

Bahnoptimierung eines Industrieroboters zur Minimierung des Energieverbrauchs

Bachelorarbeit

Studiengang Elektrotechnik

Studienrichtung Automation

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Dennis Gardy

Abgabedatum:	25.09.2023
Bearbeitungszeitraum:	01.07.2023 - 25.09.2023
Matrikelnummer:	1376352
Kurs:	TEA20
Ausbildungsfirma:	Mercedes-Benz AG
Betreuer der Ausbildungsfirma:	Mathias Ostermann M.Sc.
Gutachter der Dualen Hochschule:	Prof. Dr. Jochen M. Rieber

Sperrvermerk

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018:

Der Inhalt dieser Arbeit darf weder als Ganzes noch in Auszügen Personen außerhalb des Prüfungsprozesses und des Evaluationsverfahrens zugänglich gemacht werden, sofern keine anders lautende Genehmigung vom Dualen Partner vorliegt.

Potsdam, den 25.09.2023

Dennis Gardy

Erklärung

gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018.

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema:

Bahnoptimierung eines Industrieroboters zur Minimierung des Energieverbrauchs

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Potsdam, den 25.09.2023

Dennis Gardy

Kurzfassung

Ziel der vorliegenden Arbeit ist die Untersuchung der Energieeinsparung eines Industrieroboters vom Typ KR210 R2700-2 durch die Umsetzung einer Bahnoptimierung. Dazu wird die Forschungsfrage gestellt, ob ein für die Produktion entwickeltes Roboterprogramm durch Hinzufügen und Verschieben von Via-Punkten ohne signifikante Erhöhung der Bewegungsdauer energetisch optimiert werden kann. Die numerische Optimierung des Via-Punktes erfolgt auf Basis eines simulierten mechanischen Modells. Zur Beantwortung der Forschungsfrage wird eine Abstellbewegung des Roboters vom letzten Prozesspunkt eines Produktionsprogramms in die Grundstellung untersucht. Dieser Verfahrweg wird als repräsentativ für eine Vielzahl von Bewegungsabläufen für Industrieroboter in der Automobilproduktion angesehen, da sie in jedem Takt beim Einfahren einer Karosserie zur Kollisionsvermeidung durchgeführt wird. Nach Validierung des simulierten Modells sowie der Optimierungsergebnisse am realen System wird eine Energieeinsparung für den beschriebenen Bewegungsablauf von 11,7 % gegenüber der Initialbewegung erzielt. Basierend auf den Ergebnissen wird empfohlen, eine Potenzialanalyse durchzuführen, in der die Skalierbarkeit des Ansatzes entsprechend der Anzahl baugleicher Industrieroboter ermittelt wird. Bei positiver Bewertung ist der Versuchsumfang zu erweitern.

Abstract

This thesis aims to investigate the energy savings of an industrial robot by implementing path optimization. The research question is whether a robot programme developed for car production can be optimised in terms of energy by adding and shifting via points without significantly increasing the movement duration. The numerical optimization of the via point is based on a simulated mechanical model. Therefore, the movement of the robot from the last process point of a production programme to the home position is investigated. This path is considered representative of industrial robots' motion in automotive production since it is performed in every cycle to avoid collisions. After validation of the simulated model as well as the optimization results, an energy saving of 11.7 % is achieved for the described motion sequence compared to the initial one. Based on this, it is recommended to carry out a potential analysis in which the scalability of the approach is determined according to the number of affected industrial robots. In case of a positive evaluation, the scope of the experiment should be expanded.

Inhaltsverzeichnis

Verzeichnis verwendeter Formelzeichen und Abkürzungen	x
1 Einleitung	1
1.1 Problemstellung	1
1.2 Literaturüberblick	1
1.3 Ausgangslage	4
1.4 Zielsetzung	4
1.5 Geplantes Vorgehen	5
2 Mechanische Modellbildung	6
2.1 Vorwärtsskinematik	6
2.2 Geschwindigkeits-Kinematik	11
2.3 Dynamik	11
2.3.1 Euler-Lagrange Gleichungen	11
2.3.2 Rekursiver-Newton-Euler-Algorithmus	13
2.4 Modellannahmen und Vernachlässigungen	16
3 Implementierung der Via-Punkt Trajektorie	18
4 Validierung des Roboterdynamik-Modells	24
4.1 Messaufbau	24
4.2 Durchführung	25
4.3 Auswertung	26
5 Bahnoptimierung	33
5.1 Definition des Optimierungsproblems	33

Inhaltsverzeichnis

5.2	Zielfunktion	34
5.3	Nebenbedingungen	35
5.4	Solver- und Optimierungsalgorithmus	36
5.5	Durchführung der Optimierung	37
5.6	Auswertung der Optimierungsergebnisse	37
6	Validierung der Optimierungsergebnisse	44
6.1	Durchführung	44
6.2	Auswertung	45
7	Bewertung der Optimierungsergebnisse	48
8	Zusammenfassung und Ausblick	51
Literatur		54
Abbildungsverzeichnis		58
Tabellenverzeichnis		61
Anhang A - Planung		62
Anhang B - Theoretische Grundlagen		64
B.1	Kinematische Kette	64
B.2	Euklidische Gruppe	64
B.3	Geschwindigkeits-Kinematik	65
B.3.1	Schiefsymmetrische Matrizen	65
B.3.2	Drehwinkelgeschwindigkeit	65
B.3.3	Lineare Geschwindigkeit	66
Anhang B - Roboterdaten		68
C.1	KR210 2700-2 Datenblatt	68
C.2	Systemparameter	70
Anhang C - MATLAB-Implementierung		71
D.1	DH-Transformation	71

Inhaltsverzeichnis

D.2	Parameter	74
D.2.1	Vorwärtskinematik für die Ausgangsstellung	74
D.2.2	Trägheitsmoment I_i^0 am Massezentrum in $\frac{\text{kg}}{\text{m}^2}$	74
D.2.3	Lage der Massenschwerpunkte im Intertialkoordinatensystem KS{0}	77
D.2.4	r_{i,C_i}^i	77
D.2.5	$r_{i-1,i}^i$	78
D.2.6	Masse m_i	78
D.2.7	Getriebe-Übersetzung i_i	78
D.3	RNEA	79
D.3.1	Initialisierung	79
D.3.2	Vorwärtskinematik	79
D.3.3	Parameter in der Reihenfolge: $I_i^i, r_{i,C_i}^i, r_{i-1,i}^i, m_i, i_i$	79
D.3.4	Winkelgeschwindigkeit ω_i^i	80
D.3.5	Winkelbeschleunigung $\dot{\omega}_i^i$	80
D.3.6	Lineare Beschleunigung \ddot{p}_i^i (Siciliano) a_e (Spong)	80
D.3.7	Lineare Beschleunigung $\ddot{p}_{C_i}^i$ (Siciliano) a_c (Spong)	81
D.3.8	Gewichtsausgleich	81
D.3.9	Kraft f_i^i	82
D.3.10	Drehmoment μ_i^i	82
D.3.11	Drehmoment τ_i im KS{i-1}	83
D.4	Bahnplanung	83
D.4.1	Initialisierung	83
D.4.2	Aufstellen der Polynome	83
D.4.3	Definition der Nebenbedingungen	84
D.4.4	Berechnung der Koeffizienten a_0, \dots, a_6	84
D.4.5	Berechnung der Trajektorie	84
D.5	Testsimulation Bahnplanung	85
D.5.1	Trajektorie Kleben-Seitenwand	85
D.5.2	Bewegung 1 (home -> Vorposition)	85
D.5.3	Bewegung 3 (letzter Punkt der Trajektorie Kleben Seitenwand -> home)	85
D.5.4	Bahnplanung	86
D.5.5	Modellberechnung der Drehmomente, Leistungsaufnahme	88

Inhaltsverzeichnis

D.5.6	Daten Vorverarbeitung	88
D.5.7	Anzeige der Daten	89
D.5.8	Erzeuge die Abbildungen	89
D.6	Optimierung	90
D.6.1	KlebenSeitenwand	90
D.6.2	Bewegung 1 (home -> Vorposition)	90
D.6.3	Bewegung 3 (letzter Punkt der Trajektorie Kleben Seitenwand -> home)	90
D.6.4	Startwert Via-Punkte	91
D.6.5	Initial-Trajektorie-Definition	91
D.6.6	Definition Optimierer	91
D.7	Berechnung der Zielfunktion	92
D.7.1	Bahnplanung	93
D.7.2	Modellberechnung der Drehmomente, Leistungsaufnahme	94
D.7.3	Daten Vorverarbeitung	94
D.7.4	Anzeige der Daten	95
Anhang D - Messdaten		96
E.1	Vergleich der simulierten Verläufe für den initialen und justierten-energieoptimierten Parametervektor	96

Verzeichnis verwendeter Formelzeichen und Abkürzungen

Matrix	Großbuchstabe, fett, kursiv: \boldsymbol{A}
Punkt	Großbuchstabe: A
Skaler	Kleinbuchstabe, kursiv: a
Vektor	Kleinbuchstabe, fett, kursiv: \boldsymbol{a}
f.	folgende Seite
ff.	fortfolgende Seiten
KS <i>{i}</i>	Koordinatensystem <i>i</i>
$\dot{\omega}_i^i$	Winkelbeschleunigung des Verbindungsglieds <i>i</i> ausgedrückt in KS <i>{i}</i>
$\dot{\omega}_i$	Winkelbeschleunigung des Verbindungsglieds <i>i</i>
ω_i^i	Winkelgeschwindigkeit des Verbindungsglieds <i>i</i> ausgedrückt in KS <i>{i}</i>
ω_i	Winkelgeschwindigkeit des Verbindungsglieds <i>i</i>
$\ddot{\mathbf{p}}_i^i$	Lineare Beschleunigung im Ursprung des KS <i>{i}</i> ausgedrückt in KS <i>{i}</i>
$\ddot{\mathbf{p}}_{C_i}^i$	Lineare Beschleunigung im Masseschwerpunkt C_i ausgedrückt in KS <i>{i}</i>
$\ddot{\mathbf{p}}_i$	Lineare Beschleunigung im Ursprung des KS <i>{i}</i>
$\ddot{\mathbf{p}}_{C_i}$	Lineare Beschleunigung im Masseschwerpunkt C_i

