterationen steigen die Gewichte, wodurch der Fokus auf die Einhaltung

<sup>6</sup>Interior Point Trajectory Optimization

<sup>7</sup>Augmented Lagrangian

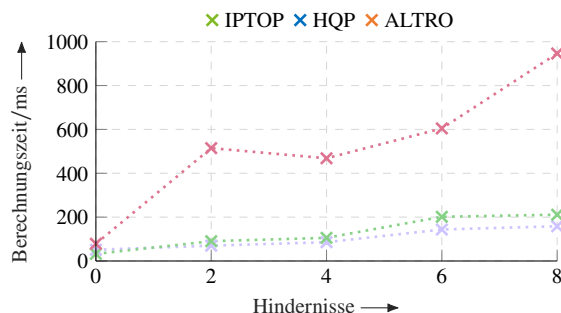


Abbildung 5.7.: Durchschnittliche Berechnungszeiten für  $K = 31$  mit Selbstkollisionsvermeidung und unterschiedlicher Hinderniszahl.

der Nebenbedingungen gelegt wird. Für ALTRO bedeutet das in diesem Fall, dass zunächst die Trajektorie in Richtung Ziel gestreckt wird, um die Kosten  $\hat{J}(\cdot)$  zu minimieren. Anschließend werden vermehrt die Nebenbedingungen bezüglich der Stellgröße und Selbstkollisionen berücksichtigt und die Trajektorie wieder gestaucht, womit ein Teil der vorherigen Optimierungsarbeit unnötig war. Während dieses Vorgehen für  $K = 11$  noch geringfügig (ca. 4,6 ms) schneller ist als IPTOP, wird die Optimierungszeit ab  $K = 21$  zunehmend schlechter.

Abbildung 5.7 zeigt die Wiederholung der Versuche für eine Horizontlänge von  $K = 31$  und variabler Hüllkörperzahl. Der erste Fall entspricht dem für  $K = 31$  aus Abbildung 5.6. Kommen Hindernisse hinzu, wird die Optimierungszeit stärker vom Aufwand der Distanzberechnungen beeinflusst und Verfahren profitieren zunehmend von weniger Funktionsaufrufen. ALTRO ist hier aufgrund der wiederholten Optimierung der erweiterten Kostenfunktion, die selbst mehrere Wiederholungen beinhaltet, im Nachteil. IPTOP und HQP haben zwar größere Gleichungssysteme als ALTRO, nähern sich dem Optimum dadurch aber meist auch mit größeren und damit weniger Schritten/Funktionsaufrufen an. Zusammen mit entsprechend schnellen Verfahren wie MA57 bei IPTOP und SpBKP<sup>8</sup> bei HQP zur Lösung dieser Gleichungssysteme unter Ausnutzung dünnbesetzter Strukturen, ergibt sich eine kürzere Optimierungszeit. Dabei skaliert IPTOP mit 14,42 ms pro Hindernis besser als HQP mit 23,41 ms. Hinzu kommt, dass HQP weniger flexibel bei der Formulierung von Teilfunktionen bei der Strukturausnutzung ist. Dadurch ist es unter anderem nicht möglich, Nebenbedingungen für die Beschränkung von Beschleunigungen anzugeben, da abseits der Systemnebenbedingungen keine Argumente von Zuständen oder Stellgrößen unterschiedlicher Zeitschritte unterstützt werden. Hierfür bedarf es die Erweiterung des Zustands um Gelenkwinkelgeschwindigkeiten und Änderung der Stellgröße in Gelenkwinkelbeschleunigungen, was  $N$  verdoppelt und die Größenordnung erhöht. Die Versuche werden daher einheitlich ohne Beschränkung der Beschleunigung durchgeführt.

## 5.2. Praktischer Umgang mit der Optimierungszeit

Der vorherige Abschnitt hat gezeigt, dass bei einer Optimierung mit verschiedenen Horizontlängen und Hinderniszahlen Berechnungszeiten im Bereich um 100 ms zu erwarten sind. Um auf Änderungen der Umgebung oder Aufgabe zügig zu reagieren, ist eine möglichst geringe Optimierungszeit nötig. Der nächste Abschnitt untersucht hierfür das Potential von vorzeitigen Optimierungsabbrüchen.

<sup>8</sup>Sparse Bunch-Kaufman-Parlett

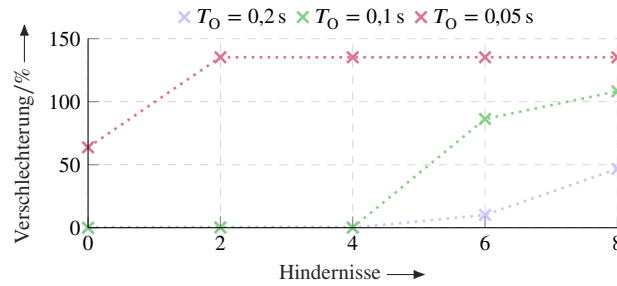


Abbildung 5.8.: Prozentuale Verschlechterung von Lösungen bei zeitlich beschränkter Optimierungszeit unter Variation der Hinderniszahl bei  $K = 51$ .

## Abbruchkriterium und suboptimale Lösungen

Für die MHP ist der Abbruch der Optimierung nach einer bestimmten Zeit  $T_O$  interessant, da es die Angabe einer unteren Schranke für die Wiederholfrequenz erlaubt. Ist die Lösung nach einer Optimierung nicht optimal aber verletzt keine Nebenbedingungen, sind die Auswirkungen aufgrund der rückgekoppelten Planung vernachlässigbar. Die folgende Auswertung untersucht, wie sich Horizontlänge und Hinderniszahlen auf die Qualität von Lösungen in Form ihrer Kosten bei unterschiedlich beschränkten Optimierungszeiten auswirken.

Abbildung 5.8 zeigt für verschiedene Hinderniszahlen den Anteil, um den die Kosten der Lösungen mit beschränkter Optimierungszeit höher ausfallen als die einer unbeschränkten Referenzlösung. Das Szenario ist dasselbe wie im vorherigen Abschnitt und verwendet die Hüllkörperform  $S_p$ . Eine Horizontlänge von  $K = 51$  ist bereits ohne Hindernisse so aufwändig, dass bei  $T_O = 0,05 \text{ s}$  nur eine Iteration auf dem Lösungskandidaten ausgeführt werden kann, was eine Erhöhung der Kosten von 63,8 % bedeutet. Die Verschlechterung schlägt sich jedoch vornehmlich im hinteren Teil des Planungshorizonts nieder, während die erste Stellgröße  $\mathbf{u}_0$  identisch mit jener aus der konvergierten Lösung ist und die Planung daher kaum beeinträchtigt. Kommen Hindernisse hinzu, dauert bereits eine Iteration länger als 0,05 s, womit effektiv keine Planung mehr möglich ist. Mit einem Zeitbudget von  $T_O = 0,1 \text{ s}$  bzw.  $0,2 \text{ s}$  lassen sich mit  $K = 51$  bei bis zu 4 Hindernissen Lösungen ohne Verschlechterung ermitteln. Bei 6 Hindernissen zeichnen sich erste Verschlechterungen von 86,2 % ( $T_O = 0,1 \text{ s}$ ) und 10,2 % ( $T_O = 0,2 \text{ s}$ ) ab. In beiden Fällen ergeben sich marginale<sup>9</sup> Abweichungen in Form des absoluten Fehlers der ersten Stellgröße in Höhe von  $0,05 \text{ rad s}^{-1}$  und  $0,02 \text{ rad s}^{-1}$ . Bei 8 Hindernissen fallen die Verschlechterungen mit 108,3 % und 46,6 % stärker aus, während die Abweichungen der ersten Stellgröße weiterhin marginal bleiben ( $0,04 \text{ rad s}^{-1}$  und  $0,01 \text{ rad s}^{-1}$ ). Erst bei 39 Hindernissen ist keine Planung für  $T_O = 0,1 \text{ s}$  mehr möglich, bzw. bei 94 Hindernissen für  $T_O = 0,2 \text{ s}$ .

Wird für eine bestimmte Anzahl an Hindernissen eine bestimmte Planungsfrequenz bei möglichst geringer Verschlechterung gesucht, ist die Verkürzung des Horizonts eine Option, wie Abbildung 5.9 für 6 Hindernisse zeigt. Ab einer Horizontlänge von  $K = 41$  ist für  $T_O = 0,05 \text{ s}$  eine Planung wieder möglich, besitzt jedoch eine Verschlechterung von 92,6 %. Diese verringert sich auf 0 %, wenn der Horizont auf  $K = 11$  verkürzt wird. Durch Verringerung der Horizontlänge auf  $K = 41$  ist die Planung für  $T_O = 0,2 \text{ s}$  bereits ohne Verschlechterung möglich. Für 0,1 s

<sup>9</sup>Werden Abweichungen in der Stellgröße als Abweichung von der Optimallösung aufgefasst, ergibt sich nach einer Optimierung je nach Wiederholfrequenz ein suboptimaler Gelenkwinkel. Bei 10 Hz sind es z. B.  $0,1 \text{ s} \cdot 0,05 \text{ rad s}^{-1} = 0,005 \text{ rad} \approx 0,29^\circ$  was im ungünstigen Fall ca. 1,3 cm am Endeffektor des UR10 entspricht (1,33 m Reichweite mit den gleichartig betroffenen Gelenken 2 und 3). Dies spiegelt nicht die Genauigkeit der Planung wieder, einen festen Punkt anzufahren.

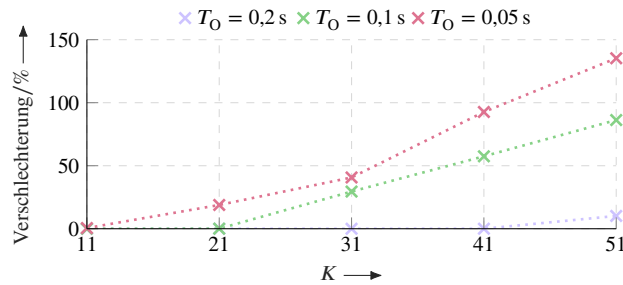


Abbildung 5.9.: Prozentuale Verschlechterung von Lösungen bei zeitlich beschränkter Optimierungszeit unter Variation der Horizontlänge bei 6 Hindernissen.

ist dies ab einer Horizontlänge von  $K = 21$  möglich. Bei einer Horizontlänge von  $K = 11$  ist ab 52 Hindernissen keine Planung mehr für  $T_O = 0,05 \text{ s}$  möglich und ab 220 Hindernissen für  $T_O = 0,1 \text{ s}$ . Diese beiden Extremfälle führen allerdings in Kombination mit dem kurzen Horizont zu einer stärkeren Beeinträchtigung der Stellgrößen in Form einer langsameren Roboterbewegung. Letztlich bestimmt das konkrete Szenario die Abwägung zwischen Hindernissen, Horizontlänge und Planungsfrequenz. Mit Verweis auf Abbildung 4.4 lassen sich Hüllkörper mitunter durch andere Formen substituieren und günstig verringern.

### Korrektur der Auswirkungen der Optimierungszeit

Die Optimierungszeit entfällt letztlich nie gänzlich, was die Auseinandersetzung mit den daraus resultierenden Folgen erfordert. Die Auswirkungen der Optimierungszeit werden von Diehl u. a. [Die+06] auch als Online-Dilemma bezeichnet und beinhalten zwei Aspekte. Abbildung 5.10 unterstützt die Erklärung dieser Aspekte mit dem zeitlichen Ablauf bei der MHP. In der oberen Zeile symbolisieren Rechtecke den Zeitraum  $[t_0, t_0 + T_O]$  der Optimierung zwischen der Messung des gegenwärtigen Zustands  $\mathbf{q}(t_0)$  und der Ausgabe der optimierten (und verspäteten) Stellgröße  $\boldsymbol{\mu}_0 = \boldsymbol{\mu}(t_0 + T_O)$ . Die untere Zeile gibt an, welche Stellgröße für welchen Zeitraum am Roboter wirkt. Unter der Annahme, dass die kommunizierte Stellgröße den Roboter ohne zusätzliche Verzögerung erreicht, wirkt sie auf einen  $T_O$  älteren Roboterzustand  $\mathbf{q}(t_0 + T_O)$ . Im Idealfall gab es keine signifikante Veränderung des Roboterzustands, z. B. bei der Bewegung aus dem Stillstand, sodass  $\mathbf{q}(t_0) \approx \mathbf{q}(t_0 + T_O)$  gilt. Dies ist jedoch spätestens nach Beginn der Bewegung für  $t_1$  nicht mehr garantiert. Das zweite Problem ist die Frage, welche Stellgröße im Intervall  $[t_1, t_1 + T_O]$  während der Optimierung wirken soll, da Stellgröße  $\boldsymbol{\mu}_0$  nur für das Intervall  $[t_0, t_1]$  vorgesehen ist. Weitere Verzögerungen wie z. B. bei der Perzeption des aktuellen Zustands von

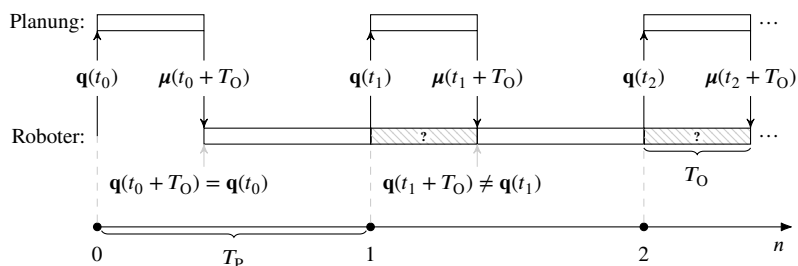


Abbildung 5.10.: Zeitlicher Ablauf bei der Planung beginnend mit der Messung des Roboterzustands, der eigentlichen Planung mit Dauer  $T_O$  und der verzögerten Ausgabe der Stellgröße mit anschließender Reaktion des Roboters.

Roboter und Hindernissen oder bei der Umsetzung der Stellgröße durch den Roboter werden nicht gesondert behandelt. Sie lassen sich für den im Folgenden skizzierten Fall mit zu der durch die Berechnungszeit bedingten Verzögerung zählen [Fin05; Krä+20b].

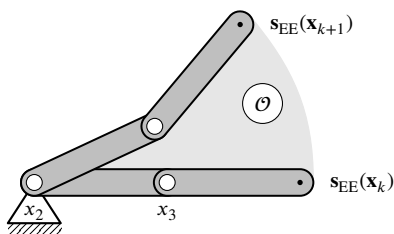
Der erste Aspekt des Online-Dilemmas wird über die Schätzung von  $\mathbf{q}(t_n + T_O)$  auf Basis von  $\mathbf{q}(t_n)$  adressiert [Die+06; Grü+17]. Er lässt sich unter Beachtung der Systemdynamik (3.1) über eine Vorwärtssimulation der Stellgrößen präzisieren:

$$\tilde{\mathbf{x}}_0 := \mathbf{q}(t_n) + \int_{t_n}^{t_n+T_O} \boldsymbol{\mu}(t) dt. \quad (5.5)$$

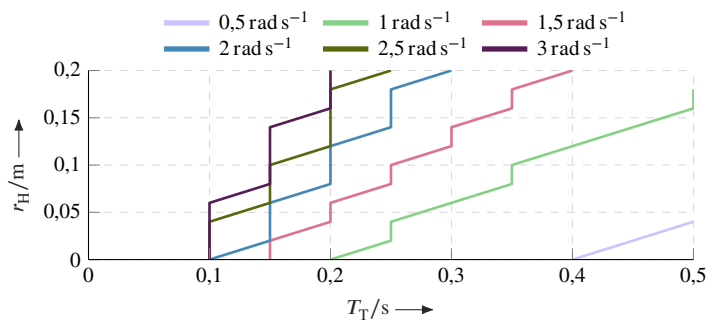
Da jedoch die Kenntnis über die am Roboter anliegende Stellgröße im Intervall  $[t_n, t_n + T_O]$  notwendig ist, wird zuvor der zweite Aspekt des Online-Dilemmas behandelt. Unter der Annahme einer für alle Planungszyklen  $n$  konstanten Zeitverzögerung  $T_O$ , kommt diese bei konstanter Fortsetzung der Stellgröße  $\boldsymbol{\mu}(t_n + T_O)$  in das Intervall  $[t_{n+1}, t_{n+1} + T_O]$  einer Totzeit gleich, welche die Planung und Ausführung verschiebt. Wird der Verlauf der Stellgrößen temporär gespeichert, bestimmt die Vorwärtssimulation eine Schätzung für  $\mathbf{q}(t_n + T_O)$ . Zur Bewertung dieser Methode wird sie beim UR10 eingesetzt und die Abweichungen von  $\mathbf{x}_0 = \mathbf{q}(t_n)$  und  $\tilde{\mathbf{x}}_0$  zum tatsächlichen Zustand  $\mathbf{q}(t_n + T_O)$  für eine Bewegung mit  $T_O = 0,1$  s bestimmt. Der geschätzte Zustand  $\tilde{\mathbf{x}}_0$  besitzt eine mittlere Abweichung von ca. 0,001 rad und liegt damit näher am realen Zustand  $\mathbf{q}(t_n + T_O)$  als  $\mathbf{x}_0 = \mathbf{q}(t_n)$  mit ca. 0,02 rad.

### Wahl der Zeitkonstanten und Tunneleffekt

Nach der Feststellung und Auseinandersetzung mit der Zeitverzögerung  $T_O$  durch die Optimierung, sind abschließend praktische Werte für das Planungszeitfenster  $T_P$  und der Diskretisierung  $T_T$  zu wählen. Es ist offensichtlich, dass  $T_P \geq T_O$  gelten muss, weil sonst die Planungszeit  $T_P$  nicht einzuhalten ist. Für eine möglichst hohe Planungsfrequenz wird daher  $T_P = T_O$  gewählt. Zur Festlegung der Zeitkonstante  $T_T$  gilt es den sogenannten Tunneleffekt zu beachten, der beschreibt, wie zwischen den diskreten Zuständen  $\mathbf{x}_k$  verletzte Nebenbedingungen unerkannt bleiben. Durch das Zusammenspiel zwischen maximaler Gelenkwinkelgeschwindigkeiten  $\mathbf{u}^+$  und  $T_T$  ergibt sich ein potentieller Tunnel zwischen zwei Zuständen  $\mathbf{x}_k$  und  $\mathbf{x}_{k+1}$ . Abbildung 5.11a zeigt diesen Tunnel beispielhaft anhand einer Vereinfachung des UR10 auf das zweite und dritte Gelenk. Ein Hindernis  $\mathcal{O}$  mit Radius  $r_H$  dringt unbemerkt ein, wenn dessen benötigter Raum



(a) Kinematische Veranschaulichung des Tunneleffekts zwischen zwei Zuständen.



(b) Phasenebene von  $T_T$  und  $r_H$  mit Grenzen des Tunneleffekts für verschiedene maximale Gelenkwinkelgeschwindigkeiten. Der Tunneleffekt tritt rechtsseitig der Grenzen ein.

Abbildung 5.11.: Analyse des Tunneleffekts für einen auf das zweite und dritte Gelenk reduzierten UR10.

kleiner ist als die Breite des Tunnels:

$$\underbrace{2r_H}_{\text{Hindernis}} \leq \underbrace{\|\mathbf{s}_{EE}(\mathbf{x}_k) - \mathbf{s}_{EE}(\mathbf{x}_{k+1})\| - 2r_R - d_C^-}_{\text{Tunnel}}. \quad (5.6)$$

Die maximale Breite des Tunnels wird über den Abstand des Endeffektors zwischen beiden Zuständen abzüglich des doppelten Radius  $r_R$  der Hüllkörper des letzten Armkörpers und des Mindestabstands  $d_C^-$  angenähert. Abbildung 5.11b variiert  $T_T$  sowie Radius  $r_H$  für  $d_C^- = 0$  und gibt für verschiedene maximale Stellgrößen die Grenzen des Tunneleffekts an. Linksseitig dieser Grenzen ist der Tunnel zu schmal für das Hindernis. Die Gelenkwinkelgeschwindigkeiten beider Gelenke addieren sich dabei wie in Abbildung 5.11a angedeutet. Wird z. B. von Hindernissen mit  $r_H = 0,05$  m ausgegangen, fordert die Abschätzung für  $T_T = 0,2$  s eine Gelenkwinkelgeschwindigkeit von maximal  $1 \text{ rad s}^{-1}$ . Für  $T_T = 0,1$  s sind hingegen bis zu  $2,5 \text{ rad s}^{-1}$  möglich. Neben einem kleineren  $T_T$  lässt sich der Tunnel auch durch Anwendung der Nebenbedingungen auf eine temporäre Überabtastung des Bereichs zwischen  $\mathbf{x}_k$  und  $\mathbf{x}_{k+1}$  eliminieren. Hierfür bieten sich kontinuierliche Darstellungen der Zwischenräume wie Polynome [Cas+09; Xia+12] oder Splines [Lo +11] bis hin zur vollständig kontinuierlichen Kollisionsvermeidung [Sch+13] an. Die vorgestellte Variante mit  $T_T = 0,1$  s besitzt allerdings sowohl bezogen auf den Tunneleffekt, als auch auf die Menge an unterstützten Hindernissen ausreichend Reserven für einen praktischen Einsatz und wird den aufwändigeren (quasi-) kontinuierlichen Ansätzen vorgezogen.

Für  $T_p = T_T$  entspricht  $\tau \in [0, T_p)$  aus dem Stellgesetz (2.6) der MHP genau dem ersten Intervall  $[0, T_T)$  der stückweise konstanten Steuerfolge  $\mathbf{u}_n^*(\tau)$ . Damit wird genau eine Stellgröße an den Roboter gesendet und für die Zeit  $T_p$  konstant gehalten. Für  $T_p > T_T$  müssen weitere Stellgrößen der optimierten Steuerfolge ausgegeben werden, womit zwingend  $KT_T \geq T_p$  gelten muss. Die Ausführung mehrerer Stellgrößen entspricht dabei faktisch einer temporären Steuerung und die MHP verschiebt sich konzeptionell zu Ansätzen wie ITOMP. Ist  $T_p < T_T$  verwendet die Planung ein restriktiveres Gitter als nötig, da sie mit zeitlichen Abständen zwischen Stelleingriffen plant, die größer sind als sie effektiv ausgeführt werden. Für den weiteren Verlauf gilt daher, wenn nicht anders angegeben  $T_p = T_T = 0,1$  s.

# 6

## Dynamische Szenen bei der Planung mit fortlaufendem Horizont

Mit Abschluss des ersten Teils und der effizienten Realisierung der MHP beginnt in diesem Kapitel der zweite Teil dieser Arbeit mit der Ausweitung der Kollisionsvermeidung auf dynamische Szenen und der Auswertung in Szenarien der MRK. Zu einer dynamischen Szene zählt die Existenz dynamischer Hindernisse gemäß Definition (4.8) sowie zeitabhängige Zielzustände  $\mathbf{x}_i(t)$ . Abschnitt 6.1 geht auf Ersteres ein und bewertet das Potential der Kollisionsvermeidung mit und ohne unterstützender Hindernisprädiktion. Das Folgen dynamischer Zielzustände mit und ohne Anwesenheit von statischen Hindernissen wird in Abschnitt 6.2 über einen Vergleich der MHP mit einem Ansatz zur Trajektorienfolgeregelung bewertet.

Die Ergebnisse dieses Kapitels basieren in Teilen auf den Publikationen [Krä+20b; Krä+21c; Ren+23a].

### 6.1. Kollisionsvermeidung bei dynamischen Hindernissen

Während statische Szenen vor der Planung im Bezug auf relevante Hindernisse und Hüllkörper ausgewertet werden, erfolgt dies bei dynamischen Hindernissen in der Regel laufend. Im weiteren Verlauf wird davon ausgegangen, dass die dynamischen Hindernisse  $\mathcal{D}$  bereits als geeignete Hüllkörper  $\mathcal{O}(t)$  modelliert sind und deren aktuelle Orientierung  $\Theta(t)$  und Translation  $\mathbf{s}(t)$  stets bekannt sind. Praktisch erfolgt dies für Personen z. B. über infrarotbasierte und markergestützte Kamerasysteme zur Bewegungserfassung.

Wie schon bei den statischen Hindernissen ergänzen sich Annäherungskosten und Nebenbedingungen auch bei den dynamischen Hindernissen. Die Nebenbedingungen werden um die der dynamischen Hindernisse erweitert:

$$\mathbf{h}(\mathbf{z}) := \begin{cases} \vdots \\ (4.21) \\ \vdots \\ \hline d_{\mathcal{D}}^- - d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2(t(n, k))) & k = 1, 2, \dots, K_{\mathcal{D}} \\ \vdots \\ \forall (\mathcal{O}_1(\cdot), \mathcal{O}_2(\cdot)) \in \mathcal{R} \times \mathcal{D} \end{cases} \leq \mathbf{0}, \quad (6.1)$$

sodass sich  $\mathbf{h}(\cdot)$  und damit auch  $N_{\mathcal{U}}$  um die Dimension  $N_{\mathcal{D}} = K_{\mathcal{D}}N_{\mathcal{RD}}$  mit  $N_{\mathcal{RD}} := |\mathcal{R} \times \mathcal{D}| = |\mathcal{D}|$  und  $K_{\mathcal{D}} \in \mathbb{N} \leq K$  vergrößert. Hierbei gilt es zu beachten, dass  $N_{\mathcal{RD}}$  potentiell zeitabhängig ist,

falls Hindernisse in den Arbeitsraum eintreten oder ihn verlassen. Diese zwischen den Planungen variable Dimension ist ein weiterer Grund, weswegen im vorherigen Kapitel die Wahl auf den Hypergraphen zur Strukturausnutzung fällt.

Über die Nebenbedingungen wird weiterhin die Gültigkeit einer Lösung unter anderem an die Kollisionsfreiheit ( $d(\cdot) \geq d_D^- \in \mathbb{R}^+$ ) geknüpft. Im Falle einer ungültigen Lösung wird der Roboter in einen sicheren Zustand, wie z. B. den nachgiebigen Modus, überführt, um potentielle Folgen einer Kollision zu reduzieren. Die Zeitfunktion  $t(n, k) := t_n + kT_T$  setzt die Zeit auf dem Horizont mit der Zeit außerhalb der Planung in Verbindung. Da nicht immer verlässliche Informationen über zukünftige Hindernisposen vorliegen und dadurch Lösungen möglicherweise zu Unrecht für ungültig erklärt werden, erstreckt sich die Kollisionsvermeidung nur über die ersten  $K_D$  Zustände des Horizonts. Dies umfasst außerdem den Teil der geplanten Trajektorie, der unmittelbar vor der Ausführung steht und daher eine höhere Relevanz für die proaktive Vermeidung von Kollisionen aufweist.

Die zukünftigen Posen der Hüllkörper für  $t(n, k > 0)$  werden zunächst als unbekannt und daher innerhalb des Horizonts als statisch angenommen ( $t(n, 0)$ ), womit sich die Schnappschussmethode ergibt. Während bei statischen Hindernissen unter Einhaltung von  $d^-$  vor einem unüberwindbaren Hindernis angehalten wird, reicht dies bei dynamischen Hindernissen nicht mehr aus, da sie dem Roboter in jedem Planungsschritt potentiell näher kommen. Ist das Hindernis so nah, dass innerhalb der ersten  $K^- - 1$  Zustände angehalten werden muss, ist keine gültige Lösung möglich. Zusätzlich besteht die Gefahr, dass Hindernisse den Horizont an beliebiger Stelle kreuzen und im schlimmsten Fall den Istzustand  $\mathbf{x}_0$  bedrohen. Damit die MHP, wie schon bei den statischen Hindernissen, auf die Sicherheitsebene sensibilisiert wird, gilt  $d_D^- > d^-$ . Zusätzlich kommt die während der Planungszeit  $T_p$  zurückgelegte Strecke des Hindernisses mit der Geschwindigkeit  $v \in \mathbb{R}$  hinzu:

$$d_D^- > d^- + T_p v. \quad (6.2)$$

Der Einfluss der Planungszeit  $T_p$  auf den Schwellwert motiviert nochmals den Vorteil höherer Planungsfrequenzen bei dynamischen Umgebungen, wie sie von der MHP erreicht werden.

Auch der Kostenterm (4.28) wird um die mit  $W_D \in \mathbb{R}^+$  gewichteten Potentiale für dynamische Hindernisse erweitert:

$$c_H(\mathbf{x}_k, k) := (4.28) \cdots + W_D \sum_{(\mathcal{O}_1(\cdot), \mathcal{O}_2(\cdot)) \in R \times D} p(d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2(t(n, k)))). \quad (6.3)$$

Während die Annäherungskosten im statischen Fall einen kompromissbereiten Abstand zu Hindernissen ermöglichen, tragen sie bei dynamischen Hindernissen eine weitere Rolle. Um das plötzliche Hineindrängen eines Hindernisses in den Horizont zu entschärfen, bieten sie einen Wahrnehmungsbereich, der über den Aktivierungsschwellwert  $d_D > d_D^-$  einstellbar ist und damit einen unterbrechungsärmeren Betrieb der Schnappschussmethode ermöglicht. Auf herannahende Hindernisse wird frühzeitiger mit einer Kurskorrektur reagiert, sodass die Nebenbedingungen seltener aktiv sind oder verletzt werden. Um dieser Rolle bestmöglich gerecht zu werden, folgt die Wahl des Gewichts  $W_D$  der Ungleichung (4.29).

Die groben Näherungen der Schwellwerte in den Gleichungen (4.20) und (6.2) sind nicht allgemeingültig, da ein dynamisches Hindernis sich dem Roboter an verschiedenen Stellen entlang der Armkörper nähern kann, wodurch stark situative Gegebenheiten für den Ausweichvorgang entstehen. Beispielsweise überlagern sich am Endeffektor die Tangentialgeschwindigkeiten sämtlicher Gelenke, wohingegen sie am ersten Armkörper nur vom ersten Gelenk beeinflusst werden. Entsprechend besitzt der Roboter am Endeffektor eine höhere Tangentialgeschwindigkeit und

Beweglichkeit als nah an der Basis, was zusätzlich auch vom aktuellen Roboterzustand abhängt. Es ist daher nicht zielführend, allgemeine theoretische Obergrenzen ohne Bezug zu Anwendungen oder Anforderungen anzugeben. Entsprechend wird der Aktivierungsschwellwert  $d_D$  und Schwellwert  $d_D^-$  im Folgenden für konkrete Anwendungsszenarien bestimmt.

### Auswertung der Schnappschussmethode

Für die folgenden Auswertungen werden zwei Szenarien aus der MRK mit jeweils einer Person gewählt. Die Person wird über eine Kugel ( $S_p$ ) für den Kopf und Rundzylinder ( $S_L$ ) für die oberen Gliedmaßen sowie je einer für Ober- und Unterkörper modelliert. In Szenario A bewegt sich die Person in den Arbeitsbereich des querenden UR10 und erzwingt damit die Kollisionsvermeidung. Das Szenario B beginnt mit einer sitzenden Person, die eine Kollisionsvermeidung durch ihr Aufstehen auslöst. Die Bewegungen entsprechen Aufnahmen einer realen Person. Um die Rückwirkung der Roboterbewegung auf das Verhalten der Person zu eliminieren, werden die Aufnahmen ohne Roboter durchgeführt und nachträglich zusammengeführt. Zusätzlich sind die Bewegungen derart abgepasst, dass sich der Mensch mehr dem gegenwärtigen Roboterzustand nähert als sich in den Horizont zu drängen, womit die MHP stärker auf den Wahrnehmungsbereich der Annäherungskosten angewiesen ist. Die maximalen Geschwindigkeiten der Person sind  $0,9 \text{ m s}^{-1}$  (A) und  $0,7 \text{ m s}^{-1}$  (B). Da der Roboter im hindernisfreien Fall nur das erste Gelenk bewegt und der Mensch als erstes auf den Endeffektor trifft, werden unter Verwendung der Gleichungen (4.20) und (6.2) sowie der Angaben aus Anhang A.1, einer Planungszeit von  $T_p = 0,1 \text{ s}$ , einer maximalen Robotergergeschwindigkeit von  $u^+ = 0,5 \text{ rad s}^{-1}$  und der höheren Geschwindigkeit des Menschen aus Szenario A folgender Wert für  $d_D^-$  berechnet:

$$d_D^- \approx 0,14 \text{ m.} \tag{6.4}$$

Für eine unterbrechungsfreie Durchführung der Versuche wird nicht auf ungünstige Lösungen reagiert und die parallele Abstandsüberwachung ist inaktiv. Die Nebenbedingungen zur Kollisionsvermeidung bei dynamischen Hindernissen agieren auf dem gesamten Horizont ( $K_D = K$ ). Weitere Angaben zu den Szenarien befinden sich in Anhang B.4.1.

Abbildung 6.1 zeigt die Verteilung der kürzesten Abstände zwischen Roboter und Mensch unter Variation des Schwellwerts  $d_D$  für beide Szenarien. Für Szenario A ist in Abbildung 6.1a

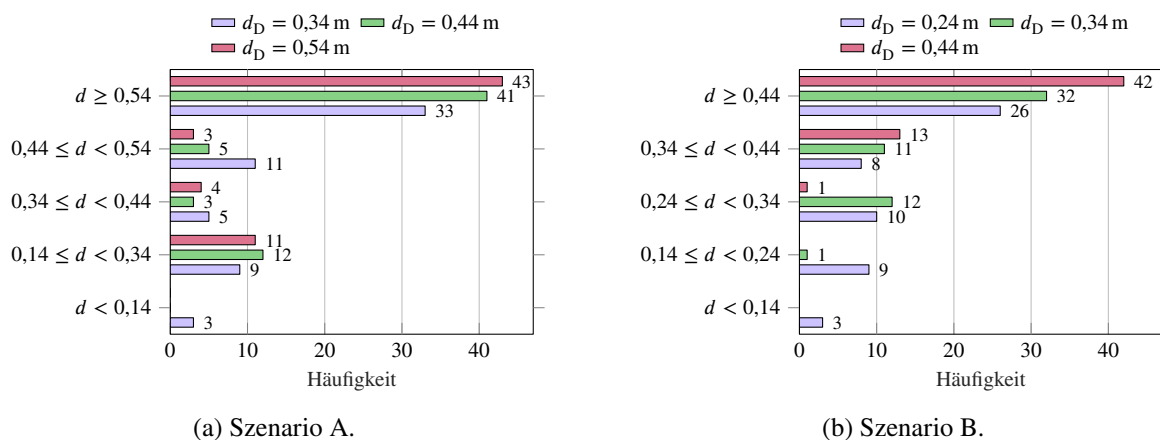


Abbildung 6.1.: Verteilung der kürzesten Abstände zwischen Roboter und Mensch unter Variation des Schwellwerts  $d_D$  der Annäherungskosten. Für  $d < 0,14 \text{ m}$  wird die Nebenbedingung verletzt.

zu erkennen, dass für  $d_D = 0,34$  m in drei Instanzen der Planung die Abstände unterhalb von  $d_D^-$  liegen. Der Wirkungsbereich der Annäherungskosten reicht nicht aus, um auf die herannahende Person frühzeitig zu reagieren. Für  $d_D = 0,44$  m und damit  $0,2$  m oberhalb von  $d_D^-$  ist der Wirkungsbereich ausreichend und die drei unterschreitenden Instanzen verlagern sich in den Bereich  $[0,14$  m,  $0,34$  m). Gleichzeitig reduzieren sich die Häufigkeiten in den beiden darüber liegenden Bereichen um 8, die nun am oberen Ende der Skala liegen. Für  $d_D = 0,54$  m ist dieser Effekt nur noch marginal und es deutet sich eine Sättigung an. Dies ist auch daran zu erkennen, dass insgesamt rund ein Drittel der Instanzen unterhalb dieses Schwellwerts liegen und damit in diesen Wirkungsbereich eindringen, während es für  $d_D = 0,44$  m nur etwa ein Viertel sind.

In Szenario B aus Abbildung 6.1b liegen die Werte für  $d_D$  kollektiv  $0,1$  m unterhalb von denen aus Szenario A. Für  $d_D = 0,24$  m ist der Wirkungsbereich zu gering und es treten Unterschreitungen von  $d_D^-$  in drei Instanzen auf. Ein höherer Wirkungsbereich von  $d_D = 0,34$  m ist ausreichend und verteilt die drei Instanzen auf die nächsten beiden Intervalle. Insgesamt profitiert dieses Szenario mehr von einem höheren Wirkungsbereich, da dieser sich aufgrund der initialen Nähe zwischen Roboter und Mensch bereits auf den ersten Planungszyklus auswirkt und eine Reaktion im Horizont der MHP auslöst. Damit stehen der MHP grundsätzlich mehr Planungsschritte bis zum Erreichen der Person zur Verfügung und die Häufigkeit, mit der Abstände in die Wirkungsbereiche eindringen, reduziert sich.

Die Planungszeit  $T_p$  hat ebenfalls Einfluss auf  $d_D$ . Für  $T_p = 0,1$  s ist der minimal mögliche Wert für  $d_D$ , für den es zu keiner Unterschreitung von  $d_D^-$  kommt  $0,4$  m (A) bzw.  $0,27$  m (B). Bei doppelter Planungszeit erhöhen sich die Werte geringfügig auf  $0,42$  m (A) bzw.  $0,3$  m (B). Bei einer Planungszeit von  $T_p = 0,5$  s betragen sie bereits  $0,55$  m (A) bzw.  $0,4$  m (B). Je größer die Schwellwerte werden, desto größer ist der Raum den Personen effektiv im Arbeitsraum einnehmen. Während Kollisionsvermeidungen zwar möglich sind, werden die Bewegungsmöglichkeiten des Roboters und damit dessen Effektivität zunehmend eingeschränkt.

## Prädiktive Kollisionsvermeidung mittels Zustandsextrapolation

Neben der Optimierung der zukünftigen Roboterbewegung bietet sich ebenso die Betrachtung der zukünftigen Hindernisbewegung innerhalb des Horizonts an. Anders als bei Szenarien aus dem Weltraum oder mit anderen Maschinen, in denen die Bewegungen konstant oder vorprogrammiert sind, ist die Vorhersage der Bewegung von Personen mit mehr Unsicherheit behaftet und erschwert eine prädiktive Kollisionsvermeidung. Lernbasierte Verfahren wie von Aksan u. a. [Aks+19] zur Bewegungsprädiktion erzielen zwar vielversprechende Ergebnisse, sind mit ca.  $7$  Hz jedoch nicht schnell genug, um mit der Planungsfrequenz von  $10$  Hz mitzuhalten [Ren+23a]. Stattdessen wird an dieser Stelle die Kombination aus Schätzung von Geschwindigkeit und Beschleunigung mit einer Extrapolation verwendet, um zu demonstrieren, welchen Nutzen die MHP bereits aus einer einfachen Prädiktionsmethode zieht.

Da sich durch die primäre Bewegung der Person nur ihre Position im Raum verändert, wird im weiteren Verlauf die Translation der Hüfte als Zustandsvektor gewählt:

$$\mathbf{x}_H(t) := \mathbf{s}_H(t). \quad (6.5)$$

Die Vorgehensweise ist jedoch analog auf andere Körperteile und die Orientierung übertragbar. Polynome  $N_p$ -ten Grades approximieren den Verlauf der Position komponentenweise innerhalb

eines Fensters  $\mathbf{b} \in \mathbb{R}^{F_S-1}$  für das gleitende Zeitintervall  $t \in [t_n - (F_S - 1)T_S, t_n]$  mit  $F_S \in \mathbb{N}$ :

$$\tilde{s}_{x,H}(t) := \sum_{j=0}^{N_p} (t - t_n + (F_S - 1)T_S)^j \psi_j, \quad (6.6)$$

mit dem zeitlichen Abstand  $T_S \in \mathbb{R}^+$  zwischen zwei Messungen. Die Zeittransformation  $t - t_n + (F_S - 1)T_S$  erlaubt eine konstante Regressormatrix  $\mathbf{I} \in \mathbb{R}^{F_S \times N_p+1}$ , was sowohl die Berechnungszeit verkürzt, als auch der numerischen Stabilität dient. Das Fenster besteht aus  $F_S - 1$  vergangenen Datenpunkten und dem aktuellen Datenpunkt  $s_{x,H,F_S-1} := s_{x,H}(t_n)$ :

$$\underbrace{\begin{bmatrix} (0T_S)^0 & (0T_S)^1 & \dots & (0T_S)^{N_p} \\ (1T_S)^0 & (1T_S)^1 & \dots & (1T_S)^{N_p} \\ \vdots & \vdots & & \vdots \\ ((F_S - 1)T_S)^0 & ((F_S - 1)T_S)^1 & \dots & ((F_S - 1)T_S)^{N_p} \end{bmatrix}}_{\mathbf{I}} \underbrace{\begin{bmatrix} \psi_0 \\ \psi_1 \\ \vdots \\ \psi_{N_p} \end{bmatrix}}_{\boldsymbol{\psi}} = \underbrace{\begin{bmatrix} s_{x,H,0} \\ s_{x,H,1} \\ \vdots \\ s_{x,H,F_S-1} \end{bmatrix}}_{\mathbf{b}}. \quad (6.7)$$

Der Parametervektor  $\boldsymbol{\psi} \in \mathbb{R}^{N_p+1}$  wird im Sinne des kleinsten quadratischen Fehlers durch Lösen von  $\boldsymbol{\psi}^* = (\mathbf{I}^T \mathbf{I})^{-1} \mathbf{I}^T \mathbf{b}$  bestimmt. Durch Ableitung der Polynome ergeben sich für  $N_p \geq 2$  Schätzungen für die gegenwärtige Geschwindigkeit  $\tilde{\dot{s}}_H(t_n)$  und Beschleunigung  $\tilde{\ddot{s}}_H(t_n)$ :

$$\tilde{\dot{s}}_{x,H}(t_n) := \sum_{j=1}^{N_p} j((F_S - 1)T_S)^{j-1} \psi_j^*, \quad (6.8)$$

$$\tilde{\ddot{s}}_{x,H}(t_n) := \sum_{j=2}^{N_p} j(j-1)((F_S - 1)T_S)^{j-2} \psi_j^*. \quad (6.9)$$

Die Komponenten für y und z sind jeweils analog.

Mittels Integration werden die Zustände  $\tilde{\mathbf{x}}_{H,k}$  auf dem Planungshorizont für  $k = 1, 2, \dots, P \leq K$  extrapoliert:

$$\tilde{\mathbf{x}}_{H,k} = \mathbf{x}_H(t_n) + kT_T \tilde{\dot{\mathbf{x}}}_H(t_n) + \frac{1}{2}(kT_T)^2 \tilde{\ddot{\mathbf{x}}}_H(t_n), \quad (6.10)$$

wobei für  $k > P$  der letzte extrapolierte Zustand  $\tilde{\mathbf{x}}_{H,P}$  für alle  $k = P + 1, P + 2, \dots, K$  im Sinne eines Halteglieds nullter Ordnung repliziert wird. Die Extrapolationen der Hüftbewegung werden unter Annahme eines starren Skeletts auf die Körperteile in ihrer momentanen Konfiguration übertragen. Für die Zustände  $\tilde{\mathbf{x}}_{K,k}$  des Kopfes bedeutet das z. B.:

$$\tilde{\mathbf{x}}_{K,k} = \mathbf{x}_K(t_n) + \Delta \mathbf{x}_{H,k} \text{ mit } \Delta \mathbf{x}_{H,k} := \tilde{\mathbf{x}}_{H,k} - \tilde{\mathbf{x}}_{H,0}. \quad (6.11)$$

Die prädierten Zustände der Hüllkörper zu den Zeitpunkten  $t(n, k > 0)$  lassen sich zeitlich synchron mit den Zuständen  $\mathbf{x}_k$  des Roboters in (6.1) und (6.3) berücksichtigen.

Die Einfachheit der Prädiktionsmethode geht zulasten der Generalisierbarkeit auf ein breites Spektrum von Bewegungen unterschiedlicher Dynamik. So ist es erforderlich, die Fensterbreite  $F_S$  und Prädiktionslänge  $P$  auf die Bewegung abzustimmen. Erstere besitzt dabei die Aufgabe einer Glättung. Dadurch, dass mehr Datenpunkte zur Bestimmung des Polynoms herangezogen werden als rechnerisch notwendig, wird der Einfluss jedes Punktes relativiert und Rauschen oder verdeckungsbedingte Sprünge bei der Messung geglättet. Gleichzeitig wird eine Trägheit eingeführt, die bei Zustandsänderungen – wie dem Beginn einer Bewegung oder dem Anhalten –

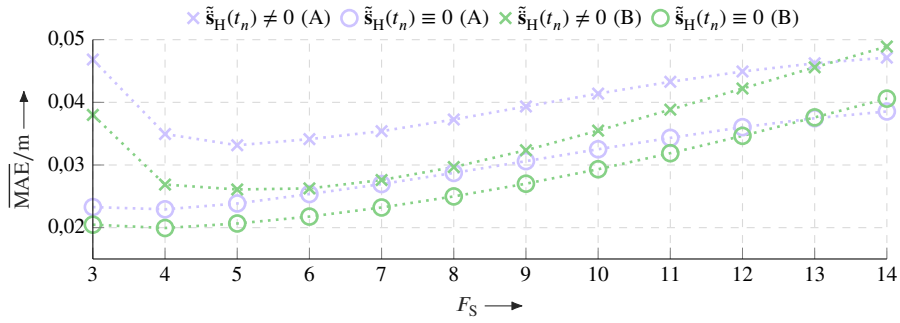


Abbildung 6.2.: Einfluss der Fensterbreite auf den  $\overline{\text{MAE}}$  über alle Prädiktionen der Bewegung für jedes Szenario unter Verwendung von  $T_S = 0,04$  s,  $P = 3$  und  $N_P = 2$ .

zu Abweichungen führt. Zur Einstellung von  $F_S$  zeigt Abbildung 6.2 den Einfluss auf die Prädiktionsgenauigkeit in beiden Szenarien. Die Prädiktionsgenauigkeit wird mit dem  $\overline{\text{MAE}}$ -Maß<sup>1</sup> angegeben, das den MAE aller Prädiktionen mittelt, die alle  $T_S = 0,04$  s entlang des Szenarios mit einer Prädiktionslänge von  $P = 3$  und Polynomgrad  $N_P = 2$  erhoben werden. Der zugrundeliegende MAE wird über die Abweichung der prädizierten Punkte zu den paarweise linear interpolierten Messwerten gebildet. Neben der Extrapolation mit konstanter Beschleunigung ist auch eine mit konstanter Geschwindigkeit ( $\tilde{s}_H(t_n) \equiv 0$ ) angegeben. Bei Annahme einer konstanten Beschleunigung ist der Einfluss von Rauschen bei einer zu kleinen Fensterbreite deutlich erkennbar. Der Einfluss der Trägheit bei längerem Fenster fällt weniger schwer ins Gewicht und das, obwohl beide Szenarien mit dem Beginn und Ende einer Bewegung wechselnde Bewegungszustände enthalten. Das Optimum liegt bei  $F_S = 5$  für beide Szenarien. Unter der Annahme einer konstanten Geschwindigkeit reduziert sich die Wichtigkeit einer Glättung, da sich Rauschen in der Geschwindigkeit weniger gravierend auf die Extrapolation auswirkt als in der Beschleunigung. Obwohl Anlaufen oder Anhalten nicht abgebildet wird, erreicht sie eine bessere Prädiktionsgenauigkeit als bei konstanter Beschleunigung. Eine fehlerbehaftete Schätzung der Beschleunigung führt zu größeren Abweichungen bei der Genauigkeit als bei einer Annahme von  $\tilde{s}_H(t_n) \equiv 0$ . Die Tendenzen für größere  $F_S$  in beiden Szenarien bleiben erhalten. Das Optimum liegt in beiden Szenarien bei  $F_S = 4$ .