Verzeichnis verwendeter Formelzeichen und Abkürzungen

$\dot{\mathbf{p}}_i^i$	Lineare Geschwindigkeit im Ursprung des KS $\{i\}$ ausgedrückt in KS $\{i\}$
$\dot{\mathbf{p}}_{C_i}^i$	Lineare Geschwindigkeit im Masseschwerpunkt C_i ausgedrückt in KS $\{i\}$
$\dot{\mathbf{p}}_i$	Lineare Geschwindigkeit im Ursprung des KS $\{i\}$
$\dot{\mathbf{p}}_{C_i}$	Lineare Geschwindigkeit im Masseschwerpunkt C_i
$-\boldsymbol{\mu}_{i+1}^{i+1}$	Drehmoment, welches vom Verbindungsglied $i + 1$ auf das Verbindungsglied i ausgeübt wird, ausgedrückt in KS $\{i + 1\}$
$-\boldsymbol{\mu}_{i+1}$	Drehmoment, welches vom Verbindungsglied $i + 1$ auf das Verbindungsglied i ausgeübt wird
$-\mathbf{f}_{i+1}^{i+1}$	Kraft, die vom Verbindungsglied $i + 1$ auf das Verbindungsglied i ausgeübt wird, ausgedrückt in KS $\{i + 1\}$
$-\mathbf{f}_{i+1}$	Kraft, die vom Verbindungsglied $i + 1$ auf das Verbindungsglied i ausgeübt wird
$\boldsymbol{\mu}_i^i$	Drehmoment, welches vom Verbindungsglied $i - 1$ auf das Verbindungsglied i ausgeübt wird, ausgedrückt in KS $\{i\}$
$\boldsymbol{\mu}_i$	Drehmoment, welches vom Verbindungsglied $i - 1$ auf das Verbindungsglied i ausgeübt wird
\mathbf{f}_i^i	Kraft, die vom Verbindungsglied $i - 1$ auf das Verbindungsglied i ausgeübt wird, ausgedrückt in KS $\{i\}$
\mathbf{f}_i	Kraft, die vom Verbindungsglied $i - 1$ auf das Verbindungsglied i ausgeübt wird
\mathbf{r}_{i,C_i}^i	Vektor vom Ursprung des KS $\{i\}$ zum Masseschwerpunkt C_i ausgedrückt in KS $\{i\}$
\mathbf{r}_{i-1,C_i}^i	Vektor vom Ursprung des KS $\{i - 1\}$ zum Masseschwerpunkt C_i des Verbindungsglieds i ausgedrückt in KS $\{i\}$
$\mathbf{r}_{i-1,i}^i$	Vektor vom Ursprung des KS $\{i - 1\}$ zum Ursprung des KS $\{i\}$ ausgedrückt in KS $\{i\}$
i_i	Übersetzungsverhältnis des Getriebes i
α_i	Verwindung
$\omega_{0,n}^0$	Endeffektor-Winkelgeschwindigkeit ausgedrückt in KS $\{0\}$
τ	generalisierte Kräfte

Verzeichnis verwendeter Formelzeichen und Abkürzungen

$C(\mathbf{q}, \dot{\mathbf{q}})$	Anteil der Corioliskraft und Zentrifugalkraft an den generalisierten Kräften
E	Einheitsmatrix
\mathbf{g}	Fallbeschleunigung
$\mathbf{g}(\mathbf{q})$	Anteil der Schwerkraft an den generalisierten Kräften
H	Endeffektor-Pose im Basis-Koordinatensystem
I_i^0	Trägheitstensor des Verbindungsglieds i ausgedrückt in KS{0}
I_i^i	Trägheitstensor des Verbindungsglieds i ausgedrückt in KS{i}
J_v	Lineare-Geschwindigkeits-Jacobi-Matrix
J_ω	Winkelgeschwindigkeits-Jacobi-Matrix
$M(\mathbf{q})$	Massenmatrix
\mathbf{o}_n^0	Koordinaten des Ursprungs des Endeffektor-Koordinatensystems im Basis-Koordinatensystem
\mathbf{q}	generalisierte Koordinaten
\mathbf{q}_e	Zielpunkt in Gelenkkoordinaten
\mathbf{q}_s	Startpunkt in Gelenkkoordinaten
\mathbf{q}_v	Parametervektor des Via-Punktes in Gelenkkoordinaten
$\mathbf{r}_{C_i}^0$	Lage des Masseschwerpunkts vom Verbindungsglied i im KS{0}
R^n_0	Rotation des Endeffektor-Koordinatensystems gegenüber dem Basis-Koordinatensystem
S	Schiefsymmetrische Matrix
T_i^{i-1}	homogene Transformationsmatrix zur Lagebeschreibung des KS{i} ausgedrückt in KS{i - 1}
\mathbf{v}_n^0	lineare Endeffektor-Geschwindigkeit ausgedrückt in KS{0}
$\ddot{\theta}$	Gelenkwinkelbeschleunigung
\ddot{q}	Gelenkwinkelbeschleunigung
$\ddot{q}_{i,max}$	maximale Gelenkwinkelbeschleunigung des i -ten Gelenks
$\dot{\theta}$	Gelenkwinkelgeschwindigkeit

Verzeichnis verwendeter Formelzeichen und Abkürzungen

\dot{q}	Gelenkwinkelgeschwindigkeit
$\dot{q}_{i,max}$	maximale Gelenkwinkelgeschwindigkeit des i -ten Gelenks
$\dot{\theta}_i$	Gelenkwinkel des i -ten Gelenks
$\tau_{i,max}$	maximales Drehmoment des i -ten Gelenks
θ	Gelenkwinkel
θ_i	Gelenkwinkel
a_i	Armelementlänge
d_i	Gelenkabstand
E_{mech}	ische Energie
J	Zielfunktion
L	Lagrange Funktion
m_i	Masse des Körpers i
P_i	Mechanische Leistung für den Antrieb des i -ten Gelenks
P_{mech}	Mechanische Leistung
$q_{i,min}$	maximaler Gelenkwinkel des i -ten Gelenks
$q_{i,max}$	minimaler Gelenkwinkel des i -ten Gelenks
t	Zeit
t_e	Endzeitpunkt der Bahnbewegung
t_s	Startzeit der Bahnbewegung
t_v	Zeitpunkt für das Erreichen des Via-Punkts
θ	Gelenkwinkel
q	Gelenkwinkel
$q_{v,i,justiert}$	Justierter, energieoptimierter Via-Punkt
K	Kinetische Energie
U	Potenzielle Energie
CO_2	Kohlenstoffdioxid
API	Application Programming Interface

Verzeichnis verwendeter Formelzeichen und Abkürzungen

ASCII	American Standard Code for Information Interchange
CAD	Computer Aided Design
dat	data-file
DC-Net	Gleichstromnetz
DH	Denavit-Hartenberg
DOE	Design of Experiments
ESB	Ersatzschaltbild
Home	Grundstellung
IR	Industrieroboter
JSON	JavaScript Object Notation
KKT-Konvergenzkriterien	Karush-Kuhn-Tucker-Konvergenzkriterien
KR C	KUKA Robot Control
KRL	KUKA Robot Language
OR	Override
PMSM	permanentmagneterregte Synchronmaschinen
Pose	Position und Orientierung
PTP	Point-to-Point
RNEA	Rekursiver-Newton-Euler-Algorithmus
RSI	RobotSensorInterface
SiL	Software-in-the-Loop
SQP	sequential quadratic programming
src	source-file
TCP	Tool-Center-Point
TP-Filter	Tiefpassfilter
UDP	User Datagram Protocol
Winsock	Windows Sockets 2

1 Einleitung

1.1 Problemstellung

Aus ökonomischer und ökologischer Perspektive ist die Mercedes-Benz AG fortwährend bestrebt die Energieeffizienz in der Automobilproduktion zu minimieren. Als Kernelement der nachhaltigen Geschäftsstrategie definiert die Ambition 2039 den Weg zur CO₂ neutralen Mobilität unter Berücksichtigung aller „Wertschöpfungsstufen des Automobils – von der Lieferkette über die Produktion bis hin zur Nutzungsphase und Entsorgung der Fahrzeuge“ [o. 22, S. 15]. Industrieroboter (IR) sind ein wesentlicher Bestandteil der Fertigung und Montage und damit ein wichtiger Stellhebel auf dem Weg zur CO₂ neutralen Automobilproduktion. Eine CO₂-Reduktion ist in der Einsparung des Roboter Energieverbrauchs zu erzielen. Damit reiht sich eine Untersuchung in die operative Umweltzielsetzung für die Planung und Produktion der Mercedes-Benz Werke ein [o. 22, S. 21]. Neben dem Energieverbrauch ist in der Produktion an erster Stelle die Ausbringungsmenge und damit die Taktzeit von Relevanz. Infolgedessen lautet die Forschungsfrage der Arbeit: Kann ein für die Produktion entwickeltes Roboterprogramm durch das Hinzufügen und Verschieben von Via-Punkten ohne eine signifikante Erhöhung der Bewegungsdauer energetisch optimiert werden?

1.2 Literaturüberblick

In der vorliegenden Bachelorarbeit wird auf die Themen mechanische Modellierung, Bahnplanung und numerische Optimierung zur Verbesserung der IR Energieeffizienz Bezug genommen. Im Folgenden wird eine Zusammenstellung der verwendeten Literatur aufgeführt. [CWV17] gibt einen Überblick über bestehende Methoden zur Verbesserung

der Energieeffizienz von Industrierobotern durch Anpassung der Hard- und/oder Software. Hardwareanpassungen werden in dieser Arbeit nicht untersucht. Aus Gründen der Vollständigkeit werden sie an dieser Stelle zusammenfassend dargestellt. Ein erster Ansatz basiert auf konstruktiven Änderungen an Bauteilen, z. B. dem Austausch schwerer Robotergreifer durch Leichtbaukomponenten. Durch die Reduzierung von Gewicht und Massenträgheit wird das aufzuwendende Drehmoment in den Antrieben des Roboters minimiert. Ein zweiter Ansatz ist die Installation von Systemen oder Komponenten zur Energierückgewinnung im Antriebsstrang während der Bremsphase. [Pel+15] erarbeitet ein Konzept für den Austausch der zurückgewonnenen Energie über ein Gleichstromnetz (DC-Netz). Dabei wird unter anderem die DC-Netzanbindung des Roboters über eine speziell entwickelte Umrichter-Schnittstelle für den bidirektionalen Energiefloss skizziert. Softwareseitig werden eine Verfahrzeit-Optimierung von IR im Kontext einer Fließfertigung und die Anpassung der Bahngeometrie für isoliert betrachtete Roboter unterschieden. In Phasen, in denen die Anlage auf einen vor- oder nachgelagerten Prozess wartet, kann eine Optimierung des Vorschubs (Override (OR)) angewandt werden. Eine Sonderfall bilden hierbei miteinander interagierende Roboter, deren Bewegungsabläufe aufeinander abgestimmt werden müssen. Einen entscheidenden Beitrag zur Untersuchung dieser Ansätze leistet [Egg19]. Der Ausarbeitung ist hinzuzufügen, dass eine Änderung der Geschwindigkeitsprofile je nach Bewegungsart Auswirkung auf die Bahnplanung und damit den Pfad des Roboters nehmen kann, woraus eine gesonderte Absicherung auf Kollisionsfreiheit erfolgen muss. Grundlage für die Optimierung des Energiebedarfs ist die Definition eines geeigneten Modells für den IR. Hervorzuheben ist, das in [Egg19] beschriebene, erstmalig in [Zia+17] publizierte und als Patent [Egg+06] angemeldete Energiemodell. Im Gegensatz zu den Robotermustern in [Pel+11], [SSP02] und [Par+15] werden nicht nur die Verluste der elektrischen Komponenten detailliert betrachtet, sondern zusätzlich die Betriebszustände des IR unterschieden. Entsprechend der Phasen MOTION¹, HOLD² und IDLE³ wird eine präzise Analyse des Energiebedarfs ermöglicht [Zia+17]. Ausgangspunkt der numerischen Optimierung ist die Formulierung eines Optimierungsproblems inklusive Definition einer Zielfunktion. [Zia+17] formuliert eine Reduktion des Energieverbrauchs durch die Anpassung der Bewegungsdauer. Ein zweiter Ansatz, der in [Egg19] untersucht wird, ist die Optimierung der Bahngeometrie.