Die Prädiktionsgüte nimmt naturgemäß mit fortlaufender Prädiktionslänge ab. Zudem kommen die Umstände, dass die Extrapolation als Prädiktionsmethode keine weiteren Informationen verarbeitet als den gegenwärtigen Zustand  $\mathbf{x}_H(t_n)$  und diesen komponentenweise isoliert betrachtet. Kontextwissen über den Gelenkraum des Menschen wie in Pereira u. a. [Per+17] oder den Arbeitsraum verbessert insbesondere bei Armbewegungen die Prädiktionsgenauigkeit. Um die Planung nicht mit entarteten Prädiktionen zu behindern, gilt es für die Bewegungen beider Szenarien geeignete Prädiktionslängen  $P$  zu ermitteln. Hierfür zeigt Abbildung 6.3 erneut den MAE als Prädiktionsfehler beider Szenarien für verschiedenen Prädiktionslängen mit jeweils optimaler Fensterbreite. Auch hier zahlt sich die Annahme einer konstanten Geschwindigkeit in Form einer höheren Genauigkeit aus. Der Prädiktionsfehler kann sich bei den Potentialen ungünstig mit deren Aktivierungsschwellwert  $d_D$  addieren und eine Reaktion der MHP hervorrufen, obwohl das Hindernis in der Zukunft einen Abstand besitzt, der keine Reaktion benötigt. Da der maximale  $\overline{\text{MAE}}$  über alle Prädiktionen pro Szenario in etwa doppelt so groß ist wie der  $\overline{\text{MAE}}$ , wird ein MAE angesetzt der kleiner ist als 0,05 m. Damit gilt im weiteren Verlauf eine Prädiktionslänge von  $P = 4$ , eine Fensterbreite von  $F_S = 4$  und die Annahme einer konstanten Geschwindigkeit

<sup>1</sup>Mittlerer MAE (Mean Absolute Error)

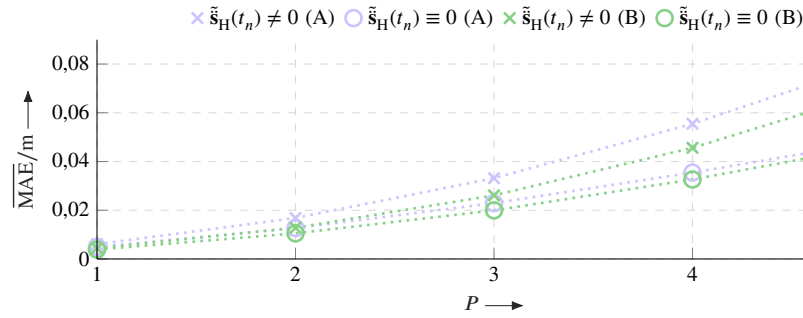


Abbildung 6.3.: Einfluss der Prädiktionslänge auf den  $\overline{\text{MAE}}$  über alle Prädiktionen der Bewegung für jedes Szenario unter Verwendung von  $T_S = 0,04$  s,  $N_P = 2$  und jeweils optimaler Fensterbreite  $F_S$ .

( $\tilde{\tilde{s}}_H(t_n) \equiv 0$ ). Abschließend demonstriert Abbildung 6.4 die auf diese Weise erzielten Prädiktionen in beiden Szenarien.

Die vorangegangene Auswertung zur Kollisionsvermeidung wird nachfolgend um die Hindernisprädiktion erweitert. Tabelle 6.1 stellt dafür die Fälle mit  $d(\cdot) \geq d_D^-$  unter Nutzung unterschiedlicher Prädiktionen mit denen der Schnappschussmethode ( $P = 0$ ) für beide Szenarien gegenüber. Ebenfalls angegeben sind Fälle einer perfekten Prädiktion, in denen die MHP die Bewegung in entsprechender Länge bereits kennt. Um den Unsicherheiten bei der Prädiktion Rechnung zu tragen, wird  $K_D = 4$  gesetzt. Weitere Angaben befinden sich in Anhang B.4.2. In beiden Szenarien erlaubt eine Prädiktion mit  $P = 4$  die Verringerung von  $d_D$  auf 0,26 m, um die Bewegung oberhalb von  $d_D^-$  zu halten. Dies entspricht in Szenario A einer Reduzierung um 0,14 m gegenüber der Schnappschussmethode. Szenario B profitiert weniger stark von der Prädiktion, da diese erst brauchbare Ergebnisse liefert, wenn die Bewegung bereits eintritt. In Anbetracht der relativ kurzen Bewegung und dichten Anordnung von Roboter und Mensch befindet sich die MHP mit dem ersten Zustand bereits aktiv in der Kollisionsvermeidung. Eine ideale Prädiktion, wie z. B. für  $P = 10$ , deckt die Entscheidung des Menschen zur unmittelbaren Ausführung einer Bewegung für beide Szenarien mit ab und erlaubt den Wirkungsbereich um 0,1 m auf  $d_D = 0,16$  m zu verkleinern. Von einer noch längeren (ebenfalls idealen) Prädiktion von  $P = 30$  wird in beiden Szenarien nicht profitiert.

Ähnlich wie für  $P = 0$  ist das minimale  $d_D$  für  $P = 4$  bei  $T_p = 0,2$  s in beiden Szenarien annähernd gleich, in diesem Fall bei 0,26 m (A) und 0,27 m (B). Erst bei  $T_p = 0,5$  s muss  $d_D$  für Szenario A auf 0,44 m erhöht werden, was dank der Prädiktion von  $P = 4$  um 0,11 m weniger ist

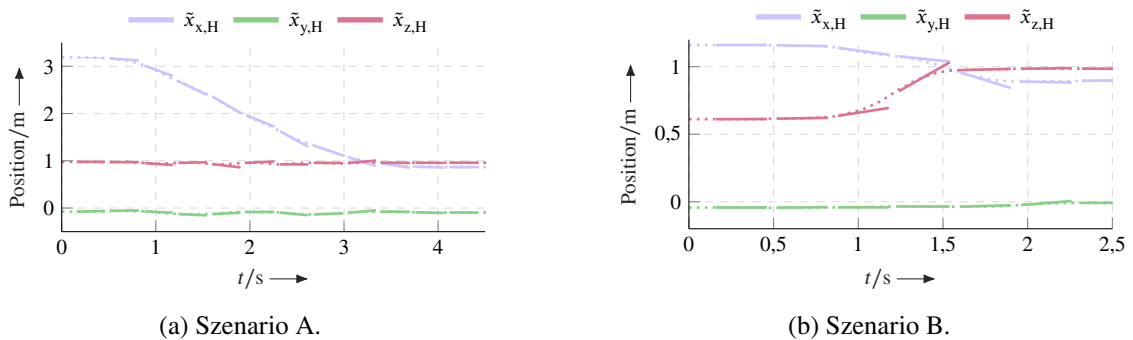


Abbildung 6.4.: Prädiktionen der x-, y- und z-Komponenten der Translation des Hüllkörpers der Hüfte beider Szenarien unter Verwendung von  $T_S = 0,04$  s,  $P = 4$ ,  $F_S = 4$  und  $N_P = 2$ . Die gemessenen Verläufe sind gepunktet hinterlegt.

Tabelle 6.1.: Fälle von eingehaltenem (✓) und verletztem (✗) Schwellwert  $d_D^- = 0,14$  m für verschiedene Aktivierungsschwellwerte  $d_D$  und Prädiktionslängen sowie -arten in den Szenarien A und B.

$d_D/m$	Szenario A				Szenario B			
	$P = 0$	$P = 4$	$P = 10$ (ideal)	$P = 30$ (ideal)	$P = 0$	$P = 4$	$P = 10$ (ideal)	$P = 30$ (ideal)
0,15	✗	✗	✗	✗	✗	✗	✗	✗
0,16	✗	✗	✓	✓	✗	✗	✓	✓
0,24	✗	✗	✓	✓	✗	✗	✓	✓
0,26	✗	✓	✓	✓	✗	✓	✓	✓
0,27	✗	✓	✓	✓	✓	✓	✓	✓
0,34	✗	✓	✓	✓	✓	✓	✓	✓
0,40	✓	✓	✓	✓	✓	✓	✓	✓
0,44	✓	✓	✓	✓	✓	✓	✓	✓

als bei  $P = 0$ . In Szenario B mildert die Prädiktion die Erhöhung von  $d_D$  nicht ab und es bleibt bei 0,4 m. Mit einer idealen Prädiktion von  $P = 10$  zeigt sich die MHP auch bei Planungszeiten von  $T_p = 0,5$  s weitestgehend robust und erlaubt einen Wirkungsbereich von 0,16 m (A) bzw. 0,17 m (B).

Die MHP profitiert damit bereits von einfachen Prädiktionen. Wie die Ergebnisse zu den idealen Prädiktionen zeigen, profitiert die MHP auch von komplexeren Ansätzen, die nicht allein auf der Basis einer Bewegung agieren, sondern aus einem größeren Kontext die Entscheidung des Menschen zu einer bevorstehenden Bewegung schätzen [Abu+19]. Damit können gewonnene Ressourcen bei den Planungszeiten in aufwändigere Planungsprobleme investiert oder die Wirkungsbereiche soweit reduziert werden, dass auch Kollaborationen auf engstem Raum im Gegensatz zu einer reinen Koexistenz möglich sind. Auch der Erhöhung der betrachteten Geschwindigkeit auf  $1,6 \text{ m s}^{-1}$ , wie sie von der ISO 13855 [ISO10] verwendet wird, lässt sich mit einer Prädiktion entgegenwirken. Da sich zum Rumpf auch die Geschwindigkeiten der Arme addieren, ist stellenweise mit noch höheren Geschwindigkeiten zu rechnen.

## 6.2. Annähern und Folgen dynamischer Zielzustände

Neben dynamischen Hindernissen besitzt die MHP die Möglichkeit, dynamische Zielzustände  $\mathbf{x}_f(t)$  anzufahren und zu folgen. Dies ist ebenfalls in Weltraumanwendungen bei der Beseitigung von Weltraumschrott [Ryb+17] oder für Übergabeszenarien in der MRK relevant, in denen sich der Übergabepunkt als Resultat einer zwischen Mensch und Roboter gegenseitig beeinflussten Bewegung ergibt [Nem+19]. Entsprechend essentiell ist die Annäherung an den konvergierenden Zielzustand und damit die Fähigkeit in Richtung dynamischer Zielzustände zu planen.

Manipulations- und Handhabungsaufgaben werden häufig im Arbeitsraum als Posen  $\zeta_f(t)$  des Endeffektors, bestehend aus Translation  $\mathbf{s}_{EE,f}(t) \in \mathbb{R}^3$  und Rotation  $\Theta_{EE,f}(t) \in \text{SO}(3)$ , definiert:

$$\zeta_f(t) := (\mathbf{s}_{EE,f}(t), \Theta_{EE,f}(t)). \quad (6.12)$$

Zur Anwendung in der MHP ist eine Überführung dieser Vorgaben in den Gelenkraum notwendig. Die naheliegendste Möglichkeit dazu ist die Nutzung einer analytischen inversen Kinematik. Sie bietet den Vorteil, dass alle gleichwertigen und gültigen Lösungen  $\mathcal{X}_{AIK}$  bestimmbar und damit für die MHP nutzbar sind:

$$\mathcal{X}_{AIK} := \{ \mathbf{x} \in \mathcal{X}_A \mid \mathbf{T}(\mathbf{x}) \rightsquigarrow (\mathbf{s}_{EE,f}, \Theta_{EE,f}) \}. \quad (6.13)$$

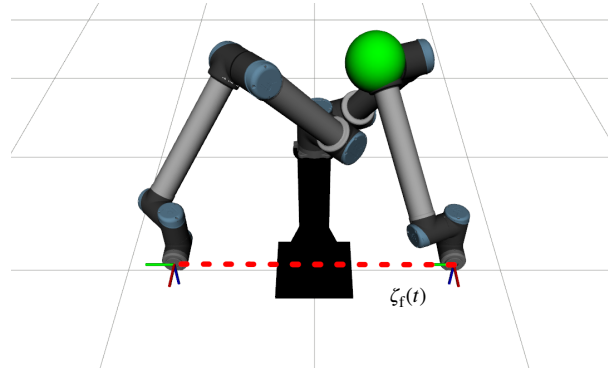


Abbildung 6.5.: Roboter im Anfangs- und Endzustand des linear verlaufenden Zielzustands  $\zeta_f(t)$ . Der Roboter startet abseits des gekennzeichneten Pfades. Das Hindernis ist nur im letzten Versuch vorhanden.

Gibt es mehrere, aber endlich viele Lösungen, wird der finale Zielzustand  $\mathbf{x}_f^*$  über ein Kriterium  $\hat{J}_f(\mathbf{x})$  für die MHP gewählt:

$$\mathbf{x}_f^* := \arg \min_{\forall \mathbf{x} \in \mathcal{X}_{\text{AIK}}} \hat{J}_f(\mathbf{x}). \quad (6.14)$$

Beispielsweise jener, der am nächsten am gegenwärtigen Roboterzustand liegt:

$$\hat{J}_f(\mathbf{x}) := \|\mathbf{x} - \mathbf{q}(t_n)\|_2. \quad (6.15)$$

Die Herleitung einer analytischen inversen Kinematik ist potentiell aufwändig und der Erfolg hängt vom kinematischen Aufbau des Roboters ab. Redundante Roboter erschweren die Herleitung zudem und benötigen sekundäre Ziele, welche die zusätzlichen Freiheitsgrade eliminieren [Bui+18]. Alternativ zur analytischen inversen Kinematik lässt sich eine lokale Lösung  $\mathbf{x}_f^*$  durch numerische Verfahren ermitteln oder durch Erweiterung der Kostenfunktion um die Definition (6.13) während der MHP implizit berechnen [Krä+21c].

Ein weiteres Verfahren, das neben der MHP zum Folgen dynamischer Endeffektorposen imstande ist, wird von Siciliano u. a. [Sic+09] als IKA<sup>2</sup> vorgestellt. Der Algorithmus arbeitet im Sinne eines Reglers mit Proportionalverstärkung  $\mathbf{K} \in \mathbb{R}^{6 \times 6}$  auf einer Regelabweichung  $\mathbf{e}(t) \in \mathbb{R}^6$  im Arbeitsraum und transformiert diese als Endeffektorgeschwindigkeit über die geometrische Jacobimatrix  $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times N}$  des Roboters in den Gelenkraum, wo sie als Stellgröße fungiert:

$$\mathbf{u}_n^*(0) = \mathbf{J}^{-1}(\mathbf{q}(t_n))(\mathbf{K}\mathbf{e}(t) + \dot{\zeta}_f(t)). \quad (6.16)$$

Über eine Vorsteuerung der Endeffektorsollgeschwindigkeit  $\dot{\zeta}_f(t)$  ist dies eine effektive und beliebte Methode Trajektorien zu folgen. Für eine detailliertere Beschreibung sei auf Siciliano u. a. [Sic+09] verwiesen.

Die Auswertung vergleicht das Folgeverhalten beider Methoden, wobei ihnen nur die momentane Zielpose  $\zeta_f(t_n)$  bekannt ist und nicht der zukünftige Verlauf. Die Vorsteuerung des IKA wird daher nicht verwendet. Auf die Auswertung einer klassischen Trajektorienfolgeregelung, die mit beiden Methoden zwar möglich ist, jedoch konzeptionell auf einer anderen Stufe als die MHP agiert, wird im Rahmen dieser Arbeit verzichtet. Die dynamische Zielpose bewegt sich mit sinusförmiger Geschwindigkeit periodisch  $\pm 0,5$  m in y-Richtung vor dem Roboter ( $x=z=1$ ). Die Orientierung ist fest und wie in Abbildung 6.5 ausgerichtet. Für einen Zyklus benötigt die Zielpose 10 s, was im Durchschnitt  $0,1 \text{ m s}^{-1}$  und im Maximum ca.  $0,31 \text{ m s}^{-1}$  entspricht. Beide Methoden beginnen

<sup>2</sup>Inverse Kinematik Algorithmus

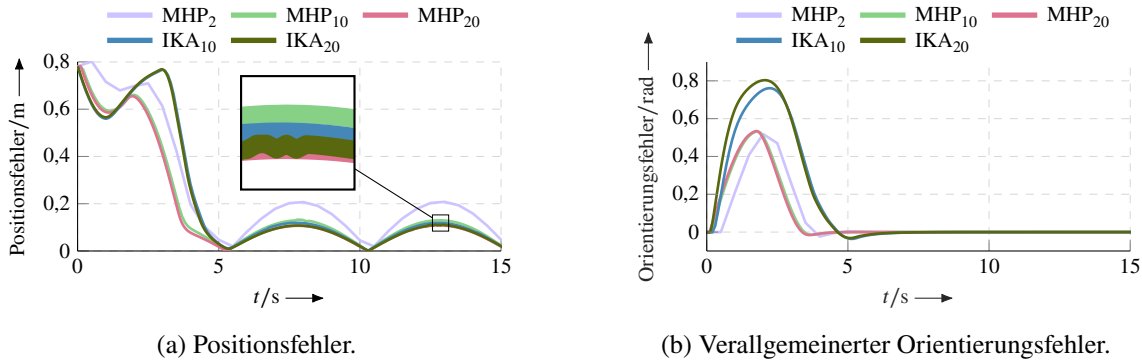


Abbildung 6.6.: Positions- und Orientierungsfehler von MHP und IKA für verschiedene Wiederhol frequencies.

im selben Startzustand, der sich nicht auf der Trajektorie befindet, um neben dem Folgeverhalten auch das Übergangsverhalten abzudecken. Weitere Angaben zu dem Szenario befinden sich in Anhang B.4.3.

Abbildung 6.6 vergleicht den Positions- und verallgemeinerten Orientierungsfehler<sup>3</sup> beider Methoden für verschiedene Wiederhol frequencies der Planungen. Unabhängig von der Wiederhol frequency fährt die MHP zielstrebig auf den beweglichen Zielzustand als der IKA, wie an der Erhöhung der Fehler im Bereich um 2,5 s in beiden Abbildungen erkennbar ist. Die Umwandlung von Arbeitsraumgeschwindigkeiten in Gelenkwinkelgeschwindigkeiten mit der Roboterjacobimatrix  $\mathbf{J}(\cdot)$  ist nur im aktuellen Arbeitspunkt  $\mathbf{q}(t_n)$  gültig, womit der IKA konzeptionell nur für kleine Schrittweiten geeignet ist. Insbesondere bei Punkt-zu-Punkt Bewegungen entstehen größere Regelabweichungen als beim dichten Folgen einer Trajektorie. Diese lassen sich über entsprechend kleine Gewichte  $\mathbf{K}$  oder Beschränkungen der Regelabweichung  $\mathbf{u}_n^*(0)$  zwar reduzieren, gehen jedoch zulasten der Folgegenauigkeit. Damit gilt es schließlich, immer einen Kompromiss zwischen der Qualität einer Punkt-zu-Punkt-Bewegung und dem Folgeverhalten zu finden, der in diesem Fall zugunsten des Folgeverhaltens ausgelegt ist. Im eingeschwungenen Zustand erreicht der IKA mit 0,12 m einen geringeren Folgefehler in der Position als die MHP mit 0,13 m bei gleicher Wiederhol frequency von 10 Hz. Die Orientierung spielt beim Folgeverhalten in diesem Beispiel eine untergeordnete Rolle und beide Methoden erreichen einen Wert nahe Null. Die Ergebnisse zeigen des Weiteren, wie das Folgeverhalten beider Methoden von einer höheren Wiederhol frequency profitiert, die aufseiten des IKA in der Implementierung für dieses Experiment theoretisch bei bis zu 5 kHz liegt. Die Grenzen der Wiederhol frequency der MHP wurden im vorherigen Kapitel ausführlich thematisiert und erreichen je nach Aufgabe 20 Hz, mit denen die MHP mit einem Positionsfehler von 0,11 m gleichauf mit dem IKA ist. Eine Wiederhol frequency von nur 2 Hz lässt den Positionsfehler auf 0,21 m ansteigen und ist für die Dynamik der Zielpose zunehmend zu langsam. Selbiger Effekt ergibt sich bei einer höheren Geschwindigkeit der Zielpose.

Abschließend wird das Szenario um eine Kugel ( $S_p$ ) als statisches Hindernis ergänzt, welche den nach oben gerichteten Ellenbogen (drittes Gelenk) des Roboters blockiert, wenn dieser nach 1,7 s die Zielpose bei  $y = 0,5$  m einnehmen möchte. Da der IKA über keine Kollisionsvermeidung verfügt, ist eine Kollision mit dem Hindernis nur dann vermeidbar, wenn der Ellenbogen des Roboters im Startzustand bereits nach unten zeigt. Durch die fortlaufende Bewertung aller

<sup>3</sup>Der verallgemeinerte Orientierungsfehler ist der Betrag des Winkels der Achse-Winkel Darstellung zwischen Soll- und Istorientierung.

möglichen Zielzustände zur gegenwärtigen Zielpose über Gleichungen (6.14) und (6.15), ist die MHP imstande in einen alternativen kollisionsfreien Zielzustand zu wechseln, in welchem der Ellenbogen des Roboters nach unten zeigt. Dadurch wird wie Abbildung 6.7 mit dem Verlauf des Positions- und verallgemeinerten Orientierungsfehlers zeigt, die Annäherung verzögert. Bis ca. 2,5 s verläuft der Positionsfehler identisch zum hindernisfreien Fall, ehe ein weiteres Abnehmen durch den Konfigurationswechsel verzögert wird. Nach 7,5 s ist die Zielpose erreicht und es beginnt das Folgeverhalten. Zu denselben Zeitpunkten weicht auch der Orientierungsfehler vom hindernisfreien Fall ab, der zwischenzeitlich bereits konvergiert ist. Aufgrund der Nähe zum aktuellen Roboterzustand verbleibt der Ellenbogen des Roboters im weiteren Verlauf nach unten gerichtet und umgeht weitere Konfrontationen.

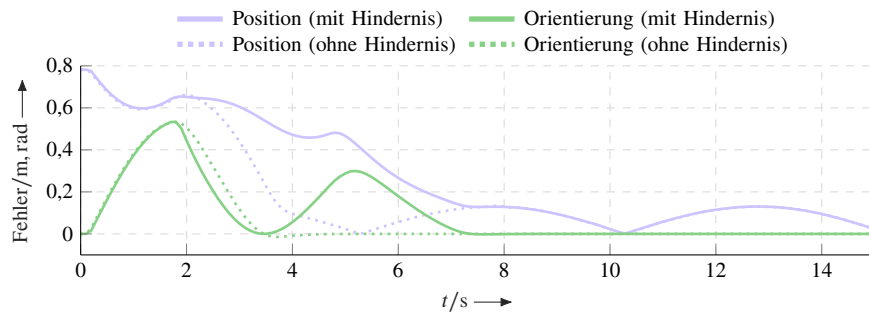


Abbildung 6.7.: Verlauf des Positions- und verallgemeinerten Orientierungsfehlers mit und ohne Hindernis.



# 7

## Dynamische Priorisierung von sekundären Kostentermen

Nachdem das vorherige Kapitel mit dynamischen Hindernissen und Zielzuständen bereits erste dynamische Aspekte eines Szenarios behandelt, folgt in diesem Kapitel die Betrachtung von dynamischen Planungskriterien. Die Kostenfunktion besteht bereits aus Termen zur Bewertung der Nähe zum Zielzustand sowie Hindernissen und zur Reduzierung von Bewegungen. Neben diesen primären Termen sind weitere sekundäre Kriterien denkbar, die jedoch potentiell mit einem oder mehreren der vorhandenen Terme konkurrieren. Abschnitt 7.1 definiert den Begriff der Konkurrenz zwischen Termen genauer und stellt zwei Varianten vor, um Hierarchien zwischen primären und sekundären Termen zu realisieren. In Abschnitt 7.2 wird dieses Konzept unter praktischen Anforderungen bewertet, indem die MHP über die dynamische Priorisierung von sekundären Kostentermen mit einem Elastizitätsbewusstsein für die Verwendung an einem Roboter mit elastischen Armkörpern ausgestattet wird.

Die Ergebnisse dieses Kapitels stammen aus den Publikationen [Krä+20a; Krä+21b].

### 7.1. Einbettung konkurrierender Kostenterme in die Planung

Die Kostenterme  $c_1(\cdot)$  und  $c_2(\cdot)$  werden als konkurrierend bezeichnet, wenn sie unterschiedliche Minima besitzen:

$$\arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_1(\mathbf{x}) \neq \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_2(\mathbf{x}). \quad (7.1)$$

Zur einfacheren Veranschaulichung sind  $c_1 : \mathbb{R}^N \rightarrow \mathbb{R}^+$  und  $c_2 : \mathbb{R}^N \rightarrow \mathbb{R}^+$  konvexe Kostenterme mit jeweils einem Minimum  $\mathbf{x}_1^*$  und  $\mathbf{x}_2^*$ . Es sind  $c_1(\cdot)$  der primäre und  $c_2(\cdot)$  der sekundäre Kostenterm. Werden beide Terme über eine Summe verknüpft, ergibt sich eine Kompromisslösung:

$$c_\Sigma(\mathbf{x}) := c_1(\mathbf{x}) + c_2(\mathbf{x}), \quad (7.2)$$

$$\begin{aligned} \Rightarrow \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_\Sigma(\mathbf{x}) &\neq \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_1(\mathbf{x}), \\ \Rightarrow \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_\Sigma(\mathbf{x}) &\neq \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_2(\mathbf{x}). \end{aligned} \quad (7.3)$$

Auf eine Gewichtung wird der Einfachheit halber verzichtet. Die Kompromisslösung wird bildlich in Abbildung 7.1 verdeutlicht, in der zwei quadratische Kostenfunktionen für  $N = 2$  mit unterschiedlichen Minima  $\mathbf{x}_1^* = (-1 \ 0)^T$  und  $\mathbf{x}_2^* = (1 \ 0)^T$  addiert werden und sich ein neues Minimum  $\mathbf{x}_\Sigma^* = \mathbf{0}$  bildet. Genaugenommen stellen der Kostenterm  $c_X(\cdot)$  aus Gleichung (3.2)

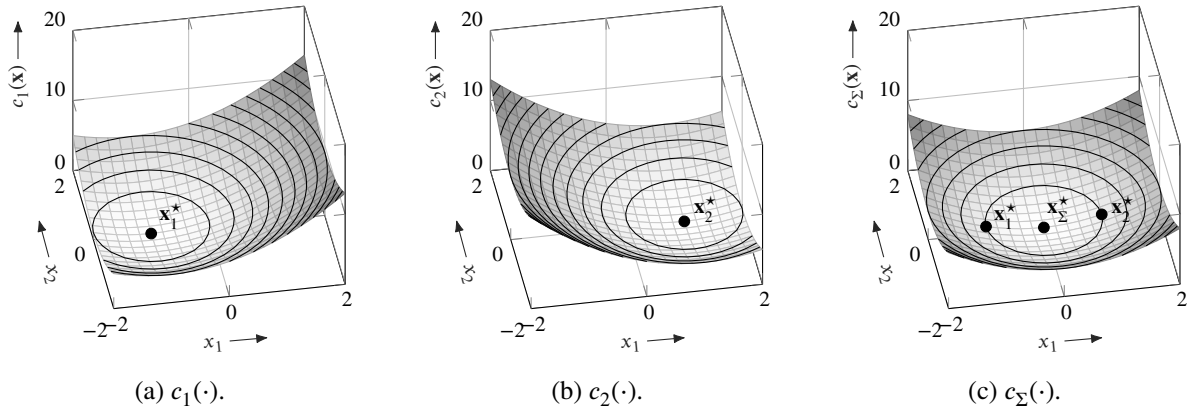


Abbildung 7.1.: Minima der Kostenterme  $c_1(\cdot)$  und  $c_2(\cdot)$  sowie das der Summe  $c_\Sigma(\cdot)$  aus beiden Termen.

und  $c_H(\cdot)$  aus Gleichung (6.3) bereits den Fall konkurrierender Kosten dar. Befindet sich ein Hindernis für den Zielzustand  $\mathbf{x}_f$  innerhalb des Aktivierungsschwellwerts, fährt die MHP eine Kompromisslösung an, die zwischen beiden Minima liegt. Dieses Verhalten ist soweit unkritisch, da es der Verantwortung einer übergeordneten Aufgabenplanung obliegt, Zielzustände so zu wählen, dass sie frei vom Einfluss der Hindernisse sind. Kapitel 6 hat gezeigt, wie mit der Hindernisprädiktion der Einflussbereich von Hindernissen gesenkt und damit mehr Raum für gültige Zielzustände ermöglicht wird.

Ein sekundärer Kostenterm ordnet sich dem primären auf unterschiedliche Weise unter. Wird  $c_2(\cdot)$  aus Gleichung (7.2) z. B. nur bei Bedarf betrachtet, bildet sich der Kompromiss nur temporär aus:

$$c_A(\mathbf{x}) := c_1(\mathbf{x}) + \sigma(\alpha(t))c_2(\mathbf{x}). \quad (7.4)$$

Dabei ist  $\sigma : \mathbb{R} \rightarrow [0, 1]$  eine über das Signal  $\alpha : \mathbb{R} \rightarrow \mathbb{R}$  bedarfsgesteuerte und problemspezifische glatte Aktivierungsfunktion. Wenn die Berücksichtigung von  $c_2(\cdot)$  aktiv dem Bedarf  $\alpha(t)$  und damit der Aktivierung entgegenwirkt ( $\sigma(\cdot) \rightarrow 0$ ), wird auch der Kompromiss zunehmend abgebaut und es gilt:

$$\arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_A(\mathbf{x}) \xrightarrow{\sigma(\cdot) \rightarrow 0} \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_1(\mathbf{x}). \quad (7.5)$$

Diese Art der Überlagerung ist zum Beispiel in Fällen nützlich, in denen das Erreichen eines Zielzustands über  $c_1(\cdot)$  allein nicht ausreicht, sondern das Abklingen des Bedarfsfalls erfordert, um die Planungsaufgabe erfolgreich zu beenden.

Die bedarfsgesteuerte Überlagerung hat einen reaktiven Charakter. Wenn  $c_2(\cdot)$  hingegen zum proaktiven Entgegenwirken des Bedarfsfalls berücksichtigt werden soll, ist eine möglichst lange Priorisierung von  $c_2(\cdot)$  erforderlich, mit einer letztendlichen Unterordnung zu  $c_1(\cdot)$  zur Vermeidung des Zielkompromisses. Realisiert wird diese Priorisierung über eine multiplikative Überlagerung:

$$c_B(\mathbf{x}) := c_1(\mathbf{x})(1 + c_2(\mathbf{x})), \quad (7.6)$$

$$\begin{aligned} \Rightarrow c_1(\mathbf{x}) > 0 \wedge c_2(\mathbf{x}) \geq 0 &\Rightarrow c_B(\mathbf{x}) \geq c_1(\mathbf{x}), \\ \Rightarrow c_1(\mathbf{x}) = 0 \wedge c_2(\mathbf{x}) \geq 0 &\Rightarrow c_B(\mathbf{x}) = 0 \Rightarrow \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_B(\mathbf{x}) = \arg \min_{\forall \mathbf{x} \in \mathcal{X}} c_1(\mathbf{x}). \end{aligned} \quad (7.7)$$

Hierbei ist  $c_2(\cdot)$  als rein sekundäres Kriterium zu verstehen, welches sich dem Erfolg der Planungsaufgabe letztendlich unterordnet. Abbildung 7.2 zeigt den Aufbau dieser Kostenfunktion für  $N = 2$  und den Minima  $\mathbf{x}_1^* = \mathbf{x}_B^* = (-1 \ 0)^T$  und  $\mathbf{x}_2^* = (1 \ 0)^T$ .

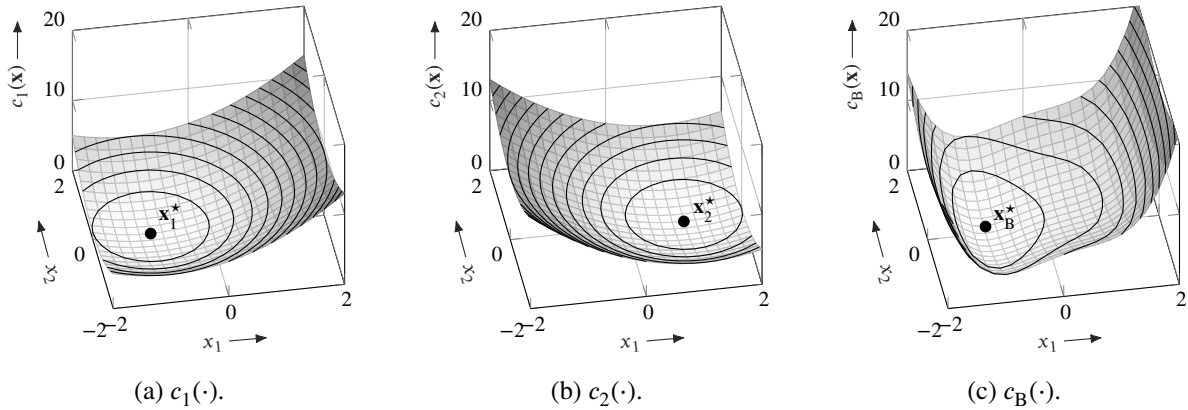


Abbildung 7.2.: Minima der Kostenterme  $c_1(\cdot)$  und  $c_2(\cdot)$  sowie das der priorisierten Überlagerung  $c_B(\cdot)$  von beiden Termen.

## 7.2. Elastizitätsbewusste Planung für Roboter mit flexiblen Armkörpern

Da Kollisionen zwischen Mensch und Roboter bei Arbeiten auf engstem Raum nicht gänzlich auszuschließen sind, bieten elastische Armkörper die Möglichkeit die Kollisionsenergie über Elastizitäten abzubauen. Diese Elastizitäten begünstigen auf der anderen Seite jedoch auch ungewollte Schwingungen und erfordern aktive Maßnahmen zur Dämpfung. Bei ausschließlich gelenkseitiger Schwingungsdämpfung hängt die Effektivität der Dämpfung vom Zugriff der Gelenke auf die Schwingungen ab. Über geometrische Maße lässt sich der Zugriff in Abhängigkeit des Roboterzustands beschreiben [Joh+17]. Diese Maße werden ursprünglich für die Schwingungsdämpfung verwendet und ermöglichen eine Adaption der Reglerparameter [Mus+19]. Darüber hinaus lassen sich diese Maße auch als sekundäre Kostenterme bei der Trajektorienplanung berücksichtigen, um bei der Bewegung zum Ziel jene Zustände zu präferieren, in denen die Effektivität der Schwingungsdämpfung möglichst hoch ist. Dadurch lassen sich spontan auftretende Schwingungen effektiver dämpfen und Manipulationsaufgaben präziser durch- oder weiterführen. Diese Idee wird im Folgenden mit Hilfe der beiden Varianten (7.4) und (7.6) zur Berücksichtigung sekundärer Kosten verfolgt.

### Definition des sekundären Kostenterms und der Aktivierungsfunktion

Die Effektivität eines Gelenks Schwingungen innerhalb einer sogenannten Schwingungsebene zu dämpfen, wird im Folgenden als Steuerbarkeit bezeichnet – nicht zu verwechseln mit der Bedeutung dieses Begriffs im systemtheoretischen Sinne. Als Schwingungsebenen  $E_i$  mit  $i = 1, 2, \dots, N_E$  werden die räumlichen Freiheitsgrade von elastischen Armkörpern bezeichnet [Joh+17]. Sie werden als Tupel eines Stütz- und Normalenvektors  $\mathbf{p} \in \mathbb{R}^3$  bzw.  $\mathbf{n} \in \mathbb{R}^3$  im Weltkoordinatensystem repräsentiert:

$$E_i := (\mathbf{p}_i(\mathbf{x}), \mathbf{n}_i(\mathbf{x})). \quad (7.8)$$

Stütz- und Normalenvektor hängen dabei über die Vorwärtskinematik vom Roboterzustand  $\mathbf{x}$  ab. Abbildung 7.3 illustriert die Schwingungsebenen und weitere im Folgenden verwendete Größen anhand eines Beispiels.

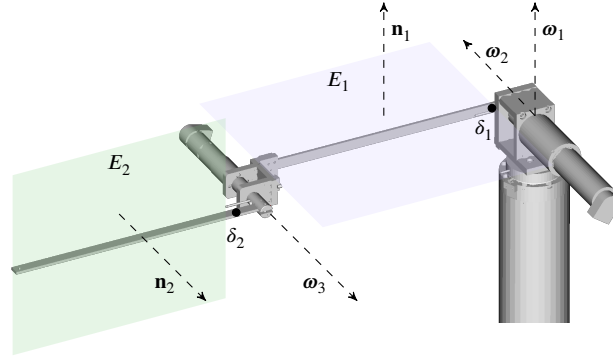


Abbildung 7.3.: Aufbau des Roboters TUDORA mit eingezeichneten Drehachsen  $\omega_j$ , Schwingungsebenen  $E_i$ , Normalenvektoren  $\mathbf{n}_i$  und Messpunkten der Dehnungsmessstreifen  $\delta_i$ .

Das im weiteren Verlauf vorgestellte Maß von John u. a. [Joh+17] zur Berechnung der Steuerbarkeit besitzt gegenüber nicht-geometrischen Maßen den Vorteil eines geringeren Rechenaufwands. Darüber hinaus wird kein dynamisches Modell des Roboters benötigt und es existiert eine eindeutige Zuordnung jeder Schwingungsebene zum jeweiligen Gelenk mit der größten Steuerbarkeit im aktuellen Zustand. Diese Vorteile ermöglichen zeitkritische Anwendungen, wie die aktive Schwingungsdämpfung oder die elastizitätsbewusste Planung in der MHP.

Sei  $\omega_j(\mathbf{x}) \in \mathbb{R}^3$  die Rotationsachse des  $j$ -ten Gelenks abhängig vom Zustand  $\mathbf{x}$ , dann ist die Steuerbarkeit  $\gamma_{j,i} : \mathbb{R}^N \rightarrow [0, 1]$  bezüglich der Schwingungsebene  $E_i$  definiert als:

$$\gamma_{j,i}(\mathbf{x}) := \left| \omega_j^T(\mathbf{x}) \mathbf{n}_i(\mathbf{x}) \right|. \quad (7.9)$$

Eine Steuerbarkeit von  $\gamma_{j,i}(\cdot) = 0$  ist dabei am schlechtesten und  $\gamma_{j,i}(\cdot) = 1$  am besten. Gilt  $\left| \omega_j(\cdot) \right| = 1$  lässt sich alternativ schreiben:

$$\gamma_{j,i}(\mathbf{x}) := \left| \cos(\phi_{j,i}) \right|, \quad (7.10)$$

wobei  $\phi_{j,i}$  der Winkel zwischen  $\omega_j$  und  $\mathbf{n}_i$  ist. Die Steuerbarkeitsmatrix  $\mathbf{\Gamma}(\mathbf{x}) \in \mathbb{R}^{N \times N_E}$  fasst die Steuerbarkeiten aller Gelenke und Schwingungsebenen zusammen:

$$\mathbf{\Gamma}(\mathbf{x}) := \begin{bmatrix} \gamma_{1,1}(\cdot) & \gamma_{1,2}(\cdot) & \cdots & \gamma_{1,N_E}(\cdot) \\ \gamma_{2,1}(\cdot) & \gamma_{2,2}(\cdot) & \cdots & \gamma_{2,N_E}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{N,1}(\cdot) & \gamma_{N,2}(\cdot) & \cdots & \gamma_{N,N_E}(\cdot) \end{bmatrix}. \quad (7.11)$$

Der sekundäre Kostenterm  $c_2(\mathbf{x})$  bestraft schließlich geringe Steuerbarkeiten und ist wie folgt definiert:

$$c_2(\mathbf{x}) := W_S \sum_{i=1}^{N_E} \sum_{j=1}^N (1 - \gamma_{j,i}(\mathbf{x})), \quad (7.12)$$

mit dem Gewicht  $W_S \in \mathbb{R}^+$ . Bei maximaler Steuerbarkeit gilt  $c_2(\mathbf{x}) = 0$  und minimaler Steuerbarkeit  $c_2(\mathbf{x}) = W_S N N_E$ .

Zur Auslegung der Kostenfunktion (7.4) wird zusätzlich eine bedarfsgesteuerte Aktivierungsfunktion  $\sigma(\cdot)$  benötigt. Der Bedarfsfall ist die Existenz von Schwingungen, welcher über die im Folgenden als Schwingungsstärke bezeichnete Größe  $\alpha_n$  zum Zeitpunkt  $t_n$  quantifiziert wird.

Zur Berechnung von  $\alpha_n$  wird zunächst das Spektrum  $(\mathbf{A}_{i,n}, \mathbf{\Omega}_{i,n})$  jedes Dehnungssignals  $\delta_i(t)$  mit  $i = 1, 2, \dots, N_E$  über eine diskrete Fouriertransformation  $\mathcal{F}\{\cdot\}$  innerhalb eines gleitenden Zeitintervalls  $t \in [t_n - F_{FT}T_{FT}, t_n]$  ermittelt:

$$\mathcal{F}\{\mathbf{b}_{FT}\mathbf{\Xi}\} \rightsquigarrow (\mathbf{A}_{i,n}, \mathbf{\Omega}_{i,n}), \quad (7.13)$$

$$\begin{aligned} \mathbf{A}_{i,n}^T &:= [A_{i,n,1} \quad A_{i,n,2} \quad \dots \quad A_{i,n,l} \quad \dots \quad A_{i,n,N_{FT}}], \\ \mathbf{\Omega}_{i,n}^T &:= [\Omega_{i,n,1} \quad \Omega_{i,n,2} \quad \dots \quad \Omega_{i,n,l} \quad \dots \quad \Omega_{i,n,N_{FT}}], \end{aligned} \quad (7.14)$$

hierbei umfasst  $\mathbf{b}_{FT} \in \mathbb{R}^{F_{FT}}$  die diskreten Messwerte mit Abtastbreite  $T_{FT} \in \mathbb{R}^+$  bestehend aus  $F_{FT} \in \mathbb{N}$  Messwerten, worin  $\delta_i(t_n)$  den aktuellsten repräsentiert. Ferner stellt  $\mathbf{\Xi} \in \mathbb{R}^{F_{FT}}$  die diskreten Werte einer Fensterfunktion dar. Zur Reduzierung des Fenstereffekts<sup>1</sup> an den für gewöhnlich nicht periodischen Intervallgrenzen wird das Hann-Fenster [Nut81] verwendet, welches das Signal an den Rändern des Intervalls auf Null blendet und damit Periodizität zum Preis eines veränderten Messsignals erzeugt. Die Schwingungsstärke wird jetzt über die Summe der Amplitudenquadrate aller Schwingungsebenen gebildet:

$$\alpha_n := \sum_{i=1}^{N_E} \sqrt{\mathbf{A}_{i,n}^T \mathbf{S} \mathbf{A}_{i,n}}, \quad (7.15)$$

mit der binären Selektionsmatrix  $\mathbf{S} \in \mathbb{B}^{N_{FT}}$ :

$$\mathbf{S} := \text{diag} \left( 0 \quad \dots \quad 0 \quad \underbrace{1}_{l^-} \quad \dots \quad \underbrace{1}_{l^+} \quad 0 \quad \dots \quad 0 \right). \quad (7.16)$$

Das Indexintervall  $[l^-, l^+]$  realisiert unter Hinzunahme von  $\mathbf{\Omega}_{i,n}^T$  einen Bandpassfilter mit  $\mathbf{\Omega}^- := \mathbf{\Omega}_{i,n,l^-}$  und  $\mathbf{\Omega}^+ := \mathbf{\Omega}_{i,n,l^+}$ , der die Möglichkeit bietet das Signal vom Mittelwert und einem Teil des Rauschens in hohen Frequenzlagen zu befreien. Auf der anderen Seite bilden die Fenster- und Abtastbreite  $F_{FT}$  bzw.  $T_{FT}$  implizite Grenzen bei der geringsten und höchsten rekonstruierbaren Frequenz und bedürfen daher eine roboter- und problemspezifische Parametrierung.

Die Schwingungsstärke  $\alpha_n$  aktiviert schließlich über die Sigmoidfunktion  $\sigma(\cdot)$  den sekundären Kostenterm:

$$\sigma(\alpha_n) := \frac{1}{2} \left( \tanh(\kappa(\alpha_n - \bar{\alpha})) + 1 \right). \quad (7.17)$$

Sie nimmt bei  $\alpha_n = \bar{\alpha} \in \mathbb{R}^+$  den Wert 0,5 an und besitzt dort die Steigung  $\frac{\kappa}{2} \in \mathbb{R}^+$ .

## Parametrierung der Aktivierungsfunktion

Für die folgenden Versuche kommt der Roboter TUDORA<sup>2</sup> (siehe Anhang A.2) aus Abbildung 7.3 zum Einsatz. Er besitzt  $N = 3$  aktuierte Freiheitsgrade und zwei elastische Armkörper, welche die  $N_E = 2$  Schwingungsebenen bilden und an denen Dehnungswerte  $\delta_i(t)$  über Dehnungsmessstreifen erfasst werden. Aufgrund des Aufbaus fallen die Gelenkwinkel  $x_2$  und  $x_3$  mit den Winkeln  $\phi_{1,1}$  und  $\phi_{2,2}$  zusammen und es ergibt sich eine Steuerbarkeitsmatrix von:

$$\mathbf{\Gamma}(\mathbf{x}) = \begin{bmatrix} |\cos(x_2)| & 0 \\ 0 & |\cos(x_3)| \\ 0 & 1 \end{bmatrix}. \quad (7.18)$$

<sup>1</sup>Durch das endliche Beobachtungsintervall werden bei der Fouriertransformation nicht-periodische Signale verwendet, die im Spektrum zu künstlichen Frequenzanteilen führen.

<sup>2</sup>TU Dortmund *Omni-Elastic Robot Adapted*

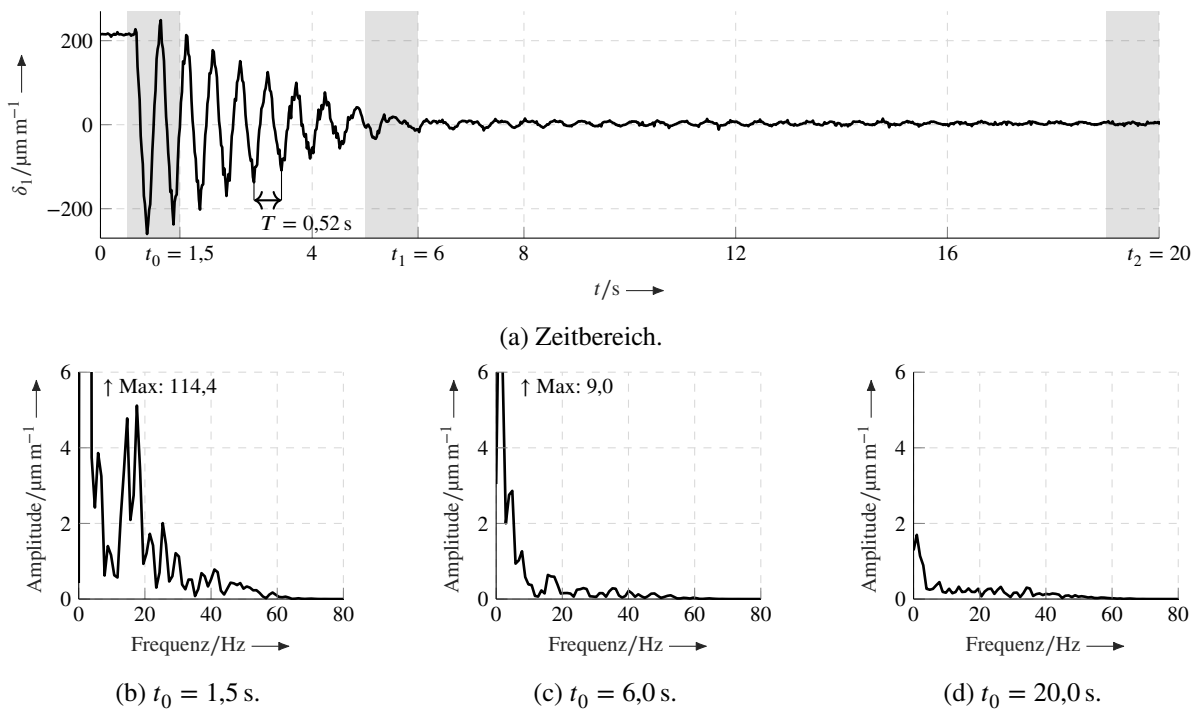


Abbildung 7.4.: Darstellung einer abklingenden Schwingung am zweiten Armkörper im Zeitbereich (a) und die Amplituden von drei Zeitfenstern im Frequenzbereich (b-d) zur Ermittlung der Periodendauer und Bandpassfrequenzen.

Da die zweite Schwingungsebene stets vom dritten Gelenk erreicht wird, reduziert sich die in  $c_2(\cdot)$  zu berücksichtigende Steuerbarkeit auf:

$$c_2(\mathbf{x}) = 1 - |\cos(x_2)| \approx 1 - \sqrt{\cos^2(x_2) + 0,001}. \quad (7.19)$$

Entsprechend werden im weiteren Verlauf auch nur die Schwingungen des ersten elastischen Armkörpers (entspricht dem zweiten Armkörper des Roboters) betrachtet, obgleich die Methode allgemein auch auf weitere Armkörper skaliert. Die Betragsfunktion  $|\cdot|$  wird über eine differenzierbare Näherung approximiert.

Die Wahl der Abtastbreite  $T_{FT}$  begrenzt gemäß Abtasttheorem die höchste rekonstruierbare Frequenz auf  $\frac{1}{2T_{FT}}$ . Bei einer Abtastzeit von 1 ms entspricht dies ausreichenden 500 Hz. Die Anzahl der Messpunkte  $F_{FT}$  hat über die resultierende zeitliche Intervallbreite von  $F_{FT}T_{FT}$  direkten Einfluss auf die kleinste rekonstruierbare Frequenz. Die Frequenz der Schwingung ist im Wesentlichen von der mechanischen Konstruktion abhängig. Abbildung 7.4 zeigt eine natürlich abklingende Schwingung, mit einer Periodendauer von 0,52 s. Um mindestens diese Frequenz im Spektrum zu erfassen, gilt  $F_{FT} \geq \frac{0,52}{T_{FT}} = 520$ . Darüber hinaus ist der Algorithmus der schnellen diskreten Fouriertransformation von Frigo u. a. [Fri+05] am effizientesten, wenn die Anzahl der Datenpunkte pro Intervall einer Zweierpotenz entspricht. Aus diesem Grund wird  $F_{FT} = 1024$  gesetzt, was einer kleinsten erkennbaren Frequenz von ungefähr 0,97 Hz entspricht.

Abbildung 7.4 zeigt zusätzlich Spektren zu drei in grau schraffierten Zeitpunkten des Zeitsignals. Diese drei Zeitpunkte sind so gewählt, dass eine einsetzende und eine fast abgeklungene Schwingung sowie Rauschen enthalten sind. Anhand des Spektrums bei 1,5 s ist zu erkennen, dass die Schwingungen die Amplituden mehrerer Frequenzen beeinflussen. Zur Elimination des Gleichanteils im Signal wird  $\Omega^- = 1,95$  Hz gesetzt. Verglichen mit dem zweiten Spektrum lässt

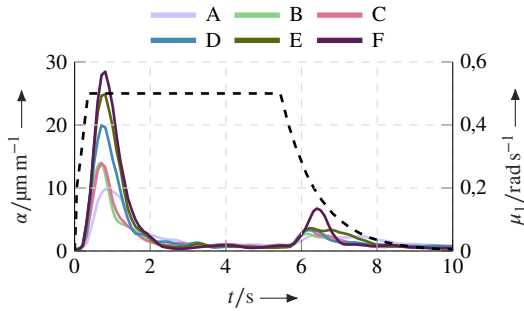


Abbildung 7.5.: Schwingungsstärken und Stellgröße (gestrichelt) für Bewegungen des ersten Gelenks bei unterschiedlichen festen Winkeln des zweiten und dritten Gelenks.

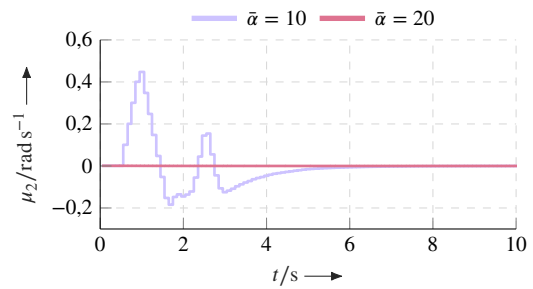


Abbildung 7.6.: Stellgröße am zweiten Gelenk, die durch unterschiedlich sensible Schwellwerte  $\bar{\alpha}$  Reaktionen der Aktivierung erfährt.

sich ab ca. 46 Hz kein wahrnehmbarer Unterschied erkennen. Es handelt sich hauptsächlich um Rauschen. Entsprechend wird  $\Omega^+ = 46,86$  Hz gesetzt.

Wird die Aktivierungsfunktion  $\sigma(\cdot)$  zu früh aktiv, löst sie bereits bei kleinsten Schwingungen aus. Abbildung 7.5 stellt hierfür die Schwingungsstärken für die freie Bewegung ohne Elastizitätsbewusstsein dar. Die sechs Start- und Zielzustände A bis F bilden je eine Konfiguration und sind so gewählt, dass nur das erste Gelenk eine Bewegung vollzieht. Die Winkel des zweiten und dritten Gelenks sind bei Start und Ziel jeweils gleich und entsprechen einem unterschiedlich weit gestreckten Arm. Genauere Angaben befinden sich in Anhang B.5.1. Die Stellgröße am ersten Gelenk ist ebenfalls in Abbildung 7.5 dargestellt und wirkt zu Beginn der Bewegung bewusst mit maximaler Beschleunigung von  $1 \text{ rad s}^{-2}$ . In Verbindung mit der Trägheit der Armkörper führt dies zu Schwingungen, die stärker ausfallen, je weiter der Arm gestreckt ist ( $A \rightarrow F$ ). Das Anhalten verläuft aufgrund der Wechselwirkungen zwischen  $Q$  und  $R$  (vgl. Abschnitt 3.1) sanfter, weshalb auch die Schwingungen kleiner ausfallen als beim Anfahren. Abbildung 7.6 verdeutlicht über die Stellgrößen zweiten Gelenk, welchen Einfluss der Schwellwert  $\bar{\alpha}$  bei  $\bar{\alpha} = 10$  oder  $\bar{\alpha} = 20$  auf die Aktivierung und damit die Bewegung hat. Abgebildet ist der Fall F. Der Wert  $\bar{\alpha} = 10$  ist zu niedrig und der Bedarfsfall wird durch kleine Schwingungen während der Beschleunigung bereits ausgelöst. Hinzu kommt, dass durch die asymmetrische Bauweise der Armkörper die Trägheit des zweiten Antriebs bei Bewegungen zu Torsionskräften führt, die kleine Schwingungen in der ersten Schwingungsebene anregen. Da das zweite Gelenk direkten Einfluss auf die Steuerbarkeit besitzt, erzeugt die Aktivierung immer wieder Auf- und Abwärtsbewegungen und wirkt dem Bedarfsfall nicht entgegen. Diese Überkompensation tritt für  $\bar{\alpha} = 20$  nicht auf, weshalb dieser Wert für die übrigen Experimente verwendet wird. Ferner wird für  $\sigma(\cdot)$  der Wert  $\kappa = 5$  gewählt, wodurch sich der Großteil der Aktivierung im Intervall  $\alpha_n \in [-0,5, 0,5]$  konzentriert.

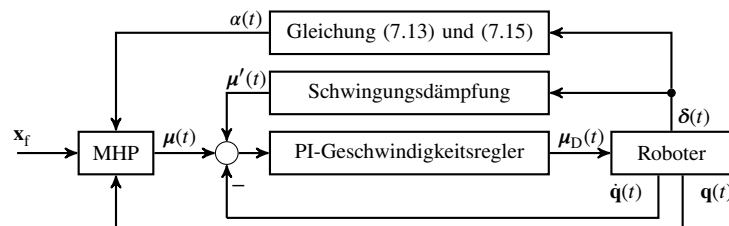


Abbildung 7.7.: Struktureller Aufbau der elastizitätsbewussten MHP mit PI-Geschwindigkeitskaskade, Schwingungsdämpfung und Berechnung der Schwingungsstärke.

## Bewertung der dynamischen Priorisierung

Die adaptive Schwingungsdämpfung von Muster u. a. [Mus+19] berechnet eine Korrekturge-  
 schwindigkeit  $\mu'(\cdot)$ , die zusammen mit der Stellgröße  $\mu(\cdot)$  der MHP als Sollwinkelgeschwin-  
 digkeit vom PI<sup>3</sup>-Geschwindigkeitsregler von Malzahn [Mal14] in Drehmomente überführt wird.  
 Abbildung 7.7 zeigt den erweiterten Aufbau. Der Einfachheit halber wird die Variante (7.4)  
 als Variante 1 und (7.6) als Variante 2 bezeichnet. Die Referenz bildet die MHP ohne Elastizi-  
 tätsbewusstsein. Es ist zu beachten, dass Variante 2 zwar ein globales Minimum im primären  
 Kostenterm besitzt, jedoch ebenfalls lokale Minima durch die Steuerbarkeit. Begegnet wird  
 diesen mit Nebenbedingung (2.5c) und einer ausreichenden Horizontlänge. Letzteres stellt ange-  
 sichts der Effizienz der MHP und der verringerten Freiheitsgrade des Roboters ( $N = 3$ ) keine  
 Einschränkungen dar.

Im ersten Versuch wird untersucht, inwieweit die Ansätze die Effektivität der Schwingungsdämp-  
 fung erhöhen. Hierfür wird der Endeffektor initial ausgelenkt und fixiert. Kurz bevor die MHP  
 mit der Planung beginnt, wird der Endeffektor abrupt aus der Fixierung gelöst und beginnt zu  
 schwingen. Es werden die Konfigurationen A, B und C von Start- und Zielzuständen aus dem  
 vorherigen Abschnitt wiederverwendet. Genauere Angaben zum Aufbau der Versuche befinden  
 sich in Anhang B.5.2.

Abbildung 7.8 vergleicht die Dehnungsmessungen zwischen der Referenz und Variante 1 zusam-  
 men mit der erreichten Steuerbarkeit in Konfiguration A. Es wird deutlich, wie die Schwingung  
 durch die erhöhte Steuerbarkeit von ca. 53 % bereits nach ca. 5 s annähernd vollständig gedämpft  
 ist. Abbildung 7.9 zeigt die Verläufe des zweiten Gelenkwinkels und demonstriert, wie diese  
 für beide Varianten vorübergehend vom Zielwert abweichen, um die Steuerbarkeit zu erhöhen.  
 Obwohl die Varianten das zweite Gelenk in unterschiedlichem Ausmaß beeinflussen, wird in  
 Abbildung 7.10 deutlich, wie ähnlich beide die Schwingungsdämpfung beeinflussen. Die Schwin-  
 gungsdämpfung nutzt die Vorteile der höheren Steuerbarkeit von 87,7 % durch Variante 2 nicht  
 aus.

Abbildung 7.11 vergleicht die Schwingungsstärke der Referenz mit der von Variante 2 in den  
 Konfigurationen B und C. Da die Schwingungsdämpfung bei ausreichender Steuerbarkeit nur  
 wenige Sekunden benötigt um den Arm zu stabilisieren, erreicht die Referenz innerhalb der  
 Bewegungszeit von ca. 5 s für eine anfängliche Steuerbarkeit von 20 % und mehr bereits ähnli-  
 che Ergebnisse wie Variante 2. Bei Restschwingungen in den letzten 4 s ermöglicht Variante 2

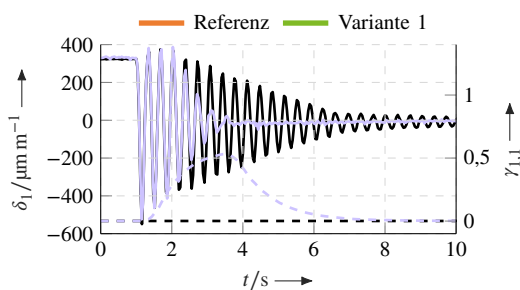


Abbildung 7.8.: Dehnungsmessungen und Steuerbarkeit (gestrichelt) zwischen Referenz und Vari-  
 ante 1 im ersten Versuch in Konfiguration A.

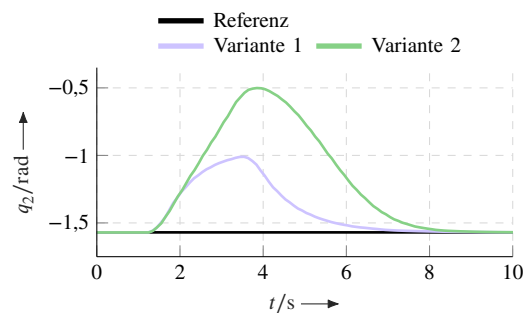


Abbildung 7.9.: Auswirkungen des Elastizitätsbe-  
 wusstseins auf den Verlauf des zweiten Gelenkwin-  
 kels für beide Varianten gegenüber der Referenz im  
 ersten Versuch in Konfiguration A.

<sup>3</sup>Proportional-Integral

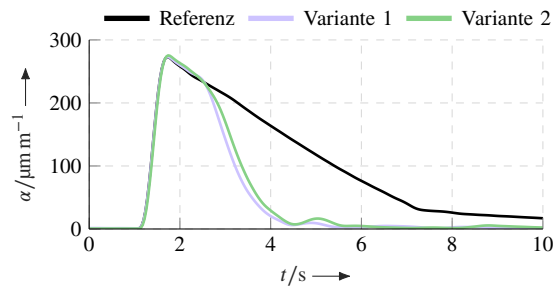


Abbildung 7.10.: Einfluss beider Varianten auf die Schwingungsstärke gegenüber der Referenz im ersten Versuch in Konfiguration A.

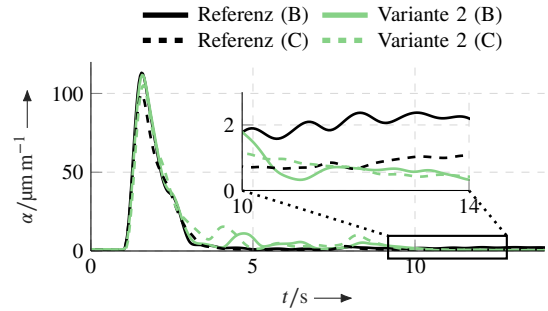


Abbildung 7.11.: Einfluss der Konfigurationen B und C auf den Verlauf der Schwingungsstärke bei Variante 2 im ersten Versuch.

eine bessere Dämpfung als die Referenz. Dieser Unterschied verschwindet allerdings ab einer anfänglichen Steuerbarkeit von 40 % und höher, wie es ab Konfiguration C der Fall ist.

Der zweite Versuch demonstriert den konzeptionellen Unterschied zwischen beiden Varianten durch Störungen beim Halten unterschiedlicher statischer Zielzustände der Konfigurationen. Die Störung wird wieder durch das Lösen des zuvor ausgelenkten Endeffektors aus der Fixierung erzeugt. Abbildung 7.12 zeigt die Dehnungsmessungen beider Varianten für den Zielzustand aus Konfiguration A. Da Variante 2 den Zielzustand priorisiert, eliminiert sie den Einfluss der Steuerbarkeit auf die Kostenfunktion, womit sie sich letztlich wie die Referenz in Abbildung 7.8 verhält. Im Gegensatz dazu verlässt Variante 1 den Zielzustand vorübergehend zugunsten einer besseren Steuerbarkeit (bis zu 53 %) und kehrt erst zurück, wenn die Schwingungen abgeklungen sind.

Abbildung 7.13 zeigt die Schwingungsstärke für die Zielzustände der Konfigurationen A und B und bestätigt die Ergebnisse aus dem vorangegangenen Versuch. Für die Schwingungsdämpfung ist eine anfängliche Steuerbarkeit von 20 % bereits ausreichend und die Vorteile einer weiteren Erhöhung auf bis zu 62 % durch Variante 1 nicht nutzbar.

Der dritte Versuch vergleicht die Effektivität der Schwingungsdämpfung hinsichtlich des proaktiven Verhaltens von Variante 2 mit dem reaktiven Verhalten von Variante 1. Die Störungen werden manuell über einen Holzhammer auf den Endeffektor eingepreßt. Abbildung 7.14 zeigt sowohl die Dehnungsmesswerte als auch die Steuerbarkeit für beide Varianten und Konfiguration A. Die Störung tritt nach etwa 3 s ein. Zu diesem Zeitpunkt hat Variante 2 eine Steuerbarkeit von 85 % und Variante 1 eine von 0 %. Sobald die Schwingungsstärke in die Nähe von  $\bar{\alpha}$  gelangt, erhöht

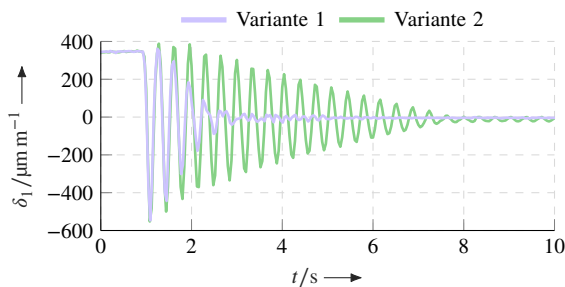


Abbildung 7.12.: Dehnungsmessungen beider Varianten im zweiten Versuch in Konfiguration A.

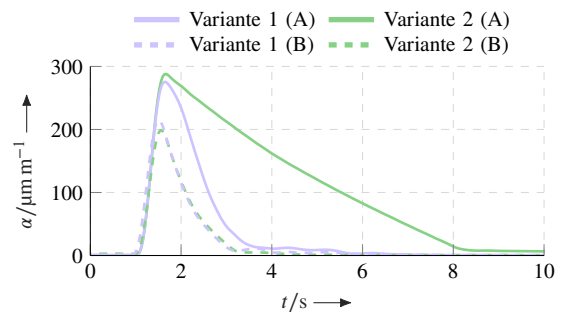


Abbildung 7.13.: Einfluss der Konfigurationen A und B auf den Verlauf der Schwingungsstärke beider Varianten im zweiten Versuch.

Variante 1 die Steuerbarkeit auf 53 %. In beiden Fällen klingen die Schwingungen gleichmäßig ab und zeigen keinen Vorteil durch das proaktive Verhalten von Variante 2. Dieses Ergebnis wird für die anderen Konfigurationen bestätigt. Die unterschiedlich hohen Schwingungsstärken zu Beginn der Störung begründen sich durch die natürliche Streuung der manuellen Ausführung.

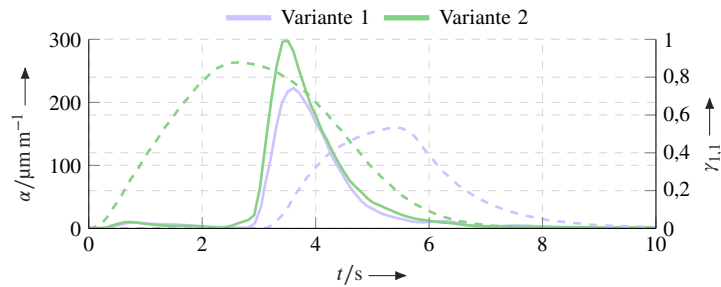


Abbildung 7.14.: Einfluss des proaktiven Verhaltens auf die Schwingungsstärke bei einer spontanen Störung während der Bewegung in Konfiguration A im dritten Versuch.

# 8

## Lokalität der Lösungsverfahren

Nachdem die MHP im zweiten Teil unter praktischen Gesichtspunkten in dynamischen Szenen bewertet wurde, folgt in diesem Kapitel einleitend für den dritten Teil dieser Arbeit die Analyse zu den Lokalitäten der Lösungsverfahren. Die vorgestellten gradientenbasierten Verfahren zur Lösung von NLP (3.29) zählen zu den lokalen Verfahren. Das bedeutet, sie nutzen lokale Informationen von Kostenfunktion und Nebenbedingungen, um den Lösungskandidaten iterativ zu verbessern. Ihnen gegenüber stehen globale Verfahren wie STOMP, die mehr Informationen nutzen, dafür aber auch mehr Berechnungszeit benötigen. Der Nachteil lokaler Verfahren bei nichtkonvexen Optimierungsproblemen ist, dass sie in die nächstbeste Lösung konvergieren, womit sie abhängig von einer geeigneten Initialisierung sind. Hinzu kommt, dass die Nebenbedingungen zur Kollisionsvermeidung und die Annäherungskosten das Planungsproblem zu einem nichtkonvexen Optimierungsproblem machen, bei dessen Lösung es sich nicht notwendigerweise um das globale, sondern potentiell um ein lokales Minimum handelt. Dies erfordert die Untersuchung der Frage nach einer geeigneten Initialisierung der Lösungsverfahren. Abschnitt 8.1 erläutert hierfür verschiedenen Formen und die Implikationen für die MHP. Abschnitt 8.2 stellt mit der erweiterten Initialisierung über eine parallele Explorationsplanung eine Methode zur Milderung der Folgen ungünstiger Initialisierungen vor.

Die Ergebnisse dieses Kapitels sind in der Publikation [Krä+22a] veröffentlicht.

### 8.1. Initialisierung des Optimierungsproblems

Abbildung 8.1 veranschaulicht den Einfluss der Initialisierungen  $\mathbf{z}^I$  und stellt die Menge  $\mathcal{Z} \subseteq \mathbb{R}^D$  aller Lösungsvektoren  $\mathbf{z}$  und dessen Untermenge  $\mathcal{Z}_A(\mathbf{x}_0, t_n)$  dar, die alle gültigen Lösungsvektoren einer Aufgabe  $A$  ausgehend von einem Anfangszustand  $\mathbf{x}_0$  zum Zeitpunkt  $t_n$  umfasst. Initialisierungen  $\mathbf{z}^I$  sind als Kreise dargestellt, die über die Pfeile in Lösungen  $\mathbf{z}_n^*$  konvergieren, die durch Punkte repräsentiert werden. Ungültige Initialisierungen konvergieren zu gültigen oder ungültigen Lösungen. Eine gültige Initialisierung führt hingegen immer auf eine gültige Lösung, da sie im Zweifel selbst die Lösung ist. Als ungültig werden Initialisierungen oder Lösungen bezeichnet, die mindestens eine Nebenbedingung verletzen. Obwohl bei der vollständigen Diskretisierung  $\mathbf{z}$  sowohl aus Zustandsfolge  $\mathbf{x}_{1:K-1}$  und Steuerfolge  $\mathbf{u}_{0:K-2}$  besteht, werden für Initialisierungen im Folgenden nur die Bildungsvorschriften der Zustände  $\mathbf{x}_k$  behandelt. Die zugehörigen Stellgrößen  $\mathbf{u}_k$  lassen sich daraus über Gleichung (3.28) ermitteln.

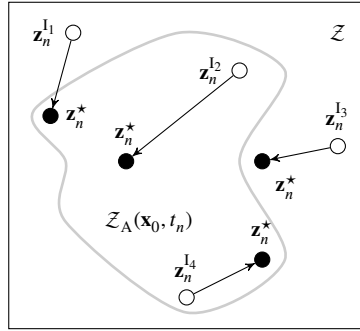


Abbildung 8.1.: Qualitative Darstellung der Lokalität der Lösungsverfahren über die Menge  $\mathcal{Z}_A(\mathbf{x}_0, t_n) \subset \mathcal{Z}$  aller zum Zeitpunkt  $t_n$  gültigen Lösungsvektoren die im Zustand  $\mathbf{x}_0$  beginnen. Abhängig von den Initialisierungen  $\mathbf{z}_n^I$  (Kreise) ergeben sich unterschiedliche Lösungen  $\mathbf{z}_n^*$  (Punkte).

Die einfachste gültige Initialisierung ohne Vorwissen der Umgebung ist  $\mathbf{z}^A$ , die dem aktuellen Zustand<sup>1</sup> entspricht:

$$\mathbf{z}^A : \quad \mathbf{x}_k = \mathbf{q}(t_n). \quad (8.1)$$

Ein Nachteil dieser Initialisierung ist, dass sie einen Stillstand repräsentiert und die Lösung bei lokalen Verfahren in dessen Nähe liegt. Im ungünstigen Fall stellt auch die Lösung einen Stillstand dar, was im nächsten Kapitel genauer betrachtet wird. Im Folgenden werden weitere bei der MHP typische Initialisierungen und ihre Eigenschaften vorgestellt.

## Lineare Interpolation

Die lineare Interpolation zwischen Start- und Zielzustand ist eine häufig gewählte Initialisierung:

$$\mathbf{z}^{LI} : \quad \mathbf{x}_k = \mathbf{q}(t_n) + \frac{k(\mathbf{x}_f - \mathbf{q}(t_n))}{K}. \quad (8.2)$$

Sie ist ohne weiteres Vorwissen und mit wenig Aufwand durchführbar. In der Regel berücksichtigt sie jedoch keine Hindernisse und stellt damit potentiell eine ungültige Initialisierung dar. Hierbei kommt es in bestimmten Fällen zu Schwierigkeiten bei der Lösung, wie anhand des Beispiels in Abbildung 8.2 erläutert wird. Abgebildet ist der Zustandsraum eines Roboters mit  $N = 2$ , einem länglichen Armkörper (siehe Anhang A.3) und vier linear interpolierten Initialisierungen (siehe Anhang B.6.1). Die ungültigen Initialisierungen führen durch ein zylindrisches Hindernis, dessen Kollisionsmenge sowie Aktivierungsbereich des Potentials durch den inneren bzw. äußeren Bereich angegeben sind. Während die Gradienten  $\nabla_x c_X(\cdot)$  einheitlich gen Zielzustand gerichtet sind, zeigen die Gradienten der Potentiale  $\nabla_x c_H(\cdot)$  und Nebenbedingungen<sup>2</sup>  $\nabla_x h_k(\cdot)$  vor und hinter dem Hindernis in entgegengesetzte Richtungen. Anstatt die Trajektorie einheitlich über oder unter das Hindernis zu führen, wird sie auseinandergezogen bis die Systemnebenbedingung (3.29c) unter Berücksichtigung der Stellgrößenbeschränkung ausgereizt ist und der Gradient  $\nabla_x g_k(\cdot)$  für  $k = 0, 1, \dots, K - 2$  sie wieder zusammenzieht. In dieser Situation lässt sich keine gültige Lösung ermitteln. Buizza Avanzini u. a. [Bui+18] nutzen Wirbelfelder, um die sternförmige Richtung der Gradienten zu vermeiden, welche allerdings nur bei einer herannahenden Trajektorie effektiv sind und die gegensätzlichen Richtungen vor und hinter dem Hindernis

<sup>1</sup>Unter der Annahme, dass der Roboter gegenwärtig keine Nebenbedingungen verletzt.

<sup>2</sup>Der Einfachheit halber nur die skalaren Ungleichungen  $d_C^- - d(\mathcal{O}_1(\mathbf{x}_k), \mathcal{O}_2) \leq 0$  der Kollisionsvermeidung für  $k = 1, 2, \dots, K - 1$ .

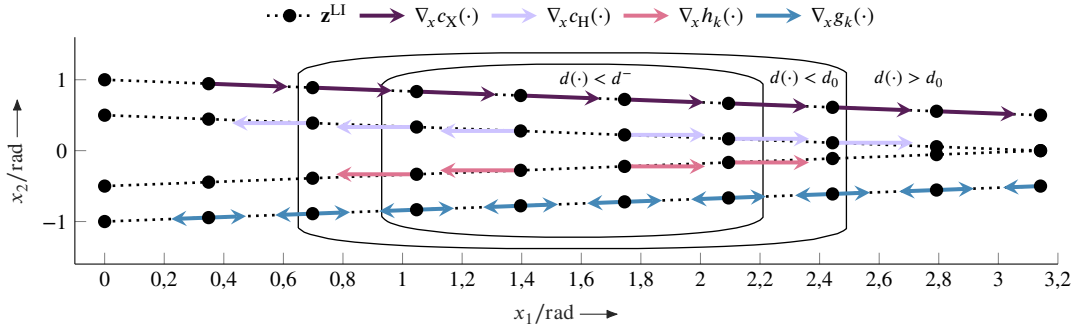


Abbildung 8.2.: Zweidimensionaler Zustandsraum mit Kollisionsmenge sowie Wirkungsbereich der Potentiale. Durch den Zustandsraum führen von links ( $\mathbf{q}(t_n)$ ) nach rechts ( $\mathbf{x}_f$ ) vier lineare Interpolationen, auf denen die Gradienten der Zielkosten, Kollisionsvermeidung, Annäherungskosten und Systemgleichung eingezeichnet sind.

bei der Initialisierung nicht auflösen. Merite- oder Filterfunktionen erlauben Lösungsverfahren eine temporäre Verschlechterung der Kosten zugunsten einer Verbesserung von verletzten Nebenbedingungen [Noc+06]. Dadurch kann es in Fällen, in denen der Horizont nur kurz hinter das Hindernis ragt, zu einer Verschiebung des hinteren Teils der Trajektorie vor das Hindernis und damit zur Auflösung des Konflikts und einer gültigen Lösung kommen. Das Eintreten dieses Falls lässt sich jedoch nur schwierig garantieren.

Alternativ zur einfachen linearen Interpolation ist die Variante  $\mathbf{z}^{\text{LI}+}$  mit einem Zwischenzustand  $\mathbf{x}_V \in \mathbb{R}^D$ :

$$\mathbf{z}^{\text{LI}+} : \quad \mathbf{x}_k = \begin{cases} \mathbf{q}(t_n) + \frac{k(\mathbf{x}_V - \mathbf{q}(t_n))}{K_V} & k = 0, 1, \dots, K_V \\ \mathbf{x}_V + \frac{(k - K_V)(\mathbf{x}_f - \mathbf{x}_V)}{K - K_V} & k = K_V + 1, K_V + 2, \dots, K \end{cases}, \quad (8.3)$$

mit  $K_V \in \mathbb{N} < K$ . Der Zwischenpunkt  $\mathbf{x}_V$  wird so gewählt, dass die Gradienten wie in Abbildung 8.3 zumindest anteilig in eine einheitliche Richtung aus der Kollision zeigen. In diesem Fall repräsentiert der Zwischenpunkt den aus Abschnitt 4.2 bekannten Fall, in dem sich die Primitive von Armkörper und Hindernis schneiden. Der Gradient der Annäherungskosten im Punkt  $\mathbf{x}_V$  resultiert aus der approximierten Penetrationsdistanz und ist ohne sie Null. Neben einem, können nach dem gleichen Prinzip auch mehrere Zwischenpunkte per Linearkombination verbunden werden. Die Wahl der Zwischenpunkte erfolgt z. B. über problemabhängige Heuris-

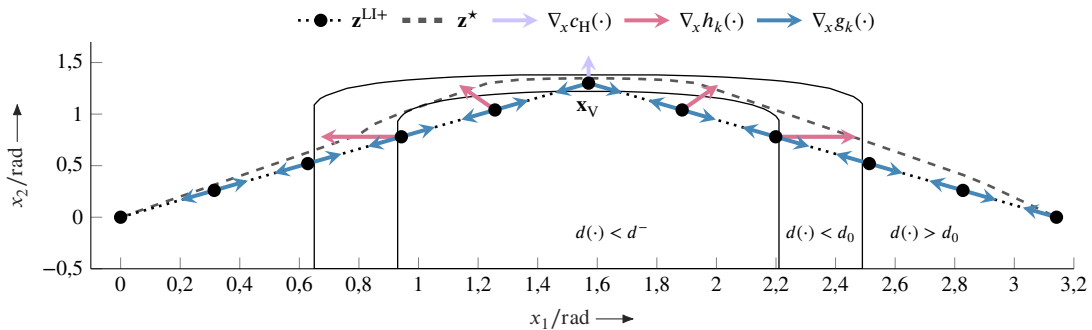


Abbildung 8.3.: Zweidimensionaler Zustandsraum mit Kollisionsmenge sowie Wirkungsbereich der Potentiale. Durch den Zustandsraum führt von links ( $\mathbf{q}(t_n)$ ) nach rechts ( $\mathbf{x}_f$ ) die lineare Interpolation mit Zwischenpunkt. Auf ihr sind die Gradienten der Kollisionsvermeidung, Annäherungskosten, und Systemgleichung eingezeichnet. Eine resultierende Lösung ist gestrichelt hinterlegt.

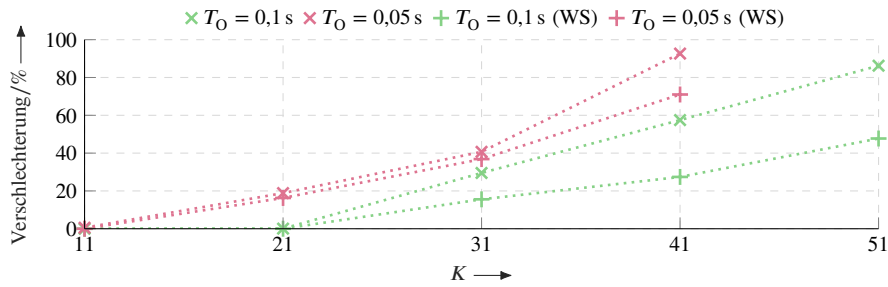


Abbildung 8.4.: Einfluss des Warmstarts auf die prozentuale Verschlechterung bezogen auf konvergierte Lösungen bei zeitlich beschränkter Optimierungszeit unter Variation der Horizontlänge bei 6 Hindernissen.

tiken [Krä+18b; Min+20; Bui+18] oder Lösungen einer vorgelagerten Pfadplanung [Lev+02; Kun+16]. Im Extremfall entspricht die Initialisierung vollständig einem zuvor geplanten Pfad.

## Warmstart

Nach der ersten Optimierung steht die Lösung  $\mathbf{z}_{n-1}^*$  des vorherigen Planungszyklus zur Verfügung. Eine Initialisierung mit dieser Lösung wird auch Warmstart ( $\mathbf{z}^{\text{WS}}$ ) genannt. Wenn sich Hindernisse und Zielzustände zwischen zwei Planungszyklen nur moderat verändern, stellt die vorherige Lösung bereits eine gute Näherung für den aktuellen Planungszyklus dar. Bei festem Zeitbudget verbleibt somit mehr Zeitreserve zur Optimierung der Trajektorie, die zu besseren Lösungen führen, wie Abbildung 8.4 für 6 Hindernisse ( $S_p$ ) verdeutlicht. Angegeben sind die prozentualen Verschlechterungen der Kosten bei gegebenem Zeitbudget von  $T_O = 0,05\text{ s}$  und  $T_O = 0,1\text{ s}$  bezogen auf die konvergierte Lösung. Weitere Angaben zum Versuch befinden sich in Anhang B.6.2. Wie in Abbildung 8.4 zu sehen, fallen die Auswirkungen eines beschränkten Zeitbudgets durch den (in diesem Fall perfekten) Warmstart geringer aus. Der Vorteil reduziert sich, je umfassender die Veränderungen der Hindernisse und Zielzustände seit der letzten Planung sind. Damit bildet der Warmstart lediglich die Möglichkeit einer verkürzten Optimierung aber keine Garantie. Neben den Zuständen der vorherigen Lösung lassen sich weitere Größen, wie die optimalen Lagrangemultiplikatoren  $\lambda_{n-1}^*$  oder die *Active Set*, in den Lösungsverfahren wiederverwenden.

Im Falle stärkerer Veränderungen von Hindernissen oder Zielzuständen ist der Warmstart allerdings nicht mehr ohne Nachteile. Eine einmal eingeschlagene Richtung wird unter Umständen beibehalten, auch wenn sie einen Umweg bedeutet. Abbildung 8.5 verdeutlicht diesen Fall beispielhaft anhand des zweiachsigen Roboters aus dem vorherigen Abschnitt und eines dynamischen Zielzustands  $\mathbf{x}_f(t)$ . Weitere Details zum Aufbau befinden sich im Anhang B.6.2. Zu Beginn der Planung liegt der Zielzustand oberhalb des Hindernisses und die Planung schlägt den direkten Weg ein. Durch die Bewegung des Zielzustands stellt sich der Weg unterhalb des Hindernisses nach einiger Zeit als besser heraus, wird aufgrund des Warmstarts aber nicht gefunden. Veränderungen wie bewegliche Zielzustände führen zu einer Aufteilung der Menge  $\mathcal{Z}_A(\cdot)$ , wie Abbildung 8.6 qualitativ veranschaulicht. Durch den Warmstart bleiben die Lösungen im oberen Teil von  $\mathcal{Z}_A(\cdot)$ , während sie in den unteren Teil nur durch eine andere Initialisierung gelangen. Obwohl  $\hat{J}(\mathbf{z}_{n+1}^*) < \hat{J}(\mathbf{z}_{n+1}^*)$  und  $\hat{J}(\mathbf{z}_{n+2}^*) < \hat{J}(\mathbf{z}_{n+2}^*)$ , werden die Lösungen  $\mathbf{z}_{n+1}^*$  und  $\mathbf{z}_{n+2}^*$  wegen der Lokalität des Lösungsverfahrens und dem Warmstart nicht gefunden.

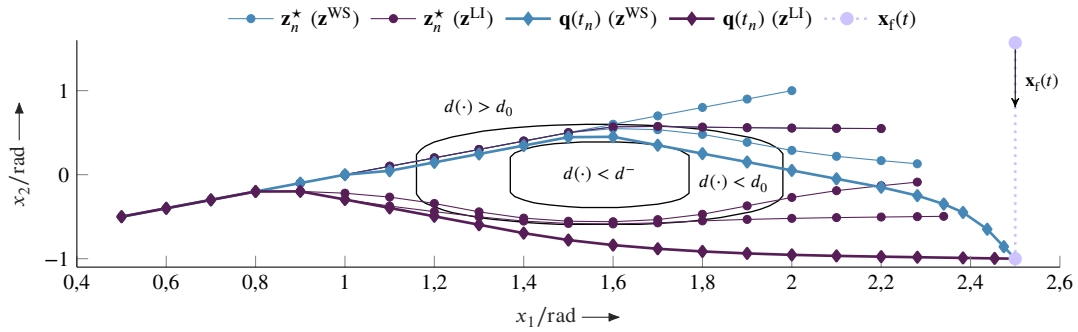


Abbildung 8.5.: Zweidimensionaler Zustandsraum mit Kollisionsmenge sowie Wirkungsbereich der Potentiale. Neben lokalen Lösungen  $\mathbf{z}_n^*$  sind auch die Gelenkwinkel  $\mathbf{q}(t_n)$  der resultierenden Gesamtbewegungen eingezeichnet. Der dynamische Zielzustand  $\mathbf{x}_f(t)$  wird einmal mit und einmal ohne Warmstart verfolgt.

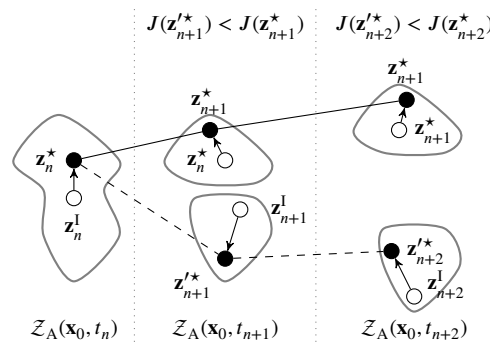


Abbildung 8.6.: Veranschaulichung des Nachteils von Warmstarts im Lösungsraum  $\mathcal{Z}$ . Die Menge  $\mathcal{Z}(\mathbf{x}_0, t_n)$  separiert sich in zwei Teile, von denen durch den Warmstart nur Lösungen aus dem oberen Teil gefunden werden, die schlechter sind als die aus dem unteren. Erst alternative Initialisierungen  $\mathbf{z}_n^I$  ermöglichen Lösungen aus dem unteren Teil.

## 8.2. Erweiterte Initialisierung über parallele Explorationsplanung

Der Warmstart ermöglicht zwar eine effiziente Nutzung der Ergebnisse des vorherigen Planungszyklus, lässt gleichzeitig jedoch das explorative Potential von alternativen Initialisierungen unge-nutzt. In diesem Abschnitt wird die als erweiterte Initialisierung bezeichnete Methode vorgestellt, welche die Vorteile von Warmstarts mit explorativem Verhalten kombiniert.

Die erweiterte Initialisierung beruht auf dem Prinzip einer zur bisherigen Planung parallelgeschalteten Explorationsplanung. Die Grundidee ist in Abbildung 8.7 anhand von fünf Planungsschritten skizziert:

- $t_0$ : Im ersten Schritt verwendet die Hauptinstanz zur Initialisierung z. B.  $\mathbf{z}_0^A$ .
- $t_1$ : Beide Instanzen lösen das gleiche Planungsproblem, wobei die Hauptinstanz mit ihrer vorherigen Lösung  $\mathbf{z}_0^*$  einen Warmstart durchführt und die Parallelinstanz eine beliebige Initialisierung  $\mathbf{z}_1^I$  (außer Warmstart) verwendet.
- $t_2$ : Die Hauptinstanz nutzt erneut den Warmstart mit ihrer vorherigen Lösung  $\mathbf{z}_1^*$ , da die Optimierung der Parallelinstanz noch nicht abgeschlossen ist.
- $t_3$ : Die Parallelinstanz bietet nun eine Lösung  $\mathbf{z}_1^I^*$  mit geringeren Kosten an, die in der Hauptinstanz der vorherigen Lösung  $\mathbf{z}_2^*$  zur Initialisierung vorgezogen wird.

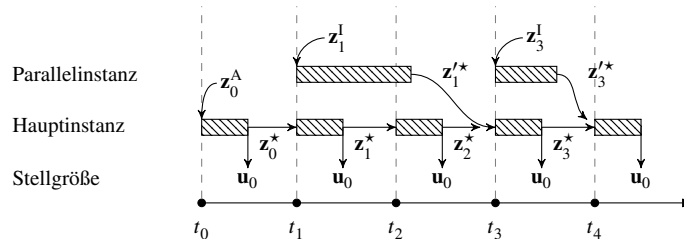


Abbildung 8.7.: Ablauf der erweiterten Initialisierung über eine parallele Explorationsplanung. Schraffierte Flächen deuten Zeiträume von Berechnungen an.

$t_4$ : Die Parallelinstanz wurde erneut mit einer beliebigen Initialisierung  $\mathbf{z}_3^I$  (außer Warmstart) initialisiert und bietet rechtzeitig die Lösung  $\mathbf{z}_3'^*$ , die jedoch höhere Kosten besitzt als die vorherige  $\mathbf{z}_3^*$  der Hauptinstanz, sodass mit letzterer ein Warmstart durchgeführt wird.

Aufgrund der Zweigleisigkeit der Methode werden die Auswirkungen der lokalen Verfahren beim nichtkonvexen NLP und die damit verbundene Kritikalität einzelner Initialisierungen reduziert. Die Lösung der Parallelinstanz wird nicht direkt zur Steuerung des Roboters verwendet, da sie ein größeres Zeitbudget besitzt als die Hauptinstanz und damit eine niedrigere Planungsfrequenz. Durch die Nachoptimierung der Lösung der parallelen Instanz durch die Hauptinstanz wird sichergestellt, dass die resultierenden Bewegungsbefehle bezüglich der aktuellen Kenntnisse über Umgebung und Aufgabe adäquat sind.

## Initialisierung der parallelen Instanz

Das Ziel der Parallelinstanz ist die Exploration einer potentiell besseren Lösung, indem der Einfluss der Initialisierung auf lokale Verfahren ausgenutzt wird. Dadurch, dass die Gültigkeit der parallelen Lösung im Rahmen des Gütevergleichs mit jener aus der Hauptinstanz überprüft wird, eignen sich auch Initialisierungen, die nicht zwingend auf eine gültige Lösung führen, wie z. B.  $\mathbf{z}^{LI}$ . Im Falle einer ungültigen Lösung der Parallelinstanz verwendet die Hauptinstanz ihre vorherige Lösung als Warmstart. Die Initialisierung der Parallelinstanz über  $\mathbf{z}^A$  eignet sich hingegen nicht, da sie am wenigsten zur Exploration anregt.