¹Roboter in Bewegung

²Roboter im Stillstand, Antriebe sind in Regelung

³Haltebremsen sind aktiv

Im Gegensatz zu früheren Ausarbeitungen werden die Ergebnisse hierbei in umfangreichen praktischen Szenarien validiert. Entscheidend für die Durchführung der Optimierung ist die Wahl eines geeigneten Algorithmus. Auf der Grundlage des Modells wird die Bahngeometrie in [Han+12] mit einem Quasi-Newton-Verfahren, siehe [PLB15, S. 49], minimiert. [Zia+17] nutzt für das selbe Ziel einen Active-Set Algorithmus [LY21, S. 445]. Bezüglich der theoretischen Grundlagen der Nichtlinearen-Optimierung wird auf [NW06] verwiesen. [CWV17, S.10 ff.] gibt eine Übersicht weiterer, für den Anwendungsfall bereits angewandter, Algorithmen. [Zia+17] formuliert eine Reduktion des Energieverbrauchs neben der Bewegungsdauer durch eine Minimierung der auftretenden Motordrehmomente. Alternativ wird in [Han+12] die vom Roboter aufgenommene DC-Netzleistung als Zielfunktion herangezogen. [Lin+18] schlägt eine Mehrzieloptimierung über den Energieverbrauch und die Verfahrdauer vor. Die häufig in den Grundlagen zitierte Ausarbeitung [SRB08] betrachtet in der Optimierung simultan die Minimierung der vom Roboter aufgenommenen Leistung, den Ruck und die Beschleunigung der einzelnen Gelenke. Abschließend sei die Ausarbeitung [Bjo+13] genannt, welche an Stelle der nichtlinearen Optimierung die Theorie der optimalen Steuerung für eine energieeffiziente und kollisionsfreie Roboterbewegung zu Grunde legt. Eine Notwendigkeit für die Optimierung einer Roboter-Bewegungsbahn ist die Definition der Bahnplanung. Hierbei werden drei wesentliche Ansätze unterschieden. In [Han+12] wird die Punkt-zu-Punkt (Point-To-Point (PTP)) Bewegungsbahn über eine B-Spline Funktion definiert. Entscheidender Nachteil dabei ist die Übertragung der Funktion auf eine industrielle Robotersteuerung aufgrund einer fehlenden Entwickler Schnittstelle zur Vorgabe von Sollwerten [Egg19, S. 55 f.]. Eine praktikablere Umsetzung bietet die Definition und Verschiebung von zusätzlichen Via-Punkten [SHV20, S. 261 ff.]. Nach der Identifikation einer energieoptimierten Gelenkwinkel-Definition wird der Via-Punkt auf die Robotersteuerung übertragen. Der vom Hersteller der Robotersteuerungen implementierte Bahnplanungsansatz wird als Blackbox angenommen. Infolgedessen sind Abweichungen der, von der Robotersteuerung berechneten Bahn gegenüber der optimierten Bewegungsbahn möglich. [Egg19] vermeidet dieses Problem durch die Berechnung der zu optimierenden Bewegungsbahn auf der Originalsteuerung in einem Software-in-the-Loop (SiL) Ansatz.

1.3 Ausgangslage

Über die Durchführung einer statistischen Versuchsplanung (Design of Experiments (DOE)) im Vorfeld der vorliegenden Ausarbeitung konnte der Energieverbrauch eines KUKA KR 270 R2700 ultra IR in einer Laborumgebung um circa 10 % gesenkt werden. Die Gelenkwinkel der Achsen zwei, drei und fünf wurden als Freiheitsgrade des DOE definiert. Der Versuch umfasst fünf hintereinander ausgeführte Bewegungsabläufe, wobei für jeden ein Via-Punkt zwischen dem Start und Zielwinkel der einzelnen Gelenkwinkel-Trajektorien festgelegt wurde. Der Via-Punkt jeder Trajektorie wurde anfänglich im Mittelpunkt zwischen Start und Zielwinkel der einzelnen Bewegungsabläufe definiert. Diese Via-Punkt Gelenkwinkel wurden mit jedem Versuch variiert. Der Gesamtumfang des Versuchsplans umfasst 20 randomisierte Konfigurationen über die drei Freiheitsgrade je Bewegungsablauf. Für jede Konfiguration wurde die Leistungsaufnahme der Antriebe in 20 Wiederholungen aufgezeichnet. Abschließend wurde für jeden Bewegungsablauf der Energieverbrauch der energieeffizientesten Via-Punkt Konfiguration mit dem Energieverbrauch der initialen Bewegungsbahn verglichen. Für die energieeffizientesten Via-Punkt Konfiguration über alle fünf Bewegungsabläufe weißt der Roboter einen um 10 % geringeren Energieverbrauch gegenüber der initialen Bewegungsbahnen auf. Kritisch wird dabei der Umfang zur Zielerreichung bewertet. Des Weiteren sind die energieeffizienten Via-Punkte nicht vorab planbar. Diese Lücke wird in der vorliegenden Bachelorarbeit geschlossen.

1.4 Zielsetzung

Die Zielsetzung der Arbeit definiert die Umsetzung einer Bahnoptimierung für einen Industrieroboter mit serieller Kinematik. Dafür ist ein in der Produktion eingesetztes Roboterprogramm durch das Hinzufügen und Verschieben von sog. Via-Punkten bezüglich der aufgenommenen Energie des Roboters bei Abfahren des Programms zu optimieren. Prämissen der Arbeit ist die Identifizierung des energieoptimierten Via-Punkts auf der Grundlage eines Modells. Des Weiteren wird aufgrund strenger Taktzeitanforderungen definiert, dass die Bewegungsdauer der optimierten Bewegungsbahn nicht signifikant höher ausfallen darf als die der initialen Bewegungsbahn. Die vorliegende Arbeit verfolgt

nicht das Ziel, dieselbe Modellgenauigkeit durch eine Parameteridentifikation abzubilden wie [Pel+11] und [GRN13]. Von einer Modellierung des Betriebsverhaltens der Antriebe, siehe [Egg19] und [Zia+17] wird ebenfalls Abstand genommen. Vielmehr liegt der Fokus darauf die Grundlagen der mechanischen Modellbildung und Optimierung darzulegen, welche in der o. g. Literatur als bekannt vorausgesetzt sind und nicht näher skizziert werden.

1.5 Geplantes Vorgehen

Zur Umsetzung dieser Anforderung erfolgt im zweiten Kapitel die Beschreibung der Vorwärtskinematik sowie eine MATLAB[®]-Implementierung des rekursiven Newton-Euler-Algorithmus, auf deren Grundlage die Dynamik des Roboters simuliert wird. Das dritte Kapitel beschreibt die Bahnplanung. Basierend auf dieser Definition wird im vierten Kapitel das dynamische Modell validiert. Im fünften Kapitel erfolgt eine Beschreibung des Optimierungsproblems und die Implementierung der Via-Punkt basierten Bahnoptimierung. Anschließend werden die Ergebnisse validiert und bewertet. Die Arbeit schließt mit einer Zusammenfassung der Ergebnisse und skizziert den Ausblick der Untersuchung. Die Zeitplanung der praktischen Bearbeitung ist im Anhang 8 hinterlegt.

2 Mechanische Modellbildung

Gegenstand der Modellbildung ist die Definition von Bewegungsgleichungen zur Simulation der Roboterdynamik [Sic+09, S. 247]. Damit soll die aufgenommene Energie des Roboters entlang einer Bewegungsbahn bestimmt werden, ohne dass das reale System angesteuert werden muss. Zunächst erfolgt eine Beschreibung der Vorwärtskinematik mithilfe der Denavit-Hartenberg (DH) Konvention. Aufbauend darauf wird die Dynamik des Systems mithilfe des Rekursiven-Newton-Euler-Algorithmus (RNEA) formuliert. Abschließend werden Annahmen und Vernachlässigungen im Kontext der Modellbildung betrachtet.

2.1 Vorwärtskinematik

Über die Vorwärtskinematik erfolgt eine geometrische Beschreibung der Endeffektor Bewegung des Roboters entlang der kinematischen Kette¹. Entsprechend der Gelenkvariablen wird dabei die Position und Orientierung (Pose) des Endeffektors im Basis-Koordinatensystem berechnet. Ausgehend vom Basis-Koordinatensystem werden homogene Transformationsmatrizen $\mathbf{A}(i) = \mathbf{T}_i^{i-1} \in \text{SE}(3)$ ² zwischen zwei aufeinanderfolgenden Koordinatensystemen $\text{KS}\{i-1\}$ und $\text{KS}\{i\}$ aufgestellt. Die Transformationsmatrizen sind abhängig von der Konfiguration der Gelenkvariablen q_i des Roboters.

$$\mathbf{A}_i = \mathbf{A}_i(q_i) \quad (2.1)$$

Eine Repräsentation der Endeffektor-Pose \mathbf{H} im Basis-Koordinatensystem wird über eine Multiplikation der homogenen Transformationsmatrizen entlang der kinematischen

¹siehe Anhang B.1

²siehe Anhang B.2

Kette erreicht.

$$\mathbf{H} = \mathbf{T}_n^0 = \mathbf{A}_1(q_1) \dots \mathbf{A}_n(q_n) \quad (2.2)$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{R}_n^0 & \mathbf{o}_n^0 \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (2.3)$$

Der Spaltenvektor \mathbf{o}_n^0 repräsentiert die Koordinaten des Ursprungs des Endeffektor-Koordinatensystems im Basis-Koordinatensystem. Die 3×3 Matrix \mathbf{R}_n^0 entspricht der Rotation des Endeffektor-Koordinatensystems gegenüber dem Basis-Koordinatensystem. [SHV20, S. 75 ff.]

Denavit-Hartenberg Konvention

Eine Methode zur Bestimmung der homogenen Transformationsmatrizen ist die nach dem Physiker Jacques Denavit und dem Ingenieur Richard Hartenberg benannte DH Konvention. Diese besagt, dass jede homogene Transformation \mathbf{A}_i als Produkt vier nacheinander ausgeführter elementarer Transformationen, gemäß 2.4 ausgedrückt werden kann.

$$\begin{aligned} \mathbf{A}_i &= \text{Rot}_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} \text{Rot}_{x,\alpha_i} \quad (2.4) \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

$$c = \cos$$

$$s = \sin$$

Die Parameter entsprechen dem Gelenkwinkel θ_i , dem Gelenkabstand d_i , der Armelementlänge a_i und der Verwindung α_i , wobei d_i , a_i und α_i konstant sind. [SHV20, S. 79]

Festlegung der Koordinatensysteme

Abbildung 2.1 zeigt den zu untersuchenden KUKA KR210 R2700-2 Industrieroboter. Die Bezeichnung der Maschine gibt Auskunft über ihre technischen Daten. Die Nenn-Traglast beträgt 210 kg. Die maximale Reichweite des Roboters liegt bei 2701 mm. Weitere Informationen sind dem Datenblatt, siehe Anhang C.1, zu entnehmen.

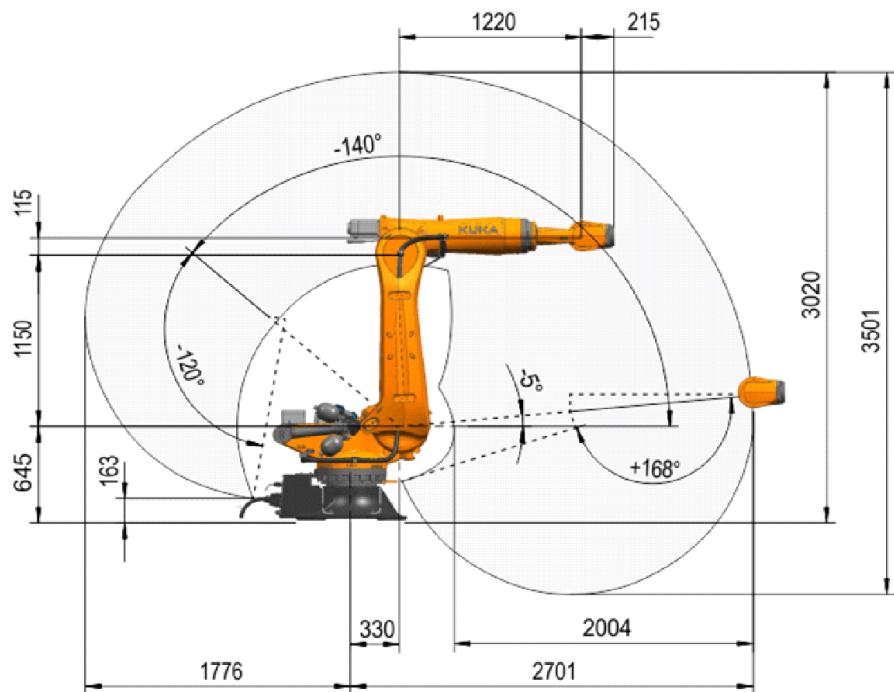


Abbildung 2.1: KUKA KR210 R2700-2

Im ersten Schritt werden die Achsen z_i , $i = 0, \dots, n - 1$, $n = 6$, siehe Abbildung 2.2, definiert. Im zweiten Schritt wird das Basiskoordinatensystem 0 festgelegt, wobei der Ursprung \mathbf{o}_0 auf der Achse z_0 platziert wird. Die Definition des ersten Koordinatensystems entspricht dem vom Hersteller festgelegten ROBROOT-Koordinatensystem. Dies erlaubt eine Überprüfung der Vorwärtskinematik mithilfe der, von der KUKA Robotersteuerung (KUKA Robot Control (KR C)) berechneten kartesischen Koordinaten. Des Weiteren

werden, siehe Abbildung 2.3, aufsteigend die Koordinatensysteme $KS\{i\}$, $i = 1, \dots, n - 1$ basierend auf $KS\{i - 1\}$, sowie das rot hervorgehobene Endeffektor-Koordinatensystem $0_6x_6y_6z_6$ platziert. Ein Werkzeug ist am Versuchsroboter nicht verbaut. Das Endeffektor-KS liegt auf dem Flansch.

Es werden dabei folgende Regeln berücksichtigt.

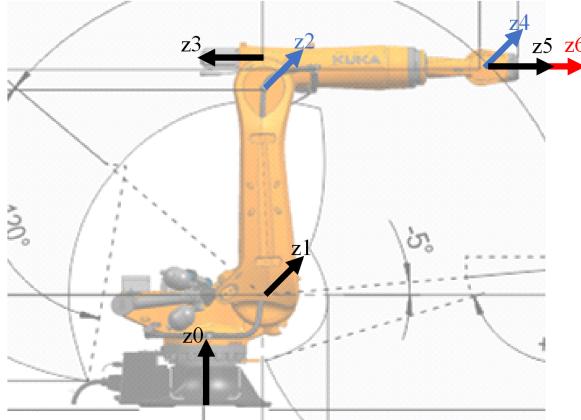


Abbildung 2.2: Festlegung der z-Achsen

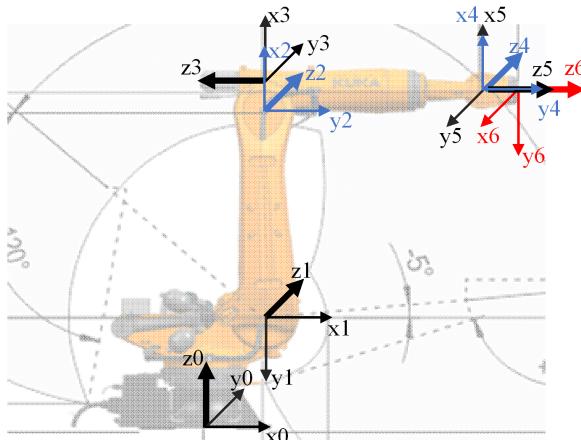


Abbildung 2.3: DH-Konvention - Festlegung der Koordinatensysteme

- Falls sich z_i und z_{i-1} schneiden, ist der Ursprung \mathbf{o}_i in den Schnittpunkt zu legen. Die Achse x_i wird rechtwinklig zur $z_{i-1} - z_i$ -Ebene definiert.³
- Falls z_i und z_{i-1} parallel sind, wird der Ursprung \mathbf{o}_i auf die Achse z_i gelegt. Die

³gilt für $KS\{4\}$ und $KS\{5\}$

Achse x_i wird parallel zur gemeinsamen Normale von z_i und z_{i-1} im Punkt \mathbf{o}_i definiert.⁴

- Falls z_i und z_{i-1} nicht koplanar sind, wird der Ursprung \mathbf{o}_i auf den Schnittpunkt der Achse z_i mit der gemeinsamen Normalen von z_i und z_{i-1} gelegt. Die Achse x_i ist die gemeinsame Normale von z_i und z_{i-1} .⁵
- die Achse y_i vervollständigt das rechtshändige, kartesische Koordinatensystem

Im nächsten Schritt erfolgt die Identifizierung der DH-Parameter. θ_i entspricht der Rotation um z_{i-1} , sodass $x_{i-1} \parallel x_i$. Der Parameter d_i drückt die Translation entlang z_{i-1} bis zum Schnittpunkt S von z_{i-1} und x_i aus. Der Parameter a_i entspricht der Translation entlang x_i , um den Schnittpunkt S in den Ursprung \mathbf{o}_i zu überführen. Abschließend ist α_i die Rotation um x_i , zur Überführung von z_{i-1} in z_i . Aus den Elementartransformationen, siehe Gleichung 2.4, wird die Gesamttransformationen entsprechend 2.2 berechnet.

Tabelle 2.1 listet die identifizierten DH-Parameter.

Tabelle 2.1: Denavit-Hartenberg Parameter

Gelenk	θ [°]	d [m]	a [m]	α [°]
1	0	0,645	0,330	-90
2	-90	0	1,150	0
3	0	0	0,115	90
4	0	-1,220	0	-90
5	0	0	0	-90
6	90	0,215	0	0

Der Roboterhersteller definiert die Drehrichtung der Achse z_1 invertiert gegenüber der o. g. Festlegung. Folglich werden die vor der Robotersteuerung bereitgestellten Winkel des ersten Gelenks mit -1 multipliziert, bevor eine Weiterverarbeitung im Modell erfolgt. Des Weiteren werden vom dritten Gelenkwinkel 90° subtrahiert, bevor der Wert dem Modell übergeben wird. Die Vorwärtskinematik wurde über drei Posen am realen System verifiziert. Dabei ist festzustellen, dass die Robotersteuerung eine Elastizität der Verbindungsglieder berücksichtigt, wohingegen in der Modellbeschreibung von starren Körpern ausgegangen wird. Die Implementierung der DH-Transformation ist im Anhang D.1 aufgeführt.

⁴gilt für KS{2}

⁵gilt für KS{1} und KS{3}

2.2 Geschwindigkeits-Kinematik

Die Geschwindigkeits-Kinematik definiert die lineare Endeffektor-Geschwindigkeit und die Endeffektor-Winkelgeschwindigkeit in Abhängigkeit der Gelenkwinkelgeschwindigkeiten des Roboters. Eine mathematische Beschreibung des Zusammenhangs erfolgt über die Jacobi-Matrix der Vorwärtsskinematik [SHV20, S. 101]. Grundlegende Definitionen zur Aufstellung der Jacobi-Matrizen \mathbf{J}_v und \mathbf{J}_ω werden im Anhang B.3 betrachtet.

2.3 Dynamik

Ziel des Kapitels ist die Berechnung der aufzubringenden Momente in den sechs Gelenken des Roboters bei Ansteuerung entlang einer Bewegungsbahn. Der dynamische Zusammenhang zwischen Gelenkkräften und Gelenkbewegungen wird zunächst formal mithilfe der Euler-Lagrange Gleichungen formuliert. Anschließend folgt die numerisch effiziente Implementierung über den RNEA [Sic+09, S. 247].