Um die Effektivität der Exploration zu erhöhen, wird die vorherige Lösung  $\mathbf{z}_{n-1}^*$ , die in der Hauptinstanz zur Bewegung des Roboters genutzt wird, bei der Initialisierung der parallelen Instanz berücksichtigt. Im Folgenden wird eine neue Initialisierung vorgestellt, welche die Idee einer Spiegelung von  $\mathbf{z}_{n-1}^*$  verfolgt, um damit eine alternative Bewegungsrichtung vorzuschlagen. Die Spiegelung erfolgt für jede Komponente  $x_{k,i}$  mit  $i = 1, 2, \dots, N$  des Zustandsvektors  $\mathbf{x}_k \in \mathbf{x}_{0:K-1,n-1}^*$  in der räumlich-zeitlichen Ebene. Die Trajektorien der Komponenten werden darin über Matrizen  $\mathbf{X}_i$  homogener Punkte repräsentiert:

$$\mathbf{X}_i := \begin{bmatrix} \tau_0 & \tau_1 & \dots & \tau_k & \dots & \tau_{K-1} \\ x_{0,i} & x_{1,i} & \dots & x_{k,i} & \dots & x_{K-1,i} \\ 1 & 1 & \dots & 1 & \dots & 1 \end{bmatrix}. \quad (8.4)$$

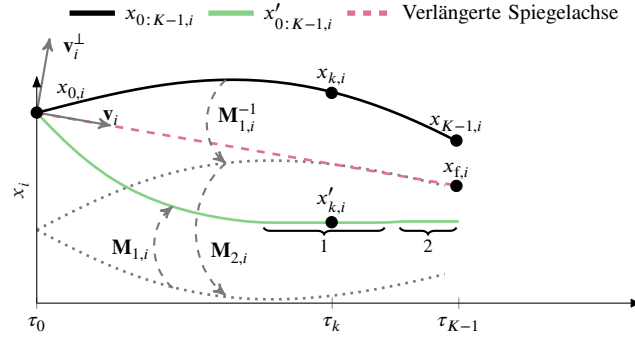


Abbildung 8.8.: Ablauf der Spiegelung der  $i$ -ten Komponente einer Zustandsfolge. Der markierte Abschnitt bei 1 beschreibt die Beschränkung der Spiegelung auf die Gelenkgrenzen und 2 die Fortsetzung über ein Halteglied nullter Ordnung.

Gespiegelt wird entlang des normierten Vektors  $\mathbf{v}_i$ , der von  $(\tau_0 \ x_{0,i})^T$  in Richtung  $(\tau_{K-1} \ x_{f,i})^T$  zeigt, worin letzterer die  $i$ -te Komponente des Zielzustands  $\mathbf{x}_f$  ist:

$$\mathbf{v}_i := \left( \begin{pmatrix} \tau_{K-1} \\ x_{f,i} \end{pmatrix} - \begin{pmatrix} \tau_0 \\ x_{0,i} \end{pmatrix} \right) \left\| \begin{pmatrix} \tau_{K-1} \\ x_{f,i} \end{pmatrix} - \begin{pmatrix} \tau_0 \\ x_{0,i} \end{pmatrix} \right\|_2^{-1}. \quad (8.5)$$

Die homogene Matrix  $\mathbf{M}_{1,i}$  beschreibt eine Transformation in ein lokales Koordinatensystem im Punkt  $(\tau_0 \ x_{0,i})^T$  mit Ausrichtung der x-Achse entlang der Spiegelachse  $\mathbf{v}_i$ :

$$\mathbf{M}_{1,i} := \begin{bmatrix} \mathbf{v}_i & \mathbf{v}_i^\perp & \begin{pmatrix} \tau_0 \\ x_{0,i} \end{pmatrix} \\ 0 & 0 & 1 \end{bmatrix}. \quad (8.6)$$

Hierbei ist  $\mathbf{v}_i^\perp$  ein Vektor orthogonal zu  $\mathbf{v}_i$ . Die Matrix  $\mathbf{M}_{2,i}$  spiegelt alle Punkte entlang der Zeitachse, indem sie das Vorzeichen der zweiten Komponente tauscht:

$$\mathbf{M}_{2,i} := \text{diag} (1 \quad -1 \quad 1). \quad (8.7)$$

Die vollständige Spiegelung ergibt sich schließlich zu:

$$\mathbf{X}'_i = \mathbf{M}_{1,i} \mathbf{M}_{2,i} \mathbf{M}_{1,i}^{-1} \mathbf{X}_i. \quad (8.8)$$

Abbildung 8.8 veranschaulicht den dreistufigen Vorgang. Zunächst wird über  $\mathbf{M}_{1,i}^{-1}$  die  $i$ -te Komponente der aktuellen Lösung in ein lokales Koordinatensystem parallel zur Zeitachse transformiert, dort über  $\mathbf{M}_{2,i}$  gespiegelt und anschließend über  $\mathbf{M}_{1,i}$  wieder in das Ursprungskoordinatensystem zurücktransformiert. Da sich nach diesem Vorgang auch die Zeitstempel  $\tau_k$  verändern, wird die Lösung mit dem Gitter (3.10) über eine lineare Interpolation neu abgetastet. Hierbei wird notfalls über ein Halteglied nullter Ordnung auf  $K$  Zustände aufgefüllt und etwaige Verletzungen der Zustandsbeschränkungen über Limitation korrigiert. Nach der Spiegelung aller  $N$  Komponenten wird schließlich die Zustandsfolge  $\mathbf{x}_{0:K-1}$  für die Initialisierung  $\mathbf{z}^M$  der parallelen Instanz wieder zusammengesetzt.

## Auswertung für ein dynamisches Hindernis

Im Folgenden wird die Effektivität der parallelen Explorationsinstanz und deren Initialisierungen mit unterschiedlichen Varianten sukzessiv evaluiert. Hierfür wird der zum Beispiel aus

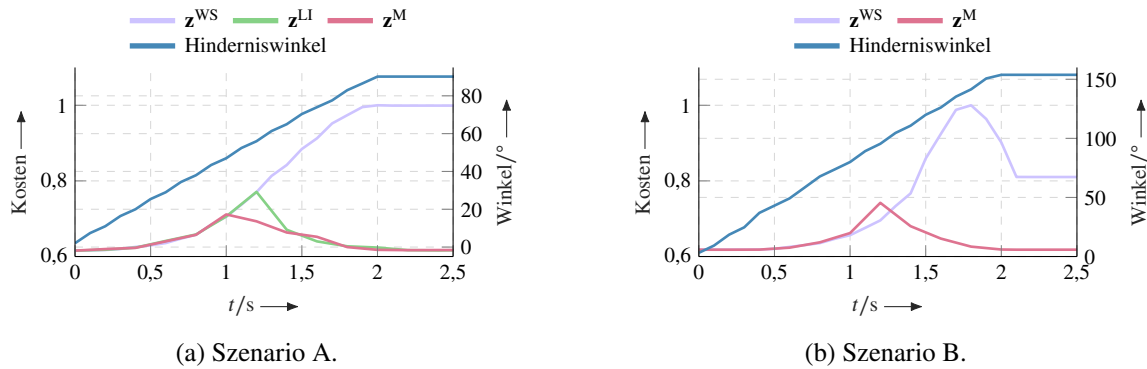


Abbildung 8.9.: Verlauf der auf den Maximalwert normierten Kosten der Lösungen nach unterschiedlichen Initialisierungen im ersten Versuch.

Abschnitt 8.1 analoge Fall eines dynamischen Hindernisses bei konstantem Zielzustand mit zwei Szenarien gewählt. In Szenario A dreht sich ein Hüllkörper ( $S_L$ ) senkrecht in die horizontale Bewegung des UR10. Szenario B tauscht die Bewegungen, sodass der Hüllkörper in einer horizontalen Drehung die senkrechte Bewegung des Roboters stört. Die Kollisionsvermeidung erfolgt über die Schnappschussmethode, um den Einfluss der Hindernisprädiktion auf die Ergebnisse zu eliminieren. Weitere Informationen zum Aufbau befinden sich im Anhang B.6.3.

Zunächst wird der Einfluss der Initialisierungen  $z^M$ ,  $z^{LI}$  und  $z^{WS}$  auf die Qualität von lokalen Lösungen untersucht, um den Nutzen der Spiegelung bei  $z^M$  zu bewerten. Alle Varianten lösen dasselbe lokale Problem und verwenden die im Anhang B.6.3 angegebenen Parameter. Die parallele Instanz kommt dabei noch nicht zum Einsatz. Zur besseren Vergleichbarkeit ignoriert der Roboter eingehende Bewegungsbefehle und verbleibt in seinem Startzustand. Abbildung 8.9 zeigt die auf den Maximalwert normierten Kosten der lokalen Lösungen und den Drehwinkel des Hüllkörpers in beiden Szenarien. In der ersten Sekunde von Szenario A (Abbildung 8.9a) werden durch einen Anstieg der Kosten bei allen drei Varianten Ausweichbewegungen unterhalb des Hindernisses deutlich. Diese Ausweichbewegung führt bei  $z^M$  zu einer gespiegelten Initialisierung, welche oberhalb des Hindernisses verläuft. Im Intervall  $t \in [1, 1,5]$  s führt diese Initialisierung zu einer Lösung mit geringeren Kosten als die der beiden anderen Varianten. Die Initialisierung über  $z^{LI}$  erkennt eine bessere Lösung oberhalb des Hindernisses ca. 0,2 s später als  $z^M$ . Der Warmstart verbleibt bei der anfänglichen Tendenz und bestätigt das in Abschnitt 8.1 dargestellte Problem fehlender Exploration. Abbildung 8.9b zeigt ähnliche Resultate für Szenario B, mit dem Unterschied, dass die Spiegelung keinen Zeitvorteil ergibt und sie zeitgleich bei ca. 1,2 s mit  $z^{LI}$  die Lösung hinter dem Hindernis präferiert. Bis dahin steigen die Kosten beider Alternativen ab 1 s gegenüber dem Warmstart sogar an, da sie ungünstige Initialisierungen verwenden, die bei gleichem Zeitbudget zu einer schlechteren Lösung führen, als die kontinuierliche Nachoptimierung beim Warmstart. Aufgrund der Ähnlichkeit zwischen  $z^M$  und  $z^{LI}$  ist nur Ersteres abgebildet.

Im zweiten Versuch werden die Auswirkungen auf die Gesamtbewegung untersucht, wenn die Hauptinstanz durch die parallelen Lösungen gemäß Abbildung 8.7 initialisiert wird. Hierfür zeigt Abbildung 8.10 die Gelenkwinkelverläufe des ersten (Szenario B) und zweiten Gelenks (Szenario A). Für Szenario A (Abbildung 8.10a) ist erkennbar, wie die Variante mit permanentem Warmstart das zweite Gelenk bis auf ca. 0,4 rad für die Ausweichbewegung auslenkt. Die Varianten mit erweiterter Initialisierung kommen mit einer kleineren Bewegung aus. Hierbei profitiert die Variante mit  $z^M$  vom früheren Finden einer lokal besseren Lösung und führt auf die

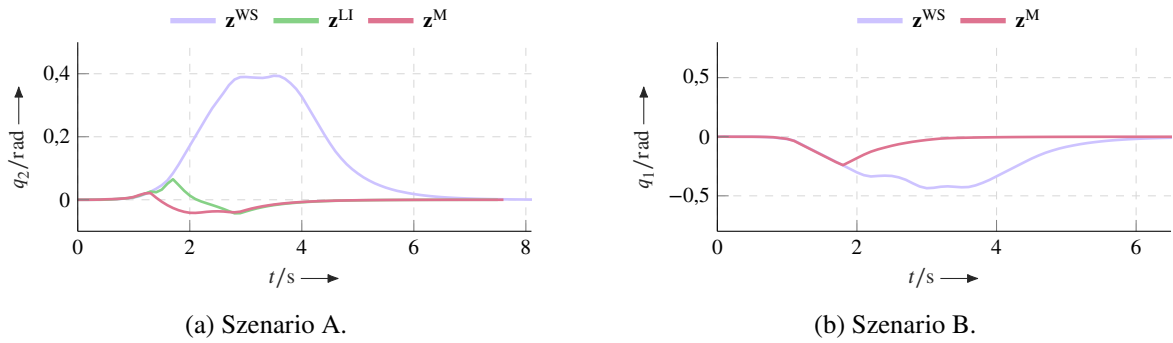


Abbildung 8.10.: Verlauf der resultierenden Gelenkwinkel des Roboters bei Nutzung der erweiterten Initialisierung über die parallele Explorationsplanung.

insgesamt geringsten Auslenkung im zweiten Gelenk. Die Variante mit  $\mathbf{z}^{\text{LI}}$  ermittelt ihre Alternativlösung ca. 0,5 s später, was zu einem längeren Weg und einer weniger „flüssigen“ Bewegung führt. Begleitend dazu sind in Tabelle 8.1 die räumlichen und zeitlichen Längen der Gesamtbewegungen angegeben. Die zeitliche Länge ist jene vom Beginn der ersten Planung bis zum Erreichen einer Hyperkugel mit einem Radius von  $1^\circ$  um den Zielzustand. Die Referenzlänge ist der kürzeste Abstand zwischen Start- sowie Zielzustand im Zustandsraum und die Referenzdauer ist die des hindernisfreien Falls. Die Ausweichbewegung der Variante ohne erweiterte Initialisierung führt auf einen 44 % längeren Pfad und eine 20 % längere Dauer im Vergleich zur Referenz. Die Auswirkungen bei der erweiterten Initialisierung fallen bei  $\mathbf{z}^{\text{LI}}$  mit 3 % bzw. 2 % geringer aus. Wird  $\mathbf{z}^{\text{M}}$  verwendet, liegen die Unterschiede bereits außerhalb der Rundungsgenauigkeit. Abbildung 8.10b ergänzt die Ergebnisse für Szenario B mit den Verläufen des ersten Gelenks. Die erweiterte Initialisierung erzielt keinen bedeutsamen Unterschied zwischen  $\mathbf{z}^{\text{LI}}$  und  $\mathbf{z}^{\text{M}}$ , wie auch Tabelle 8.1 zeigt, weshalb nur letzterer Fall angegeben ist. Die Ausweichbewegung ohne erweiterte Initialisierung ist dezenter als in Szenario A, aber dennoch mit 25 % räumlicher und 9 % zeitlicher Länge umfangreicher als die der erweiterten Initialisierung. Obwohl die erweiterte Initialisierung die Pfadlänge um 11 % gegenüber der Referenz erhöht, verlängert sich nicht die Dauer der Gesamtbewegung, welche von der gesättigten Geschwindigkeit des zweiten und dritten Gelenks bestimmt wird.

Der vorherige Versuch hat gezeigt, dass die parallele Explorationsinstanz zu einer räumlich und zeitlich kürzeren Gesamtbewegung gegenüber dem Warmstart führt. Der dritte Versuch untersucht daher, inwiefern die lokalen Lösungen durch die erweiterte Initialisierung an die von globalen Verfahren heranreichen, ohne dabei auf die niedrigen Laufzeiten der MHP zu verzichten. Die globalen Verfahren sind STOMP und GTO. Sie decken durch ihre unterschiedlichen Ansätze ein breites Spektrum an globalen Lösungen ab. STOMP ähnelt der Funktionsweise einer begrenzt globalen evolutionären Optimierung und sucht stichprobenhaft nach dem Zufallsprinzip nach

Tabelle 8.1.: Gegenüberstellung der räumlichen und zeitlichen Länge der resultierenden Bewegungen bei erweiterter Initialisierung in beiden Szenarien.

Szenario	Räumliche Länge				Zeitliche Länge			
	Referenz	Warmstart	$\mathbf{z}^{\text{LI}}$	$\mathbf{z}^{\text{M}}$	Referenz	Warmstart	$\mathbf{z}^{\text{LI}}$	$\mathbf{z}^{\text{M}}$
A	1,00 (1,2 rad)	1,44	1,03	1,00	1,00 (5,1 s)	1,20	1,02	1,00
B	1,00 (1,7 rad)	1,25	1,11	1,11	1,00 (5,7 s)	1,09	1,00	1,00

Tabelle 8.2.: Ähnlichkeiten zwischen lokalen und globalen Lösungen zu verschiedenen Zeitschritten.

Szenario		$t = 0$		$t = 0.5$		$t = 1$		$t = 1.5$		$t = 2$		$t = 2.5$		$t = 3$	
		GTO	STOMP	GTO	STOMP	GTO	STOMP	GTO	STOMP	GTO	STOMP	GTO	STOMP	GTO	STOMP
		A	$\mathbf{z}^{\text{WS}}$	0,18	0,21	0,27	0,98	0,86	1,36	9,69	9,28	8,24	8,02	0,07	1,28
	$\mathbf{z}^{\text{LI}}$	0,18	0,21	0,72	1,07	7,35	1,37	9,68	9,33	0,16	0,31	0,01	0,18	0,01	0,17
	$\mathbf{z}^{\text{M}}$	0,19	0,22	0,39	0,97	7,28	1,37	0,35	0,93	0,17	0,11	0,02	0,19	0,01	0,16
B	$\mathbf{z}^{\text{WS}}$	0,00	0,40	0,47	0,35	0,98	1,01	14,46	12,73	9,85	7,62	6,50	3,60	2,77	2,38
	$\mathbf{z}^{\text{LI}}$	0,00	0,40	0,25	0,57	1,03	0,97	14,50	12,79	1,38	2,21	1,40	0,89	1,88	0,79
	$\mathbf{z}^{\text{M}}$	0,00	0,40	0,45	0,35	1,10	1,01	14,45	12,76	1,36	2,29	1,39	0,92	1,88	0,81

kompletten Trajektorien. GTO ist formal am ähnlichsten zur MHP aber durch die Lokalität nach wie vor abhängig von der Initialisierung, die in diesem Fall über  $\mathbf{z}^{\text{LI}}$  realisiert wird. Beide Verfahren verwenden, sofern unterstützt, die gleiche Kostenfunktion und Nebenbedingungen wie die MHP. Die zugehörigen Parameter und weitere Details zur Implementierung sind in Anhang B.6.3 zusammengefasst.

Zur Bewertung der Ähnlichkeit zwischen lokaler Lösung und der von globalen Verfahren wird der Pfad der Zustände als naheliegendste gemeinsame Repräsentation verwendet. Das Zeitprofil und die Stellgrößen werden nicht berücksichtigt, da sie in STOMP und der GTO/MHP aus verschiedenen Kriterien entstehen. Das Ähnlichkeitsmaß  $\ell \in \mathbb{R}^+$  ist angelehnt an die Frechet- und Hausdorff-Distanz:

$$\ell := \sum_{k=0}^{K-1} \min_{\substack{t \in [0,1] \\ j \in [2, K_G]}} \left\| \mathbf{x}_k - (\mathbf{x}_{j-1}^G + t(\mathbf{x}_j^G - \mathbf{x}_{j-1}^G)) \right\|_2, \quad (8.9)$$

und beschreibt die Summe aller kürzesten Abstände zwischen den Zuständen  $\mathbf{x}_k$  der lokalen Lösung und der linearen Interpolation der  $K_G \in \mathbb{N}$  Zustände  $\mathbf{x}_j^G$  eines globalen Verfahrens. Niedrigere Werte bedeuten eine stärkere Ähnlichkeit. Tabelle 8.2 fasst die Ähnlichkeiten zwischen lokalen und globalen Lösungen der Zeitschritte  $t = 0 \text{ s}, 0,5 \text{ s}, \dots, 3 \text{ s}$  zusammen. Die lokalen Lösungen stammen von der Variante mit reinem Warmstart  $\mathbf{z}^{\text{WS}}$  sowie von der über  $\mathbf{z}^{\text{LI}}$  und  $\mathbf{z}^{\text{M}}$  initialisierten parallelen Explorationsinstanz der erweiterten Initialisierung. Alle drei Varianten führen außerdem zu Gesamtrajektorien am Roboter, von denen die Zustände zu den jeweiligen Zeitschritten als Startzustände der globalen Verfahren dienen.

In den ersten zwei Zeitschritten von Szenario A nähern sich die Varianten dem Hindernis an und liegen dabei jeweils nah an den globalen Lösungen. Bei  $t = 1 \text{ s}$  gilt dies nur noch für den Warmstart, der bereits eine Bewegungstendenz unterhalb des Hindernisses eingeschlagen hat. Bei der erweiterten Initialisierung hingegen führt die Anwesenheit des Hindernisses zu gegensätzlichen Bewegungen der globalen Verfahren. GTO präferiert den Weg oberhalb und STOMP unterhalb des Hindernisses, wie Abbildung 8.11a für  $\mathbf{z}^{\text{M}}$  anhand des zweiten Gelenks veranschaulicht. Die Ähnlichkeit der lokalen Lösung bei erweiterter Initialisierung mit der von STOMP bedeutet, dass beide explorative Varianten zunächst weiter den Weg unterhalb des Hindernisses nehmen und nicht der besseren Lösung von GTO folgen. Dies ändert sich für  $\mathbf{z}^{\text{M}}$  im Zeitschritt  $t = 1,5 \text{ s}$ , in welchem die lokale Lösung nah an den beiden globalen Lösungen liegt, welche inzwischen einheitlich den Weg oberhalb des Hindernisses empfehlen. Auch dieser Zeitschritt ist in Abbildung 8.11a dargestellt. Die erweiterte Initialisierung mit  $\mathbf{z}^{\text{LI}}$  sowie der Warmstart ermitteln

mangels explorativer Eigenschaften lokale Lösungen, die entgegen der Empfehlung der globalen Verfahren weiterhin unterhalb des Hindernisses führen. Während der Warmstart dieses Verhalten auch im Zeitschritt  $t = 2$  s beibehält, erkennt die erweiterte Initialisierung mit  $\mathbf{z}^{\text{LI}}$  die bessere lokale Lösung oberhalb des Hindernisses und folgt mit entsprechender Ähnlichkeit den globalen Lösungen. Die erweiterte Initialisierung hat das Hindernis damit bereits überwunden und fährt in den übrigen Zeitschritten den Zielzustand an, während dieses Verhalten beim Warmstart erst ab  $t = 2,5$  s einsetzt.

In Szenario B nähern sich die Varianten dem Hindernis innerhalb der ersten drei Zeitschritte an und folgen dabei den globalen Lösungen. Abbildung 8.11b zeigt mit der erweiterten Initialisierung und  $\mathbf{z}^{\text{M}}$  stellvertretend, wie bei  $t = 1,5$  s die globalen Lösungen bereits vor dem Hindernis herführen, während die lokalen Lösungen (aller Varianten) noch dahinter planen. Einen Zeitschritt weiter folgt die erweiterte Initialisierung (in beiden Varianten) den globalen Lösungen vor dem Hindernis, während der Warmstart (nicht eingezeichnet) bei seiner bisherigen Richtung bleibt (siehe Tabelle 8.2). Diese Richtung behält er bis zum letzten Zeitschritt bei, während die erweiterte Initialisierung in beiden Ausführungen das Hindernis bereits überwunden hat und in den übrigen Zeitschritten im Einklang mit den globalen Lösungen den Zielzustand anfährt.

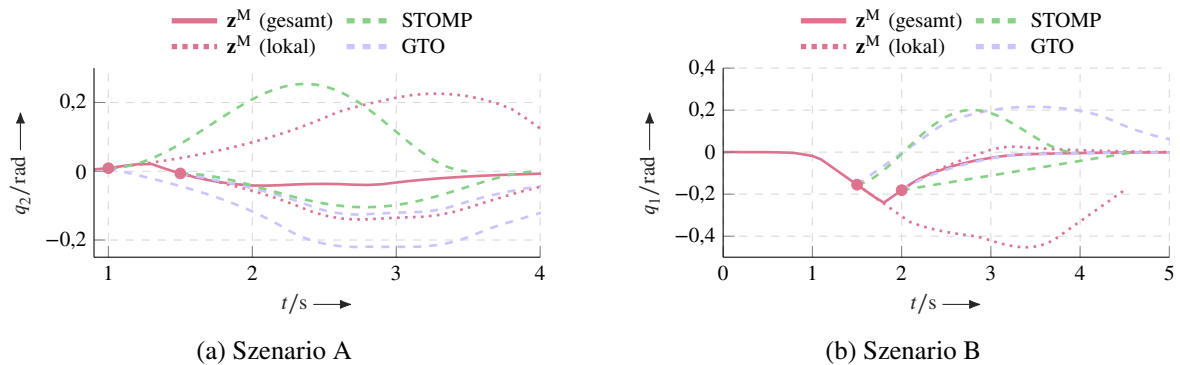


Abbildung 8.11.: Verlauf resultierender Gelenkwinkel des Roboters mit lokalen Lösungen der erweiterten Initialisierung ( $\mathbf{z}^{\text{M}}$ ) und den globalen Lösungen an jeweils zwei Zeitpunkten.



# 9

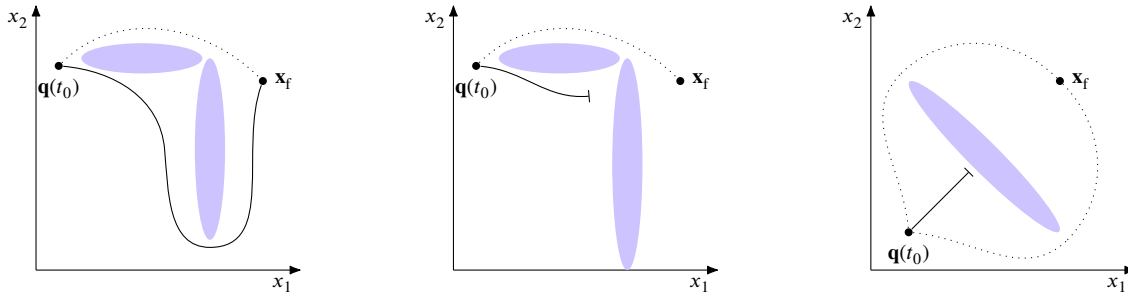
## Lokalität des Planungsproblems

Nachdem im vorherigen Kapitel die Lokalität der Lösungsverfahren behandelt wurde, folgt in diesem Kapitel die Betrachtung der Lokalität des Planungsproblems der MHP. In Kapitel 2 wird das Ziel der optimierungsbasierten Trajektorienplanung als Lösung des globalen OSP (2.5) definiert. Der Ansatz der MHP verfolgt zur Reduzierung des Berechnungsaufwands mit Hilfe der Beschränkung des Horizonts jedoch die Lösung des lokalen Problems (3.29). Die resultierende Gesamtrajektorie ist das Ergebnis der Ausführung der jeweils ersten Stellgröße der lokalen Lösungen. Während bei der Lösung des globalen Problems die Informationen im Zustandsraum entlang der gesamten Trajektorie zugleich genutzt werden, ist die MHP auf die lokalen Informationen der Teillösungen jedes Planungsschritts beschränkt. Je kürzer der Horizont ist, desto beschränkter sind diese Informationen und desto stärker die Abkehr vom globalen Planungsproblem. Dies bedeutet, dass eine lokal optimale Lösung nicht zwingend optimal im Sinne des globalen Planungsproblems ist. Eine lokale Lösung ist optimal im Sinne des globalen Problems, wenn sie eine Teilstrategie der global optimalen Lösung darstellt [Bel03].

Im freien Zustandsraum mit ausschließlich konvexen Zustandsbeschränkungen sind die Informationen gleich und die Gesamtrajektorie der MHP gleicht jener der globalen Lösung. Dies ändert sich bei Hinzunahme der Kollisionsvermeidung und führt auf zwei Situationen, die anhand von Abbildung 9.1 veranschaulicht werden. Beschränkte Horizonte reichen unter Umständen nicht weit genug, um den weiteren Verlauf der Szene zu erfassen und in der Optimierung zu bewerten, sodass ein Umweg wie in Abbildung 9.1a eingeschlagen wird. Häufig entscheidet zusätzlich die Initialisierung darüber, welchen Weg die Lösung an so einem Entscheidungspunkt annimmt. Obwohl Umwege zu einer suboptimalen Gesamtbewegung führen, wird der Zielzustand erreicht. Anders verhält es sich in der Situation aus Abbildung 9.1b, in welcher die Optimierung in ein lokales Minimum fällt, aus dem heraus alle Bewegungen von geringerer Güte sind, als stehen-zubleiben. Abbildung 9.1c verallgemeinert die Situation auf nur ein Hindernis. Lösungen dieser Art werden auch strukturelle lokale Minima [Kim+92; Bro+02], *Reach Around Local Minima* [Byr+12] oder im Rahmen der APF-Methode auch Kräftegleichgewicht genannt.

Welchen Einfluss Horizontlänge, Initialisierung und Kostengewichte auf strukturelle lokale Minima (fortan lokale Minima genannt) haben, wird im nächsten Abschnitt genauer untersucht. Dabei wird sich auf (quasi-)statische Hindernisse beschränkt, unter der vereinfachenden Annahme, dass dynamische Hindernisse der Planung aufgrund ihrer Bewegung früher oder später einen Korridor ermöglichen, solange sie den Zielzustand nicht aktiv abschirmen. Abschnitt 9.2 stellt eine Methode vor, welche den genannten Herausforderungen der lokalen Planung mit einem temporären Zielzustand begegnet.

Die Ergebnisse dieses Kapitels sind in Teilen in der Publikation [Krä+22a] veröffentlicht.



(a) Suboptimale Lösung in Form eines längeren Wegs. (b) Suboptimale Lösung in Form eines strukturellen lokalen Minimums. (c) Einfachster Fall eines strukturellen lokalen Minimums.

Abbildung 9.1.: Fälle von suboptimalen Gesamtbewegungen (durchgezogen) verglichen mit der global optimalen Lösung (gepunktet) im Zustandsraum mit verschiedenen Hinderniskonfigurationen. Die Hindernisse können beliebige Formen annehmen.

### 9.1. Strukturelle lokale Minima bei beschränkten Horizonten

Neben der minimalen Horizontlänge  $K^-$  aus (3.11) als absolute untere Schranke führen lokale Minima des Planungsproblems zu einer üblicherweise höher anzusetzenden theoretischen Schranke  $K_{LM}^-$ , wie im Folgenden gezeigt wird. Zur Veranschaulichung der Entstehung von lokalen Minima, die zum Stillstand führen, wird in Abbildung 9.2 eine Trajektorie des zweidimensionalen Roboters aus dem vorherigen Kapitel (siehe Anhang A.3) geplant. Die Trajektorie wird zwischen Start- und Zielzustand durch ein statisches Hindernis gestört. Für weitere Informationen zum Aufbau dieses Beispiels siehe Anhang B.7.1. Die globale Lösung wird über GTO mit der Initialisierung  $\mathbf{z}^{LI+}$  und ausreichender Horizontlänge ermittelt. Über die Vierecke sind die Endpunkte lokaler Lösungen mit unterschiedlicher Horizontlänge angegeben. Passend dazu wird in Abbildung 9.3 der Verlauf der Kosten dieser lokalen Lösungen in Abhängigkeit ihrer Skalierungen entlang der globalen Lösung betrachtet. Dafür wird die Zustandsfolge  $\mathbf{x}_{0:K-1}$  für jede Horizontlänge räumlich entlang der globalen Lösung zwischen Start- und Zielzustand mit dem Parameter  $\rho \in [0, 1]$  skaliert und die zugehörige Steuerfolge  $\mathbf{u}_{0:K-2}$  über Gleichung (3.1) gebildet. Hierbei entspricht  $\rho = 0$  dem Anfangs- und  $\rho = 1$  dem Zielzustand. Diese Darstellung ähnelt äußerlich der Entwicklung eines Lösungskandidaten  $\mathbf{z}$  während der Optimierung und erlaubt eine eindimensionale Auswertung der sonst  $D$ -dimensionalen Argumente der Kostenfunktion. Der gestrichelte Teil markiert die Skalierung  $\rho > \rho^+$ , für welche die lokale Lösung aufgrund von

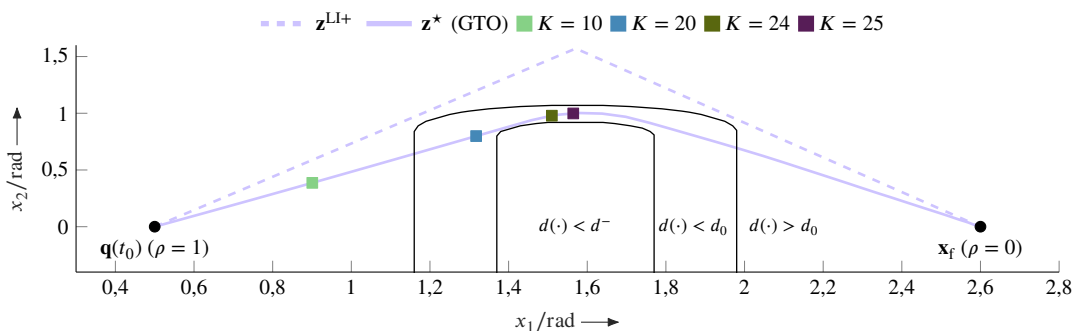


Abbildung 9.2.: Initialisierung und globale Lösung mit markierten Endpunkten skaliert lokaler Lösungen auf Basis unterschiedlicher Horizontlängen.

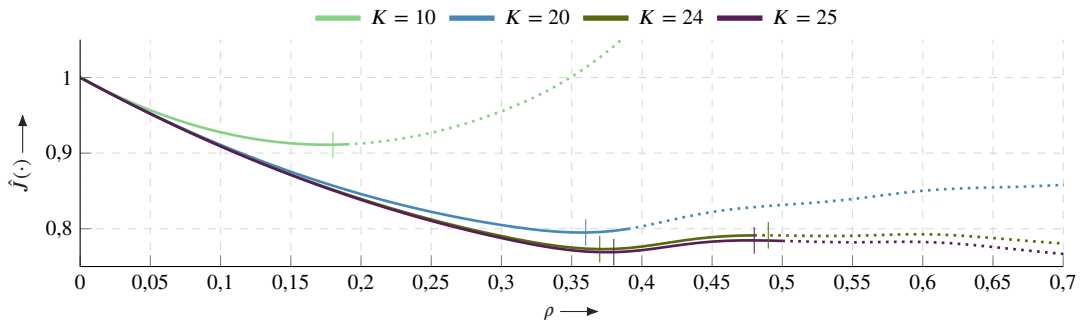


Abbildung 9.3.: Kosten lokaler Lösungen in Abhängigkeit ihrer Skalierung. Striche markieren lokale Extrema. Skalierungen mit ungültiger Lösung sind gepunktet dargestellt.

Stellgrößenbeschränkungen ungültig ist. Der vollständige Verlauf dient nur der Orientierung, da ein längerer Horizont nicht nur die Reichweite entlang der Kostenfunktion, sondern auch ihre Beschaffenheit verändert.

In Abbildung 9.3 sind für  $K = 10$  und  $K = 20$  lokale Minima im Intervall  $\rho \in [0,18, 0,36]$  zu erkennen, die nach einer anfänglichen Annäherung an das Hindernis einen Stillstand repräsentieren. Bei größeren Horizontlängen reichen zum einen die Skalierungen weiter und zum anderen prägt sich im Bereich  $\rho \approx 0,48 = \rho_{\max}$  für  $K = 24$  und  $K = 25$  zunehmend ein lokales Maximum aus. Rechtsseitig davon befinden sich Lösungen zum Umfahren des Hindernisses bis hin zu einem absoluten Minimum. Die kritische Horizontlänge  $K_{\text{LM}}^- \in \mathbb{N}_0$  ist jetzt die maximale Länge, für welche sich entweder kein lokales Maximum  $\rho_{\max}$  ausprägt oder die Zustandsfolge nicht über dieses lokale Maximum hinausreicht ( $\rho^+ < \rho_{\max}$ ). Im Beispiel ist  $K_{\text{LM}}^- = 24$ , da für  $K = 25$  ein lokales Maximum vorliegt und die Skalierung über dieses hinweg reicht. Für  $K \leq K_{\text{LM}}^-$  ist das lokale Minimum vor dem Hindernis das einzige und ein Stillstand ist unvermeidbar. Für  $K > K_{\text{LM}}^-$  und eine Initialisierung mit einer Zustandsfolge die  $\rho^+ > \rho_{\max}$  entspricht (z. B.  $\mathbf{z}^{\text{LI}+}$ ), wird der Lösungskandidat in Richtung globales Minimum gestreckt, sodass er ausreichend weit um das Hindernis herum reicht und ein Stillstand vermieden wird.

Nicht immer beginnt die Bewegung unmittelbar vor einem Hindernis. Befindet sich das Hindernis anfangs in einiger Entfernung, sodass es in den ersten Planungszyklen außerhalb des Horizonts liegt, bewegt sich die Planung zunächst direkt darauf zu. Sobald der  $K - 1$ -te Zustand des Horizonts den Aktivierungsbereich der Potentiale in das statische Hindernis verlegt, reagiert die Planung. Die effektive Horizontlänge zum Ausweichen entspricht daher der Anzahl an Zuständen, die sich in der Nähe des Hindernisses befinden. Mit einer geeigneten Initialisierung wie z. B.  $\mathbf{z}^{\text{LI}+}$  wird mit dem Umfahren begonnen, sobald sich ausreichend viele Zustände in der Nähe des Hindernisses befinden. Beim Warmstart verbleibt der Roboter tendenziell vor dem Hindernis, da die vorherigen Lösungen zunehmend einen Stillstand repräsentieren und einer Initialisierung mit  $\mathbf{z}^{\text{A}}$  gleichen.

## Bedeutung der Kostengewichte

Die folgende Auswertung untersucht die Einflüsse der Kostengewichte  $Q$ ,  $R$ , und  $W_C$  auf  $K_{\text{LM}}^-$ . Auch die individuelle Bewertung des letzten Zustands auf dem Horizont durch  $Q_K - 1$ , die häufig bei MPCs aus Stabilitätsgründen Verwendung findet, wird untersucht. Eine Verschiebung ( $\rho_{\max} \rightarrow 0$ ) des lokalen Maximums führt zu einem kleineren  $K_{\text{LM}}^-$  und damit entweder zu einer robusteren Planung gegenüber Stillständen oder der Möglichkeit einen kürzeren Horizont zu wählen. Da der Absolutwert der Kosten nicht von Bedeutung ist, steuern die Gewichte die Wich-

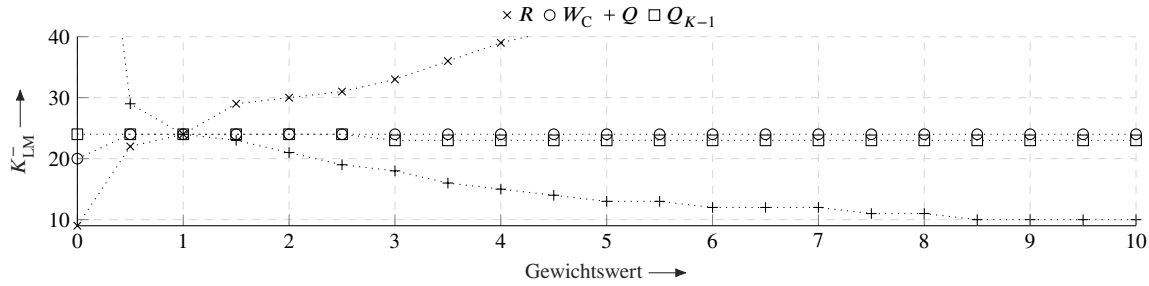


Abbildung 9.4.: Veränderung von  $K_{LM}^{-}$  bei individueller Veränderung der Kostengewichte.

tigkeit der Summanden von  $\hat{J}(\cdot)$  relativ zueinander. Die Erhöhung eines Gewichts ist äquivalent zur Verringerung aller anderen Gewichte, solange das Gewichtsverhältnis konstant bleibt. Die folgenden Ergebnisse sind daher hinsichtlich der Variation der Gewichte symmetrisch.

Abbildung 9.4 fasst die Einflüsse der Gewichte auf  $K_{LM}^{-}$  des vorherigen Beispiels zusammen. Zur besseren Vergleichbarkeit werden die Gewichte einzeln in 0,5-Schritten zwischen  $[0, 10]$  skaliert. Entsprechend führen bei 1 alle Gewichte zu  $K_{LM}^{-} = 24$ . Für kleinere Gewichte  $R$  sinkt  $K_{LM}^{-}$ , da die Bewegung weniger Kosten verursacht und die Gesamtkosten beim Ausweichen schwächer ansteigen. Als Konsequenz wird das lokale Maximum  $\rho_{max}$  auch von kürzeren Horizontlängen erreicht. Durch hohe Werte für  $Q$  setzen sich die Zielkosten  $c_x(\cdot)$  immer mehr von den Übrigen ab und es dominieren ihre kreisförmigen Isolinien. Ist der Horizont so lang, dass sich ein ausreichend langer vorderer Abschnitt hinter dem Hindernis einwärts dieser Isolinien bewegt, sinken die Kosten durch die Zielannäherung stärker als sie durch die Ausweichbewegung steigen. Die Kosten werden geringer, je länger die Trajektorie gestreckt wird und je mehr Zustände hinter das Hindernis und damit näher an das Ziel gelangen. Dieselbe Tendenz lässt sich auch bei  $Q_{K-1}$  in abgeschwächter Form beobachten. Die Erhöhung von  $W_C$  bewirkt einen größeren Umweg um das Hindernis, womit sich ein Kompromiss zwischen Potentialen und Zielkosten ergibt. Der Einfluss der Potentiale ist jedoch auf ihre Aktivierungsbereiche beschränkt, womit der Einfluss auf die Horizontlänge für höhere  $W_C$  im Wesentlichen von  $d_C$  bestimmt wird.

### Systematische Festlegung der Kostengewichte

Aus den Kapiteln 3 und 4 sind bereits Vorgaben  $R < Q < W_C$  für die relativen Größen der Gewichte  $R$ ,  $Q$  (und implizit  $Q_{K-1}$ ) und  $W_C$  bekannt. Davon ausgehend lässt sich in Kombination mit der Analyse von  $K_{LM}^{-}$  eine Methode zum Einstellen der Gewichte der MHP ableiten.

Das Gewicht  $R$  stellt die untere Schranke und wird gemäß dem Einfluss auf  $K_{LM}^{-}$  aus Abbildung 9.4 auf  $R = 0,5$  gesetzt. Als Nächstes wird  $W_C$  als obere Schranke festgelegt. Hierfür wird der Einfluss verschiedener Werte unter  $R = 0,5$  auf  $K_{LM}^{-}$  neu erfasst und in Abbildung 9.5a dargestellt. Da  $K_{LM}^{-}$  nach einem anfänglichen Anstieg konstant bleibt, wird  $W_C = 10$  gesetzt, sodass genügend Spielraum für die dazwischenliegenden Gewichte  $Q$  und  $Q_K$  bleibt. Im dritten Schritt werden diese beiden Gewichte nacheinander innerhalb der Schranken variiert und der Einfluss auf  $K_{LM}^{-}$ , wie ebenfalls in Abbildung 9.5a dargestellt, ermittelt. Da  $K_{LM}^{-}$  für  $Q \geq 1$  nicht weiter sinkt, wird  $Q = 1$  gesetzt. Wie Abbildung 9.5a zeigt, ist für  $Q_{K-1} \geq 1$  beim Gewicht keine Verbesserung erkennbar und es gilt  $Q = Q_{K-1}$  und folglich  $K_{LM}^{-} = 22$ . Abbildung 9.5b zeigt abschließend den Verlauf der Kosten entlang der globalen Lösung unter den ermittelten Gewichten für verschiedene Horizontlängen. Durch die Abstimmung der Gewichte ist  $K_{LM}^{-}$  zwar gesunken, eine Initialisierung über das Maximum  $\rho_{max}$  hinweg ist jedoch nach wie vor erforderlich.

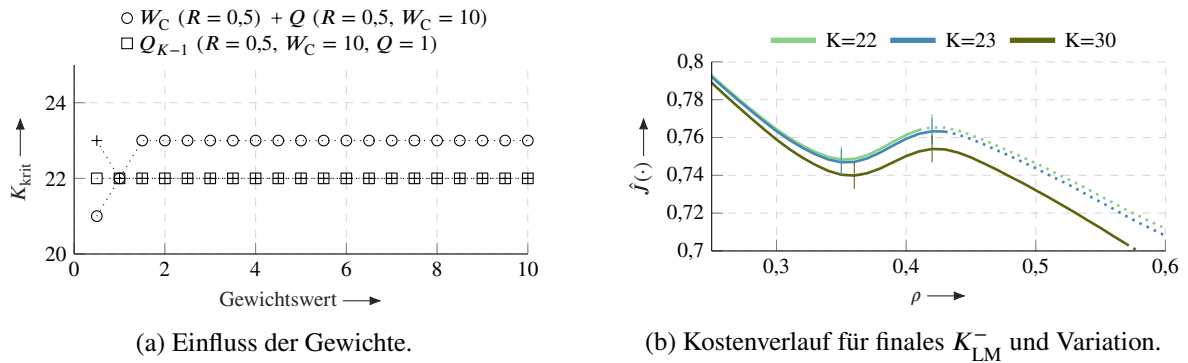


Abbildung 9.5.: Auslegung der Gewichte für möglichst kleines  $K_{LM}^{-}$ .

Die Auswertung von  $K_{LM}^{-}$  erlaubt nur qualitative Schlussfolgerungen und gibt keine feste Zahl für die Horizontlänge vor. Letztlich variieren die Zusammenhänge in ihrer Stärke, je nach Start- und Zielzustand sowie Anzahl, Form und Verteilung der Hindernisse. Hinzu kommt, dass – auch wenn  $K > K_{LM}^{-}$  gilt – eine notwendige Initialisierung über z. B.  $\mathbf{z}^{LI+}$  zum Umfahren des Hindernisses nicht völlig auszuschließen ist. Das Verfahren löst damit das eingangs eingeführte Problem nicht ganzheitlich, sondern bietet eine grobe Orientierung bei der Gewichtswahl und eine Basis für weiterführende Arbeiten.

## 9.2. Temporärer Zielzustand über globale Punktführung

Der vorherige Abschnitt zeigt eine theoretische untere Schranke  $K_{LM}^{-}$  für die Horizontlänge zur Vermeidung von lokalen Minima auf und analysierte deren Zusammenhang mit den Gewichten der Kostenterme. Durch die Abhängigkeit von einem konkreten Szenario sind die Erkenntnisse in den wechselhaften Umgebungen der kollaborativen Robotik nur begrenzt nutzbar. Zusätzlich sind einer Erhöhung von  $K$  auch praktische Grenzen durch die Laufzeit gesetzt. Unberücksichtigt bleibt außerdem die Herausforderung der lokalen Planung hinsichtlich der Umwege aus Abbildung 9.1a.