2.3.1 Euler-Lagrange Gleichungen

Die Lagrange Funktion L eines mechanischen Systems entspricht der Differenz der kinetischen Energie K und der potentiellen Energie U [SHV20, S. 175].

$$L = K - U \quad (2.5)$$

Die Euler-Lagrange-Gleichung wird über den folgenden Ausdruck beschrieben. Dabei entspricht die Ordnung n des Systems der Anzahl generalisierter Koordinaten⁶, bzw. im Anwendungsfall den sechs rotatorischen Gelenkkoordinaten $(\theta_1, \dots, \theta_6)$. Die Größe τ repräsentiert generalisierte Kräfte⁷. Im Anwendungsfall werden die Antriebsmomente in

⁶Voneinander unabhängige Koordinaten, welche den aktuellen Systemzustand vollständig beschreiben [Eng08]

⁷Kräfte und Momente

den Gelenken betrachtet.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau_i ; \quad i = 1, \dots, n \quad (2.6)$$

$$\frac{\partial L}{\partial \dot{\mathbf{q}}} = \frac{\partial K}{\partial \dot{\mathbf{q}}} \quad (2.7)$$

$$\frac{\partial L}{\partial \mathbf{q}} = -\frac{\partial U}{\partial \mathbf{q}} \quad (2.8)$$

Die kinetischen Energie des Roboters entspricht der Gleichung 2.9

$$K = \frac{1}{2} \dot{\mathbf{q}}^T \left[\sum_{i=1}^n (m_i \mathbf{J}_{v_i}(\mathbf{q})^T \mathbf{J}_{v_i}(\mathbf{q}) + \mathbf{J}_{\omega_i}(\mathbf{q})^T \mathbf{R}_i^0(\mathbf{q}) \mathbf{I}_i^i \mathbf{R}_i^0(\mathbf{q})^T \mathbf{J}_{\omega_i}(\mathbf{q})) \right] \dot{\mathbf{q}} \quad (2.9)$$

Der Ausdruck in der Klammer wird als Massenmatrix $\mathbf{M}(\mathbf{q})$ bezeichnet. Die Massen m_i wurden über ein Computer Aided Design- (CAD) Modell, welches vom Hersteller bereitgestellt wird, identifiziert. Die Trägheitstensoren \mathbf{I}_i^i , bezogen auf den Masseschwerpunkt des Verbindungsglieds i , sind relativ zu den Gelenkkoordinatensystemen orientiert und damit für alle Konfigurationen des Roboters konstant. Über das CAD-Modell werden die Trägheitstensoren \mathbf{I}_i^0 relativ zum KS{0} bereitgestellt. Die gewünschte Umrechnung erfolgt mithilfe der Ähnlichkeitstransformation 2.10.

$$\mathbf{I}_i^i = \mathbf{R}_i^{0T} \mathbf{I}_i^0 \mathbf{R}_i^0 \quad (2.10)$$

Die potentielle Energie wird über den Ausdruck 2.11 berechnet.

$$U = \sum_{i=1}^n m_i \mathbf{g}^T \mathbf{r}_{C_i}^0 \quad (2.11)$$

Der Vektor $\mathbf{g} = [0 \ 0 \ -9,81]^T \frac{\text{m}}{\text{s}^2}$ ist die Fallbeschleunigung, ausgedrückt in KS{0}. Der Vektor $\mathbf{r}_{C_i}^0$ definiert die Lage des Masseschwerpunkts vom Verbindungsglied i im KS{0}. Für die Herleitung der Bewegungsgleichungen 2.12 wird auf die Literatur [SHV20, S. 180 ff.] verwiesen. Hierbei ist ersichtlich, dass die numerische Bestimmung von $\boldsymbol{\tau}$ aufgrund der partiellen Ableitungen von K und U insbesondere in den letzten Gliedern der kinematischen Kette einen hohen Berechnungsaufwand erfordert, weshalb das Modell über

den RNEA implementiert ist.

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (2.12)$$

Der Term $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ berücksichtigt die Einflüsse der Corioliskraft und Zentrifugalkraft⁸.

Der Vektor $\mathbf{g}(\mathbf{q})$ berücksichtigt die Gravitation. [SHV20, S. 180 ff.]

2.3.2 Rekursiver-Newton-Euler-Algorithmus

Der Ansatz sieht vor, jedes Verbindungsglied der kinematischen Kette und die auf darauf wirkenden Kräfte $-\mathbf{f}_{i+1}$ und \mathbf{f}_i im Einzelfall zu betrachten. Ziel ist die Berechnung der im zugehörigen Gelenk wirkenden Drehmomente $\boldsymbol{\tau}$. Zur Berechnung der Drehmomente $\boldsymbol{\tau}$ in den Gelenken wird der Vektor $\boldsymbol{\mu}$ eingeführt. $\boldsymbol{\mu}_i$ hat die Dimension $3 \times n$ und repräsentiert das Drehmoment im Gelenk i des Verbindungsglieds i . $\boldsymbol{\tau}_i$ entspricht dem Betrag des Drehmoments je Gelenk i . Abbildung 2.4 zeigt die, zu berücksichtigenden Größen am Beispiel eines generischen Verbindungsglieds i .

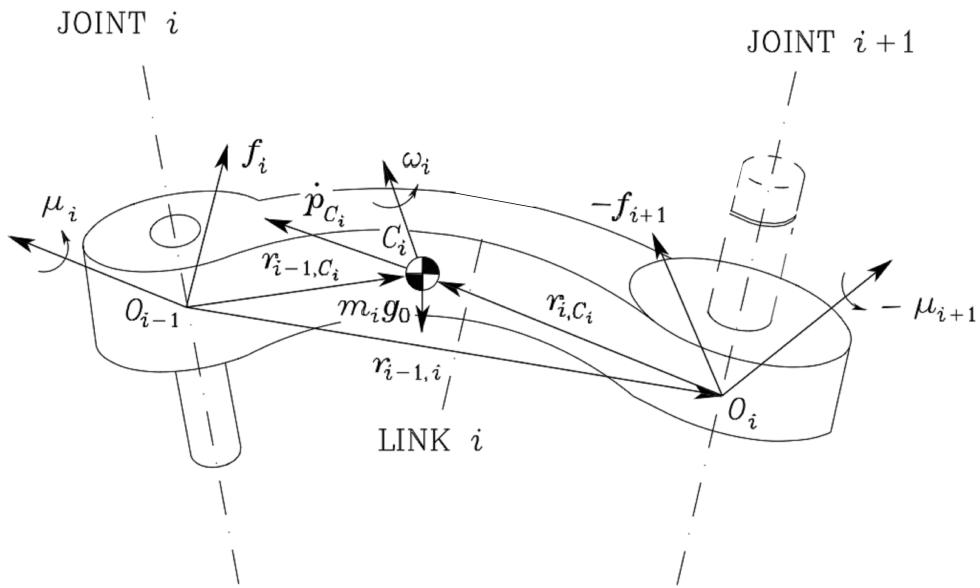


Abbildung 2.4: Generisches Verbindungsglied [Sic+09, S. 283]

⁸Eine detaillierte Betrachtung von Scheinkräften in rotierenden Bezugssystemen erfolgt in [RS16, S. 159]

Kräfte und Momente

- \mathbf{f}_i - Kraft, die vom Verbindungsglied $i - 1$ auf das Verbindungsglied i ausgeübt wird
- $-\mathbf{f}_{i+1}$ - Kraft, die vom Verbindungsglied $i + 1$ auf das Verbindungsglied i ausgeübt wird
- $\boldsymbol{\mu}_i$ - Drehmoment, welches vom Verbindungsglied $i - 1$ auf das Verbindungsglied i ausgeübt wird
- $-\boldsymbol{\mu}_{i+1}$ - Drehmoment, welches vom Verbindungsglied $i + 1$ auf das Verbindungsglied i ausgeübt wird

In der MATLAB[®] Implementierung des RNEA werden alle Größen des Verbindungsglieds i auf das fest damit verbundenen KS $\{i\}$ bezogen. Daraus resultieren die Kräfte und Momente in der Notation \mathbf{f}_i^i und $\boldsymbol{\mu}_i^i$. Nachfolgend werden die Parameter der Verbindungsglieder eingeführt. Die Werte der Systemparameter sind im Anhang C.2 aufgeführt.

Parameter

- m_i - Masse des Verbindungsglieds i
- \mathbf{I}_i^i - Trägheitstensor des Verbindungsglieds i im fest damit verbundenen KS $\{i\}$
- \mathbf{r}_{i-1,C_i}^i - Vektor vom Ursprung des KS $\{i - 1\}$ zum Masseschwerpunkt C_i des Verbindungsglieds i ausgedrückt in KS $\{i\}$
- \mathbf{r}_{i,C_i}^i - Vektor vom Ursprung des KS $\{i\}$ zum Masseschwerpunkt C_i ausgedrückt in KS $\{i\}$
- $\mathbf{r}_{i-1,i}^i$ - Vektor vom Ursprung des KS $\{i - 1\}$ zum Ursprung des KS $\{i\}$ ausgedrückt in KS $\{i\}$
- i_i - Übersetzungsverhältnis des Getriebes i

Geschwindigkeiten und Beschleunigungen

- $\dot{\mathbf{p}}_{C_i}$ - Lineare Geschwindigkeit im Masseschwerpunkt C_i
- $\dot{\mathbf{p}}_i$ - Lineare Geschwindigkeit im Ursprung des KS $\{i\}$
- $\boldsymbol{\omega}_i$ - Winkelgeschwindigkeit des Verbindungsglieds i
- $\ddot{\mathbf{p}}_{C_i}$ - Lineare Beschleunigung im Masseschwerpunkt C_i

- $\ddot{\mathbf{p}}_i$ - Lineare Beschleunigung im Ursprung des KS*{i}*
- $\dot{\boldsymbol{\omega}}_i$ - Winkelbeschleunigung des Verbindungsglieds *i*

Mit Kenntnis der Größen $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ und unter Festlegung der Startbedingungen $\boldsymbol{\omega}_0 = 0$, $\ddot{\mathbf{p}}_0 - g = [0 \ 0 \ -9.81]^T$ und $\dot{\boldsymbol{\omega}}_0 = 0$ erfolgt sukzessive die Berechnung der Geschwindigkeiten und Beschleunigungen für jedes Verbindungsglied, beginnend mit dem ersten. Die Gleichungen 2.13, 2.14, 2.15 und 2.16 sind der Literatur [Sic+09, S. 287 f.] entnommen. Der betrachtete KUKA KR210 R2700-2 weist eine RRR-Kinematik⁹ mit Zentralhand¹⁰ auf weshalb eine Beschreibung für rotatorische Gelenke gewählte wurde.

$$\boldsymbol{\omega}_i^i = \mathbf{R}_i^{i-1T} \left(\boldsymbol{\omega}_{i-1}^{i-1} + \dot{\theta}_i \mathbf{z}_0 \right) \quad (2.13)$$

$$\dot{\boldsymbol{\omega}}_i^i = \mathbf{R}_i^{i-1T} \left(\dot{\boldsymbol{\omega}}_{i-1}^{i-1} + \ddot{\theta}_i \mathbf{z}_0 + \dot{\theta}_i \boldsymbol{\omega}_{i-1}^{i-1} \times \mathbf{z}_0 \right) \quad (2.14)$$

$$\ddot{\mathbf{p}}_i^i = \mathbf{R}_i^{i-1T} \ddot{\mathbf{p}}_{i-1}^{i-1} + \dot{\boldsymbol{\omega}}_i^i \times \mathbf{r}_{i-1,i}^i + \boldsymbol{\omega}_i^i \times (\boldsymbol{\omega}_i^i \times \mathbf{r}_{i-1,i}^i) \quad (2.15)$$

$$\ddot{\mathbf{p}}_{C_i}^i = \ddot{\mathbf{p}}_i^i + \dot{\boldsymbol{\omega}}_i^i \times \mathbf{r}_{i,C_i}^i + \boldsymbol{\omega}_i^i \times (\boldsymbol{\omega}_i^i \times \mathbf{r}_{i,C_i}^i) \quad (2.16)$$

Im nächsten Schritt erfolgt die rekursive Berechnung der Kräfte und Drehmomente. Der Algorithmus beginnt mit der Berechnung der Größen für das sechste Gelenk, gemäß der Gleichungen 2.17, 2.18 und 2.19. \mathbf{f}_{n+1}^{n+1} und $\boldsymbol{\mu}_{n+1}^{n+1}$ werden zu 0 angenommen.