Die im Folgenden vorgeschlagene Methode basiert auf der Idee, den Verlauf der Kosten derart zu verändern, dass kein lokales Minimum entsteht. In Abbildung 9.6 sind globale Lösungen in vorübergehende Zielzustände  $\mathbf{x}'_f$  dargestellt, die von verschiedenen Stellen der Initialisierung stammen. Ferner sind auf den globalen Lösungen jeweils die maximal gültigen Skalierungen von lokalen Planungen mit  $K = 22$  – also einer gemäß dem vorherigen Abschnitt unzureichenden Horizontlänge – über Vierecke eingezeichnet. Die Gewichte entsprechen den zuvor ermittelten. Abbildung 9.7 zeigt, wie sich der Verlauf der Kosten der skalierten Lösungen durch Verschiebung des Planungsziels verändert. Während sich für  $\mathbf{x}'_{f,A}$  noch ein lokales Minimum vor dem Hindernis ausbildet, dessen Überwinden obendrein abhängig von der Initialisierung ist, sinken die Kosten der Planung für  $\mathbf{x}'_{f,B}$  stetig. Die lokale Lösung reicht zwar nicht bis zum vorübergehenden Zielzustand, wird allerdings nur durch die Horizontlänge beschränkt und nicht durch ein lokales Minimum. Die Kosten einer weiteren lokalen Planung (gepunktet in Abbildung 9.6) ausgehend vom erreichten Punkt bis zum eigentlichen Zielzustand  $\mathbf{x}_f$  innerhalb der Horizontlänge, nehmen schließlich ebenfalls stetig ab. Mit Hilfe des vorübergehenden Zielzustands lässt sich das Hindernis damit auch für  $K \leq K_{LM}^{-}$  überwinden. Je näher das lokale Planungsziel am aktuellen Zustand des Roboters liegt, desto weniger Einfluss hat das Hindernis.

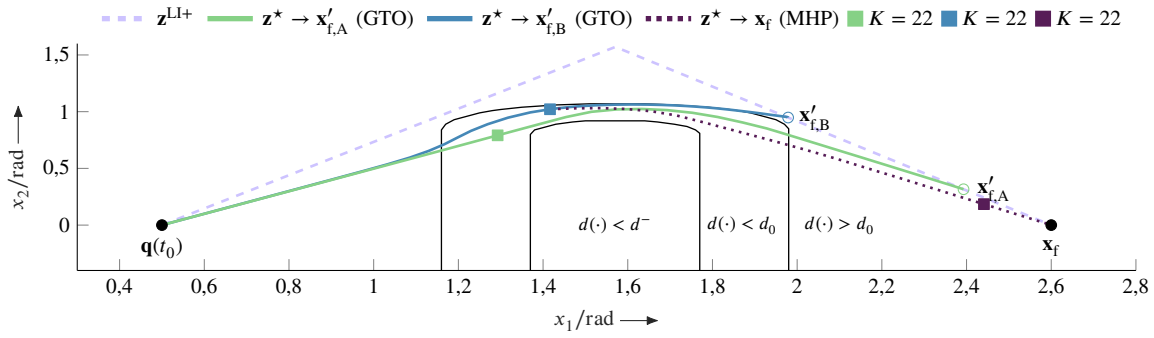


Abbildung 9.6.: Globale Lösungen  $\mathbf{z}^*$  zu den vorübergehenden Zielzuständen  $\mathbf{x}'_{f,A}$  und  $\mathbf{x}'_{f,B}$  auf der Initialisierung  $\mathbf{z}^{LI+}$ . Vierecke markieren die Endpunkte der lokalen Lösungen für  $K = 22$ . Von einem dieser Endpunkte ist die lokale Lösung in den eigentlichen Zielzustand  $\mathbf{x}_f$  gepunktet angegeben.

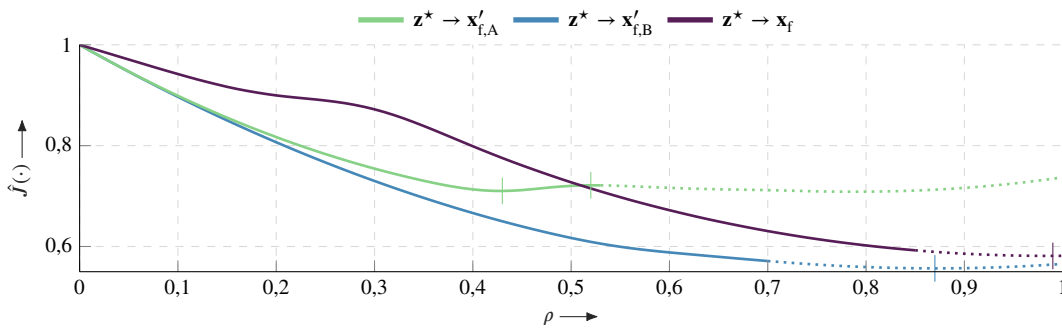


Abbildung 9.7.: Kosten der skalierten lokalen Lösungen zu verschiedenen (temporären) Zielzuständen. Durch Einfügen eines temporären Zielzustands ergeben sich für beide Teilplanungen keine lokalen Minima.

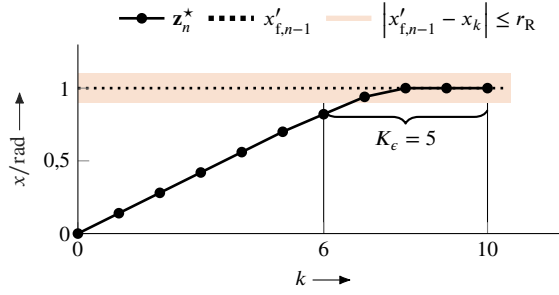
Mit einer ausreichend feinen Aufteilung der globalen Lösung in eine Abfolge von Zwischenpunkten lässt sich das globale Planungsproblem auf mehrere lokale verteilen. Da die globale Lösung a priori jedoch nicht bekannt und nur aufwändig zu generieren ist, stammt der vorübergehende Zielzustand  $\mathbf{x}'_{f,n}$  aus einer globalen Kandidatenmenge  $\mathcal{P} \subseteq \mathbb{R}^N$  und ist lokal erreichbar über mindestens eine gültige Trajektorie. Die Kandidatenmenge sollte möglichst nah an der globalen Lösung liegen, um die Diskrepanz zwischen lokalem und globalem Problem zu reduzieren. Sie wird daher über einen geometrischen Pfad repräsentiert, dessen  $K_p$  linear interpolierte Wegpunkte  $\mathbf{x}_j^P$  aus einer globalen Planungsmethode gewonnen werden:

$$\mathcal{P} := \left\{ \mathbf{x}_0^P, \mathbf{x}_1^P, \dots, \mathbf{x}_j^P, \dots, \mathbf{x}_{K_p-1}^P \right\}. \quad (9.1)$$

Der erste Zustand  $\mathbf{x}_0^P = \mathbf{q}(t_n)$  entspricht dem Roboterzustand zum Zeitpunkt  $t_n$  der Generierung von  $\mathcal{P}$  und der letzte Zustand entspricht dem aktuellen Zielzustand  $\mathbf{x}_{K_p-1}^P = \mathbf{x}_f(t_n)$ . Ferner sei  $\mathcal{P}$  über seine genormte Länge mit  $\rho$  parametrisiert, sodass die Notation  $\mathcal{P}(\rho = 0) = \mathbf{x}_0^P$  und  $\mathcal{P}(\rho = 1) = \mathbf{x}_{K_p-1}^P$  gilt.

Anders als bei Imran u. a. [Imr+16] werden die Punkte  $\mathbf{x}_j^P$  nicht nacheinander abgefahren. Vielmehr stellt  $\mathcal{P}$  eine kontinuierliche Menge dar, aus welcher der vorübergehende Zielzustand  $\mathbf{x}'_{f,n}$  zu jedem Planungsschritt  $t_n$  ermittelt wird:

$$\mathbf{x}'_{f,n} = \mathcal{P}(\rho_n). \quad (9.2)$$


 Abbildung 9.8.: Beispielhafte Berechnung der Horizontreserve  $K_\epsilon$  für  $N = 1$ .

Hierfür wird die passende Skalierung  $\rho_n$  auf dem Pfad bestimmt:

$$\rho_n = \rho_{n-1} + \Delta\rho_n. \quad (9.3)$$

Die Verschiebung  $\Delta\rho_n$  entlang von  $\mathcal{P}$  wird in Abhängigkeit der aktuellen lokalen Lösung  $\mathbf{x}_{0:K-1,n-1}^*$  berechnet:

$$\Delta\rho_n := \max \{0, \beta(K_\epsilon - K_\epsilon^-)\} \quad (9.4)$$

und basiert auf der Anzahl  $K_\epsilon \in \mathbb{N}_0$  an Zuständen  $\mathbf{x}_k$  die zusammenhängend für  $k = K - 1, K - 2, \dots, 0$  innerhalb einer  $\epsilon$ -Umgebung mit Radius  $r_R \in \mathbb{R}^+$  um das bisherige Planungsziel  $\mathbf{x}'_{f,n-1}$  liegen und eine Horizontreserve bilden. Abbildung 9.8 demonstriert die Berechnung von  $K_\epsilon$  anhand eines Beispiels. Der Faktor  $\beta \in \mathbb{R}^+$  skaliert die Schrittweite, mit der sich der Roboter theoretisch in einem Zeitschritt bewegen kann. Der Parameter  $K_\epsilon^- \in \mathbb{N}_0$  beschreibt die Mindestanzahl von Zuständen innerhalb der  $\epsilon$ -Umgebung und stellt damit eine Mindestreserve dar, bei dessen Unterschreiten  $\Delta\rho_n = 0$  gilt. Im Falle dynamischer Hindernisse besteht damit Raum für Ausweichbewegungen, ohne direkt den Kontakt zu  $\mathbf{x}'_{f,n-1}$  zu verlieren. Die Einführung dieser Reserve unterscheidet den vorgestellten Ansatz zur klassischen Pfadfolgeregelung [Zub15; Fau+17] und zur Methode von Imran u. a. [Imr+16]. Blockiert ein dynamisches Hindernis den eben berechneten vorübergehenden Zielzustand  $\mathbf{x}'_{f,n}$ , wird über *Backstepping*  $\rho_n$  soweit iterativ um  $\Delta\rho_{BS} \in \mathbb{R}^+$  reduziert, bis die Kollision aufgelöst ist.

Im ersten Planungsschritt ( $n = 0$ ) gibt es keine vorherige Lösung und es gilt  $\rho_0 = 0$  und  $\mathbf{x}'_{f,0} = \mathbf{x}_0^P = \mathbf{q}(t_0)$ . Die resultierende lokale Lösung führt zu  $K_\epsilon = K$ , da alle Zustände bereits auf  $\mathbf{q}(t_0)$  liegen. Im darauffolgenden Planungsschritt ( $n = 1$ ) verschiebt sich  $\rho_0$  um  $\Delta\rho_1$  gemäß Definition (9.4) und damit auch  $\mathbf{x}'_{f,1}$  entlang von  $\mathcal{P}$ . Die MHP plant und bewegt den Roboter in Richtung dieses vorübergehenden Zielzustands, ehe sich dieser Vorgang für alle anschließenden Planungszyklen wiederholt.

Da die Aufgabe der globalen Zielführung primär das Umfahren von Bereichen mit lokalen Minima ist, übernimmt die MHP in den übrigen Bereichen die alleinige Kontrolle. Um dabei ein Hin- und Herspringen zwischen  $\mathbf{x}'_{f,n}$  und  $\mathbf{x}_f$  zu vermeiden, wird der Wechsel auf die MHP nur in der Nähe von  $\mathbf{x}_f$  gestattet. Liegt der aktuelle Zustand  $\mathbf{q}(t_n)$  des Roboters in einer von  $r_{MHP} \in \mathbb{R}^+$  um den aktuellen Zielzustand  $\mathbf{x}_f$  aufgespannten Hyperkugel  $\mathcal{X}_F(\mathbf{x}_f) \in \mathbb{R}^N$ , plant die MHP auf direktem Weg zu  $\mathbf{x}_f$ :

$$\mathcal{X}_F(\mathbf{x}_f) := \{ \mathbf{x} \in \mathbb{R}^N \mid \|\mathbf{x} - \mathbf{x}_f\|_2 \leq r_{MHP}, c_H(\mathbf{x}) = 0, \mathbf{h}(\mathbf{x}) \leq \mathbf{0} \}. \quad (9.5)$$

Der Radius  $r_{MHP}$  wird so gewählt, dass für alle Zustände die Potentiale sowie Nebenbedingungen  $\mathbf{h}(\cdot)$  Null bzw. inaktiv sind. Damit ist die direkte Planung von allen  $\mathbf{x} \in \mathcal{X}_F(\cdot)$  hindernisfrei. Wie das Beispiel in Abbildung 9.6 zeigt, ist die direkte Planung auch aus Zuständen  $\mathbf{x} \notin \mathcal{X}_F(\cdot)$

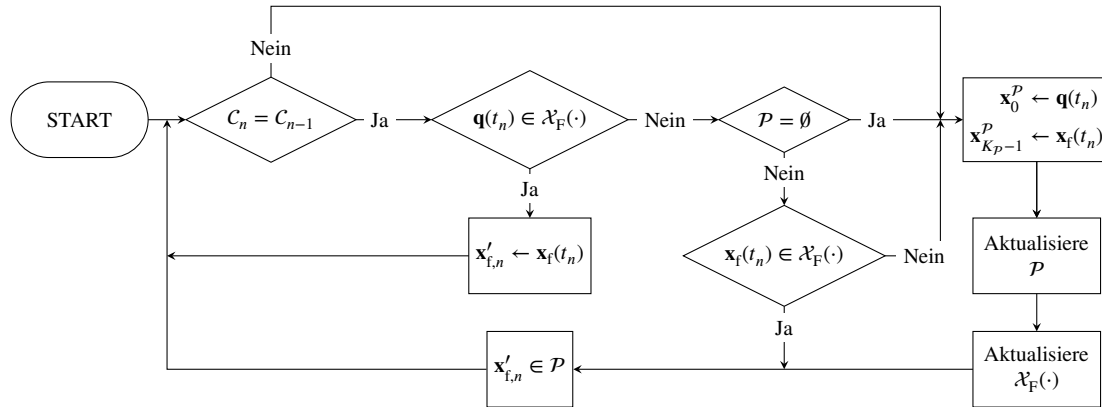


Abbildung 9.9.: Ablauf bei der Berechnung von  $\mathbf{x}'_{f,n}$  bzw. der Aktualisierung von  $\mathcal{P}$  und  $\mathcal{X}_F(\cdot)$ .

möglich. Die Herleitung eines präziseren und weniger restriktiven Bereichs  $\mathcal{X}_F(\cdot)$  bietet sich in weiterführenden Arbeiten über die Gradienten der Kostenterme an.

Da  $\mathcal{X}_F(\cdot)$  hindernisfrei ist, impliziert dies auch, dass nicht nur  $\mathbf{x}_f$  im Zentrum über die direkte Planung erreicht wird, sondern ebenfalls alle anderen Zustände  $\mathbf{x} \in \mathcal{X}_F(\cdot)$  untereinander. Daher wird  $\mathcal{P}$  im Falle eines dynamischen Zielzustands  $\mathbf{x}_f(t)$  nicht laufend erneuert, sondern nur, wenn  $\mathbf{x}_f(t_n) \notin \mathcal{X}_F(\cdot)$ . Diese Hysterese verringert die Häufigkeit mit der  $\mathcal{P}$  aktualisiert wird und reduziert unnötige Variationen für ähnliche Zielzustände, die bei stochastischen globalen Planern auftreten. Sobald  $\mathcal{P}$  erneuert wird, wird auch  $r_{\text{MHP}}$  für den aktuellen Zielzustand  $\mathbf{x}_f(t_n)$  neu bestimmt.

Neben dynamischer Zielzustände wird eine Neuplanung von  $\mathcal{P}$  ebenfalls angestoßen, wenn sich die Mengen der statischen Hindernisse  $C_{n-1} \neq C_n$  zwischen den Planungsschritten ändern, weil z. B. dynamische Hindernisse temporär an einem Ort verharren. Abbildung 9.9 fasst abschließend den Ablauf der Neuplanung und den jeweiligen Ursprung des (vorübergehenden) Zielzustands anhand eines Ablaufdiagramms zusammen.

### Auswahl eines Verfahrens zur Ermittlung des globalen Pfads

Die (Neu-)Planung von  $\mathcal{P}$  erfolgt über globale Planungsverfahren, um das Wissen der gesamten Szene zu nutzen. Die Literatur bietet ein breites Spektrum an globalen Pfadplanern wie RRT oder PRM, die stichprobenbasiert nach dem Zufallsprinzip arbeiten und damit frei von lokalen Minima sind. Da  $\mathcal{P}$  nur zur Hilfestellung dient und nicht akkurat gefolgt wird, werden weniger Kriterien gestellt, als in der Literatur allgemein zum Vergleich von Pfadplanern üblich. Die Pfadplanung findet parallel zur MHP statt, um dessen Ablauf nicht zu verzögern. Da die MHP allein bereits in der Lage ist Kollisionen zu vermeiden, verwendet sie den bisherigen und potentiell leeren Pfad  $\mathcal{P}$ , bis die globale Planung abgeschlossen ist. Trotzdem sollte die Planung von  $\mathcal{P}$  rasch erfolgen, da im Extremfall die MHP für die Dauer der globalen Planung vor einem Hindernis verharret, einen falschen Weg einschlägt oder ein altes Ziel verfolgt. Daher fokussiert sich die Bewertung darauf einen Planer zu identifizieren, der bei geringem Zeitbudget eine möglichst hohe Erfolgsquote für einen beliebigen aber gültigen Pfad aufweist.

Geplant werden jeweils 100 Pfade für den UR10 in zwei Szenarien. Szenario A beinhaltet die Situation einer Hindernisquerung, in der ein längliches senkrecht Hindernis ( $\mathcal{S}_F$ ) die Direktbewegung des Roboters blockiert und z. B. eine Trennwand oder einen temporär statischen Rumpf eines Menschen repräsentiert. Um die Planer dabei stärker herauszufordern, wird der Arbeitsraum über eine unendlich ausgedehnte Ebene auf eine Höhe von 1,5 m beschränkt. Szenario

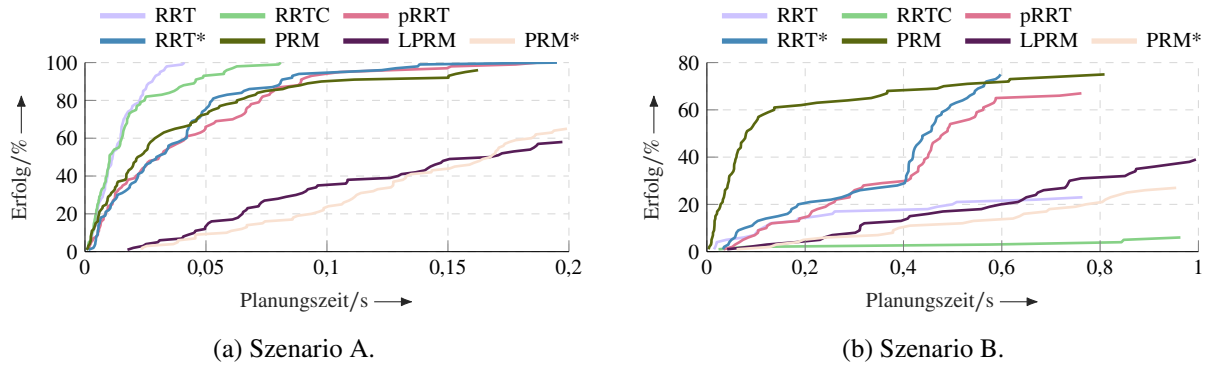


Abbildung 9.10.: Erfolgsquoten globaler Planer bei unterschiedlich hohem Zeitbudget.

B beschreibt die Situation zweier Engstellen, die den Roboter in seiner Bewegungsfreiheit einschränken, wie sie z. B. bei der Aufnahme und dem Ablegen von Objekten aus Regalfächern auftreten. Verwendet werden der klassische RRT, dessen *Connect*- (RRTC [Kuf+00]) und Sternenerweiterung (RRT\* [Kar+11]) sowie die parallele Variante pRRT [Kuf+00]. Hinzu kommen der klassische Roadmap-basierte Planer PRM sowie dessen Lazy- (LPRM [Boh+00]) und Sternenerweiterung (PRM\* [Kar+11]). Weitere Details befinden sich im Anhang B.7.2.

Abbildung 9.10a stellt die Erfolgsquote der Planer in Abhängigkeit des Zeitbudgets in Szenario A dar. Eine Planung ist nicht erfolgreich, wenn sie die jeweilige Planungszeit übersteigt, der Pfad Kollisionen aufweist oder der Zielzustand nicht erreicht wird. Die RRT-basierten Planer erreichen eine Erfolgsquote von 100 % in unter 0,2 s. PRM ist mit 96 % annähernd auf gleichem Niveau, während die anderen PRM-Varianten im Bereich um ca. 60 % abgeschlagen sind. RRT und die Connect-Erweiterung besitzen auch für nur 0,1 s Planungszeit noch eine Erfolgsquote von 100 % und laufen damit synchron zur MHP. Mit durchschnittlich 14 ms für die Planungen bis 0,1 s Planungszeit und damit einer Wiederholffrequenz von ca. 71 Hz, eignet sich der klassische RRT am besten für dieses Szenario. Die Connect-Erweiterung liegt mit durchschnittlich 18 ms dicht dahinter. Die parallele Variante pRRT leidet unter dem Nachteil, dass sich Parallelität erst ab einer bestimmten Problemkomplexität auszahlt, die in Szenario A nicht erreicht wird. RRT\* erreicht das gleiche Niveau, da es durch das Bestreben, die Pfadlänge kleinzuhalten, einen inhärent höheren Aufwand betreibt. PRM schafft es auf ein ähnliches Niveau wie pRRT, wohingegen die Lazy und Star Varianten darunter liegen.

Abbildung 9.10b stellt die Erfolgsquote der Planer in Abhängigkeit des Zeitbudgets in Szenario B dar. Szenario B ist durch die beiden Engstellen komplizierter mit randomisierten Stichproben zu planen und besitzt obendrein 11 Hindernisse vom Typ  $S_F$ . Entsprechend länger dauern die Planungen und senken die erreichbare Erfolgsrate bei 1 s Planungszeit. RRT\* und PRM platzieren sich mit 75 % vor pRRT (67 %), der im Gegensatz zu Szenario A in diesem Fall von seiner Parallelität profitiert. Die anderen RRT Varianten erzielen ähnlich wie die anderen PRM Varianten deutlich weniger Erfolge. Bei geringeren Planungszeiten baut PRM seinen Vorsprung aus und erzielt für 0,15 s eine Erfolgsquote von 61 % gegenüber 15 % für RRT\* als zweitbesten. Die Wiederholffrequenz von PRM ist mit 6,6 Hz damit knapp halb so schnell wie die der MHP.

Die resultierenden Pfadlängen sind aufgrund fehlender Kriterien in der Regel länger als nötig. Grundsätzlich ist jeder der genutzten Planer mit einem Gütemaß erweiterbar, welches der Auswahl des kürzesten aller Pfadkandidaten dient. Hierfür ermittelt der Planer innerhalb eines Zeitbudgets so viele gültige Pfadkandidaten wie möglich. Während dieses Vorgehen bei den ohnehin geringen Planungszeiten in Szenario A durchaus eine Option darstellt, ist dies für längere Planungszeiten

Tabelle 9.1.: Vergleich der räumlichen Pfadlängen unterschiedlicher Planer mit und ohne gesonderter Nachbearbeitung.

	Szenario A				Szenario B	
	RRT	RRTC	pRRT	RRT*	PRM	PRM
Nominell	17,48	13,45	12,47	11,95	12,44	15,45
Nachbearbeitet	8,14	8,00	8,30	8,38	8,27	8,70

in Szenario B unvorteilhaft, da es der Idee einer beliebigen aber zügigen Lösung zunehmend widerspricht. Die Sternerweiterungen RRT\* (und PRM\*) berücksichtigen bei der Ermittlung des Pfades die Pfadlänge und tendieren damit bereits zu kürzeren Pfaden, wie Tabelle 9.1 in Form der durchschnittlichen Pfadlänge zeigt. Alternativ lässt sich der Lösungspfad über eine einmalige Nachbearbeitung kürzen, die schneller zu berechnen ist als neue Pfadkandidaten zu erzeugen aber den Pfadverlauf aus topologischer Sicht nicht derart verändern kann, wie es für einen optimalen Pfad oft notwendig ist. Tabelle 9.1 gibt die resultierenden Pfadlängen der Nachbearbeitung an und verdeutlicht die konsistent erzielten Kürzungen. Hinsichtlich der eher weichen Forderung nach einem kurzen Pfad stellt die Nachbearbeitung daher einen günstigen Kompromiss gegenüber der Suche nach optimalen Pfaden dar.

Da sich die Nachbearbeitung nicht merklich in der Berechnungszeit des Pfades niederschlägt, wird sie standardmäßig bei der Planung von  $\mathcal{P}$  aktiviert. Im weiteren Verlauf wird für Szenario A RRT und für Szenario B PRM verwendet. Der Einheitlichkeit halber wird in beiden Szenarien eine Planungszeit von 0,2 s angesetzt. Damit ist die Planung halb so schnell wie die MHP und besitzt Reserve. Steht nach Ablauf dieser Zeit kein Ergebnis fest, wird die Planung für weitere 0,2 s fortgesetzt. In diesen regelmäßigen Unterbrechungen der Planung wird auf Änderungen der Umgebung und Zielzustände reagiert. Durch Gewährung des Zeitfensters wird verhindert, dass die globale Planung aufgrund von permanenten Änderungen an Umgebung oder Zielzustand nie zu einem Ergebnis kommt. Nach Ablauf der Planungszeit ist dieses Ergebnis zwar potentiell nicht mehr aktuell, bietet der Planung aber dennoch eine Richtung, die insbesondere dann brauchbar ist, wenn der Zielzustand noch weit entfernt ist oder sich die Umgebung nur punktuell verändert hat.

### Parametrisierung von $\beta$ und $K_\epsilon^-$

Der Wert  $\beta$  steuert, wie weit  $\mathbf{x}'_{f,n}$  für jeden Zustand des Horizonts, der in der  $\epsilon$ -Umgebung um den bisherigen Wert von  $\mathbf{x}'_{f,n-1}$  liegt, verschoben wird. Ein zu hoher Wert führt dazu, dass  $\mathbf{x}'_{f,n}$  außer Reichweite des Horizonts gerät. Ein zu kleiner Wert beeinflusst hingegen die Geschwindigkeit des Roboters negativ, wenn durch zunehmendes  $K_\epsilon$  der langsame Teil der Bewegung auf dem Horizont in den Bereich der ersten Stellgröße gelangt und damit permanent das langsame Abbremsen vor dem Zielzustand aus Abbildung 3.2 ausgelöst wird. Entsprechend gilt es, einen Wert zu wählen, der einerseits  $K_\epsilon$  möglichst konstant und oberhalb von  $K_\epsilon^-$  hält und andererseits die Gesamtbewegung nicht ausbremst. Genauere Angaben zu den folgenden Versuchen befinden sich in Anhang B.7.3.

Abbildung 9.11 zeigt den Verlauf von  $K_\epsilon$  für verschiedene  $\beta$  in den Szenarien A und B aus dem vorherigen Abschnitt. Zu Beginn beider Szenarien liegen alle Zustände auf  $\mathbf{x}'_{f,0} = \mathbf{q}(t_0)$ , sodass  $\Delta\rho_0$  am größten ist. Ein höherer Wert für  $\beta$  vergrößert  $\Delta\rho_n$  zusätzlich, sodass für  $\beta > 0,03$  die Horizontreserve  $K_\epsilon$  zunehmend unterhalb von  $K_\epsilon^- = 10$  fällt. Für  $\beta = 0,09$  wird  $\mathbf{x}'_{f,n}$

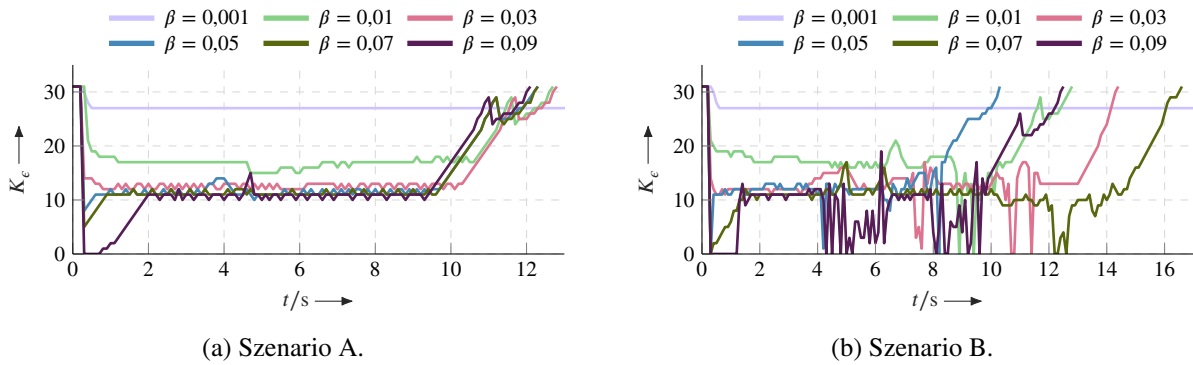


Abbildung 9.11.: Verlauf von  $K_e$  für verschiedene  $\beta$  in beiden Szenarien.

außer Reichweite des Horizonts verschoben und die Gefahr für lokale Minima steigt. Da für  $K_e < K_e^-$  der vorübergehende Zielzustand  $\mathbf{x}'_{f,n}$  nicht weiter verschoben wird und sinngemäß auf die MHP wartet, führt dies dazu, dass  $\mathbf{x}'_{f,n}$  in den darauffolgenden Iterationen wieder eingeholt und die Mindestreserve von  $K_e^- = 10$  eingehalten wird. Im weiteren Verlauf von Szenario A ist erkennbar, wie ein höherer Wert von  $\beta$  dazu führt, dass die Mindestreserve  $K_e^-$  zunehmend angenähert wird. Die Hinderniskonstellation ist weniger komplex als in Szenario B und die Reaktion der Kollisionsvermeidung daher nur für  $\beta \geq 0,05$  im Bereich um  $t = 4$  s leicht sichtbar. Deutlicher ist die Reaktion in Szenario B, in dem mehrere – und mit steigendem  $\beta$  stärkere – Einbrüche von  $K_e$  auftreten. Nur für  $\beta = 0,001$  wird  $K_e^-$  eingehalten, jedoch geht dies zu Lasten der durchschnittlichen Geschwindigkeit, die mit  $0,16 \text{ rad s}^{-1}$  nur noch knapp ein Drittel der von  $\beta = 0,01$  ( $0,56 \text{ rad s}^{-1}$ ) entspricht. Ähnliches gilt auch für Szenario A. Als Kompromiss wird daher im weiteren Verlauf in beiden Szenarien  $\beta = 0,01$  gewählt.

Mit  $K_e^-$  wird eine weiche untere Schranke vorgegeben, die als Mindestreserve bei geeignetem  $\beta$  für Ausweichbewegungen zur Verfügung steht. Im Folgenden wird Szenario A um ein dynamisches Hindernis ( $S_L$ ) erweitert, das sich mit  $0,2 \text{ m s}^{-1}$  von unten in die lokale Lösung der MHP drängt. Daran wird untersucht, wie weit das Hindernis eindringt, bevor  $\mathbf{x}'_{f,n}$  außer Reichweite des Horizonts gerät. Die Stellgrößen werden dafür vom Roboter nicht umgesetzt. Ferner verwendet die MHP die Schnappschussmethode, um die Ergebnisse bezüglich der momentanen Hindernisposition nicht zu verfälschen. Abbildung 9.12 zeigt den Verlauf von  $K_e$  nebst z-Koordinate des dynamischen Hindernisses für verschiedene  $K_e^-$  und  $\beta = 0,01$ . Nach 4 s beginnt die MHP jeweils auf das Hindernis zu reagieren. Ohne Mindestreserve ist bereits nach 2,6 s  $K_e = 1$  und fällt nach 4 s auf Null. Danach entfernt sich der letzte Zustand der lokalen Lösung zunehmend von  $\mathbf{x}'_{f,n}$ . Bei einer Mindestreserve von  $K_e^- = 10$  fällt  $K_e$  zunächst in den ersten 3 s auf genau diesen Wert, da

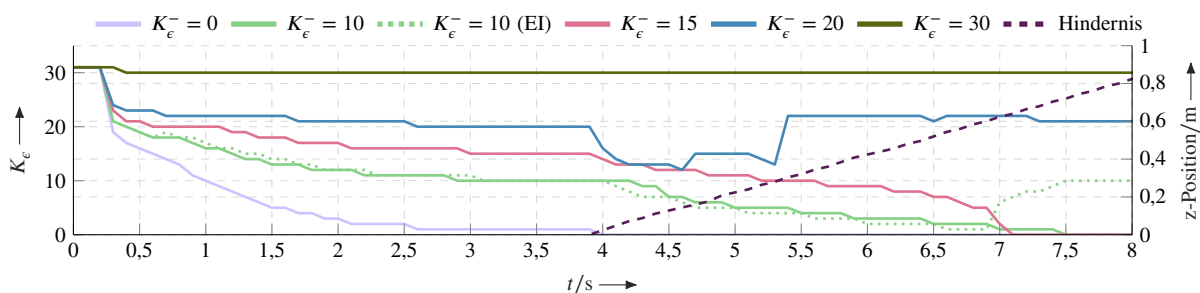


Abbildung 9.12.: Verlauf von  $K_e$  für verschiedene  $K_e^-$  und  $\beta = 0,01$ . Ebenfalls angegeben ist der Verlauf der z-Position des Hindernisses. EI steht für die erweiterte Initialisierung aus Kapitel 8.

$\mathbf{x}'_{f,n}$  langsam entlang  $\mathcal{P}$  verschoben wird. Dringt das Hindernis dann in die lokale Lösung ein, fällt  $K_e$  innerhalb der nächsten 3,5 s auf 0, während sich das Hindernis auf 72 cm Höhe bewegt. Für  $K_e^- = 15$  ist ein ähnliches Verhalten erkennbar, wobei durch die Variation von  $\mathcal{P}$ ,  $K_e = 0$  0,4 s früher erreicht wird. Für ein  $K_e^- = 20$  befindet sich  $\mathbf{x}'_{f,n}$  bereits so nah am Roboter, dass es vom Hindernis touchiert wird und das *Backstepping* auslöst. Zuerst nach 4,6 s und anschließend nochmals bei 5,4 s. Das *Backstepping* verschiebt  $\mathbf{x}'_{f,n}$  soweit zurück, bis es kollisionsfrei ist, was den jeweiligen Anstieg von  $K_e$  erklärt. Ist das Hindernis vorbeigezogen, wird  $\mathbf{x}'_{f,n}$  wieder normal entlang von  $\mathcal{P}$  verschoben bis nach 8,3 s die Mindestreserve von  $K_e^- = 20$  erreicht wird. Bei  $K_e^- = 30$  erstreckt sich effektiv fast keine lokale Lösung durch den Zustandsraum, die von einem Hindernis beeinträchtigt werden könnte. Je größer  $K_e^-$ , desto weniger Horizont steht für das Folgen von  $\mathbf{x}'_{f,n}$  zur Verfügung, wodurch sich die MHP zunehmend im Sinne einer Pfadfolge entlang von  $\mathcal{P}$  bewegt. Setzt der Roboter die Stellgrößen um und wird die Geschwindigkeit mit einbezogen, provoziert ein  $K_e^- \rightarrow K$  genauso wie  $\beta \rightarrow 0$  den Effekt des langsamen Abbremsens. Somit ist die Durchschnittsgeschwindigkeit für  $K_e^- = 30$  mit  $0,05 \text{ rad s}^{-1}$  ca. zehnmal kleiner als für  $K_e^- = 15$  ( $0,53 \text{ rad s}^{-1}$ ) oder  $K_e^- = 10$  ( $0,56 \text{ rad s}^{-1}$ ).

In Anbetracht der ähnlichen Ergebnisse für  $K_e^- = 10$  und  $K_e^- = 15$  wird zugunsten einer höheren Geschwindigkeit für die weiteren Versuche  $K_e^- = 10$  festgelegt. Wird anstelle des Warmstarts die erweiterte Initialisierung aus Kapitel 8 verwendet, ermittelt die MHP gegebenenfalls noch vor dem Aufbrauchen der Reserve eine alternative Lösung unter dem Hindernis hindurch, wodurch die Horizontreserve wieder aufgebaut wird. Dieser Fall ist in Abbildung 9.12 für  $K_e^- = 10$  (EI) dargestellt.

## Neuplanung bei dynamischem Zielzustand

Im Folgenden werden die Auswirkungen der Neuplanung bei einem dynamischen Zielzustand ausgewertet. Hierfür wird der Zielwert des dritten Gelenks aus Szenario A (zuvor 0 rad) durch ein sinusförmiges Signal mit einer Amplitude von 0,7 rad und Periodendauer von 10 s ersetzt. Wie zuvor ermittelt, ist  $\beta = 0,01$  und  $K_e^- = 10$ . Weitere Angaben befinden sich in Anhang B.7.4. Abbildung 9.13 zeigt die Verläufe des dritten Gelenkwinkels des UR10 für Regionen  $\mathcal{X}_F(\cdot)$  mit unterschiedlichen Radien. Je kleiner  $r_{\text{MHP}}$ , desto häufiger wird  $\mathcal{P}$  erneuert, was zeitlich durch die Punkte auf den Trajektorien markiert ist. Für  $r_{\text{MHP}} = 0,1 \text{ rad}$  und  $r_{\text{MHP}} = 0,5 \text{ rad}$  verzögert die häufige Aktualisierung von  $\mathcal{P}$  die Gesamtbewegung, da  $\mathbf{x}'_{f,n}$  nach jeder Aktualisierung bei  $\rho_n = 0$  beginnt und die MHP für einen Planungszyklus ausbremst. Für  $r_{\text{MHP}} = 0,9 \text{ rad}$  wird  $\mathcal{P}$  nur zu Beginn einmalig ermittelt, womit keine Verzögerungen auftreten. Ist das Hindernis überwunden, beeinträchtigt bei  $r_{\text{MHP}} = 0,1 \text{ rad}$  das häufige Anhalten das Folgerverhalten und führt auf einen

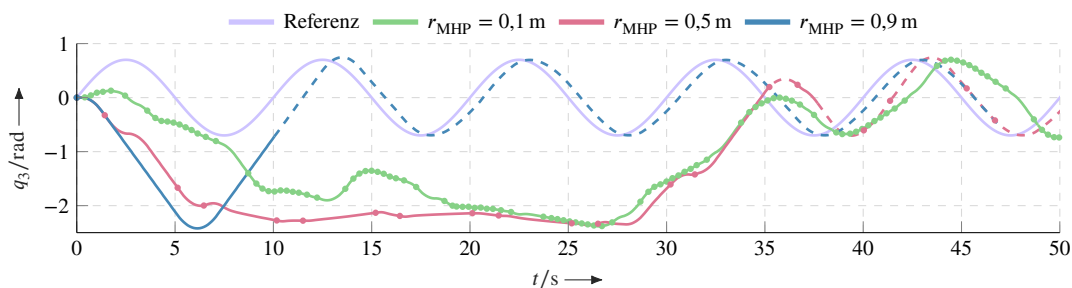
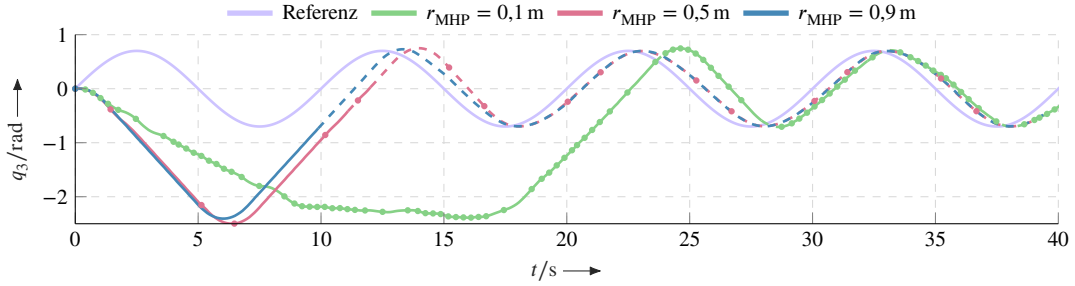


Abbildung 9.13.: Folgerverhalten im dritten Gelenk bei unterschiedlichen Radien von  $\mathcal{X}_F(\cdot)$ . Punkte kennzeichnen Neuplanungen von  $\mathcal{P}$ . Abschnitte direkter Planung sind gestrichelt dargestellt.


 Abbildung 9.14.: Wiederholung des Versuchs mit Vorkalierung von  $\rho_n$ .

Folgefehler von 1,06 rad. Bei  $r_{\text{MHP}} = 0,5$  rad wird dies durch Einsetzen der direkten Planung (gestrichelter Abschnitt) verhindert und es resultiert ein Folgefehler von 0,28 rad. Dieser ist auf ähnlichem Niveau (0,25 rad) wie für den Fall  $r_{\text{MHP}} = 0,9$  rad, der sich bereits nach ca. 10 s in der Direktplanung befindet.

Wird  $\rho_n$  nach Aktualisierung von  $\mathcal{P}$  nicht auf 0 gesetzt, sondern vorkaliert auf:

$$\rho_n = (K - K_e^-)\beta, \quad (9.6)$$

wird das Abbremsen der MHP abgemildert. Über Gleichung (9.6) wird der für die Punktfolge verfügbare Teil des Horizonts bereits zum Verschieben von  $\rho_n$  eingesetzt, ohne auf eine lokale Lösung zu warten, die bei  $\mathbf{x}'_{f,n} = \mathbf{q}(t_n)$  einen Planungszyklus später ohnehin zu  $K_e = K$  führt. Abbildung 9.14 zeigt die Ergebnisse der Vorkalierung für die zuvor gezeigten Fälle und verdeutlicht, wie das Hindernis in den drei Fällen früher überwunden wird. Der Fall  $r_{\text{MHP}} = 0,5$  rad ist fast gleich schnell wie bei  $r_{\text{MHP}} = 0,9$  rad, während für  $r_{\text{MHP}} = 0,1$  rad immerhin eine Zeitersparnis<sup>1</sup> von ca. 11 s erreicht wird. Der konvergierte Folgefehler für  $r_{\text{MHP}} = 0,1$  rad wird auf 0,34 rad gesenkt. Damit wird durch die initiale Verschiebung von  $\rho_n$  bei neuem  $\mathcal{P}$  der Notwendigkeit einer größeren Region  $\mathcal{X}_F(\cdot)$  entgegengewirkt.

## Qualitativer Vergleich zu anderen Methoden

Abschließend folgt anhand von Szenario B ein qualitativer Vergleich der vorgestellten Methode mit der Variante von Imran u. a. [Imr+16] (im Folgenden HPP<sup>2</sup> genannt), GTO und STOMP. GTO und STOMP sind mit der konventionellen Initialisierung über  $\mathbf{z}^{\text{LI}}$  nicht in der Lage Lösungen zu ermitteln, weshalb sie mit dem Pfad  $\mathcal{P}$  initialisiert werden, der auch für  $\mathbf{x}'_{f,n}$  dient. HPP wird über die MHP mit globaler Punktfolge und den Parameterwerten  $K = 2$ ,  $K_e^- = 0$  und  $\beta = 0,1$  imitiert. Dies entspricht einem sequentiellen Abfahren von äquidistanten Punkten auf  $\mathcal{P}$ .

Abbildung 9.15 vergleicht die zweite Komponente von  $\mathbf{x}'_{f,n}$  für HPP und MHP. Zu erkennen sind die äquidistanten Sprünge bei HPP, die immer dann eintreten, wenn sich der aktuelle Roboterzustand in der  $\epsilon$ -Umgebung befindet. Je kleiner die Sprünge, desto langsamer bewegt sich der Roboter aufgrund des kurzen Horizonts von  $K = 2$ , womit diese Methode mit 23,7 s am längsten braucht. Für  $\beta > 0,1$  entstehen direkt zu Beginn lokale Minima, da sich  $\mathbf{x}'_{f,n}$  zu weit vom Roboter entfernt. Für  $\beta = 0,1$  hält sich HPP aufgrund des fehlenden Horizonts dicht am Verlauf von  $\mathcal{P}$ . Dies wird von Abbildung 9.16 verdeutlicht, in der jeweils zeitlich passend<sup>3</sup> skalierte Varianten

<sup>1</sup>Jeweils gemessen am Zeitpunkt des minimalen Gelenkwinkels.

<sup>2</sup>Hybrid Path Planning

<sup>3</sup>Ein Pfad besitzt kein Zeitprofil. Daher wird er räumlich äquidistant aufgelöst und so skaliert, dass sein Endpunkt mit dem Erreichen der Gesamtbewegung am Zielzustand zusammenfällt.

von  $\mathcal{P}$  den Gesamtbewegungen des zweiten Gelenks gegenübergestellt sind. Die MHP mit globaler Punktfolge folgt der Form von  $\mathcal{P}$  zwar im ähnlichen Maße, ist dabei jedoch ca. 11 s eher am Ziel. Der glattere Verlauf von  $\mathbf{x}'_{f,n}$  schlägt sich außerdem in eine ruhigere Steuerfolge nieder. STOMP kann in ungleichmäßig dichten Arbeitsräumen seine Stärke in Form der gradientenfreien Variation von Trajektorienkandidaten nicht effektiv einsetzen. Damit die Variation im beengten Bereich (Regalfächer) gültige Lösungen ermittelt, ist diese entsprechend klein zu wählen, was sich in Bereichen mit mehr Freiraum (vor den Regalen) wiederum als zu gering erweist. Wie Abbildung 9.16 zeigt, entspricht die Lösung erwartungsgemäß einer Variation der Punkte aus  $\mathcal{P}$ . Obwohl beide Verläufe damit ähnlich sind, ist die Lösung von STOMP mit 8,1 rad länger als  $\mathcal{P}$  (7,7 rad). GTO erzielt mit der Optimierung der globalen Trajektorie ausgehend von  $\mathcal{P}$  als Initialisierung den kürzesten Pfad (5,7 rad) und erreicht das Ziel nach 8,7 s am schnellsten. Sofern eine gültige Initialisierung vorliegt, profitiert GTO von der durch Gradienten zielgerichteten Veränderung/Optimierung der Lösung, wie insbesondere die Unterschiede zu  $\mathcal{P}$  in den ersten 3 s demonstrieren.