$$\mathbf{f}_i^i = \mathbf{R}_{i+1}^i \mathbf{f}_{i+1}^{i+1} + m_i \ddot{\mathbf{p}}_{C_i}^i \quad (2.17)$$

$$\boldsymbol{\mu}_i^i = -\mathbf{f}_i^i \times (\mathbf{r}_{i-1,i}^i + \mathbf{r}_{i,C_i}^i) + \mathbf{R}_{i+1}^i \boldsymbol{\mu}_{i+1}^{i+1} + \mathbf{R}_{i+1}^i \mathbf{f}_{i+1}^{i+1} \times \mathbf{r}_{i,C_i}^i + \mathbf{I}_i^i \dot{\boldsymbol{\omega}}_i^i + \boldsymbol{\omega}_i^i \times (\mathbf{I}_i^i \boldsymbol{\omega}_i^i) \quad (2.18)$$

$$\tau_i = i_i \boldsymbol{\mu}_i^{iT} \mathbf{R}_i^{i-1T} [0 \ 0 \ 1]^T \quad (2.19)$$

Aufgrund der Lagerausführung des ersten Gelenks wird angenommen, dass das nachfolgende Drehmoment nicht auf das erste Gelenk übertragen wird, sodass der Term $\mathbf{R}_{i+1}^i \boldsymbol{\mu}_{i+1}^{i+1}$ hierbei entfällt. Aus den identifizierten Drehmomenten τ_i erfolgt durch Multiplikation mit den Winkelgeschwindigkeiten $\dot{\boldsymbol{\omega}}_i^i$ die Berechnung der mechanischen Leistung P_{mech} . Die Implementierung des Modells ist dem Anhang D.3 zu entnehmen. [Sic+09, S. 282 ff.]

⁹Drei rotatorische Gelenke

¹⁰Funktionsumfang Rollen, Nicken, Gieren

2.4 Modellannahmen und Vernachlässigungen

Bei der Entwicklung des mechanischen Modells zur Simulation der Roboterdynamik wurden Annahmen getroffen um die Komplexität zu reduzieren. Ein wesentlicher Einfluss ist der Vernachlässigung des hydraulischen Gewichtsausgleichs am zweiten Gelenk zuzuschreiben. Der Gewichtsausgleich ist aus einem Hydraulikzylinder mit Membranspeicher aufgebaut und wirkt dem am zweiten Gelenk auftretenden Drehmoment entgegen. Im Modell wurde dieser aufgrund fehlender Konstruktionsdaten bezüglich des Kolbendurchmessers und Kolbenstangendurchmessers nicht berücksichtigt. Fehlende Daten sind auch der Grund für die Auslassung des Gewichts und der Steifigkeit der Schlauchpakete. Das Gewicht der Elektromotoren¹¹ wird ebenfalls vernachlässigt. Aufgrund des Verhältnisses zwischen der Masse der Motoren und dem Gewicht der Achsen wird diese Vernachlässigung toleriert. Am Beispiel des dritten Verbindungsglieds beträgt das Verhältnis 1:15. Die gleiche Schlussfolgerung gilt für die Massenträgheit der Motoren¹². [Sic+09, S. 287 f.] berücksichtigt bei der Berechnung der Drehmomente zudem die Einflüsse der viskosen und der coloumbschen Reibung. Die entsprechenden Reibungskoeffizienten werden im Modell nicht berücksichtigt. Neben der Berechnung der mechanisch aufzuwendenden Leistung erfordert eine hohe Modellgenauigkeit die Betrachtung der elektrischen Komponenten in Abbildung 2.5. Bei den Motoren handelt es sich um dreiphasige permanentmagneterregte Synchronmaschinen (PMSM). Die Ansteuerung erfolgt über Frequenzumrichter. Im Ersatzschaltbild (ESB) sind alle Antriebe über einen gemeinsamen Gleichrichter vom Netz auf einen Gleichspannungszwischenkreis gekoppelt. PMSM können sowohl motorisch als auch generatorisch betrieben werden. Für die Berechnung des Energieverbrauchs in der vorliegenden Arbeit wird angenommen, dass die generatorisch umgewandelte Energie über einen Bremswiderstand als Wärme abgeführt wird. Von einer Spannungs-Schwellwert gesteuerten Zu- und Abschaltung des Bremswiderstands wird abgesehen. [Egg19, S. 21 ff.]

¹¹Gemeint ist der Stator

¹²Gemeint ist der Rotor

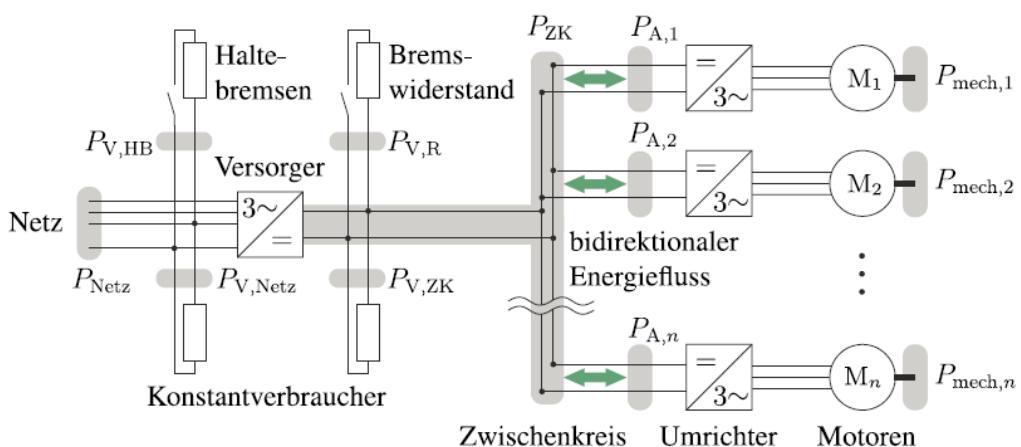


Abbildung 2.5: Ersatzschaltbild der aufgenommenen Netzleistung [Egg19, S. 20]

3 Implementierung der Via-Punkt Trajektorie

Die Berechnung der Bewegungsbahn erfolgt für alle sechs Gelenkwinkel q_i nach demselben Ansatz, einem Polynom sechster Ordnung.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 + a_6 t^6 \quad (3.1)$$

$$\dot{q}(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 + 6a_6 t^5 \quad (3.2)$$

$$\ddot{q}(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 + 30a_6 t^4 \quad (3.3)$$

Maßgeblich für die Ordnung des Polynoms ist die Anzahl von sieben Nebenbedingungen. Definiert sind die Startwinkel $q_i(t_s)$, die Anfangsgeschwindigkeiten $\dot{q}_i(t_s) = 0$, die Anfangsbeschleunigungen $\ddot{q}_i(t_s) = 0$, die Via-Punkte $q_i(t_v)$ sowie die Winkel im Zielpunkt $q_i(t_e)$, die Endgeschwindigkeiten $\dot{q}_i(t_e) = 0$ und die Endbeschleunigungen $\ddot{q}_i(t_e) = 0$. Darüber hinaus sind der Startzeitpunkt t_s , der Zeitpunkt für das Erreichen des Via-Punkts, sowie die Dauer der Bewegung über den Endzeitpunkt t_e vorgegeben. Die Bestimmung der Parameter a_0, \dots, a_6 erfolgt durch Lösen des linearen Gleichungssystems 3.4. Die Implementierung in MATLAB® ist im Anhang D.4 hinterlegt.

$$\begin{bmatrix} 1 & t_s & t_s^2 & t_s^3 & t_s^4 & t_s^5 & t_s^6 \\ 0 & t_s & 2t_s & 3t_s^2 & 4t_s^3 & 5t_s^4 & 6t_s^5 \\ 0 & 0 & 2 & 6t_s & 12t_s^2 & 20t_s^3 & 30t_s^4 \\ 1 & t_v & t_v^2 & t_v^3 & t_v^4 & t_v^5 & t_v^6 \\ 1 & t_e & t_e^2 & t_e^3 & t_e^4 & t_e^5 & t_e^6 \\ 0 & 1 & 2t_e & 3t_e^2 & 4t_e^3 & 5t_e^4 & 6t_e^5 \\ 0 & 0 & 2 & 6t_e & 12t_e^2 & 20t_e^3 & 30t_e^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} q_s \\ \dot{q}_s \\ \ddot{q}_s \\ q_v \\ q_e \\ \dot{q}_e \\ \ddot{q}_e \end{bmatrix} \quad (3.4)$$

Die Bestimmung der Roboterdynamik setzt voraus, dass der simulierte Gelenkwinkel $q(t)$ und die zeitlichen Verläufe $\dot{q}(t)$, $\ddot{q}(t)$ die, von der KUKA Robotersteuerung berechnete Bahnverläufen möglichst genau approximieren. Hierzu wird eine Bewegung aus dem Programm Kleben-Seitenwand, welches für den Roboter bereits auf Kollisionsfreiheit validiert ist, getestet. Die Programmierung erfolgte in der Bewegungsart PTP und sieht vor, den Roboter vom letzten Prozesspunkt auf seine Grundstellungsposition (Home) zu verfahren. Tabelle 3.1 definiert die vorgegebenen Winkel. Die MATLAB®-Implementierung der Bewegung ist im Anhang D.5 hinterlegt.

Tabelle 3.1: Winkeldefinition der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Startwert Gelenkwinkel	Zielwert Gelenkwinkel	Via-Punkte
$q_{s,1} = -53,8$	$q_{e,1} = -7,6^\circ$	$q_{v,1} = \frac{q_{s,1} + q_{e,1}}{2}$
$q_{s,2} = -70,3^\circ$	$q_{e,2} = -119,3^\circ$	$q_{v,2} = \frac{q_{s,2} + q_{e,2}}{2}$
$q_{s,3} = 98,8^\circ$	$q_{e,3} = 88,5^\circ$	$q_{v,3} = \frac{q_{s,3} + q_{e,3}}{2}$
$q_{s,4} = -69,9^\circ$	$q_{e,4} = 10,3^\circ$	$q_{v,4} = \frac{q_{s,4} + q_{e,4}}{2}$
$q_{s,5} = -58,7^\circ$	$q_{e,5} = 32,4^\circ$	$q_{v,5} = \frac{q_{s,5} + q_{e,5}}{2}$
$q_{s,6} = 55,7^\circ$	$q_{e,6} = -10,2^\circ$	$q_{v,6} = \frac{q_{s,6} + q_{e,6}}{2}$