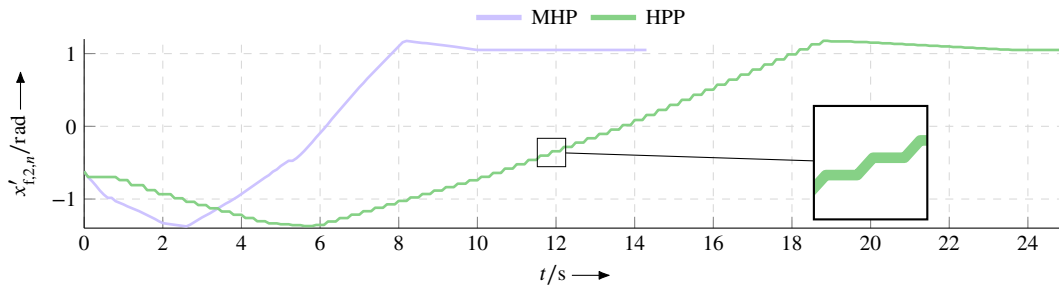


Abbildung 9.15.: Verlauf der zweiten Komponente von  $\mathbf{x}'_{f,n}$  in Szenario B.

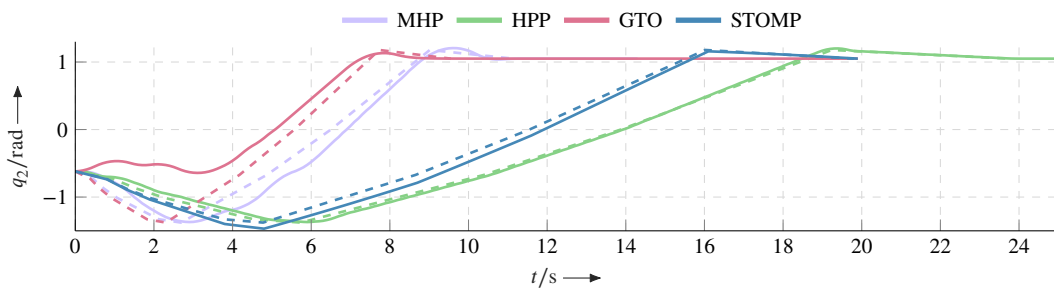


Abbildung 9.16.: Verlauf des zweiten Gelenkwinkels des Roboters in Szenario B gestrichelt hinterlegt mit skalierten Varianten von  $\mathcal{P}$ .

# 10

## Zusammenfassung und Ausblick

Das übergeordnete Ziel der Arbeit ist die Anwendbarkeit der MHP hinsichtlich Planungsfrequenz, Kollisionsvermeidung und Lokalität zu bewerten und zu erhöhen. Dieses Ziel wird in Form von drei Teilzielen verfolgt:

Im ersten Teil werden zunächst technische Herausforderungen bezüglich einer effizienten Trajektorienoptimierung zur Erhöhung der Planungsfrequenz am Beispiel des kollaborativen Roboters UR10 behandelt. Hierunter fällt die Formulierung dreier unterschiedlich transkribierter NLPs, in welchen Zustand und Stellgröße unter Ausnutzung der unterlagerten Reglerkaskaden von Servoantrieben jeweils von Gelenkwinkeln und deren Geschwindigkeiten repräsentiert werden. Die NLPs werden anschließend mit distanzbasierten Kosten und Nebenbedingungen zur Kollisionsvermeidung vervollständigt. Als Hüllkörper werden SSVs aufgrund ihres vorteilhaften Kompromisses zwischen Komplexität und der zur Modellierung günstigen Formen eingesetzt. Die Auswertung der Berechnungszeiten von Distanzfunktionen mit unterschiedlicher Implementierung empfiehlt eine partielle Reimplementierung bestimmter Fälle zur Vermeidung von Zeitverlusten an den Schnittstellen zu bestehenden Bibliotheken. Die vorgestellte approximative Penetrationsdistanz sorgt dafür, dass sich in Lösungsverfahren stets ein Gradient zur Auflösung von Kollision berechnen lässt und leistet dies dank der Ausnutzung von Zwischenergebnissen der Distanzberechnung ohne nennenswerten Mehraufwand.

Mit dem Ziel einer finalen Auswahl von Transkriptionsmethode und Lösungsverfahren für die MHP erfolgt eine Bewertung der Effizienz verschiedener Kombinationen sowie die Verkürzung der Berechnungszeit durch die Repräsentation dünnbesetzter Strukturen und dessen Nutzung in spezialisierten Lösungsverfahren. Während sich ohne Strukturausnutzung noch der Vorteil des Mehrfachschießverfahrens zur Abwägung zwischen Sensitivität und Größenordnung für IP- und SQP-Verfahren auszahlt, lässt sich die Auswirkung der Größenordnung bei der vollständigen Diskretisierung mit Strukturausnutzung soweit kompensieren, dass zusammen mit dem IP-Verfahren die niedrigste Optimierungszeit erzielt wird. Auch in weiteren Vergleichen setzt sich die finale Kombination IPTOP, bestehend aus dem IP-Verfahren IPOpt (MA57), Hypergraph und vollständiger Diskretisierung, gegen andere spezialisierte Verfahren der Trajektorienoptimierung und optimalen Regelung wie HQP und ALTRO hinsichtlich der Optimierungszeit durch. Hierbei skalieren IPTOP und HQP ähnlich mit der Horizontlänge bei einem nahezu konstantem Versatz und IPTOP mit Vorteilen bei steigender Hinderniszahl.

Über ein Zeitbudget wird ein zeitliches Abbruchkriterium realisiert, welches die Planung mit einer festen Frequenz erlaubt. Die Ergebnisse für den Extremfall, in dem das Zeitbudget nur für eine Iteration ausreicht, zeigen, dass bis zu 220 kugelförmige Hindernisse bei einer Horizontlänge von  $K = 11$  und Planungsfrequenz von 10 Hz möglich sind. Zu beachten ist, dass die Qualität der Lösungen für sinkende Iterationen zunehmend abnimmt, auch wenn sich dies

durch eine geeignete Initialisierung, wie z. B. den Warmstart, abmildern lässt. Die Bewertung von suboptimalen Lösungen erfolgt in der Arbeit nur individuell für jeden Planungszyklus und ist daher potentiell restriktiv. Von weiterem Interesse ist die Qualität der resultierenden Gesamtbewegung als Ergebnis aller Planungszyklen. Hierbei können sich stärkere Abweichungen in den Einzellösungen über den gesamten Verlauf relativieren, weshalb eine Ausweitung dieser Analyse für weiterführende Arbeiten empfohlen wird. Die Lösung des Online-Dilemmas erfolgt über eine Prädiktion des eigentlichen Startzustandes der Optimierung, die aufgrund der Exklusivität der MHP bezüglich der Stellgrößen mit ausreichend hoher Genauigkeit erfolgt.

IPTOP ist primär modularisiert aufgebaut, um verschiedene Kombinationen von Methoden, deren Wechselwirkungen und Verbesserungen zu untersuchen. Gleichzeitig steht bei den Realisierungen die Möglichkeit im Vordergrund, für Auswertungen auf möglichst viele Daten und Signale Zugriff zu erhalten. Beides begrenzt die Effizienz der Implementierung und bildet Optimierungspotential. Eine Laufzeitanalyse der Nebenbedingungen und individuellen Komponenten der Kostenfunktion zeigt die Dominanz der Distanzberechnungen und Vorwärtskinematik. Durch einen Vergleich mit theoretischen Laufzeiten wird Einsparpotential durch eine Optimierung der Reihenfolge von Berechnungen im Lösungsverfahren und Speicherung von Zwischenergebnissen motiviert, die im Rahmen einer Spezialisierung von Lösungsverfahren auf die Trajektorienoptimierung als Teil weiterführender Arbeiten vielversprechend ist.

Der zweite Teil der Arbeit demonstriert die Möglichkeiten der MHP unter dynamischen Szenen vor dem Hintergrund der erzielten Planungsfrequenzen und ermöglicht eine Einordnung hinsichtlich der Anwendbarkeit im Kontext kollaborativer Roboter. An erster Stelle steht dabei die Auswertung anhand eines Szenarios aus der MRK, in dem sich die Bewegungen von Roboter und Mensch kreuzen und die Kollisionsvermeidung von dynamischen Hindernissen auf die Probe gestellt wird. Hierbei werden die Vorgaben und Empfehlungen zur Auslegung von Gewichten und Schwellwerten um die der Kollisionsvermeidung zu dynamischen Hindernissen komplettiert. Die Aktivierungsschwellwerte der Annäherungskosten werden stets oberhalb der jeweiligen Nebenbedingungen angesetzt und folgen bei statischen Hindernissen praktischen Vorgaben zur Bewegungsfreiheit des Roboters. Bei dynamischen Hindernissen übernehmen sie die Aufgabe eines Wahrnehmungsbereichs der insbesondere ohne Hindernisprädiktion essentiell für einen kollisionsfreien Betrieb ist. Eine Hindernisprädiktion erlaubt die relative Reduzierung des Aktivierungsschwellwerts und steigert die Praktikabilität in Form größerer Bewegungsfreiräume von Roboter und Mensch. Dabei erzielt bereits eine einfache Prädiktion in Form eines extrapolierten Zustands, der über einen Polynomfilter geschätzt wird, merkliche Verbesserungen. Wenn Armbewegungen des Menschen bei der Prädiktion zu berücksichtigen sind, bietet sich wie von Pereira u. a. [Per+17] der Weg über den Gelenkraum des Menschen an. Durch die effiziente Optimierung im ersten Teil werden mit und ohne Hindernisprädiktion bei einer Horizontlänge von  $K = 31$  und der Kollisionsvermeidung zu einer Person eine Planungsfrequenz von 10 Hz eingehalten und Kollisionen erfolgreich vermieden.

Die Ergebnisse zeigen das grundsätzliche Potential von IPTOP auf und motivieren den Transfer auf konkrete Anwendungsfälle, wie es z. B. das Forschungsprojekt [Krä+22b] durchführt, das aus dieser Arbeit hervorgeht. Das Forschungsprojekt verfolgt ferner die Weiterführung dieser Arbeit hinsichtlich der Berücksichtigung von Prädiktionsunsicherheiten in Form eines Multi-Stage-Ansatzes sowie den Nutzen von neuartigen netzbasierten Bewegungsprädiktoren [Ren+23c]. Eine Ausweitung auf normgestützte Bedingungen bezüglich Hindernisgeschwindigkeiten nach ISO 13855 [ISO10] ist ebenfalls naheliegend. Der Quelltext von IPTOP ist im Rahmen der Publikation [Ren+24] öffentlich zugänglich.

---

Als zweiter dynamischer Aspekt wird das Folgeverhalten eines variablen Zielzustands untersucht und mit den Ergebnissen des IKA verglichen. Die Ergebnisse zeigen, dass beim reinen Folgen einer Pose mit unbekannter Richtung die MHP an die Qualität des IKA heranreicht. Insbesondere das Übergangsverhalten in eine entfernte Zielpose ist Stärke der MHP, was beim IKA nur über einen Kompromiss mit dem Folgeverhalten zu verbessern ist. Die MHP stellt demnach eine flexible Methode für das Folge- und Übergangsverhalten inklusive Kollisionsvermeidung dar, benötigt für Zielposen allerdings eine vorherige Umwandlung in den Gelenkraum oder die Ausweitung der Annäherungskosten wie in [Krä+21c].

Neben dynamischen Hindernissen und Zielkonfigurationen wird die Nutzung von zur Planungszeit variierenden Optimierungskriterien in Form der zu maximierenden Effektivität von Schwingungsregelkreisen beim gliedelastischen Roboter TUDORA betrachtet. Die Berücksichtigung der Effektivität über sekundäre Kosten trägt insgesamt zu einer schnelleren Dämpfung von Schwingungen bei, als die Verwendung eines konventionellen Gütemaßes. Die bedarfsgesteuerte Aktivierung ist für die betrachtete Aufgabe besser geeignet, da sie eine temporäre Verbesserung der Effektivität auch in der Nähe des Zielzustands bietet, wo sie die proaktive Variante bereits verstärkt ignoriert. Wenn die Bewegung zur Erhöhung der Effektivität selbst Schwingungen verursacht, führt ein zu niedriger Aktivierungsschwellwert zu Selbstanregungen von Oszillationen. Auf der anderen Seite untergräbt ein zu hoher Schwellwert das bedarfsgesteuerte Konzept. Die Wahl des Aktivierungsschwellwerts ist damit entscheidend und erfordert eine vorherige Analyse von Schwingungen am realen System. Eine prophylaktische Erhöhung der Effektivität durch die Priorisierung bietet für den gegebenen Roboter keine signifikanten Vorteile. Diese Variante bleibt dennoch eine valide Alternative und besitzt gegenüber dem Nullraum den Vorteil, dass sie Anwendungen unterstützt, in denen der Zielzustand zwar final oberste Priorität hat, aber während der Bewegung andere Ziele gleichwertig sind.

Die Möglichkeit von variablen Planungskriterien ist ein Vorteil der MHP, birgt allerdings auch die Gefahr lokaler Minima. Während sie in der gegebenen Aufgabe über eine Endzustandsnebenbedingung und ausreichende Horizontlänge eliminiert werden, empfiehlt sich für eine allgemeinere Lösung die Kombination mit der globalen Punktführung aus Kapitel 9. In weiterführenden Arbeiten lässt sich darüber hinaus die Aktivierung der bedarfsgesteuerten Terme für jede Schwingungsebene individuell durchführen, um den Bewegungsaufwand zu reduzieren und damit potentiellen Selbstanregungen vorzubeugen.

Im dritten Teil geht die Arbeit auf Herausforderungen hinsichtlich der Lokalität von Lösungsverfahren und Optimierungsproblem ein. Zur Adressierung der Lokalität des Lösungsverfahrens wird die MHP mit explorativem Verhalten über eine erweiterte parallele Initialisierung ausgestattet, die gleichzeitig den Warmstart zur potentielle Verbesserung der Berechnungszeiten erlaubt. Ein Vergleich mit Lösungen globaler Verfahren zeigt, wie die lokal beschränkten Informationen effektiver ausgenutzt werden. Über die Initialisierung der explorativen Parallelinstanz mit gespielten Altlösungen lassen sich Umwege in Form von ausschweifenden Ausweichbewegungen in den gezeigten Szenarien früher vermeiden, als bei einer herkömmlichen linearen Interpolation. Weiterführend lässt sich die Idee der explorativen Parallelinstanz auf die Bewertung von unterschiedlichen Zielzuständen erweitern, wie es z. B. im Rahmen des Forschungsprojekts [Krä+22b] zum Finden und Anfahren eines optimalen Zielzustands verfolgt wird, aus welchem Tätigkeiten von Personen an einem Werkstück verdeckungsfrei zu dokumentieren sind. Im gleichen Zug lassen sich weitere Formen von explorativen Initialisierungen evaluieren, die neben einer Spiegelung feste Gegebenheiten des Arbeitsraums, wie z. B. freigehaltene Bereiche, berücksichtigen. Unterstützt wird dies durch das größere Zeitbudget der parallelen Instanz, welches das Aufstellen aufwändiger Initialisierungen erlaubt.

Über die Einführung temporärer Zielzustände aus einer globalen Punktfolge wird abschließend eine Methode zur Auflösung struktureller lokaler Minima vorgestellt, welche die Lokalität des Optimierungsproblems adressiert. Strukturelle lokale Minima werden erfolgreich überwunden, während der MHP gleichzeitig Raum zum lokal optimalen Ausweichen vor dynamischen Hindernissen gegeben wird. Dynamische Zielzustände werden ebenfalls unterstützt und die Ergebnisse zum Folgeverhalten sind entsprechend anwendbar. Die Parametrisierung steuert die Nähe zum verfolgten Punkt sowie die Horizontreserve und beeinflusst damit die Anfälligkeit für strukturelle lokale Minima und Ausweichreserven für dynamische Hindernisse auf Kosten der Ausnutzung der verfügbaren Robotergeschwindigkeit. In Szenen mit steigender statischer Hindernisbelastung und Engstellen sind niedrige Geschwindigkeiten zunehmend vertretbarer und eine Nähe zum verfolgten Punkt aufgrund der Gefahr von strukturellen lokalen Minima ohnehin zu präferieren. Auf gleiche Weise lässt sich für Szenen mit höherer Belastung von dynamischen Hindernissen argumentieren.

Die globale Punktfolge basiert auf der Annahme, dass strukturelle lokale Minima nur von statischen Hindernissen ausgehen. In bestimmten Situationen stellen allerdings auch dynamische Hindernisse eine Gefahr dar. Die Bewertung des konkreten Ausmaßes und etwaige Lösungsansätze empfehlen sich in zukünftigen Arbeiten. Ferner basiert die Methode auf der Annahme und Kenntnis einer hindernisfreien Region um den Zielzustand. In weiterführenden Arbeiten bietet sich daher eine Ausweitung der Systematik zur präziseren Ermittlung dieser Region an, da eine größere Region die Häufigkeit der globalen Neuplanung bei dynamischen Zielzuständen senkt. Die Erstellung der Punktmenge, aus welcher die temporären Zielzustände entnommen werden, basiert auf globalen Pfadplanern. Obwohl eine Auswahl anhand zweier unterschiedlich komplexer Szenarien hinsichtlich Planungserfolg und Planungszeit erfolgt und ausreichende Geschwindigkeiten erzielt werden, empfiehlt sich dennoch eine Bewertung der Vorteile der Verfahren von Leven u. a. [Lev+02] und Pomarlan u. a. [Pom+13], die ihre Datenstrukturen bei Änderungen der Umgebung nur lokal aktualisieren. Die globale Punktfolge ermöglicht außerdem die Synchronisation von Arbeitsabläufen zwischen Mensch und Roboter, die sich nach Huber u. a. [Hub+13] und Someshwar u. a. [Som+12] positiv auf eine flüssige Zusammenarbeit auswirkt. Die Bewertung des Potentials in diesem Anwendungsbereich stellt ebenfalls eine interessante Aufgabenstellung dar.

Aus der Analyse der Kostenfunktion für verschiedene Horizontlängen beim Umfahren statischer Hindernisse wird eine Systematik zur Auswahl der Gewichte der Planung abgeleitet. Während diese Systematik in der aktuellen Form unter anderem von Aspekten wie Hindernisposition und Zielzustand abhängt, lässt sich ihr Nutzen in weiterführenden Arbeiten durch Entflechtung der Abhängigkeiten potentiell steigern und die Methode verallgemeinern.

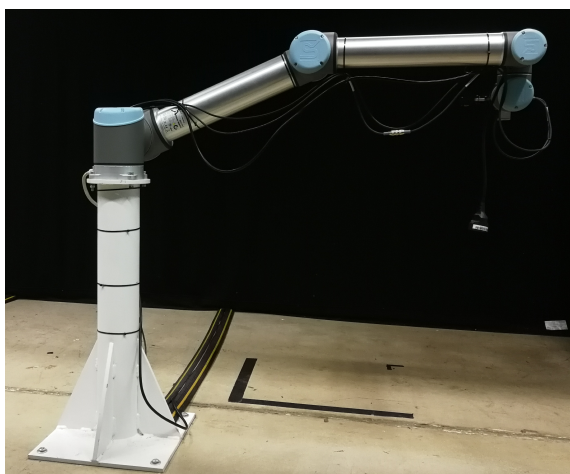
# A

## Daten zu den verwendeten Robotersystemen

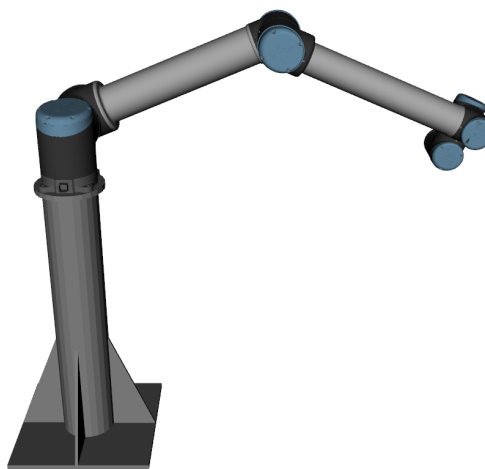
In diesem Anhang befinden sich Begleitinformationen zu den verwendeten Robotersystemen.

### A.1. Universal Robot 10 (UR10)

Der Roboterarm *Universal Robot 10* besteht aus  $N = 6$  Gelenken und ist in Abbildung A.1 sowohl als Foto, als auch als Modell in der Simulation abgebildet.



(a) UR10 in der Laborumgebung



(b) UR10 in der Simulationsumgebung

Abbildung A.1.: *Universal Robot 10*.

Die Simulation erfolgt im ROS<sup>1</sup> (Melodic) mit den Programmen RViz (Visualisierung und kinematische Simulation) und Gazebo (Physiksimulation). Das ROS stellt unter anderem eine Reihe von Kommunikationsdiensten zur Interaktion mit anderen Anwendungen (Knoten) bereit und ermöglicht dadurch die dezentrale Verteilung und flexible Kombination von Funktionen. Zur detailgetreuen Darstellung des Roboters wurde ein für die Ziele dieser Arbeit angepasstes Simulationspaket auf der Basis eines Pakets von ROS-Industrial [ROS17] erstellt. Zur Kommunikation mit dem realen Roboter kommt das ROS-Paket von Andersen [And15] zum Einsatz, das

---

<sup>1</sup>Robot Operating System

durch die Implementierungen zur Abstandsberechnung (Kapitel 4) um eine Sicherheitsebene zur treiberseitigen Kollisionsvermeidung im Labor erweitert wurde. Herstellerseitig läuft auf dem Roboter die Controlsoftware CB3 3.12.1.90940.

Die Transformationsmatrizen der direkten Kinematik lauten:

$$\begin{aligned}
 \mathbf{T}_B &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,8 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_1(q_1) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0,1273 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 \mathbf{T}_2(q_2) &= \begin{bmatrix} -\sin(q_2) & 0 & \cos(q_2) & 0 \\ 0 & 1 & 0 & 0,221 \\ -\cos(q_2) & 0 & -\sin(q_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_3(q_3) = \begin{bmatrix} \cos(q_3) & 0 & \sin(q_3) & 0 \\ 0 & 0 & 0 & -0,1719 \\ -\sin(q_3) & 0 & \cos(q_3) & 0,612 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 \mathbf{T}_4(q_4) &= \begin{bmatrix} -\sin(q_4) & 0 & \cos(q_4) & 0 \\ 0 & 0 & 0 & 0 \\ -\cos(q_4) & 0 & -\sin(q_4) & 0,5723 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_5(q_5) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(1_1) & 0 & 0,1149 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\
 \mathbf{T}_6(q_6) &= \begin{bmatrix} \cos(q_2) & 0 & \sin(q_2) & 0 \\ 0 & 0 & 0 & 0 \\ -\sin(q_2) & 0 & \cos(q_2) & 0,1157 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_E = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0,0922 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.
 \end{aligned}$$

Translationen sind in Meter angegeben. Zugehörige Denavit-Hartenberg Parameter werden nicht angegeben, da sie die Bezugssysteme der Hüllkörper verändern würden und lediglich eine andere (minimale) Repräsentation der Kinematik darstellen.

Die Modellierung mit Hüllkörpern erfolgt gemäß Tabelle A.1 mit dem Resultat in Abbildung A.2. Es ergeben sich  $|\mathcal{R}| = 6$  und  $N_{RR} = 8$ . Boden und Decke werden über unendlich ausgedehnte Ebenen mit  $\mathbf{p}_0^T = \mathbf{0}$ ,  $\mathbf{p}_1^T = (1 \ 0 \ 0)$  und  $\mathbf{p}_2^T = (0 \ 1 \ 0)$  bzw.  $\mathbf{p}_0^T = (0 \ 0 \ 2,5)$ ,  $\mathbf{p}_1^T = (1 \ 0 \ 0)$  und  $\mathbf{p}_2^T = (0 \ -1 \ 0)$  (jeweils in Meter) modelliert.

Aus dem Handbuch geht eine maximale Zeit bis zum Stillstand des ersten Gelenks von  $T_{B,\max} = 750$  ms hervor [Uni21]. Das zweite und dritte Gelenk liegen bei 240 ms bzw. 280 ms. Für das erste Gelenk entspricht dies in grober Näherung einem Bremsmoment  $M$  von:

$$M = \frac{\omega I}{T_{B,\max}} = \frac{3,2 \cdot 10 \cdot 1,33^2}{0,75} \approx 75,5 \text{ N m}. \quad (\text{A.1})$$

Diese Zahlen basieren auf der Annahme von  $m = 10$  kg Endeffektorlast, einer Winkelgeschwindigkeit von  $\omega = 3,2 \text{ rad s}^{-1}$  und einer effektiven Armlänge von  $d_{\text{eff}} = 1,33$  m. Die MHP beschränkt die maximale Gelenkwinkelgeschwindigkeit allerdings auf  $0,5 \text{ rad s}^{-1}$  und die Experimente verwenden eine Endeffektorlast von ca. 1,5 kg. Wird das Bremsmoment auf diese Werte umgelegt ergibt sich in grober Näherung eine Bremszeit von:

$$T_B = \frac{\omega I}{M} = \frac{0,5 \cdot 1,5 \cdot 1,33^2}{75,5} \approx 18 \text{ ms}. \quad (\text{A.2})$$

Um Signallaufzeiten zu berücksichtigen wird die Bremszeit mit  $T_B = 0,15$  s abgeschätzt. Unter der Annahme von  $Q = R = 1$  und den symmetrischen Beschränkungen  $x^+ = 3,1$  rad sowie  $u^+ = 0,5 \text{ rad s}^{-1}$  ergibt sich nach Gleichung (4.29) eine Abschätzung des Potentialgewichts von  $W_C = 155$ .

Tabelle A.1.: Kinematischer Aufbau und Verteilung der Hüllkörper für den UR10

ID	Name	T	Hüllkörper			Kollisionsvermeidung	
			$\mathbf{p}_0$ in m	$\mathbf{p}_1$ in m	$r$ in m	Selbst	Hindernis
0	<i>pedestal/world</i>	$\mathbf{T}_B$	-	-	-	-	×
1	<i>base_link</i>	$\mathbf{T}_1$	(0 0 -0,8)	(0 0 -0,04)	0,1	-	×
2	<i>shoulder_link</i>	$\mathbf{T}_2$	(0 0 0)	-	0,11	-	✓
3	<i>upper_arm_link</i>	$\mathbf{T}_3$	(0 -0,05 0)	(0 -0,05 0,6)	0,11	-	✓
4	<i>forearm_link</i>	$\mathbf{T}_4$	(0 0 0)	(0 0 0,56)	0,08	0;2	✓
5	<i>wrist_1_link</i>	$\mathbf{T}_5$	(0 0,1149 0)	-	0,07	3	✓
6	<i>wrist_2_link</i>	$\mathbf{T}_6$	(0 -0,04 0,1157)	(0 0,04 0,1157)	0,05	3	✓
7	<i>wrist_3_link</i>	$\mathbf{T}_E$	(0 0,1 0)	-	0,05	0;2;3;4	✓
8	<i>ee_link</i>	-	-	-	-	-	×



Abbildung A.2.: Hüllkörper (grau durchsichtig) am UR10.

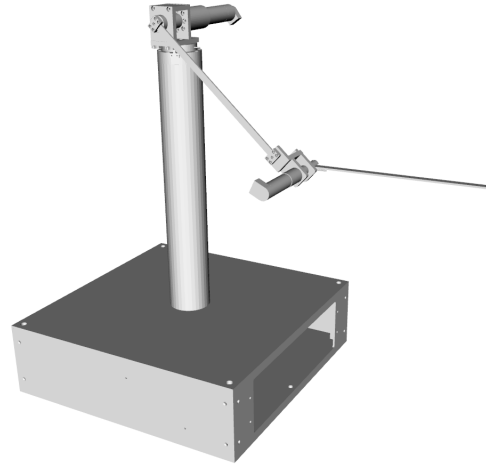
## A.2. TU Dortmund Omni-elastic Robot Adapted (TUDORA)

Der Roboterarm TUDORA besteht aus  $N = 3$  Gelenken,  $N_E = 2$  Schwingungsebenen und ist in Abbildung A.3 sowohl als Foto, als auch als Modell in der Simulation abgebildet. Er ist eine Selbstanfertigung des Lehrstuhls für Regelungssystemtechnik der TU-Dortmund.

Die Simulation erfolgt analog zum UR10 im ROS. Zur detailgetreuen Darstellung des Roboters wurde ein Simulationspaket auf Basis der Konstruktionsdaten erstellt. Zur Kommunikation mit dem realen Roboter kommt Matlab<sup>®</sup> zum Einsatz, das als Gateway zwischen dem ROS-Netzwerk und dem Echtzeitsystem von Simulink<sup>®</sup> fungiert.



(a) TUDORA in der Laborumgebung



(b) TUDORA in der Simulationsumgebung

Abbildung A.3.: TUDORA

Die Transformationsmatrizen der direkten Kinematik lauten:

$$\mathbf{T}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,18 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_1(q_1) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0,625 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{T}_2(q_2) = \begin{bmatrix} -\sin(q_2) & 0 & \cos(q_2) & 0 \\ 0 & 1 & 0 & -0,065 \\ -\cos(q_2) & 0 & -\sin(q_2) & 0,064 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_3(q_3) = \begin{bmatrix} \cos(q_3) & 0 & \sin(q_3) & 0,4385 \\ 0 & 0 & 0 & 0,052 \\ -\sin(q_3) & 0 & \cos(q_3) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{T}_E = \begin{bmatrix} 1 & 0 & 0 & 0,382 \\ 0 & 1 & 0 & 0,0245 \\ 0 & 0 & 1 & 0,002 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Translationen sind in Meter angegeben. Zugehörige Denavit-Hartenberg Parameter werden nicht angegeben, da sie die Bezugssysteme der Hüllkörper verändern würden und lediglich eine andere (minimale) Repräsentation der Kinematik darstellen.

Die Modellierung mit Hüllkörpern erfolgt gemäß Tabelle A.2 mit dem Resultat in Abbildung A.4. Es ergeben sich  $|\mathcal{R}| = 3$  und  $N_{RR} = 3$ . Auf die Modellierung des Motors des letzten Gelenks wurde verzichtet, da TUDORA primär in Kapitel 7 zum Einsatz kommt und dort neben der Vermeidung von Selbstkollisionen keine weiteren Hindernisse vorkommen. Boden und Decke werden über unendlich ausgedehnte Ebenen mit  $\mathbf{p}_0^T = (0 \ 0 \ 0,2)$ ,  $\mathbf{p}_1^T = (1 \ 0 \ 0)$  und  $\mathbf{p}_2^T = (0 \ 1 \ 0)$  bzw.  $\mathbf{p}_0^T = (0 \ 0 \ 1,8)$ ,  $\mathbf{p}_1^T = (1 \ 0 \ 0)$  und  $\mathbf{p}_2^T = (0 \ -1 \ 0)$  (jeweils in Meter) modelliert. Der Standfuß befindet sich aus Sicht der Kollisionsvermeidung damit unterhalb des Bodens.

Der Roboter TUDORA weicht von den symmetrischen Gelenkwinkelgrenzen ab und besitzt  $\mathbf{x}^+ = (2,97 \ 0,79 \ 2,62)$  rad und  $\mathbf{x}^- = (-2,97 \ -4,01 \ -2,62)$  rad. Unter der Annahme von  $Q = R = 1$  und  $x^+ = 3$  rad sowie  $u^+ = 0,5$  rad s<sup>-1</sup> ergäbe sich nach Gleichung (4.29) eine

Abschätzung des Potentialgewichts von  $W_C = 146$ . Der Einfachheit halber wird  $W_C = 155$  vom UR10 übernommen.

Tabelle A.2.: Kinematischer Aufbau und Verteilung der Hüllkörper für TUDORA

ID	Name	T	Hüllkörper			Kollisionsvermeidung	
			$\mathbf{p}_0$ in m	$\mathbf{p}_1$ in m	$r$ in m	Selbst	Hindernis
0	<i>pedestal/world</i>	$\mathbf{T}_B$	-	-	-	-	×
1	<i>base</i>	$\mathbf{T}_1$	(0 0 0)	(0 0 0,6)	0,07	-	×
2	<i>link_1</i>	$\mathbf{T}_2$	(0 -0,05 0,06)	(0 0,22 0,06)	0,05	-	✓
3	<i>link_2</i>	$\mathbf{T}_3$	(0,075 0,02 0)	(0,425 0,02 0)	0,05	1	✓
4	<i>link_3</i>	$\mathbf{T}_E$	(0,015 0,02 0)	(0,385 0,02 0)	0,03	1;2	✓
5	<i>ee_link</i>	-	-	-	-	-	×

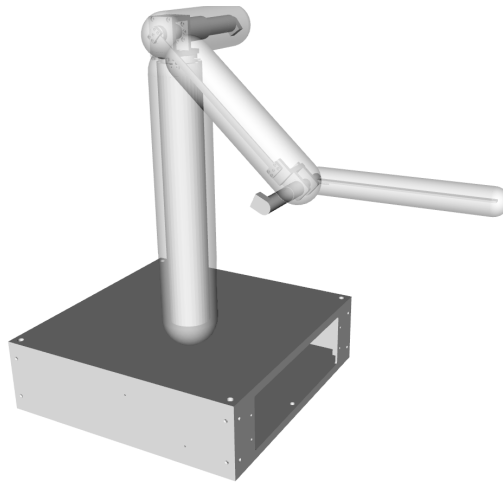


Abbildung A.4.: Hüllkörper (grau durchsichtig) von TUDORA.

### A.3. Roboterarm mit zwei Gelenken

Der Roboterarm mit  $N = 2$  Gelenken ist ein rein fiktiver Roboter zu Anschauungszwecken in Beispielen. Da beide Gelenke ineinander fallen, besitzt er effektiv einen Armkörper und ist schematisch in Abbildung A.5 dargestellt.

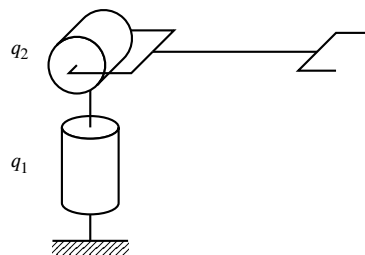


Abbildung A.5.: Roboterarm mit zwei Armkörpern.

Die Transformationsmatrizen der direkten Kinematik lauten:

$$\mathbf{T}_B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_1(q_1) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{T}_2(q_2) = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(q_2) & -\cos(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{T}_E = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Translationen sind in Meter angegeben. Zugehörige Denavit-Hartenberg Parameter werden nicht angegeben, da sie die Bezugssysteme der Hüllkörper verändern würden und lediglich eine andere (minimale) Repräsentation der Kinematik darstellen.

Die Modellierung mit Hüllkörpern erfolgt gemäß Tabelle A.3 Es ergeben sich  $|\mathcal{R}| = 1$  und  $N_{RR} = 0$ .

Tabelle A.3.: Kinematischer Aufbau und Verteilung der Hüllkörper des zweigliedrigen Roboters

ID	Name	$\mathbf{T}$	Hüllkörper			Kollisionsvermeidung	
			$\mathbf{p}_0$ in m	$\mathbf{p}_1$ in m	$r$ in m	Selbst	Hindernis
0	<i>pedestal/world</i>	$\mathbf{T}_B$	-	-	-	-	×
1	<i>base</i>	$\mathbf{T}_1$	-	-	-	-	×
2	<i>link_1</i>	$\mathbf{T}_2$	-	-	-	-	×
3	<i>link_2</i>	$\mathbf{T}_E$	(0 0 0)	(1 0 0)	0	-	✓
4	<i>ee_link</i>	-	-	-	-	-	×

# B

## Begleitinformationen zu den Versuchen

Mit den Begleitinformationen zu den Versuchen liefert dieses Kapitel ausführlichere Informationen, die nicht direkt für das Verständnis im Fließtext notwendig sind, jedoch der Reproduzierbarkeit dienen. Die Angaben sind aufgeteilt nach Kapiteln und dem jeweiligen Abschnittsnamen.

### B.1. Allgemeine Angaben

Einige Details gelten durchgehend und werden an dieser Stelle übergeordnet angegeben. Abweichungen davon sind entsprechend vermerkt.

**Rechner B.1.1:** *Angaben zum Rechner bei Simulationen:*

- *Intel® i7-8700K CPU @4,8 GHz (6 Kerne, 12 Threads)*
- *32 GB DDR4 Arbeitsspeicher*
- *Ubuntu 18.04*

Um die Kommunikation und Lastverteilung bei Simulationen möglichst authentisch zu gestalten, wird ein zweiter Rechner verwendet auf dem die physikalische Simulation der Roboterbewegung (Gazebo, RViz) ausgeführt wird.

**Rechner B.1.2:** *Angaben zum Rechner für Physiksimulationen:*

- *Intel® i7-4790 CPU @4,0 GHz (4 Kerne, 8 Threads)*
- *16 GB DDR3 Arbeitsspeicher*
- *Ubuntu 18.04*

Bei realen Versuchen entfällt die Physiksimulation und es verbleibt die Ausführung der MHP:

**Rechner B.1.3:** *Angaben zum Rechner bei realen Versuchen mit dem UR10:*

- *Intel® i7-6900K CPU @3,7 GHz (8 Kerne, 16 Threads)*
- *32 GB DDR4 Arbeitsspeicher*
- *Ubuntu 18.04*

**Rechner B.1.4:** *Angaben zum Rechner bei realen Versuchen mit TUDORA:*

- *Intel® i5-6500 CPU @3,2 GHz (4 Kerne, 4 Threads)*
- *16 GB DDR4 Arbeitsspeicher*

- *Ubuntu 18.04*

Bei TUDORA kommt zusätzlich zum Rechner für die MHP ein Gateway zwischen ROS und Simulink<sup>®</sup>-*Realtime* zum Einsatz:

**Rechner B.1.5:** *Angaben zum Matlab Gateway bei realen Versuchen mit TUDORA:*

- *Intel<sup>®</sup> i5-7600 CPU @3,5 GHz (4 Kerne, 4 Threads)*
- *16 GB DDR4 Arbeitsspeicher*
- *Windows 10*

Alle in dieser Arbeit erfolgten Implementierungen sind in C++ unter Verwendung des Standards C++17 umgesetzt. Die Kompilierung erfolgt mit dem Kompilationsprogramm GCC (7.5.0) und dem Optimierungslevel *O3*. Ausführbare Anwendungen sind als Knoten im ROS realisiert. Darüber hinaus wurden bei der Planung und Umsetzung die Möglichkeiten zur objektorientierten Programmierung und Polymorphie genutzt, um dem Framework eine Reihe von Schnittstellen zur Erweiterung zu geben. In den Implementierungen verwendete Bibliotheken:

- *Eigen* [Gue+21]
- *EigenMVN* [Ben14]
- *ThreadPool* [Pro14]
- *NatNet SDK* [Opt21b]
- *yaml-cpp* [Bed15]

Anmerkung: Nicht immer stammen die Implementierungen von den Autoren der ursprünglichen Veröffentlichungen. Stellenweise gibt es nur eine Quelltext-Veröffentlichung. Zur Anbindung an das Framework dieser Arbeit waren stellenweise Adaptierungen nötig.

Zur Optimierung der Rechenzeit wird die Intel<sup>®</sup> MKL<sup>1</sup> [Int22] verwendet. Sie beinhaltet hardwareoptimierte Rechenroutinen sowie das Lösungsverfahren PARDISO für lineare Gleichungssysteme. Folgende Bibliotheken und Anwendungen sind mit der MKL verlinkt:

- *Eigen*
- *EigenMVN*
- IPOpt
- MUMPS
- MA27, MA57, MA77 und MA97
- PARDISO

## B.2. Kapitel 4

### B.2.1. Spezialfall bei sich schneidenden Primitiven

Die ausführlichen Details des NLP (4.17) bei der Untersuchung zu sich schneidenden Primitiven in Abschnitt 4.2 umfassen:

- Optimierungsvariable  $s_x \in [-1, 1]$  m

---

<sup>1</sup>*Math Kernel Library*

- Zielfunktion ist die negative Distanz  $-d(S_1, S_2(s_x))$
- Hindernis  $S_1$  mit  $\mathbf{p}_0^{(1)} = (-0,5 \ 0)^T$  m,  $\mathbf{p}_1^{(1)} = (0,5 \ 0)^T$  m und  $r^{(1)} = 0,1$  m
- Hindernis  $S_2(s_x)$  mit  $\mathbf{p}_0^{(2)} = (s_x \ -0,5)^T$  m,  $\mathbf{p}_1^{(2)} = (s_x \ 0,5)^T$  m und  $r^{(2)} = 0,1$  m
- Initialisierung mit  $s_x = 0$
- Lösungsverfahren *fmincon* (IP-Verfahren in Standardkonfiguration, Matlab<sup>®</sup> R2022a)
- Versuch auf Rechner B.1.1 durchgeführt

### B.2.2. Vergleich der Berechnungszeit verschiedener Implementierungen

Die ausführlichen Details des Laufzeitvergleichs zwischen der eigenen Implementierung der Distanzfunktion  $d(\cdot)$  und denen der GTE und FCL in Abschnitt 4.2 umfassen:

- 100 000 Wiederholungen
- Referenzkugel mit  $\mathbf{p}_0 = (0 \ 0 \ 0)^T$  m und  $r = 0,1$  m
- Referenzzylinder mit  $\mathbf{p}_0 = (1 \ 1 \ 1)^T$  m,  $\mathbf{p}_1 = (2 \ 2 \ 2)^T$  m und  $r = 0,1$  m
- Referenzfläche mit  $\mathbf{p}_0 = (1 \ 1 \ 1)^T$  m,  $\mathbf{p}_1 = (2 \ 2 \ 2)^T$  m,  $\mathbf{p}_2 = (0 \ 2 \ 2)^T$  m und  $r = 0,1$  m
- Referenzebene mit  $\mathbf{p}_0 = (0 \ 0 \ 0)^T$  m und Normalenvektor  $\mathbf{n} = \left\| (1 \ 1 \ 1)^T \right\|_2$
- Die Elemente der Positionen der randomisierten Hindernisse werden zufällig (gleichverteilt) aus dem Intervall  $[-1, 1]$  m gewählt
- GTE [Ebe20]
- FCL [Pan+12]
- Versuche auf Rechner B.1.1 durchgeführt

### B.2.3. Einfluss der Gewichte

Die ausführlichen Details zum Versuch aus Abschnitt 4.3 über den Einfluss der Gewichte der Annäherungskosten umfassen:

- Roboter UR10
- Anfangszustand  $\mathbf{x}_0 = (-\frac{\pi}{2} \ 0 \ 0 \ 0 \ 0 \ 0)^T$  rad
- Zielzustand  $\mathbf{x}_f = (\frac{\pi}{2} \ 0 \ 0 \ 0 \ 0 \ 0)^T$  rad
- Kugel 1 mit  $\mathbf{p}_0 = (0,9 \ -0,8 \ 0,75)^T$  m und  $r = 0,1$  m
- Kugel 2 mit  $\mathbf{p}_0 = (0,9 \ 0,8 \ 0,75)^T$  m und  $r = 0,1$  m
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0$  und  $W_C \in \{0,1; 1; 10\}$
- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>

- Lösungsverfahren IPOpt [Coi23] (MatSolver: MA27, *Adaptive Mu*, *Auto Scaling*)
- Initialisierung über  $\mathbf{z}^A$
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Zusätzlich zeigt Abbildung B.1 den Aufbau mit den beiden Hindernissen sowie den Roboter im Start- und Zielzustand.

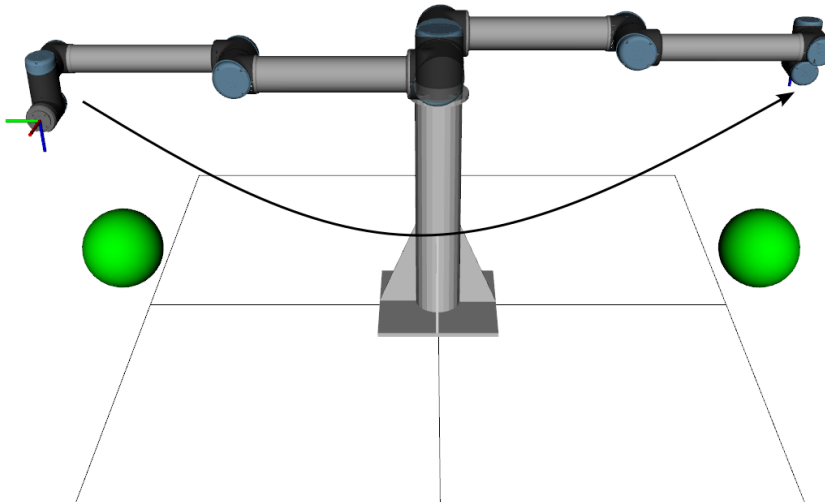


Abbildung B.1.: Versuchsaufbau zur Bewertung des Einflusses der Gewichte.

#### B.2.4. Berechnungszeiten von Kostenfunktion und Nebenbedingung

Weitere Angaben zur Erhebung der Berechnungszeiten von Kostenfunktion  $\hat{J}(\cdot)$  und Nebenbedingung  $\mathbf{h}(\cdot)$  in Abschnitt 4.3 sind:

- Roboter UR10
- 10 000 Wiederholungen
- Die Elemente der x-Positionen der randomisierten Hindernisse werden zufällig (gleichverteilt) aus dem Intervall  $[-1, 1]$  m gewählt
- Die übrigen Komponenten betragen jeweils 1 m
- Radius der Kugeln  $r = 0,1$  m
- Die Zustände (und Stellgrößen) werden ebenfalls zufällig aus dem Intervall  $[-1, 1]$  rad ( $\text{rad s}^{-1}$ ) gewählt (der Versuch erfordert keine sinnvollen Trajektorien)
- Durchführung auf dem Rechner B.1.1

Die linearen Modelle nach der Methode der kleinsten Quadrate für das erste Szenario sind:

$$\begin{aligned} T_f(K) &= 4,77K - 0,38 \quad \text{RMSE: } 0,41 \mu\text{s}, \\ T_h(K) &= 4,60K - 0,24 \quad \text{RMSE: } 0,63 \mu\text{s}, \end{aligned} \tag{B.1}$$

und für das zweite Szenario:

$$\begin{aligned} T_f(H) &= 11,41H + 29,34 \quad \text{RMSE: } 0,49 \mu\text{s}, \\ T_h(H) &= 11,73H + 20,75 \quad \text{RMSE: } 0,32 \mu\text{s}, \end{aligned} \tag{B.2}$$

mit der Anzahl  $H$  von Hindernissen.