Abbildung 3.1 visualisiert den simulierten Verlauf der Gelenkwinkel. Abbildung 3.2 zeigt den zeitlichen Verlauf der Gelenkwinkel am realen System. Die Färbung der simulierten und aufgezeichneten Daten sind im weiteren Verlauf der Arbeit konsistent. Aus Gründen der Darstellung wird auf eine Legende in den Abbildungen der aufgezeichneten Daten verzichtet. Diese werden des Weiteren in den Abbildungen als gemessene Daten bezeichnet. Die Werte sind an der Robotersteuerung mit einem einem Zeitabstand $\Delta t = 4$ ms abgetastet. Zur Visualisierung der Winkelgeschwindigkeit erfolgt zunächst die Bildung des Differenzenquotienten für zwei benachbarte Abtastwerte. Durch eine anschließende Tiefpassfilterung (TP-Filterung) wird der zeitliche Verlauf geglättet, siehe Abbildung 3.4. Die Abtastfrequenz beträgt 250 Hz. Die Eckfrequenz des Butterworth-Filters wird auf 10 Hz definiert. Die Daten der Winkelbeschleunigung werden für dasselbe Vorgehen durch eine Anwendung des zentralen Differenzenquotienten für die zweite Ableitung berechnet. Der zeitliche Verlauf der Gelenkwinkel ist näherungsweise identisch. Da der

3 Implementierung der Via-Punkt Trajektorie

Via-Punkt in der Simulation im Mittel zwischen den Start- und Zielpunkten und damit auf der Originalbahn definiert ist, wurde die Bewegung am realen Roboter ohne Definition des initialen Via-Punktes gefahren. Der Verlauf der Beschleunigung ist detaillierter zu betrachten. Die Maximalwerte der Winkelbeschleunigung aller sechs Gelenke fallen in der simulierten Bewegung während den ersten zwei Beschleunigungsphasen circa 10-20 % höher aus als die berechneten Beschleunigungen des realen Systems. Mithilfe der Originaldaten vor Anwendung der TP-Filterung wurde verifiziert, dass der Filteransatz keine Auswirkung auf die Maxima nimmt. Während der dritten und vierten Beschleunigungsphase sind die Maximalbeschleunigungen in den ersten drei Gelenken nahezu identisch. Aufgrund der hohen Massen der ersten drei Verbindungsglieder bzw. Armelemente sind diese Werte von hoher Relevanz. Qualitativ ist im Beschleunigungsverlauf des realen Systems ein höherer Ruck beim Anfahren und Abbremsen erkennbar. Der Nulldurchgang der Beschleunigung und damit der Zeitpunkt der maximalen Geschwindigkeit¹ ist in der Simulation ca. 0,13 s nach links verschoben. In der Schlussfolgerung wird die Bewegungsbahn ausreichend genau simuliert, um die Parameter $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ dem RNA zu übergeben.

¹Zeitpunkt bei Erreichen des Via-Punkts

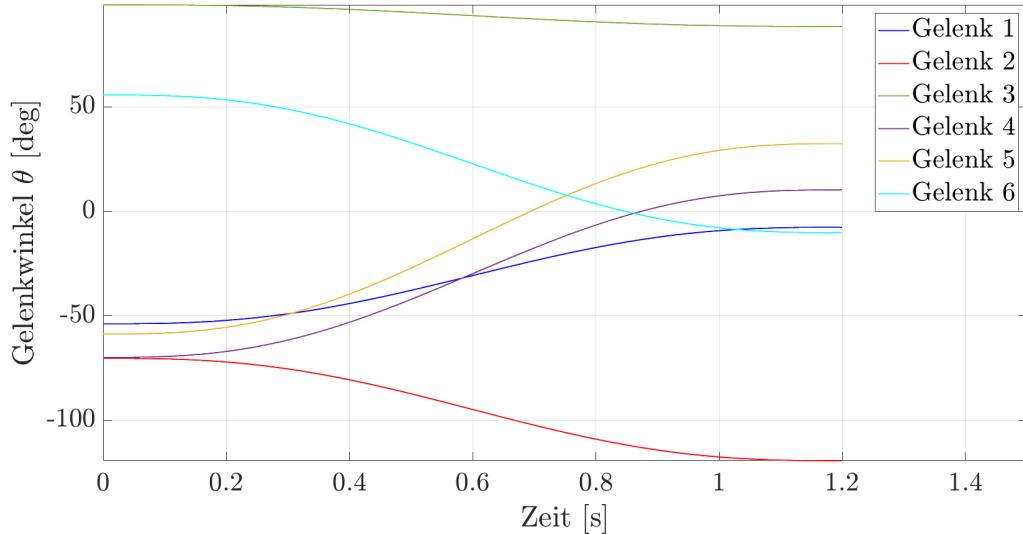


Abbildung 3.1: Simulierte Gelenkwinkel der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

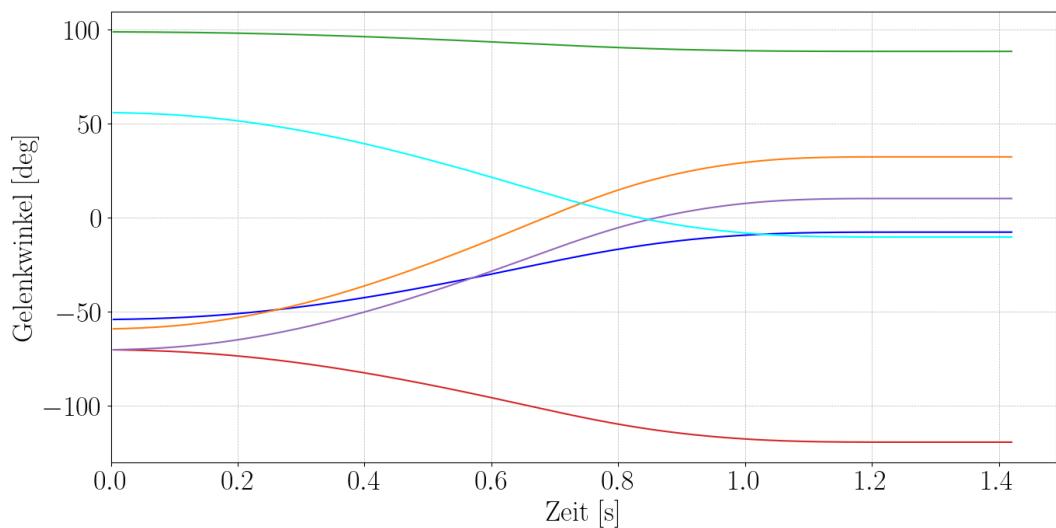


Abbildung 3.2: Gemessene Gelenkwinkel der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

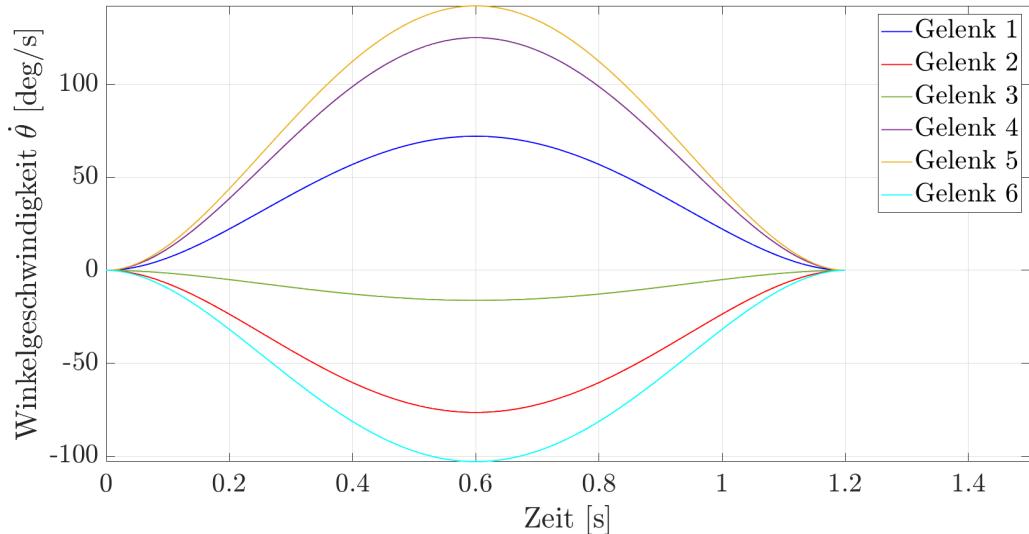


Abbildung 3.3: Simulierte Winkelgeschwindigkeit der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

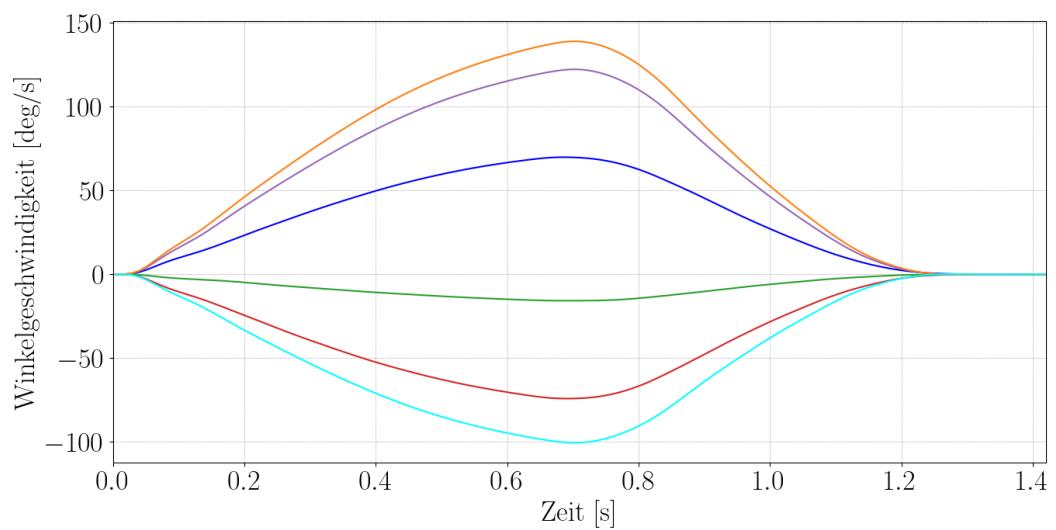


Abbildung 3.4: Gemessene Winkelgeschwindigkeit der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

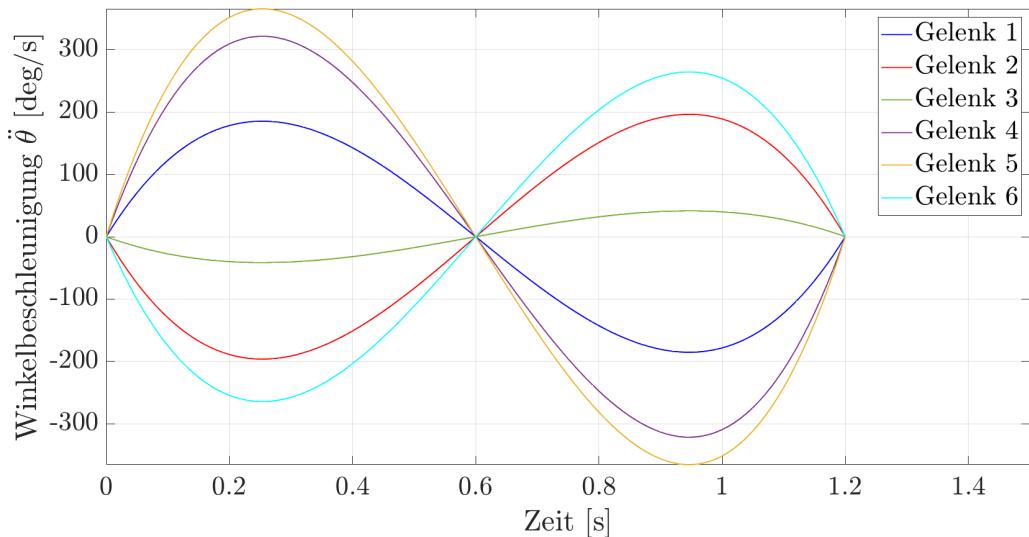


Abbildung 3.5: Simulierte Winkelbeschleunigung der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