Der Aufwand der Bestandteile von Kostenfunktion und Nebenbedingungen ist in Tabelle B.1 in Form der über 10 000 Aufrufe gemittelten Berechnungszeiten angegeben. Aus ihnen lässt sich ein Modell für die untere Schranke der mittleren Laufzeit in Abhängigkeit der Häufigkeiten  $N_i$  unterschiedlicher Paarungen von Hüllkörpern und Horizontlänge  $K$  aufstellen:

$$T_{\hat{J}} = K \left( T_{c_X} + T_{c_U} + T_{\mathbf{T}} + \sum_{i \in \mathbb{L}} N_i T_i + T_{c_H} \sum_{i \in \mathbb{L}} N_i \right), \quad (\text{B.3})$$

$$T_{\mathbf{h}} = K \left( T_{\mathbf{T}} + \sum_{i \in \mathbb{L}} N_i T_i \right), \quad (\text{B.4})$$

mit der Bezeichnermenge  $\mathbb{L} := \{\text{PP, PL, PF, PE, LL, LF, LE}\}$  für die verschiedenen Hüllkörperkombinationen. Die Laufzeiten  $T_i$  entsprechen denen der gewählten Distanzfunktionen aus Tabelle 4.1. Für das gewählte Szenario und den Roboter UR10 gilt  $N_{\text{PP}} = 1 + 3N_H$ ,  $N_{\text{PL}} = 5 + 3N_H$  und  $N_{\text{LL}} = 2$ .

Tabelle B.1.: Über 10 000 Wiederholungen gemittelte Laufzeiten der isolierten Komponenten  $c_X$ ,  $c_U$  und  $c_H$  sowie der Vorwärtskinematik  $\mathbf{T}$ .

$T_{c_X}$	$T_{c_U}$	$T_{c_H}$	$T_{\mathbf{T}}$
72 ns	72 ns	13 ns	445 ns

## B.3. Kapitel 5

### B.3.1. Spezialisierung auf die Trajektorienoptimierung

Die ausführlichen Details zum Versuch aus Abschnitt 5.1 über Größenordnungen für unterschiedliche Transkriptionen umfassen:

- Roboter UR10
- 10 Wiederholungen pro Versuch
- Anfangszustand  $\mathbf{x}_0 = \left(-\frac{\pi}{2} \ 0 \ 0 \ 0 \ 0 \ 0\right)^T \text{ rad}$
- Zielzustand  $\mathbf{x}_f = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T \text{ rad}$
- Kugel 1 mit  $\mathbf{p}_0 = (0,7 \ -1,0 \ 0,6)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 2 mit  $\mathbf{p}_0 = (0,7 \ -1,0 \ 1,3)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 3 mit  $\mathbf{p}_0 = (1,1 \ -0,4 \ 0,6)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 4 mit  $\mathbf{p}_0 = (1,1 \ -0,4 \ 1,3)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- $d_R = 0,05 \text{ m}$  und  $d_R^- = 0,02 \text{ m}$
- $d_C = 0,1 \text{ m}$  und  $d_C^- = 0,05 \text{ m}$
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$

- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>
- IPOpt [Coi23] (MatSolver: MA27, *Adaptive Mu*, *Auto Scaling*)
- SLSQP aus der Softwaresammlung NLOpt<sup>2</sup> [Joh21]
- Initialisierung über  $\mathbf{z}^A$
- Der Roboter ignoriert die optimierten Stellgrößen
- Kostenschwellwerte in Abhängigkeit von  $I$ : 1:603; 2:590; 3:573,5; 6:524,5; 30:132,5
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Zusätzlich zeigt Abbildung B.2 den Aufbau mit allen Hindernissen sowie den Roboter im Start- und Zielzustand.

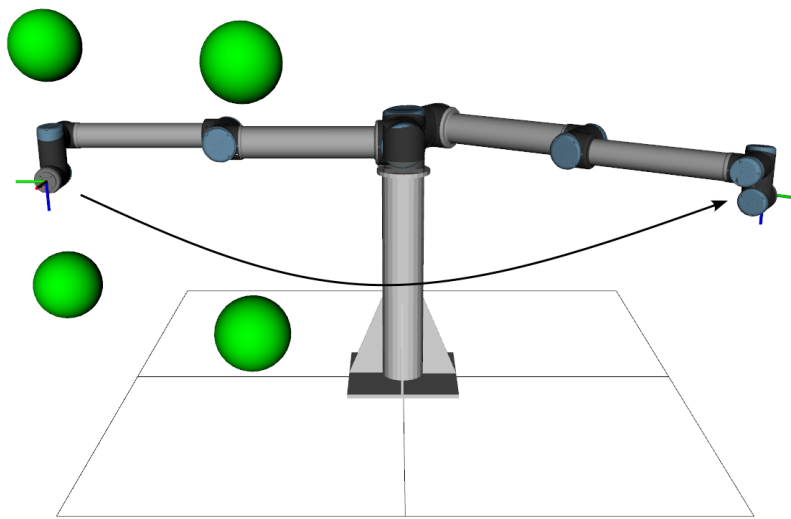


Abbildung B.2.: Aufbau zum Versuch über Größenordnungen für unterschiedliche Transkriptionen.

### B.3.2. Repräsentation von Abhängigkeiten mittels Hypergraphen

Weiterführende Auswertung aus Abschnitt 5.1 mit demselben Aufbau wie B.3.1, nur diesmal mit anderen linearen Solvern bei IPOpt:

- MA27, MA57, MA77 und MA97 [Har21b]
- MUMPS [Coi22]
- PARDISO aus der MKL [Int22]

### B.3.3. Vergleich mit anderen Frameworks zur Trajektorienoptimierung

Die letzte Auswertung aus dem Abschnitt 5.1 zum Vergleich von IPTOP mit anderen Ansätzen umfasst folgenden Aufbau:

- Roboter UR10

---

<sup>2</sup>Sammlung von Lösungsverfahren für *Nonlinear Optimization*

- 10 Wiederholungen pro Versuch
- Anfangszustand  $\mathbf{x}_0 = (-1,5 \ 0 \ 0 \ 0 \ 0 \ 0)^T$  rad
- Zielzustand  $\mathbf{x}_f = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$  rad
- Kugel 1 mit  $\mathbf{p}_0 = (0,7 \ -1,0 \ 0,6)^T$  m und  $r = 0,1$  m
- Kugel 2 mit  $\mathbf{p}_0 = (0,7 \ -1,0 \ 1,3)^T$  m und  $r = 0,1$  m
- Kugel 3 mit  $\mathbf{p}_0 = (1,1 \ -0,4 \ 0,6)^T$  m und  $r = 0,1$  m
- Kugel 4 mit  $\mathbf{p}_0 = (1,1 \ -0,4 \ 1,3)^T$  m und  $r = 0,1$  m
- Kugel 5 mit  $\mathbf{p}_0 = (0,95 \ -0,75 \ 0,6)^T$  m und  $r = 0,1$  m
- Kugel 6 mit  $\mathbf{p}_0 = (0,95 \ -0,75 \ 1,3)^T$  m und  $r = 0,1$  m
- Kugel 7 mit  $\mathbf{p}_0 = (1,15 \ -0,05 \ 0,6)^T$  m und  $r = 0,1$  m
- Kugel 8 mit  $\mathbf{p}_0 = (1,15 \ -0,05 \ 1,3)^T$  m und  $r = 0,1$  m
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$
- $T_T = T_O = 0,1$  s und  $K \in \{11; 21; 31; 41; 51\}$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = \infty$  (Da HQP bei  $N = 6$  keine Beschleunigung abbilden kann)
- IPOpt [Coi23] (MatSolver: MA57, *Adaptive Mu*, *Auto Scaling*)
- HQP (MatSolver: SpBKP, QPSolver: Franke, *DScale*) [Fra98]
- ALTRO [Opt21a]
- Initialisierung über  $\mathbf{z}^A$
- Der Roboter ignoriert die optimierten Stellgrößen
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Hindernisverteilungen:

- Bei 2 Hindernissen: Nr. 1 und 3
- Bei 4 Hindernissen: Nr. 1, 3, 5 und 7
- Bei 6 Hindernissen: Nr. 1 bis 6
- Bei 8 Hindernissen: Nr. 1 bis 8

Zusätzlich zeigt Abbildung B.3 den Aufbau mit allen Hindernissen sowie den Roboter im Start- und Zielzustand.

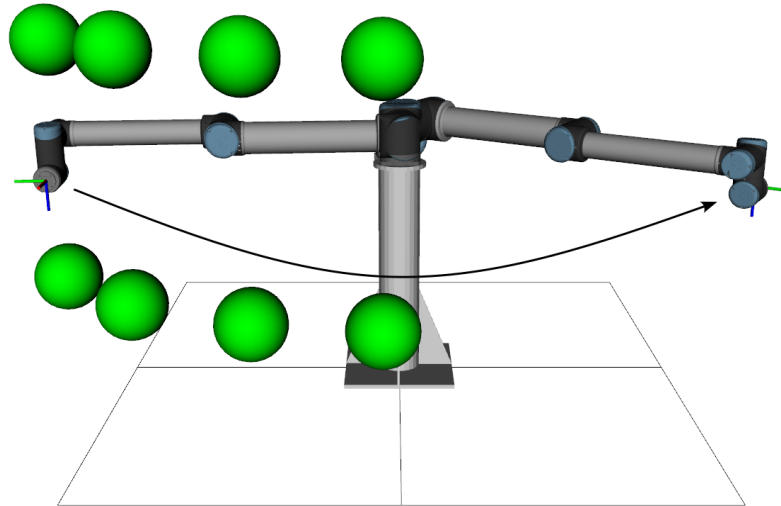


Abbildung B.3.: Versuchsaufbau zum Vergleich von IPTOP mit anderen Ansätzen.

## B.4. Kapitel 6

### B.4.1. Auswertung der Schnappschussmethode

Die Auswertung der Schnappschussmethode aus Abschnitt 6.1 ist wie folgt aufgebaut:

- Roboter UR10
- Anfangszustand (A)  $\mathbf{x}_0 = (-1,8 \quad -1,3 \quad 1,6 \quad -2 \quad 0 \quad 0)^T$  rad
- Zielzustand (A)  $\mathbf{x}_f = (1 \quad -1,3 \quad 1,6 \quad -2 \quad 0 \quad 0)^T$  rad
- Anfangszustand (B)  $\mathbf{x}_0 = (-1,3 \quad -1,3 \quad 1,3 \quad 0 \quad 0 \quad 0)^T$  rad
- Zielzustand (B)  $\mathbf{x}_f = (1 \quad -1,3 \quad 1,3 \quad 0 \quad 0 \quad 0)^T$  rad
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_D \in \{0,24; 0,34; 0,44; 0,54\}$  m und  $d_D^- = 0,14$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_D = 155$
- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>
- IPOpt [Coi23] (MatSolver: MA57, Adaptive Mu, Auto Scaling)
- Erste Initialisierung über  $\mathbf{z}^A$  dann Warmstart
- Testperson (Autor der Arbeit) wird über einen mit passiven Markern bestückten Anzug (Jacke und Mütze) erfasst
- Bewegungserfassung über das infrarotbasierte Kamerasystem OptiTrack [Opt21b] auf dem Rechner B.1.3
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Zusätzlich zeigt Abbildung B.4 die Person in ihrer Anfangs- und Endpose sowie den Roboter im Start- und Zielzustand für beide Szenarien.

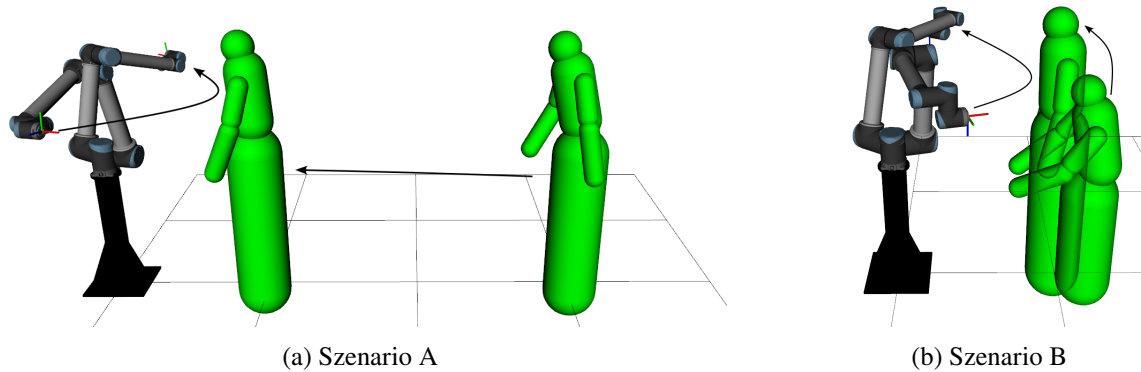


Abbildung B.4.: Start- und Zielzustand von Roboter und Mensch.

### B.4.2. Prädiktive Kollisionsvermeidung mittels Zustandsextrapolation

Die Auswertung aus Abschnitt 6.1 wird um die Zustandsschätzung und -extrapolation erweitert. Der Aufbau ist bis auf die folgenden Angaben identisch zu Abschnitt B.4.1.

- Index  $K_D = 4$ , bis zu welchem Nebenbedingungen zur Kollisionsvermeidung bei dynamischen Hindernissen berücksichtigt werden sollen
- Fensterbreite  $F_S = 4$  des Schätzers
- $d_D \in \{0,15; 0,16; 0,24; 0,26; 0,27; 0,34; 0,4; 0,44\}$  m
- Prädiktionlänge  $P \in \{0; 4; 10; 30\}$

### B.4.3. Annähern und Folgen dynamischer Zielkonfigurationen

Die Auswertung aus Abschnitt 6.2 über das Folgen dynamischer Zielkonfigurationen ist wie folgt aufgebaut:

- Roboter UR10
- Zielpose  $\mathbf{s}_{EE,f}(t) = \left(1 \quad \frac{1}{2} \sin(0,2\pi t) \quad 1\right)^T$  m und  $\mathbf{\Theta}_{EE,f}(t) = \text{diag}(1 \quad -1 \quad -1)$
- Anfangszustand  $\mathbf{x}_0 = (-1,5 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0)^T$  rad
- Kugel mit  $\mathbf{p}_0 = (0,35 \quad 0,2 \quad 1,4)^T$  m und  $r = 0,1$  m
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$
- $T_T = 0,1$  s,  $T_O \in \{0,5; 0,1; 0,05\}$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = \infty$  (da der IKA keine Beschleunigungen unterstützt)
- IPOpt [Coi23] (MatSolver: MA57, Adaptive Mu, Auto Scaling)
- Erste Initialisierung über  $\mathbf{z}^A$  dann Warmstart
- IKA Proportionalverstärkung  $\mathbf{K} = \text{diag}(3 \quad 3 \quad 3 \quad 3 \quad 3 \quad 3)$
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

## B.5. Kapitel 7

### B.5.1. Parametrierung der Aktivierungsfunktion

Die Auswertung aus Abschnitt 7.2 über die Parametrierung der Aktivierungsfunktion ist wie folgt aufgebaut:

- Roboter TUDORA
- Start- und Zielzustände siehe Tabelle B.2
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $Q = 1$ ,  $R = 1$  und  $W_R = 155$
- Kein Elastizitätsbewusstsein ( $W_S = 0$ )
- $T_T = T_O = 0,1$  s und  $K = 71$
- $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 1$  rad s<sup>-2</sup>
- Der Roboter TUDORA weicht von den symmetrischen Gelenkwinkelgrenzen ab:  $\mathbf{x}^+ = (2,97 \ 0,79 \ 2,62)$  rad und  $\mathbf{x}^- = (-2,97 \ -4,01 \ -2,62)$  rad
- IPOpt [Coi23] (MatSolver: MA57, Adaptive Mu, Auto Scaling)
- Erste Initialisierung über  $\mathbf{z}^A$  dann Warmstart
- Ausführung der MHP auf Rechner B.1.4 mit dem Matlab<sup>®</sup>-Gateway auf Rechner B.1.5

Beim Versuch zur Auslegung von  $\bar{\alpha}$  kommt das Elastizitätsbewusstsein und damit die folgenden Parameter hinzu:

- $W_S = 1,5$
- $\kappa = 5$
- $\bar{\alpha} \in \{10; 20\}$

Tabelle B.2.: Start- und Zielzustände für den Versuch. Das Minuszeichen repräsentiert gegenüber dem Startzustand konstante Komponenten.

Kombination	$\mathbf{x}_0$ in °	$\mathbf{x}_f$ in °
A	$(-45 \ -90 \ 90)^T$	$(135 \ - \ -)^T$
B	$(-46,98 \ -78,5 \ 78,5)^T$	$(132,93 \ - \ -)^T$
C	$(-49,27 \ -66,5 \ 66,5)^T$	$(131,21 \ - \ -)^T$
D	$(-49,85 \ -53,3 \ 53,3)^T$	$(130,06 \ - \ -)^T$
E	$(-50,99 \ -36,7 \ 36,7)^T$	$(128,92 \ - \ -)^T$
F	$(-53,86 \ 0 \ 0)^T$	$(126,05 \ - \ -)^T$

### B.5.2. Bewertung der dynamischen Priorisierung

Die Auswertung aus Abschnitt 7.2 zur dynamischen Priorisierung ergänzt die vorherigen Parameter um  $\bar{\alpha} = 20$ . Zusätzlich sind in Tabelle B.3 die initialen Dehnungsmesswerte des eingespannten Roboters angegeben.

Tabelle B.3.: Initiale Dehnungsmesswerte des eingespannten Roboters für die Start- und Zielzustände A, B und C.

	A	B	C
$\delta_1/\mu\text{m m}^{-1}$	325	160	160
$\delta_2/\mu\text{m m}^{-1}$	78	90	88

## B.6. Kapitel 8

### B.6.1. Lineare Interpolation

Angaben zur Auswertung der Gradienten nach Initialisierung per linearer Interpolation (mit Zwischenzustand) in Abschnitt 8.1:

- Roboterarm mit zwei Gelenken
- Anfangszustände  $\mathbf{x}_0 = (0 \ 1)^T \text{ rad}, (0 \ 0,5)^T \text{ rad}, (0 \ -0,5)^T \text{ rad}, (0 \ -1)^T \text{ rad}$
- Zielzustände  $\mathbf{x}_f = (\pi \ 0,5)^T \text{ rad}, (\pi \ 0)^T \text{ rad}, (\pi \ 0)^T \text{ rad}, (\pi \ -0,5)^T \text{ rad}$
- Zylinder mit  $\mathbf{p}_0 = (0 \ 0,5 \ 0,5)^T \text{ m}, \mathbf{p}_1 = (0 \ 0,5 \ 1,5)^T \text{ m}$  und  $r = 0,3 \text{ m}$
- $K = 10$
- $d_C = 0,1 \text{ m}$  und  $d_C^- = 0$
- $Q = 1$  und  $W_C = 10$
- Initialisierung über  $\mathbf{z}^{\text{LI}}$  bzw.  $\mathbf{z}^{\text{LI+}}$
- Bei der Initialisierung mit Zwischenzustand ist  $\mathbf{x}_v = (\frac{\pi}{2} \ 1,3)^T \text{ rad}$
- Durchführung auf dem Rechner B.1.1

### B.6.2. Warmstart

Angaben zur Auswertung der prozentualen Verschlechterung mit und ohne Warmstart in Abschnitt 8.1:

- Roboter UR10
- Anfangszustand  $\mathbf{x}_0 = (-1,5 \ 0 \ 0 \ 0 \ 0 \ 0)^T \text{ rad}$
- Zielzustand  $\mathbf{x}_f = (1 \ 0 \ 0 \ 0 \ 0 \ 0)^T \text{ rad}$
- Kugel 1 mit  $\mathbf{p}_0 = (0,7 \ -1,0 \ 0,6)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 2 mit  $\mathbf{p}_0 = (1,1 \ -0,4 \ 0,6)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 3 mit  $\mathbf{p}_0 = (0,95 \ -0,75 \ 0,6)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 4 mit  $\mathbf{p}_0 = (1,15 \ -0,05 \ 0,6)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 5 mit  $\mathbf{p}_0 = (0,7 \ -1,0 \ 1,3)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- Kugel 6 mit  $\mathbf{p}_0 = (0,95 \ -0,75 \ 1,3)^T \text{ m}$  und  $r = 0,1 \text{ m}$
- $d_R = 0,05 \text{ m}$  und  $d_R^- = 0,02 \text{ m}$

- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$
- $T_T = T_O = 0,1$  s
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>
- IPOpt [Coi23] (MatSolver: MA57, Adaptive Mu, Auto Scaling)
- Initialisierung über  $\mathbf{z}^A$  mit anschließendem optionalem Warmstart
- Der Roboter ignoriert die optimierten Stellgrößen
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Zusätzlich zeigt Abbildung B.5 den Aufbau mit allen Hindernissen sowie den Roboter im Start- und Zielzustand.

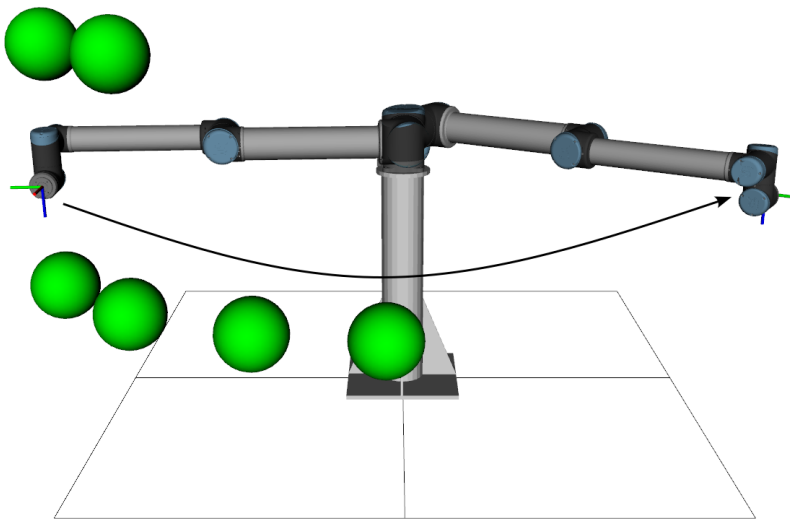


Abbildung B.5.: Aufbau zur Auswertung der prozentualen Verschlechterung mit und ohne Warmstart.

Die Auswertung zur Problematik bei Warmstarts in Abschnitt 8.1 basiert auf folgendem Aufbau:

- Roboterarm mit zwei Gelenken
- Anfangszustand  $\mathbf{x}_0 = (0,5 \quad -0,5)^T$  rad
- Zielzustand  $\mathbf{x}_f(t) = (2,5 \quad (2 + \pi)t)^T$  rad
- Zylinder mit  $\mathbf{p}_0 = (0 \quad 0,5 \quad 0,9)^T$  m,  $\mathbf{p}_1 = (0 \quad 0,5 \quad 1,1)^T$  m und  $r = 0$
- $K = 25$
- $d_C = 0,1$  m und  $d_C^- = 0$
- $Q = 1$ ,  $R = 0,1$  und  $W_C = 1$
- $T_T = 0,1$  s
- $u^+ = 1$  rad s<sup>-1</sup>
- Der Roboter weicht von den symmetrischen Gelenkwinkelgrenzen ab:  $\mathbf{x}^+ = (0 \quad -\frac{\pi}{2})^T$  rad und  $\mathbf{x}^- = (\pi \quad \frac{\pi}{2})^T$  rad
- Im Problemfall: Erste Initialisierung über  $\mathbf{z}^A$  und dann Warmstart

- Im erfolgreichen Fall: Initialisierung über  $\mathbf{z}^A$
- Lösungsverfahren *fmincon* (IP-Verfahren in Standardkonfiguration, Matlab<sup>®</sup> R2022a)
- Durchführung auf dem Rechner B.1.1

### B.6.3. Auswertung für ein dynamisches Hindernis

Die Versuche 1 und 2 zur Bewertung der erweiterten Initialisierung aus Abschnitt 8.2 besitzen folgenden Aufbau:

- Roboter UR10
- Anfangszustand (A)  $\mathbf{x}_0 = (-1 \ 0 \ 0 \ 0 \ 0 \ 0)^T$  rad
- Zielzustand (A)  $\mathbf{x}_f = (0,2 \ 0 \ 0 \ 0 \ 0 \ 0)^T$  rad
- Anfangszustand (B)  $\mathbf{x}_0 = (0 \ 0,7 \ -0,7 \ 0 \ 0 \ 0)^T$  rad
- Zielzustand (B)  $\mathbf{x}_f = (0 \ -0,5 \ 0,5 \ 0 \ 0 \ 0)^T$  rad
- Zylinder (A) mit  $\mathbf{p}_0 = (1,6 \ 0 \ 0,9)^T$  m,  $\mathbf{p}_1 = (1,25 \ 0 \ 0,55)^T$  m und  $r = 0,1$  m
- Zylinder (A) dreht sich mit  $45^\circ \text{s}^{-1}$  relativ um die y-Achse des Arbeitsraums
- Zylinder (B) mit  $\mathbf{p}_0 = (1,5 \ 0,25 \ 0,9)^T$  m,  $\mathbf{p}_1 = (0,39 \ 0,74 \ 0,9)^T$  m und  $r = 0,1$  m
- Zylinder (B) dreht sich mit  $76,4^\circ \text{s}^{-1}$  relativ um die z-Achse des Arbeitsraums
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 2$ ,  $R = 1$  und  $W_R = W_C = 155$
- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,3$  rad s und  $a^+ = \infty$
- IPOpt [Coi23] (MatSolver: MA57, *Adaptive Mu*, *Auto Scaling*)
- Der Roboter ignoriert in Versuch 1 die optimierten Stellgrößen
- Die parallele Instanz kommt erst in Versuch 2 zum Einsatz und besitzt ein festes Iterationslimit von 50 bei unbeschränkter Optimierungszeit
- Initialisierungen sind  $\mathbf{z}^{\text{WS}}$ ,  $\mathbf{z}^{\text{LI}}$  und  $\mathbf{z}^{\text{M}}$
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Zusätzlich zeigt Abbildung B.6 den Roboter und das Hindernis im Start- und Zielzustand für beide Szenarien.

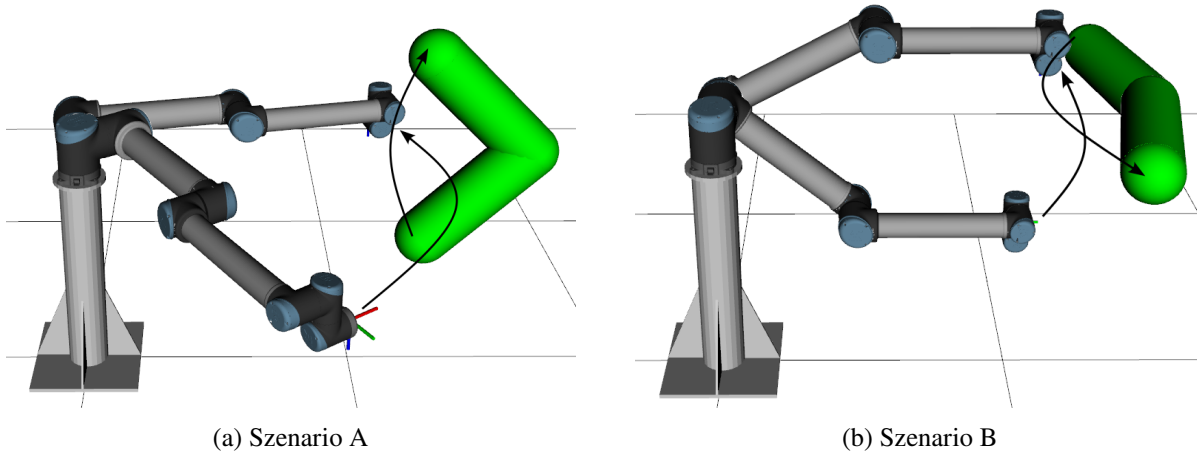


Abbildung B.6.: Versuchsaufbau zur Auswertung der erweiterten Initialisierung bei dynamischen Hindernissen

Der dritte Versuch aus Abschnitt 8.2 ergänzt den Aufbau der vorherigen Versuche wie folgt:

- STOMP von [ROS20] und um Gelenkwinkelbeschränkungen sowie der Differentiationsmatrix aus der ursprünglichen Veröffentlichung von Kalakrishnan u. a. [Kal+11] ergänzt
- STOMP nutzt  $K = 200$ ,  $T_T = 0,1$  s, 300 Iterationen, 400 *Rollouts*,  $R = 0,1$  und Kostensensitivität 1
- GTO als einmalige MHP Instanz mit 200 Iterationen von IPOpt ohne Zeitlimit und  $K = 200$  sowie  $T_T = 0,05$  s

## B.7. Kapitel 9

### B.7.1. Strukturelle lokale Minima bei beschränkten Horizonten

Das einführende Beispiel aus Abschnitt 9.1 ist wie folgt aufgebaut:

- Roboterarm mit zwei Gelenken
- Anfangszustand  $\mathbf{x}_0 = (0,5 \ 0)^T$  rad
- Zielzustand  $\mathbf{x}_f(t) = (2,6 \ 0)^T$  rad
- Zylinder mit  $\mathbf{p}_0 = (0 \ 0,5 \ 0,5)^T$  m,  $\mathbf{p}_1 = (0 \ 0,5 \ 1,5)^T$  m und  $r = 0,1$  m
- $d_C = 0,1$  m und  $d_C^- = 0$
- $Q = 1$ ,  $R = 1$  und  $W_C = 1$
- $T_T = 0,1$  s
- $u^+ = 0,6$  rad s<sup>-1</sup>
- Der Roboter weicht von den symmetrischen Gelenkwinkelgrenzen ab:  $\mathbf{x}^+ = (0 \ -\frac{\pi}{2})$  rad und  $\mathbf{x}^- = (\pi \ \frac{\pi}{2})$  rad
- GTO verwendet  $K = 70$  und eine Initialisierung über  $\mathbf{z}^{\text{LI}^+}$  mit dem Zwischenzustand  $\mathbf{x}_V = (\frac{\pi}{2} \ \frac{\pi}{2})^T$  rad

- Lösungsverfahren *fmincon* (IP-Verfahren in Standardkonfiguration, Matlab® R2022a)
- Durchführung auf dem Rechner B.1.1

### B.7.2. Auswahl eines Verfahrens zur Ermittlung des globalen Pfads

Die Auswertung zur Auswahl eines Verfahrens zur Ermittlung von  $\mathcal{P}$  in Abschnitt 9.2 ist wie folgt aufgebaut:

- Roboter UR10
- Anfangszustand (A)  $\mathbf{x}_0 = \left(-\frac{\pi}{2} \ 0 \ 0 \ 0 \ 0 \ 0\right)^T \text{ rad}$
- Zielzustand (A)  $\mathbf{x}_f = \left(\frac{\pi}{2} \ 0 \ 0 \ 0 \ 0 \ 0\right)^T \text{ rad}$
- Anfangszustand (B)  $\mathbf{x}_0 = (0,62 \ -0,62 \ 0,62 \ 0 \ 2,23 \ 0)^T \text{ rad}$
- Zielzustand (B)  $\mathbf{x}_f = (-0,56 \ 1,05 \ -0,93 \ -0,19 \ 1,05 \ 0)^T \text{ rad}$
- 100 Wiederholungen
- Planungsverfahren stammen aus der OMPL<sup>3</sup> [Suc+12], die an das MHP-Framework angebunden wurde
- Die Kollisionsdetektion ist über einen binären Vergleich der Nebenbedingungen realisiert und nutzt daher  $d_R^- = 0,02 \text{ m}$  bzw.  $d_C^- = 0,05 \text{ m}$
- Die übrigen Parameter liegen auf den Standardwerten der Bibliothek
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Die Hindernisanordnung in Szenario A ist wie folgt:

- Quader mit  $\mathbf{p}_0 = (0,7 \ 0,15 \ 0)^T \text{ m}$ ,  $\mathbf{p}_1 = (1,2 \ 0,15 \ 0)^T \text{ m}$ ,  $\mathbf{p}_2 = (0,7 \ 0,15 \ 1,5)^T \text{ m}$  und  $r = 0,05 \text{ m}$
- Ebene mit  $\mathbf{p}_0 = (0 \ 0 \ 1,5)^T \text{ m}$ ,  $\mathbf{p}_1 = (1 \ 0 \ 1,5)^T \text{ m}$ ,  $\mathbf{p}_2 = (0 \ -1 \ 1,5)^T \text{ m}$

Die Schränke in Szenario B sind vollständig über insgesamt 11 Quader ( $\mathcal{S}_p$ ) modelliert mit den folgenden Eigenschaften:

- Insgesamt 4 Wände je  $0,38 \text{ m} \times 1,98 \text{ m}$  mit  $r = 0,01 \text{ m}$
- Insgesamt 7 Böden (3 links, 4 rechts) je  $0,38 \text{ m} \times 0,78 \text{ m}$  mit  $r = 0,01 \text{ m}$
- Koordinate der zum Roboter gerichteten unteren Ecke der linken Wand des linken Schranks:  
 $\mathbf{p}_0 = (0,76 \ 1,11 \ 0,01)^T \text{ m}$
- Koordinate der zum Roboter gerichteten unteren Ecke der linken Wand des rechten Schranks:  
 $\mathbf{p}_0 = (0,76 \ 0,07 \ 0,01)^T \text{ m}$
- Alle Elemente sind jeweils ohne Überlappung aneinandergereiht

Zusätzlich zeigt Abbildung B.7 den Roboter und das Hindernis im Start- und Zielzustand für beide Szenarien.

<sup>3</sup>Open Motion Planning Library

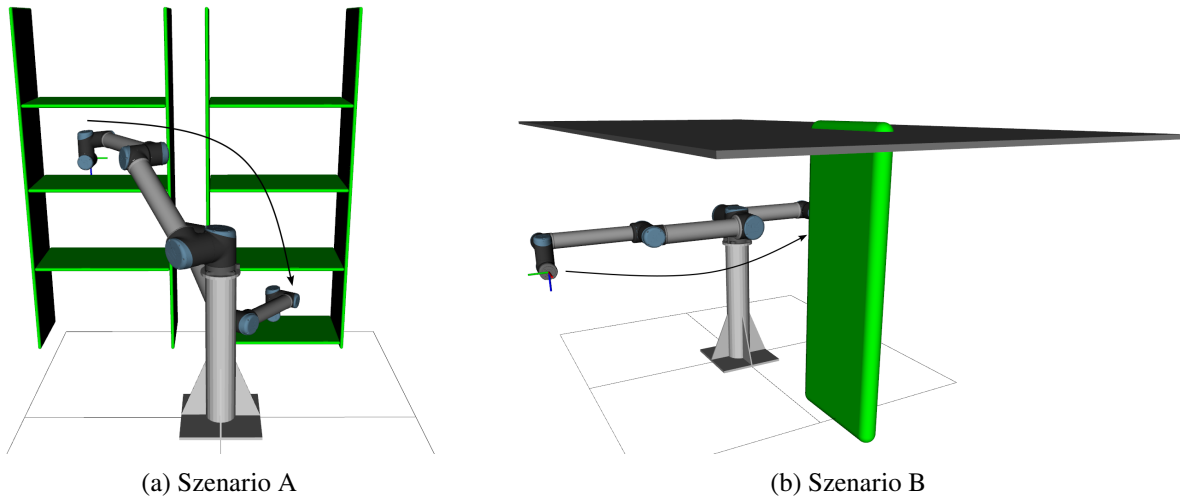


Abbildung B.7.: Aufbau zur Auswahl eines globalen Pfadplaners.

### B.7.3. Parametrisierung von $\beta$ und $K_\epsilon^-$

Die Auswertung zur Parametrisierung von  $\beta$  aus Abschnitt 9.2 ist wie folgt aufgebaut:

- Roboter UR10
- Anfangszustand, Zielzustand und Hindernisse je nach Szenario gemäß Abschnitt B.7.2
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$
- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>
- $K_\epsilon^- = 10$ ,  $r_{MHP} = 0,1$  rad,  $\Delta\rho_{BS} = 0,01$  rad und  $\beta \in \{0,001; 0,014; 0,03; 0,05; 0,07; 0,09\}$
- Das globale Verfahren (RRT) nutzt die Nachbearbeitung und besitzt ein Zeitbudget von 0,2 s, das bis zu 4 mal verlängert wird bevor die Planung abgebrochen wird
- IPOpt [Coi23] (MatSolver: MA57, Adaptive Mu, Auto Scaling)
- Erste Initialisierung über  $\mathbf{z}^A$  und dann Warmstart
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Die Auswertung zur Parametrisierung von  $K_\epsilon^-$  erweitert Szenario A um ein dynamisches Hindernis:

- Zylinder mit  $\mathbf{p}_0(t) = (0,45 \quad -0,9 \quad 0,2t)^T$  m,  $\mathbf{p}_1 = (0,45 \quad -0,4 \quad 0,2t)^T$  m und  $r = 0,03$  m
- Das Hindernis bewegt sich für 10 s
- $\beta = 0,01$  und  $K_\epsilon^- \in \{0; 10; 15; 20; 30\}$
- Stellgrößen werden vom Roboter ignoriert
- Ein Fall mit erweiterter Initialisierung

Zusätzlich zeigt Abbildung B.8 den Verlauf des Hindernisses sowie den Roboter im Start- und Zielzustand. Der übrige Aufbau gleicht dem zur Auswertung der Parametrisierung von  $\beta$ .

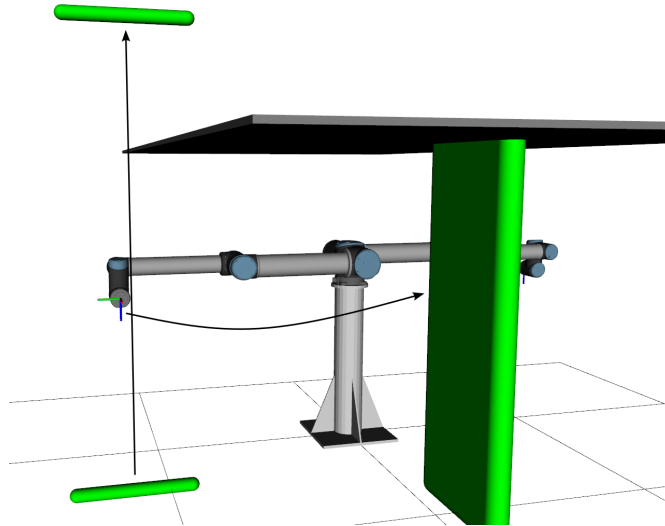


Abbildung B.8.: Die MHP plant eine Bewegung von links in den Zielzustand während ein Hindernis die Planung von unten nach oben durchkreuzt.

#### B.7.4. Neuplanung bei dynamischem Zielzustand

Die Auswertung zur Neuplanung bei dynamischem Zielzustand aus Abschnitt 9.2 ist wie folgt aufgebaut:

- Roboter UR10
- Anfangszustand  $\mathbf{x}_0 = \left(-\frac{\pi}{2} \ 0 \ 0 \ 0 \ 0 \ 0\right)^T$  rad
- Dynamischer Zielzustand  $\mathbf{x}_i(t) = \left(\frac{\pi}{2} \ 0 \ 0,7 \sin(0,2\pi t) \ 0 \ 0 \ 0\right)^T$  rad
- Hindernis gemäß Szenario A aus Abschnitt B.7.2
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$
- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>
- $K_\epsilon^- = 10$ ,  $\beta = 0,01$ ,  $\Delta\rho_{BS} = 0,01$  rad und  $r_{MHP} \in \{0,1; 0,5; 0,9\}$  rad
- Das globale Verfahren (RRT) nutzt die Nachbearbeitung und besitzt ein Zeitbudget von 0,2 s, das bis zu 4 mal verlängert wird, bevor die Planung abgebrochen wird
- IPOpt [Coi23] (MatSolver: MA57, Adaptive Mu, Auto Scaling)
- Erste Initialisierung über  $\mathbf{z}^A$  und dann Warmstart
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

Zusätzlich zeigt Abbildung B.9 den Verlauf der des Zielzustands.

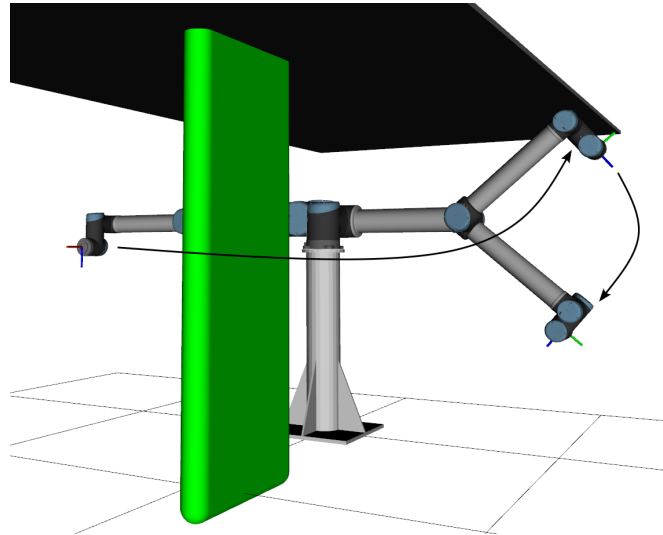


Abbildung B.9.: Der Roboter beginnt links und bewegt sich am Hindernis vorbei in Richtung dynamischem Zielzustand (auf und ab).

### B.7.5. Qualitativer Vergleich zu anderen Methoden

Die Auswertung zur Neuplanung bei dynamischem Zielzustand aus Abschnitt 9.2 ist wie folgt aufgebaut:

- Roboter UR10
- Anfangszustand, Zielzustand und Hindernisse gemäß Szenario B aus Abschnitt B.7.2
- $d_R = 0,05$  m und  $d_R^- = 0,02$  m
- $d_C = 0,1$  m und  $d_C^- = 0,05$  m
- $Q = 1$ ,  $R = 0,1$  und  $W_R = W_C = 155$
- $T_T = T_O = 0,1$  s und  $K = 31$
- $x^+ = 3,1$  rad,  $u^+ = 0,5$  rad s<sup>-1</sup> und  $a^+ = 0,5$  rad s<sup>-2</sup>
- $K_e^- = 10$  und  $\beta = 0,01$ ,  $r_{MHP} = 0,1$  rad und  $\Delta\rho_{BS} = 0,01$  rad
- Das globale Verfahren (RRT) nutzt die Nachbearbeitung und besitzt ein Zeitbudget von 0,2 s, das bis zu 4 mal verlängert wird, bevor die Planung abgebrochen wird
- IPOpt [Coi23] (MatSolver: MA57, *Adaptive Mu*, *Auto Scaling*)
- Erste Initialisierung über  $\mathbf{z}^A$  und dann Warmstart
- HPP über  $\beta = 0,1$ ,  $K = 2$  und  $K_e^- = 0$  initiiert
- STOMP nutzt  $K = 200$ ,  $T_T = 0,1$  s, 300 Iterationen, 400 *Rollouts*,  $R = 0,1$  und Kostensensitivität 1
- GTO als einmalige MHP Instanz mit maximal 300 Iterationen von IPOpt ohne Zeitlimit und  $K = 200$  bei ansonsten gleichen Parametern
- Rechner B.1.1 für die MHP und Rechner B.1.2 für die Physiksimulation

# Literatur

- [Abu+19] **Y. Abu Farha und J. Gall:** „Uncertainty-Aware Anticipation of Activities“. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, S. 1197–1204.
- [Aks+19] **E. Aksan, M. Kaufmann und O. Hilliges:** „Structured Prediction Helps 3D Human Motion Modelling“. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, S. 7143–7152.
- [Ame+01] **P. R. Amestoy, I. S. Duff, J.-Y. L’Excellent und J. Koster:** „MUMPS: A General Purpose Distributed Memory Sparse Solver“. In: *Applied Parallel Computing. New Paradigms for HPC in Industry and Academia*. Bd. 1947. Lecture Notes in Computer Science. 2001, S. 121–130.
- [And+16] **O. Andersson, M. Wzorek, P. Rudol und P. Doherty:** „Model-Predictive Control with Stochastic Collision Avoidance using Bayesian Policy Optimization“. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, S. 4597–4604.
- [And+18] **J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings und M. Diehl:** „CasADi: A Software Framework for Nonlinear Optimization and Optimal Control“. In: *Mathematical Programming Computation* 20.3 (2018), S. 1655.
- [And15] **T. T. Andersen:** *Optimizing the Universal Robots ROS Driver*. Techn. Ber. Technical University of Denmark, Department of Electrical Engineering, 2015.
- [Ata+16] **A. Ataka, P. Qi, A. Shiva, A. Shafti, H. Wurdemann, H. Liu und K. Althoefer:** „Real-Time Pose Estimation and Obstacle Avoidance for Multi-Segment Continuum Manipulator in Dynamic Environments“. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, S. 2827–2832.
- [AW 18] **A+W Software GmbH:** *LÖWE Fenster: Automatisierte Fertigung – handwerkliche Perfektion*. 2018. URL: <https://www.loewe-fenster.de/automatisierte-fertigung-handwerkliche-perfektion-pr/> (besucht am 12.07.2023).
- [Bak23] **Bakers Machinery:** *Kuchenanlage – Tromp*. 2023. URL: <https://bakersmachinery.com/de/produkt/kuchenanlage-tromp/> (besucht am 24.03.2024).
- [Bal+06] **L. Balan und G. Bone:** „Real-time 3D Collision Avoidance Method for Safe Human and Robot Coexistence“. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2006, S. 276–282.
- [Bar+00] **R. A. Bartlett, A. Wachter und L. T. Biegler:** „Active Set vs. Interior Point Strategies for Model Predictive Control“. In: *Proceedings of the 2000 American Control Conference (ACC)*. 2000, 4229–4233 vol.6.

- [Bec+17] **D. Beckert, A. Pereira und M. Althoff:** „Online Verification of Multiple Safety Criteria for a Robot Trajectory“. In: *2017 IEEE Annual Conference on Decision and Control (CDC)*. 2017, S. 6454–6461.
- [Bed15] **J. Beder:** *yaml-cpp - A YAML Parser and Emitter in C++ Matching the YAML 1.2 Spec*. Version 0.5.2. 2015. URL: <https://github.com/jbeder/yaml-cpp>.
- [Beh+15] **R. Behrens, J. Saenz, C. Vogel und N. Elkmann:** „Upcoming Technologies and Fundamentals for Safeguarding all Forms of Human-Robot Collaboration“. In: *8th International Conference Safety of Industrial Automated Systems - SIAS 2015*. 2015, S. 18–23.
- [Bel03] **R. E. Bellman:** *Dynamic Programming*. Dover ed. Dover Publications, 2003.
- [Bél16] **M. Bélanger-Barrette:** *Cobot, Force Limited Robot, Collaborative Robot, What's the Deal?* 2016. URL: <https://blog.robotiq.com/bid/73295/Cobot-Force-Limited-Robot-Collaborative-Robot-What-s-the-Deal> (besucht am 12. 07. 2023).
- [Ben14] **E. Benazera:** *EigenMVN - Multivariate Normal Distribution Sampling using C++11 and Eigen Matrices*. Commit 616a1ac. 2014. URL: <https://github.com/beniz/eigenmvn>.
- [Bet98] **J. T. Betts:** „Survey of Numerical Methods for Trajectory Optimization“. In: *Journal of Guidance, Control, and Dynamics* 21.2 (1998), S. 193–207.
- [Bie+18] **M. Biel und M. Norrlof:** „Efficient Trajectory Reshaping in a Dynamic Environment“. In: *2018 IEEE 15th International Workshop on Advanced Motion Control (AMC)*. 2018, S. 54–59.
- [Bob+85] **J. E. Bobrow, S. Dubowsky und J. S. Gibson:** „Time-Optimal Control of Robotic Manipulators Along Specified Paths“. In: *The International Journal of Robotics Research* 4.3 (1985), S. 3–17.
- [Bob88] **J. E. Bobrow:** „Optimal Robot Plant Planning using the Minimum-Time Criterion“. In: *IEEE Journal on Robotics and Automation* 4.4 (1988), S. 443–450.
- [Boh+00] **R. Bohlin und L. E. Kavraki:** „Path Planning using Lazy PRM“. In: *Robotics and Automation, 2000 IEEE International Conference*. 2000, 521–528 vol.1.
- [Bos+11] **P. Bosscher und D. Hedman:** „Real-Time Collision Avoidance Algorithm for Robotic Manipulators“. In: *Industrial Robot: An International Journal* 38.2 (2011), S. 186–197.
- [Bou+23] **K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauza, T. Davchev, Y. Zhou, A. Gupta, A. Raju, A. Laurens, C. Fantacci, V. Dalibard, M. Zambelli, M. Martins, R. Pevceviciute, M. Blokzijl, M. Denil, N. Batchelor, T. Lampe, E. Parisotto, K. Żoła, S. Reed, S. G. Colmenarejo, J. Scholz, A. Abdolmaleki, O. Groth, J.-B. Regli, O. Sushkov, T. Rothörl, J. E. Chen, Y. Aytar, D. Barker, J. Ortiz, M. Riedmiller, J. T. Springenberg, R. Hadsell, F. Nori und N. Heess:** *RoboCat: A Self-Improving Foundation Agent for Robotic Manipulation*. 2023.

- [Bro+02] **O. Brock und O. Khatib:** „Elastic Strips: A Framework for Motion Generation in Human Environments“. In: *The International Journal of Robotics Research* 21.12 (2002), S. 1031–1052.
- [Bui+18] **G. Buizza Avanzini, A. M. Zanchettin und P. Rocco:** „Constrained Model Predictive Control for Mobile Robotic Manipulators“. In: *Robotica* 36.01 (2018), S. 19–38.
- [Byr+12] **S. Byrne, W. Naeem und R. S. Ferguson:** „Efficient Local Sampling for Motion Planning of a Robotic Manipulator“. In: *Advances in Autonomous Robotics*. Bd. 7429. Lecture Notes in Computer Science. 2012, S. 164–175.
- [Byr+14] **A. Byravan, B. Boots, S. S. Srinivasa und D. Fox:** „Space-Time Functional Gradient Optimization for Motion Planning“. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014, S. 6499–6506.
- [Cas+09] **J. Cascio, M. Karpenko, Q. Gong, P. Sekhavat und I. M. Ross:** „Smooth Proximity Computation for Collision-Free Optimal Control of Multiple Robotic Manipulators“. In: *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2009, S. 2452–2457.
- [Cas+19] **A. Casalino, D. Bazzi, A. M. Zanchettin und P. Rocco:** „Optimal Proactive Path Planning for Collaborative Robots in Industrial Contexts“. In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. 2019, S. 6540–6546.
- [Coh+10] **B. J. Cohen, S. Chitta und M. Likhachev:** „Search-Based Planning for Manipulation with Motion Primitives“. In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. 2010, S. 2902–2908.
- [Coi22] **Coin-OR: ThirdParty-Mumps.** Version 5.1.1. 2022. URL: <https://github.com/coin-or-tools/ThirdParty-Mumps>.
- [Coi23] **Coin-OR: Ipop.** Version 3.14.11. 2023. URL: <https://github.com/coin-or/Ipop>.
- [Cor+12] **J. A. Corrales, G. J. G. Gómez, F. Torres und V. Perdereau:** „Cooperative Tasks between Humans and Robots in Industrial Environments“. In: *International Journal of Advanced Robotic Systems* 9.3 (2012), S. 94.
- [Die+06] **M. Diehl, H. G. Bock, H. Diedam und P.-B. Wieber:** „Fast Direct Multiple Shooting Algorithms for Optimal Robot Control“. In: *Fast Motions in Biomechanics and Robotics*. Bd. 340. Lecture Notes in Control and Information Sciences. 2006, S. 65–93.
- [Die+09] **M. Diehl, H. J. Ferreau und N. Haverbeke:** „Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation“. In: *Nonlinear Model Predictive Control*. Bd. 384. Lecture Notes in Control and Information Sciences. 2009, S. 391–417.
- [Dra+15] **A. D. Dragan, S. Bauman, J. Forlizzi und S. S. Srinivasa:** „Effects of Robot Motion on Human-Robot Collaboration“. In: *2015 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 2015, S. 51–58.

- [Duf+83] **I. S. Duff und J. K. Reid:** „The Multifrontal Solution of Indefinite Sparse Symmetric Linear“. In: *ACM Transactions on Mathematical Software* 9.3 (1983), S. 302–325.
- [Duf04] **I. S. Duff:** „MA57—a code for the solution of sparse symmetric definite and indefinite systems“. In: *ACM Transactions on Mathematical Software* 30.2 (2004), S. 118–144.
- [Ebe20] **D. Eberly:** *Geometric Tools Engine*. Version 5.4. 2020. URL: [www.github.com/davideberly/GeometricTools](http://www.github.com/davideberly/GeometricTools).
- [Eng+19] **T. Englert, A. Völz, F. Mesmer, S. Rhein und K. Graichen:** „A Software Framework for Embedded Nonlinear Model Predictive Control using a Gradient-based Augmented Lagrangian Approach (GRAMPC)“. In: *Optimization and Engineering* 20.3 (2019), S. 769–809.
- [Fan+16] **Y. Fang, J. Hu, W. Liu, B. Chen, J. Qi und X. Ye:** „A CPG-based Online Trajectory Planning Method for Industrial Manipulators“. In: *Proceedings of 2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS 2016)*. 2016, S. 41–46.
- [Far02] **J. Farkas:** „Theorie der einfachen Ungleichungen“. In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1902.124 (1902), S. 1–27.
- [Fau+17] **T. Faulwasser, T. Weber, P. Zometa und R. Findeisen:** „Implementation of Nonlinear Model Predictive Path-Following Control for an Industrial Robot“. In: *IEEE Transactions on Control Systems Technology* 25.4 (2017), S. 1505–1511.
- [Fer+14] **H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock und M. Diehl:** „qpOASES: A Parametric Active-Set Algorithm for Quadratic Programming“. In: *Mathematical Programming Computation* 6.4 (2014), S. 327–363.
- [Fin05] **R. Findeisen:** „Nonlinear Model Predictive Control : A Sampled Data Feedback Perspective“. Diss. Universität Stuttgart, 2005.
- [Fis+20] **A. Fishman, C. Paxton, W. Yang, D. Fox, B. Boots und N. Ratliff:** „Collaborative Interaction Models for Optimized Human-Robot Teamwork“. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, S. 11221–11228.
- [Fla+12] **F. Flacco, T. Kroger, A. de Luca und O. Khatib:** „A Depth Space Approach to Human-Robot Collision Avoidance“. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*. 2012, S. 338–345.
- [Fra98] **R. Franke:** *Omuses: A Tool for the Optimization of Multistage Systems, and HQP: A Solver for Sparse Nonlinear Optimization*. Commit ae4d4d6. 1998. URL: <https://github.com/omuses/hqp>.
- [Fri+05] **M. Frigo und S. G. Johnson:** „The Design and Implementation of FFTW3“. In: *Proceedings of the IEEE* 93.2 (2005), S. 216–231.

- [Gaf+22] **N. Gafur, G. Kanagalingam, A. Wagner und M. Ruskowski:** „Dynamic Collision and Deadlock Avoidance for Multiple Robotic Manipulators“. In: *IEEE Access* 10 (2022), S. 55766–55781.
- [Gec+05] **T. Gecks und D. Henrich:** „Human-Robot Cooperation: Safe Pick-and-Place Operations“. In: *2005 IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*. 2005, S. 549–554.
- [Gil+88] **E. G. Gilbert, D. W. Johnson und S. S. Keerthi:** „A Fast Procedure for Computing the Distance Between Complex Objects in Three-Dimensional Space“. In: *IEEE Journal on Robotics and Automation* 4.2 (1988), S. 193–203.
- [Gin01] **Gino van den Bergen:** „Proximity Queries and Penetration Depth Computation on 3D Game Objects“. In: *Game Developers Conference*. 2001.
- [Grü+17] **L. Grüne und J. Pannek:** *Nonlinear Model Predictive Control: Theory and Algorithms*. Second edition. Communications and control engineering. Springer, 2017.
- [Gue+21] **G. Guennebaud, B. Jacob u. a.:** *Eigen*. Version 3.4.0. 2021. URL: <http://eigen.tuxfamily.org>.
- [Har21b] **Harwell Subroutine Library:** *A Collection of Fortran Codes for Large Scale Scientific Computation*. Version 2021.05.05. 2021. URL: <http://www.hsl.rl.ac.uk/>.
- [Her+19] **A. Hernandez, J. Gall und F. Moreno:** „Human Motion Prediction via Spatio-Temporal Inpainting“. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, S. 7133–7142.
- [Hof+16] **A. Hoffmann, A. Poepfel, A. Schierl und W. Reif:** „Environment-Aware Proximity Detection with Capacitive Sensors for Human-Robot-Interaction“. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, S. 145–150.
- [Hog+11] **J. Hogg und J. A. Scott:** *HSL\_MA97 : A Bit-Compatible Multifrontal Code for Sparse Symmetric Systems*. Techn. Ber. 2011.
- [Hou+11] **B. Houska, H. J. Ferreau und M. Diehl:** „ACADO Toolkit-An Open-Source Framework for Automatic Control and Dynamic Optimization“. In: *Optimal Control Applications and Methods* 32.3 (2011), S. 298–312.
- [Hub+13] **M. Huber, C. Lenz, C. Wendt, B. Färber, A. Knoll und S. Glasauer:** „Increasing Efficiency in Robot-Supported Assemblies through Predictive Mechanisms: An Experimental Evaluation“. In: *2013 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2013, S. 503–508.
- [Imr+16] **M. Imran und F. Kunwar:** „A Hybrid Path Planning Technique Developed by Integrating Global and Local Path Planner“. In: *2016 International Conference on Intelligent Systems Engineering (ICISE)*. 2016, S. 118–122.
- [Int22] **Intel:** *oneAPI - Math Kernel Library*. Version 4.4.20(1). 2022. URL: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html>.

- [ISO10] **ISO13855: Sicherheit von Maschinen - Anordnung von Schutzeinrichtungen im Hinblick auf Annäherungsgeschwindigkeiten von Körperteilen.** Techn. Ber. DIN EN ISO 13855:2010-10. International Organization for Standardization, 2010.
- [Jar+17] **P. T. Jardine, S. Givigi und S. Yousefi:** „Planar Inequality Constraints for Stable, Collision-free Model Predictive Control of a Quadcopter“. In: *IFAC-PapersOnLine* 50.1 (2017), S. 9095–9100.
- [Joh+17] **F. I. John, J. Malzahn und T. Bertram:** „Controllability and Accessibility of Vibrations in Multiple Planes on Link-Elastic Robot Arms“. In: *IEEE ROBIO 2017*. 2017, S. 1491–1496.
- [Joh21] **S. G. Johnson:** *NLopt*. Version 2.7.1. 2021. URL: <http://github.com/stevengj/nlopt>.
- [Kal+11] **M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor und S. Schaal:** „STOMP: Stochastic Trajectory Optimization for Motion Planning“. In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. 2011, S. 4569–4574.
- [Kal+15] **J. Kalmari, J. Backman und A. Visala:** „A Toolkit for Nonlinear Model Predictive Control using Gradient Projection and Code Generation“. In: *Control Engineering Practice* 39 (2015), S. 56–66.
- [Kan+19] **A. Kanazawa, J. Kinugawa und K. Kosuge:** „Adaptive Motion Planning for a Collaborative Robot Based on Prediction Uncertainty to Enhance Human Safety and Work Efficiency“. In: *IEEE Transactions on Robotics* 35.4 (2019), S. 817–832.
- [Kar+11] **S. Karaman und E. Frazzoli:** „Sampling-based Algorithms for Optimal Motion Planning“. In: *The International Journal of Robotics Research* 30.7 (2011), S. 846–894.
- [Kat+13] **R. Katzschmann, T. Kroger, T. Asfour und O. Khatib:** „Towards Online Trajectory Generation Considering Robot Dynamics and Torque Limits“. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, S. 5644–5651.
- [Kav+96] **L. E. Kavraki, P. Svestka, J.-C. Latombe und M. H. Overmars:** „Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces“. In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), S. 566–580.
- [Kha+02] **W. Khalil und E. Dombre:** „Trajectory Generation“. In: *Modeling, Identification and Control of Robots*. 2002, S. 313–345.
- [Kha85] **O. Khatib:** „Real-Time Obstacle Avoidance for Manipulators and Mobile Robots“. In: *1985 IEEE International Conference on Robotics and Automation (ICRA)*. 1985, S. 500–505.
- [Kim+92] **J.-O. Kim und P. K. Khosla:** „Real-Time Obstacle Avoidance using Harmonic Potential Functions“. In: *IEEE Transactions on Robotics and Automation* 8.3 (1992), S. 338–349.

- [Kiv+17] **T. Kivelä, J. Mattila, J. Puura und S. Launis:** „Redundant Robotic Manipulator Path Planning for Real-Time Obstacle and Self-Collision Avoidance“. In: *Advances in Service and Industrial Robotics*. 2017.
- [Ko+93] **N. Y. Ko und B. H. Lee:** „Motion Planning of an Articulated Robot Manipulator Avoiding Time-Varying Obstacles“. In: *1993 IEEE International Conference on Robotics and Automation (ICRA)*. 1993, S. 1015.
- [Krä+18b] **M. Krämer, C. Rösmann, F. I. John und T. Bertram:** „Zeitoptimaler Mastaufbau einer mobilen Autobetonpumpe unter Nebenbedingungen“. In: *Forschung im Ingenieurwesen* 82.1 (2018), S. 45–57.
- [Krä+20a] **M. Krämer, F. I. Muster, C. Rösmann und T. Bertram:** „Elasticity-Aware Online Motion Optimization for Link-Elastic Manipulators“. In: *IFAC-PapersOnLine* 53.2 (2020), S. 9980–9985.
- [Krä+20b] **M. Krämer, C. Rösmann, F. Hoffmann und T. Bertram:** „Model Predictive Control of a Collaborative Manipulator Considering Dynamic Obstacles“. In: *Optimal Control Applications and Methods* (2020), S. 1211–1232.
- [Krä+21b] **M. Krämer, F. I. Muster, C. Rösmann und T. Bertram:** „An Optimization-based Approach for Elasticity-Aware Trajectory Planning of Link-Elastic Manipulators“. In: *Mechatronics* 75 (2021).
- [Krä+21c] **M. Krämer, R. J. Velasco-Guillen, P. Beckerle und T. Bertram:** „Comparing Online Robot Joint Space Trajectory Optimization for Task Space Applications“. In: *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2021, S. 223–230.
- [Krä+22a] **M. Krämer und T. Bertram:** „Improving Local Trajectory Optimization by Enhanced Initialization and Global Guidance“. In: *IEEE Access* 10 (2022), S. 29633–29645.
- [Krä+22b] **M. Krämer, F. Hoffmann und T. Bertram:** „Model Predictive Motion Planning for Robot-Assisted Observation and Recording of Human Activities“. In: *DFG Projektnummer 497071854*. 2022.
- [Kra88] **D. Kraft:** *A Software Package for Sequential Quadratic Programming*. Techn. Ber. DFVLR Institut für Dynamik der Flugsysteme Oberpfaffenhofen, 1988.
- [Krö10] **T. Kröger:** *On-Line Trajectory Generation in Robotic Systems: Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events*. Bd. 58. Springer Tracts in Advanced Robotics. Springer-Verlag Berlin Heidelberg, 2010.
- [Kuf+00] **J. J. Kuffner und S. M. LaValle:** „RRT-Connect: An Efficient Approach to Aangle-Query Path Planning“. In: *2000 IEEE International Conference on Robotics and Automation (ICRA)*. 2000, S. 995–1001.
- [Kul+05] **D. Kulić und E. A. Croft:** „Safe Planning for Human-Robot Interaction“. In: *Journal of Robotic Systems* 22.7 (2005), S. 383–396.
- [Kul+06] **D. Kulić und E. A. Croft:** „Real-Time Safety for Human-Robot Interaction“. In: *Robotics and Autonomous Systems* 54.1 (2006), S. 1–12.

- [Kun+16] **A. Kuntz, C. Bowen und R. Alterovitz:** „Interleaving Optimization with Sampling-Based Motion Planning (IOS-MP): Combining Local Optimization with Global Exploration“. 2016. URL: <https://arxiv.org/pdf/1607.06374>.
- [Kyr+88] **K. J. Kyriakopoulos und G. N. Saridis:** „Minimum Jerk Path Generation“. In: *1988 IEEE International Conference on Robotics and Automation (ICRA)*. 1988, S. 364–369.
- [Lan+12] **C. Landry, D. Hömberg, R. Henrion und M. Gerds:** „Path Planning and Collision Avoidance for Robots“. In: *Numerical Algebra, Control and Optimization 2.3* (2012), S. 437–463.
- [Lan+16] **F. Lange und A. Albu-Schäffer:** „Path-Accurate Online Trajectory Generation for Jerk-Limited Industrial Robots“. In: *IEEE Robotics and Automation Letters 1.1* (2016), S. 82–89.
- [Lar+99] **E. Larsen, S. Gottschalk, M. C. Lin und D. Manocha:** *Fast Proximity Queries with Swept Sphere Volumes*. Techn. Ber. Department of Computer Science, UNC Chapel Hill, 1999.
- [Las+15] **P. A. Lasota und J. A. Shah:** „Analyzing the Effects of Human-Aware Motion Planning on Close-Proximity Human-Robot Collaboration“. In: *Human Factors 57.1* (2015), S. 21–33.
- [Las+17] **P. A. Lasota und J. A. Shah:** „A Multiple-Predictor Approach to Human Motion Prediction“. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, S. 2300–2307.
- [LaV98] **S. M. LaValle:** „Rapidly-Exploring Random Trees : A new Tool for Path Planning“. In: *The Annual Research Report* (1998), S. 98–11.
- [Lee+13] **S.-D. Lee, B.-S. Kim und J.-B. Song:** „Human-Robot Collision Model with Effective Mass and Manipulability for Design of a Spatial Manipulator“. In: *Advanced Robotics 27.3* (2013), S. 189–198.
- [Lev+02] **P. Leven und S. Hutchinson:** „A Framework for Real-time Path Planning in Changing Environments“. In: *The International Journal of Robotics Research 21.12* (2002), S. 999–1030.
- [Li+19] **S. Li und J. A. Shah:** „Safe and Efficient High Dimensional Motion Planning in Space-Time with Time Parameterized Prediction“. In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. 2019, S. 5012–5018.
- [Llo+93] **J. Lloyd und V. Hayward:** „Trajectory Generation for Sensor-Driven and Time-Varying Tasks“. In: *The International Journal of Robotics Research 12.4* (1993), S. 380–393.
- [Lo +11] **C. Lo Guarino Bianco und O. Gerelli:** „Online Trajectory Scaling for Manipulators Subject to High-Order Kinematic and Dynamic Constraints“. In: *IEEE Transactions on Robotics 27.6* (2011), S. 1144–1152.
- [Luc+12] **A. de Luca und F. Flacco:** „Integrated Control for pHRI: Collision Avoidance, Detection, Reaction and Collaboration“. In: *2012 IEEE RAS & EMBS International*

- Conference on Biomedical Robotics and Biomechatronics (BioRob)*. 2012, S. 288–295.
- [Lum85] **V. J. Lumelsky**: „On Fast Computation of Distance Between Line Segments“. In: *Information Processing Letters* 21.2 (1985), S. 55–61.
- [Luo+19] **R. C. Luo und L. Mai**: „Human Intention Inference and On-Line Human Hand Motion Prediction for Human-Robot Collaboration“. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, S. 5958–5964.
- [Mac+03] **S. Macfarlane und E. A. Croft**: „Jerk-Bounded Manipulator Trajectory Planning: Design for Real-Time Applications“. In: *IEEE Transactions on Robotics and Automation* 19.1 (2003), S. 42–52.
- [Mag+19] **B. Magyar, N. Tsiogkas, J. Deray, S. Pfeiffer und D. Lane**: „Timed-Elastic Bands for Manipulation Motion Planning“. In: *IEEE Robotics and Automation Letters* 4.4 (2019), S. 3513–3520.
- [Mal14] **J. Malzahn**: „Modeling and Control of Multi-Elastic-Link Robots under Gravity“. Diss. Technische Universität Dortmund, 2014.
- [Mar+23] **T. Marcucci, P. Nobel, R. Tedrake und S. Boyd**: *Fast Path Planning through Large Collections of Safe Boxes*. 2023.
- [Mat+19] **E. Matheson, R. Minto, E. G. G. Zampieri, M. Faccio und G. Rosati**: „Human-Robot Collaboration in Manufacturing Applications: A Review“. In: *Robotics* 8.4 (2019), S. 100.
- [Mat14] **B. Matthias**: „Industrial Safety Requirements for Collaborative Robots and Applications“. In: *ERF 2014 Workshop: Workspace Safety in Industrial Robotics: Trends, Integration, and Standards*. 2014.
- [McC+92] **J. M. McCarthy und J. E. Bobrow**: „The Number of Saturated Actuators and Constraint Forces During Time-Optimal Movement of a General Robotic System“. In: *1992 IEEE International Conference on Robotics and Automation (ICRA)*. 1992, S. 542–546.
- [Min+20] **M. Minelli, A. Sozzi, G. de Rossi, F. Ferraguti, F. Setti, R. Muradore, M. Bonfe und C. Secchi**: „Integrating Model Predictive Control and Dynamic Waypoints Generation for Motion Planning in Surgical Scenario“. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, S. 3157–3163.
- [Moh+95] **A. Mohri, X. D. Yang und M. Yamamoto**: „Collision Free Trajectory Planning for Manipulator using Potential Function“. In: *1995 IEEE International Conference on Robotics and Automation (ICRA)*. 1995, S. 3069–3074.
- [Mus+19] **F. I. Muster, J. Malzahn und T. Bertram**: „Exploiting Controllability for Vibration Damping in Multiple Planes of Link-Elastic Robot Arms\*“. In: *2019 IEEE International Conference on Mechatronics (ICM)*. 2019, S. 187–192.
- [Nem+19] **H. Nemlekar, D. Dutia und Z. Li**: „Object Transfer Point Estimation for Fluent Human-Robot Handovers“. In: *2019 IEEE International Conference on Robotics and Automation (ICRA)*. 2019, S. 2627–2633.

- [Noc+06] **J. Nocedal und S. J. Wright:** *Numerical Optimization*. Second Edition. Springer Series in Operations Research and Financial Engineering. Springer Science + Business Media LLC, 2006.
- [Nut81] **A. Nuttall:** „Some Windows with very good Sidelobe Behavior“. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29.1 (1981), S. 84–91.
- [Opt21a] **Optimus Ride, Inc.:** *Altro-Cpp*. Commit d5e8cfe. 2021. URL: <https://github.com/optimusride/altro-cpp>.
- [Opt21b] **OptiTrack:** *NatNet SDK*. Version 4.0.0. 2021. URL: <https://optitrack.com/support/downloads/developer-tools.html>.
- [Oya+04] **T. Oyama und Y. Uno:** „Estimation of a Human Planned Trajectory from a Measured Trajectory“. In: *2004 IEEE International Conference on Systems, Man & Cybernetics*. 2004, S. 728–734.
- [Pan+12] **J. Pan, S. Chitta und D. Manocha:** „FCL: A General Purpose Library for Collision and Proximity Queries“. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*. Version 0.7.0. 2012, S. 3859–3866. URL: <https://github.com/flexible-collision-library/fcl>.
- [Par+13a] **C. Park, J. Pan und D. Manocha:** „ITOMP: Incremental Trajectory Optimization for Real-time Replanning in Dynamic Environments“. In: *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling*. ICAPS’12. 2013, S. 207–215.
- [Par+13b] **C. Park, J. Pan und D. Manocha:** „Real-Time Optimization-based Planning in Dynamic Environments using GPUs“. In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. 2013, S. 4090–4097.
- [Par+19] **J. S. Park, C. Park und D. Manocha:** „I-Planner: Intention-Aware Motion Planning using Learning-based Human Motion Prediction“. In: *The International Journal of Robotics Research* 38.1 (2019), S. 23–39.
- [Pav+17] **D. Pavlichenko und S. Behnke:** „Efficient Stochastic Multicriteria Arm Trajectory Optimization“. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, S. 4018–4025.
- [Per+15] **C. Perez-D’Arpino und J. A. Shah:** „Fast Target Prediction of Human Reaching Motion for Cooperative Human-Robot Manipulation Tasks using Time Series Classification“. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. 2015, S. 6175–6182.
- [Per+17] **A. Pereira und M. Althoff:** „Calculating Human Reachable Occupancy for Guaranteed Collision-free Planning“. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, S. 4473–4480.
- [Pet+05] **S. Petti und T. Fraichard:** „Safe Motion Planning in Dynamic Environments“. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2005, S. 2210–2215.

- [Pfe+87] **F. Pfeiffer und R. Johanni:** „A Concept for Manipulator Trajectory Planning“. In: *IEEE Journal on Robotics and Automation* 3.2 (1987), S. 115–123.
- [Pom+13] **M. Pomarlan und I. A. Sutan:** „Motion Planning for Manipulators in Dynamically Changing Environments using Real-Time Mapping of Free Workspace“. In: *2013 IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2013)*. 2013, S. 483–487.
- [Pon+86] **L. S. Pontryagin, V. G. Boltianskiĭ, R. V. Gamkrelidze und E. F. Mishchenko:** *The Mathematical Theory of Optimal Processes*. Bd. volume 4. Classics of Soviet mathematics. CRC Press, Taylor & Francis Group, 1986.
- [Pro14] **J. Progsch:** *ThreadPool - A simple C++11 Thread Pool Implementation*. Commit 9a42ec1. 2014. URL: <https://github.com/progschj/ThreadPool>.
- [Rag+15] **M. Ragaglia, A. M. Zanchettin und P. Rocco:** „Safety-Aware Trajectory Scaling for Human-Robot Collaboration with Prediction of Human Occupancy“. In: *2015 International Conference on Advanced Robotics (ICAR)*. 2015, S. 85–90.
- [Rat+09] **N. Ratliff, M. Zucker, J. A. Bagnell und S. Srinivasa:** „CHOMP: Gradient Optimization Techniques for Efficient Motion Planning“. In: *2009 IEEE International Conference on Robotics and Automation (ICRA)*. 2009, S. 489–494.
- [Rei+09] **J. K. Reid und J. A. Scott:** „An Out-Of-Core Sparse Cholesky Solver“. In: *ACM Transactions on Mathematical Software* 36.2 (2009), S. 1–33.
- [Ren+23a] **H. Renz, M. Krämer und T. Bertram:** „Comparing Human Motion Forecasts in Moving Horizon Trajectory Planning of Collaborative Robots“. In: *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2023, S. 1–6.
- [Ren+24] **H. Renz, M. Krämer und T. Bertram:** „Moving Horizon Planning for Human-Robot Interaction“. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction. HRI '24*. 2024, S. 939–943. URL: <https://github.com/rst-tu-dortmund/mhp4hri>.
- [Rob+17] **S. Robla-Gomez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torreferrero und J. Perez-Oria:** „Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments“. In: *IEEE Access* 5 (2017), S. 26754–26773.
- [Roc23] **Rockwell Automation:** *Intelligente, sichere Fertigung*. 2023. URL: <https://www.rockwellautomation.com/de-de/capabilities/industrial-cybersecurity/industry-services/manufacturing.html> (besucht am 12.07.2023).
- [Rog+01] **V. Rogozin, Y. Edan und T. Flash:** „A Real-Time Trajectory Modification Algorithm“. In: *Robotica* 19.4 (2001), S. 395–405.
- [Rös+17a] **C. Rösmann, F. Hoffmann und T. Bertram:** „Integrated Online Trajectory Planning and Optimization in Distinctive Topologies“. In: *Robotics and Autonomous Systems* 88 (2017), S. 142–153.

- [Rös+17b] **C. Rösmann, M. Oeljeklaus, F. Hoffmann und T. Bertram:** „Online Trajectory Prediction and Planning for Social Robot Navigation“. In: *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2017, S. 1255–1260.
- [Rös+18] **C. Rösmann, M. Krämer, A. Makarow, F. Hoffmann und T. Bertram:** „Exploiting Sparse Structures in Nonlinear Model Predictive Control with Hypergraphs“. In: *2018 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2018, S. 1332–1337.
- [ROS17] **ROS-Industrial:** *universal\_robot*. 2017. URL: [https://github.com/ros-industrial/universal\\_robot](https://github.com/ros-industrial/universal_robot).
- [Rös19] **C. Rösmann:** „Time-Optimal Nonlinear Model Predictive Control“. Diss. Technische Universität Dortmund, 2019.
- [ROS20] **ROS-Industrial:** *stomp\_ros*. Commit: 201c16c. 2020. URL: [https://github.com/ros-industrial/stomp\\_ros](https://github.com/ros-industrial/stomp_ros).
- [Ryb+17] **T. Rybus, K. Seweryn und J. Z. Sasiadek:** „Control System for Free-Floating Space Manipulator Based on Nonlinear Model Predictive Control (NMPC)“. In: *Journal of Intelligent & Robotic Systems* 85.3-4 (2017), S. 491–509.
- [Saf+19] **M. Safeea, P. Neto und R. Bearee:** „Efficient Calculation of Minimum Distance Between Capsules and Its Use in Robotics“. In: *IEEE Access* 7 (2019), S. 5368–5373.
- [San+01] **L. O. Santos, P. A. Afonso, J. A. Castro, N. M. Oliveira und L. T. Biegler:** „Online Implementation of Nonlinear MPC: An Experimental Case Study“. In: *Control Engineering Practice* 9.8 (2001), S. 847–857.
- [San+18] **B. Sangiovanni, A. Rendiniello, G. P. Incremona, A. Ferrara und M. Piastra:** „Deep Reinforcement Learning for Collision Avoidance of Robotic Manipulators“. In: *2018 European Control Conference (ECC)*. 2018, S. 2063–2068.
- [Sch+01] **O. Schenk, K. Gärtner, W. Fichtner und A. Stricker:** „PARDISO: A High-Performance Serial and Parallel Sparse Linear Solver in Semiconductor Device Simulation“. In: *Future Generation Computer Systems* 18.1 (2001), S. 69–78.
- [Sch+13] **J. Schulman, J. Ho, A. Lee, I. Awwal, H. Bradlow und P. Abbeel:** „Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization“. In: *Robotics: Science and Systems IX*. 2013.
- [Sic+09] **B. Siciliano, L. Sciavicco, L. Villani und G. Oriolo:** *Robotics: Modelling, Planning and Control*. Advanced textbooks in control and signal processing. Springer, 2009.
- [Som+12] **R. Someshwar, J. Meyer und Y. Edan:** „Models and Methods for H-R Synchronization“. In: *IFAC Proceedings Volumes* 45.6 (2012), S. 829–834.
- [Ste+16] **R. Steffens, M. Nieuwenhuisen und S. Behnke:** „Continuous Motion Planning for Service Robots with Multiresolution in Time“. In: *Intelligent Autonomous Systems 13*. Bd. 302. Advances in Intelligent Systems and Computing. 2016, S. 203–215.

- [Ste+20] **B. Stellato, G. Banjac, P. Goulart, A. Bemporad und S. Boyd:** „OSQP: An Operator Splitting Solver for Quadratic Programs“. In: *Mathematical Programming Computation* 12.4 (2020), S. 637–672.
- [Sto+13] **M. F. Stoelen, V. F. Tejada, J. G. Victores, A. Jardoan Huete, F. Bonsignorio und C. Balaguer:** „Adaptive Collision-Limitation Behavior for an Assistive Manipulator“. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2013, S. 1143–1148.
- [Str+94] **O. Stryk und M. Schlemmer:** „Optimal Control of the Industrial Robot Manutec r3“. In: *Computational Optimal Control*. 1994, S. 367–382.
- [Stu+21] **F. Stuhlenmiller, D. Clever, S. Rinderknecht, M. Lutter und J. Peters:** „Trajectory Optimization of Energy Consumption and Expected Service Life of a Robotic System“. In: *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2021, S. 842–847.
- [Suc+12] **I. A. Sucan, M. Moll und L. E. Kavraki:** „The Open Motion Planning Library“. In: *IEEE Robotics & Automation Magazine* 19.4 (2012). Version 1.4.2, S. 72–82. URL: <https://ompl.kavrakilab.org/>.
- [T A+19] **T. A. Howell, B. E. Jackson und Z. Manchester:** „ALTRO: A Fast Solver for Constrained Trajectory Optimization“. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, S. 7674–7679.
- [Tas+19] **B. Tasseff, C. Coffrin, A. Wächter und C. Laird:** *Exploring Benefits of Linear Solver Parallelism on Modern Nonlinear Optimization Applications*. 2019.
- [Tea+14] **T. A. V. Teatro, J. M. Eklund und R. Milman:** „Nonlinear Model Predictive Control for Omnidirectional Robot Motion Planning and Tracking With Avoidance of Moving Obstacles“. In: *Canadian Journal of Electrical and Computer Engineering* 37.3 (2014), S. 151–156.
- [Ter+04] **S. Terwen, M. Back und V. Krebs:** „Predictive Powertrain Control for Heavy Duty Trucks“. In: *IFAC Proceedings Volumes* 37.22 (2004), S. 105–110.
- [Uni21] **Universal Robots:** *Universal Robots User Manual*. UR10/CB3 Original instructions (en) Version 9.3.116. 2021.
- [Wäc+06] **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.
- [Wan+16] **M. Wang, J. Luo und U. Walter:** „A Non-Linear Model Predictive Controller with Obstacle Avoidance for a Space Robot“. In: *Advances in Space Research* 57.8 (2016), S. 1737–1746.
- [Wan+17] **Y. Wang, X. Ye, Y. Yang und W. Zhang:** „Collision-Free Trajectory Planning in Human-Robot Interaction through Hand Movement Prediction from Vision“. In: *2017 IEEE International Conference on Humanoid Robotics (Humanoids)*. 2017, S. 305–310.

- [Wan+18] **Y. Wang, Y. Sheng, J. Wang und W. Zhang:** „Optimal Collision-Free Robot Trajectory Generation Based on Time Series Prediction of Human Motion“. In: *IEEE Robotics and Automation Letters* 3.1 (2018), S. 226–233.
- [Wan+21] **C. Wang, J. Bingham und M. Tomizuka:** „Trajectory Splitting: A Distributed Formulation for Collision Avoiding Trajectory Optimization“. 2021. URL: <https://arxiv.org/pdf/2111.01899>.
- [Wei+18] **R. Weitschat, J. Ehrensperger, M. Maier und H. Aschemann:** „Safe and Efficient Human-Robot Collaboration Part I: Estimation of Human Arm Motions“. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, S. 1993–1999.
- [Xia+12] **Y. Xiao, Z. Du und W. Dong:** „Smooth and Near Time-Optimal Trajectory Planning of Industrial Robots for Online Applications“. In: *Industrial Robot: An International Journal* 39.2 (2012), S. 169–177.
- [Xue+17] **H. Xue, D. Q. Huynh und M. Reynolds:** „Bi-Prediction: Pedestrian Trajectory Prediction Based on Bidirectional LSTM Classification“. In: *DICTA 2017*. 2017, S. 1–8.
- [Zan+17] **A. M. Zanchettin und P. Rocco:** „Probabilistic Inference of Human Arm Reaching Target for Effective Human-Robot Collaboration“. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, S. 6595–6600.
- [Zha+17] **X. Zhang, A. Liniger und F. Borrelli:** „Optimization-Based Collision Avoidance“. 2017. URL: <https://arxiv.org/pdf/1711.03449>.
- [Zha+18] **X. Zhao und J. Pan:** „Considering Human Behavior in Motion Planning for Smooth Human-Robot Collaboration in Close Proximity“. In: *2018 IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2018, S. 985–990.
- [Zub15] **A. Zube:** „Cartesian Nonlinear Model Predictive Control of Redundant Manipulators Considering Obstacles“. In: *2015 IEEE International Conference on Industrial Technology (ICIT 2015)*. 2015, S. 137–142.
- [Zuc+13] **M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell und S. S. Srinivasa:** „CHOMP: Covariant Hamiltonian Optimization for Motion Planning“. In: *The International Journal of Robotics Research* 32.9-10 (2013), S. 1164–1193.

# Wissenschaftlicher Tätigkeitsnachweis

## Wissenschaftliche Publikationen als Erstautor

**M. Krämer, C. Rösmann, F. I. John und T. Bertram:** „Zeitoptimale Regelung des Mastaufbaus einer Autobetonpumpe unter statischen, kinodynamischen und geometrischen Nebenbedingungen“. In: *Dritte IFToMM D-A-CH Konferenz 2017*. 2017, S. 28–35.

**M. Krämer, C. Rösmann, F. I. John und T. Bertram:** „Zeitoptimaler Mastaufbau einer mobilen Autobetonpumpe unter Nebenbedingungen“. In: *Forschung im Ingenieurwesen* 82.1 (2018), S. 45–57.

**M. Krämer, F. I. Muster, C. Rösmann und T. Bertram:** „Elasticity-Aware Online Motion Optimization for Link-Elastic Manipulators“. In: *IFAC-PapersOnLine* 53.2 (2020), S. 9980–9985.

**M. Krämer, C. Rösmann, F. Hoffmann und T. Bertram:** „Model Predictive Control of a Collaborative Manipulator Considering Dynamic Obstacles“. In: *Optimal Control Applications and Methods* (2020), S. 1211–1232.

**M. Krämer und T. Bertram:** „Automatisierte 3D-Umfeldmodellierung für eine effiziente Trajektorienplanung kollaborativer Roboter“. In: *Digital-Fachtagung VDI MECHATRONIK 2021*. 2021, S. 7–13.

**M. Krämer, F. I. Muster, C. Rösmann und T. Bertram:** „An Optimization-based Approach for Elasticity-Aware Trajectory Planning of Link-Elastic Manipulators“. In: *Mechatronics* 75 (2021).

**M. Krämer, R. J. Velasco-Guillen, P. Beckerle und T. Bertram:** „Comparing Online Robot Joint Space Trajectory Optimization for Task Space Applications“. In: *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2021, S. 223–230.

**M. Krämer und T. Bertram:** „Improving Local Trajectory Optimization by Enhanced Initialization and Global Guidance“. In: *IEEE Access* 10 (2022), S. 29633–29645.

## Wissenschaftliche Publikationen als Co-Autor

**C. Rösmann, M. Krämer, A. Makarow, F. Hoffmann und T. Bertram:** „Exploiting Sparse Structures in Nonlinear Model Predictive Control with Hypergraphs“. In: *2018 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. 2018, S. 1332–1337.

**R. J. Velasco-Guillen, M. Krämer, F. Hoffmann, T. Bertram und P. Beckerle:** „Robot-Human Object Handover with Haptic Input and Visual Feedback“. In: *Proceedings - 29. Workshop Computational Intelligence, Dortmund, 28. - 29. November 2019*. 2019, S. 177–187.

**M. Waldner, M. Krämer und T. Bertram:** „Ein Hardware-in-the-Loop-Prüfstand zur Verifikation und Validierung von Matrixscheinwerfern“. In: *Fünfte IFToMM D-A-CH Konferenz 2019*. 2019, S. 118–119.

**M. Waldner, M. Krämer und T. Bertram:** „Hardware-in-the-Loop-Simulation of the Light Distribution of Automotive Matrix-LED-Headlights“. In: *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2019, S. 1311–1316.

**M. Waldner, M. Krämer und T. Bertram:** „Optimierung des Entwicklungsprozesses von Scheinwerfersystemen durch mechanische Nachbildung der Fahrdynamik“. In: *Fachtagung VDI MECHATRONIK 2019*. 2019, S. 91–96.

**F. I. Muster, M. Krämer und T. Bertram:** „Trajektorienplanung für einen strukturelastischen Leichtbauroboter mit mehreren Schwingungsebenen“. In: *Sechste IFToMM D-A-CH Konferenz 2020*. 2020, S. 67–68.

**M. Waldner, M. Krämer und T. Bertram:** „Digitization of Matrix-Headlights that Move as in the Real Test Drive“. In: *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. 2020, S. 264–269.

**H. Renz, M. Krämer und T. Bertram:** „Comparing Human Motion Forecasts in Moving Horizon Trajectory Planning of Collaborative Robots“. In: *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2023, S. 1–6.

**H. Renz, M. Krämer und T. Bertram:** „Reduktion eines menschlichen Körpermodells für eine echtzeitfähige Kollisionsvermeidung in der Mensch-Roboter-Kollaboration“. In: *Neunte IFToMM D-A-CH Konferenz 2023*. 2023.

**H. Renz, M. Krämer und T. Bertram:** „Uncertainty Estimation for Predictive Collision Avoidance in Human-Robot Collaboration“. In: *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2023, S. 1–6.

**H. Renz, M. Krämer und T. Bertram:** „Moving Horizon Planning for Human-Robot Interaction“. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. HRI '24. 2024, S. 939–943. URL: <https://github.com/rst-tu-dortmund/mhp4hri>.

## Patente

**T. Bertram, C. Rösmann, F. Hoffmann, F. I. John, J. Henikl und M. Krämer:** „Großmanipulator mit automatisiertem Mastenaufbau“. DE102016125145A1. 2018.

**T. Bertram, C. Rösmann, F. Hoffmann, F. I. John, J. Henikl und M. Krämer:** „Large manipulator with automated boom construction“. WO2018115248A1. 2018.

## Betreute studentische Abschlussarbeiten

**L. Meier-Ebert:** „Reinforcement Learning an einem Roboterarm zum Ballancieren eines Stabs“. Bachelorthesis. Technische Universität Dortmund, 2017.

- J. Stockermann:** „Bildbasierte Regelung des Sawyer Roboters im Robot Operating System“. Bachelorthesis. Technische Universität Dortmund, 2017.
- M. Waldner:** „Modellprädiktive Regelung der Vorfeldausschleuchtung eines Kraftfahrzeugs bei verschiedenen Fahrscenarien“. Masterthesis. Technische Universität Dortmund, 2017.
- Q. T. Bui:** „Online-Trajektorienplanung auf Basis einer Initialisierung mittels Homotopieklassen für einen Roboterarm mit Hindernissen“. Masterthesis. Technische Universität Dortmund, 2018.
- Y. Jiang:** „Task-Parameterized Movement Learning“. Masterthesis. Technische Universität Dortmund, 2018.
- S. Kurzawe:** „Modellprädikatives Ballancieren eines Stabs mit einem seriellen Roboterarm“. Masterthesis. Technische Universität Dortmund, 2018.
- S. Lai:** „Visual Servoing for Grasping of Moving Objects“. Masterthesis. Technische Universität Dortmund, 2018.
- A. Overmann:** „Lernen durch Demonstration mit Hilfe eines Kraft-Momentensensors auf dem Universal Robot Arm“. Masterthesis. Technische Universität Dortmund, 2018.
- A. Puzicha:** „Modellprädiktive Regelung eines strukturelastischen Roboterarms im Gelenkraum“. Bachelorthesis. Technische Universität Dortmund, 2018.
- V. F. Vallejo Rodríguez:** „Vision-based Simultaneous Localization and Mapping with Multiple Mobile Robots“. Masterthesis. Technische Universität Dortmund, 2018.
- R. J. Velasco-Guillen:** „Online Trajectory Optimization for Task Space Pick and Place Tasks using Model Predictive Control“. Masterthesis. Technische Universität Dortmund, 2018.
- B. Yuan:** „Policy Gradient Reinforcement Learning for Pole Balancing with a Robot Arm“. Masterthesis. Technische Universität Dortmund, 2018.
- C. Hams:** „Simulation und dezentrale Regelung eines Roboterarms mit zentraler Störgrößenkompensation“. Bachelorthesis. Technische Universität Dortmund, 2019.
- F. Menebröker:** „Robot Task Planning in the Context of Human-Robot Collaboration“. Masterthesis. Technische Universität Dortmund, 2019.
- X. Ren:** „Visual Servoing for Object Manipulation in Health Care“. Masterthesis. Technische Universität Dortmund, 2019.
- H. Renz:** „Trajektorienplanung für einen strukturelastischen Leichtbauroboter unter Nebenbedingungen“. Bachelorthesis. Technische Universität Dortmund, 2019.
- R. Sicking:** „Echtzeit Umgebungserfassung mittels 3D-Sensorik zur Unterstützung der Mensch-Roboter Kollaboration“. Masterthesis. Technische Universität Dortmund, 2019.
- Y. Sun:** „Human-Robot-Interaction in Hand-Over Task using Force Torque Sensor in Robotic Gripper“. Masterthesis. Technische Universität Dortmund, 2019.
- D. Feng:** „Kontaktlose Teleoperation eines kollaborativen Roboters mittels Eyetracking“. Masterthesis. Technische Universität Dortmund, 2021.

**P. Hartmann:** „Vergleich von parallelen und sequentiellen Algorithmen zur Echtzeit-Trajektorienoptimierung in dynamischen Umgebungen“. Masterthesis. Technische Universität Dortmund, 2021.

**M. Lemanov:** „Automatisierte Umfeldmodellierung des Arbeitsraums kollaborativer Roboter mittels Next-Best-View“. Bachelorthesis. Technische Universität Dortmund, 2021.

**O. Perkasa:** „Active Compliance of a Collaborative Robot Based on Admittance Control“. Masterthesis. Technische Universität Dortmund, 2021.

**H. Al-Zubaidy:** „A Substitutional-Reality-based Benchmark System for Evaluating Human-Robot Interaction Scenarios“. Masterthesis. Technische Universität Dortmund, 2021.

**H. Renz:** „Human Movement Prediction for Collision Avoidance in Human-Robot Collaboration“. Masterthesis. Technische Universität Dortmund, 2022.

## **Betreute studentische Projektgruppen**

*Online Motion Planning and Control of Cooperative Robotic Manipulators.* 2017.

*Model Predictive Control for Robotic Pole Balancing.* 2018.

*Dynamischer Scheinwerferprüfstand.* 2018.

*Programming and Simulation of CoRobots.* 2019.

*Human-Robot-Interaction I.* 2020.

*Human-Robot-Interaction II.* 2020.

*Model Predictive Control and Planning for CoRobot Assistant.* 2023.