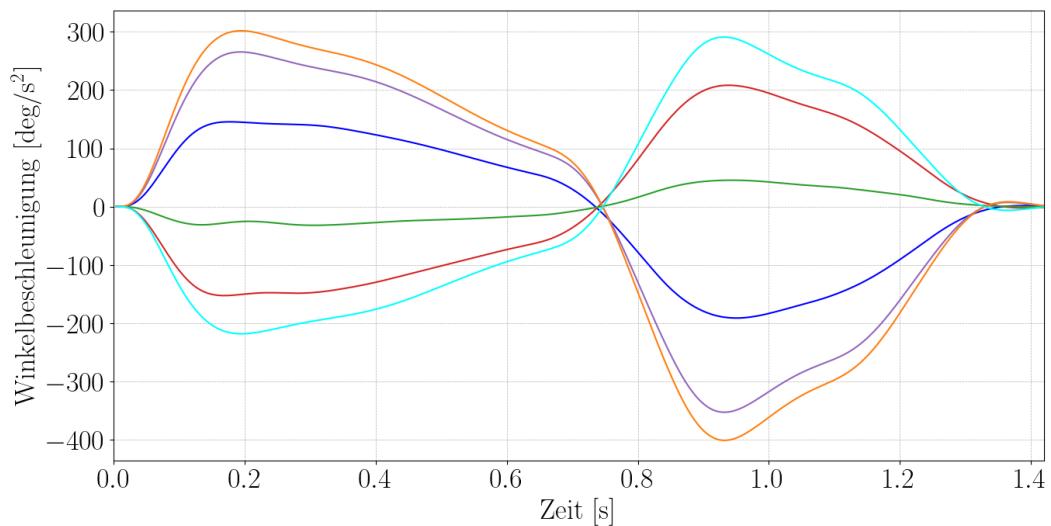


Abbildung 3.6: Gemessene Winkelbeschleunigung der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

4 Validierung des Roboterdynamik-Modells

Unter Berücksichtigung der getroffenen Annahmen wird untersucht, ob das implementierte Modell die Roboterdynamik hinreichend genau simuliert. Dazu werden die simulierten Drehmomente qualitativ und quantitativ mit den gemessenen Werten verglichen.

4.1 Messaufbau

Als Testumgebung dient eine Roboterzelle in der digitalen Anlauffabrik des Mercedes-Benz Werks Berlin. Der abgegrenzte Zellbereich ist mit Sicherheitseinrichtungen zur Gewährleistung der Personen- und Maschinensicherheit ausgestattet. Der Roboter besitzt kein Werkzeug sodass ein Kollisionsrisiko in den Versuchen, im Vergleich zu einem EE mit komplexen Geometrien, geringer ausfällt. Darüber hinaus ist eine signifikante Abweichung von der initialen Bahn durch Anwendung der Bahnoptimierung, wie in 3.1 gezeigt, nicht zu erwarten. In der PTP-Bewegungsart berechnet die KR C5 den schnellsten Weg des Tool-Center-Point (TCP) zwischen Start- und Zielpunkt [o. 23b, S. 429]. Eine stark abweichende Bahn im kartesischen Raum würde zu einem längeren Weg im Gelenkraum und damit zu einem höheren Energieverbrauch führen [Egg19, S. 59]. Der installierte KUKA KR 210 wird über eine KR C5 angesteuert, welche die KUKA.RobotSensorInterface 5.0 (RSI) Technologie unterstützt. RSI ist eine Schnittstelle zur zyklischen Kommunikation zwischen dem Industrieroboter und einem Sensorsystem. Die Robotersteuerung kommuniziert mit dem Sensorsystem über eine echtzeitfähige Netzwerkverbindung. Die

Daten werden dabei über das Ethernet UDP/IP-Protokoll¹ übertragen [o. 20, S. 11]. Der Roboter verfügt damit über eine Schnittstelle zur Aufzeichnung von Bewegungsdaten und den Messwerten interner Messeinrichtungen. Auf der Robotersteuerung ist dafür ein RSI-Kontext zu parametrieren, wobei festgelegt wird, welche Daten an eine festgelegte Ziel-IP-Adresse inklusive Port übertragen werden [o. 20, S. 43]. Der Datenaustausch findet unidirektional zwischen zwei Netzwerkteilnehmern statt. Die KR C5 hat die Funktion eines sendenden Clients. Der Empfänger² stellt den Server dar. Auf dem Empfänger wird die Ethernet Verbindung mit der definierten IP Adresse eingerichtet. Des Weiteren wird der im RSI Kontext festgelegte UDP-Port freigegeben. Der Port-Zugriff wird auf dem Windows Betriebssystem des Servers über die Programmierschnittstelle (Application Programming Interface (API)) Windows Sockets 2 (Winsock) realisiert [Whi+06]. Der entsprechende RSI-Kontext wird im Initialisierungsschritt des KUKA Robot Language KRL-Programms aktiviert [o. 20, S. 50]. Anschließend werden die Daten der erfassten Winkelverläufe und Drehmomente der einzelnen Gelenke in einem Takt von 4 ms in der American Standard Code for Information Interchange (ASCII) Darstellung an den Server übertragen. Die Daten werden in einem Textfile gespeichert und mithilfe der Skriptsprache Python in die JavaScript Object Notation (JSON) konvertiert, verarbeitet und visualisiert. Neben den Winkelverläufen werden die Winkelgeschwindigkeit, Winkelbeschleunigung sowie die mechanische Leistung in den jeweiligen Gelenken benötigt. Da Winkelgeschwindigkeit und Winkelbeschleunigung nicht explizit zur Verfügung stehen, werden sie über den Differenzenquotient, siehe Kapitel 3 gebildet. Durch Multiplikation der Winkelgeschwindigkeit mit dem Getriebe-Drehmoment wird der zeitliche Verlauf der mechanischen Leistung berechnet.

4.2 Durchführung

Die Validierung des Modells erfolgt exemplarisch für die, in der Tabelle 3.1 definierte, initiale Trajektorie aus dem Programm Kleben-Seitenwand. Die Betriebstemperatur des Roboters wird aufgrund der Vernachlässigung von Reibungseinflüssen im Modell nicht berücksichtigt. Die simulierten sowie gemessenen Verläufe der Gelenkwinkel, Winkelge-

¹User Datagram Protocol (UDP) ist ein verbindungsloses Protokoll für den Datenaustausch von Netzwerkteilnehmern [o. 20].

²Hierbei ein Notebook

schwindigkeit und Winkelbeschleunigung sind bereits im letzten Kapitel aufgeführt. Die simulierten Drehmomente werden in der Abbildung 4.2 gezeigt. Die gemessenen Werte sind in der Abbildung 4.4 dargestellt. In beiden Fällen werden die Drehmomente auf der Getriebeseite betrachtet. Dies erfolgt mit dem Ziel die Zusammenhänge zwischen Drehmoment, Winkelgeschwindigkeit und mechanischer Leistung erkennbar nachzuweisen. Eine Darstellung der Drehmomente auf der Motorseite wird vermieden. Durch eine Multiplikation mit der Getriebe-Übersetzung ergibt sich eine Änderung des Vorzeichens für die Drehmomente in den Gelenken eins, zwei, vier und fünf, wie im Anhang D.1 angegeben, so dass die Zusammenhänge nicht ad hoc erkennbar sind.

4.3 Auswertung

Die gemessenen Drehmomente werden im Zeitintervall 0 s bis 1,3 s betrachtet. Daten für den Zeitraum $t > 1,3$ s entfallen auf den Vorgang der Antriebsregelung bis zum Einfallen der Haltebremsen, um den Roboter in Position zu halten.

Quantitative Auswertung der Drehmomente

Die simulierten Drehmomente fallen signifikant größer aus als die gemessenen. Der negative Maximalwert von τ_2 liegt in der Simulation bei ca. -12 kNm. Dem gegenüber steht ein gemessener Wert von ca. -2,8 kNm. Die Differenz wird der Vernachlässigung des Gewichtsausgleichs zugeschrieben. Dieser kompensiert insbesondere den Einfluss der Gewichtskraft nachfolgender Glieder am zweiten Gelenk. Exemplarisch zeigt die Abbildung 4.1 den simulierten Anteil der Gewichtskraft bei Stillstand des Roboters in der Ausgangsstellung $q_{s,i}$, siehe Tabelle 3.1. Für die Drehmomente am zweiten Gelenk ist erkennbar, dass der Anteil der Gewichtskraft von -7743 Nm genau dem Offset des Drehmoments beim Start der Bewegung entspricht. In der Konsequenz fällt τ_2 im Gütekriterium der Optimierung zu stark ins Gewicht. Daraus wird abgeleitet, dass der Gewichtsausgleich in der Modellbildung zu berücksichtigen ist. Der Gewichtsausgleich wurde so gewählt, dass er den Anteils der Gewichtskraft des zweiten und dritten Verbindungsglieds am Drehmoment des zweiten Gelenks ausgleicht. Das Roboter-Dynamikmodell wird um die

Vorschrift 4.1 erweitert.

$$\boldsymbol{\mu}_i^i = \boldsymbol{\mu}_i^i - \boldsymbol{\mu}_{i,grav}^i \quad \forall i = 2 \quad (4.1)$$

wobei

$$\boldsymbol{\mu}_{i,grav}^i = -\mathbf{f}_{i,grav}^i \times (\mathbf{r}_{i-1,i}^i + \mathbf{r}_{i,C_i}^i) + \mathbf{R}_{i+1}^i \boldsymbol{\mu}_{i+1,grav}^{i+1} + \mathbf{R}_{i+1}^i \mathbf{f}_{i+1,grav}^{i+1} \times \mathbf{r}_{i,C_i}^i \quad \forall i = 2 \quad (4.2)$$

$$\boldsymbol{\mu}_{i,grav}^i = -\mathbf{f}_{i,grav}^i \times (\mathbf{r}_{i-1,i}^i + \mathbf{r}_{i,C_i}^i) \quad \forall i = 3 \quad (4.3)$$

$$\mathbf{f}_{i,grav}^i = \mathbf{R}_{i+1}^i \mathbf{f}_{i+1,grav}^{i+1} + m_i \ddot{\mathbf{p}}_{C_i}^i - \mathbf{R}_i^{0T} - m_i \mathbf{g} \quad \forall i \in \{2, 3\}. \quad (4.4)$$

Es folgt ein Drehmomentverlauf gemäß Abbildung 4.3.

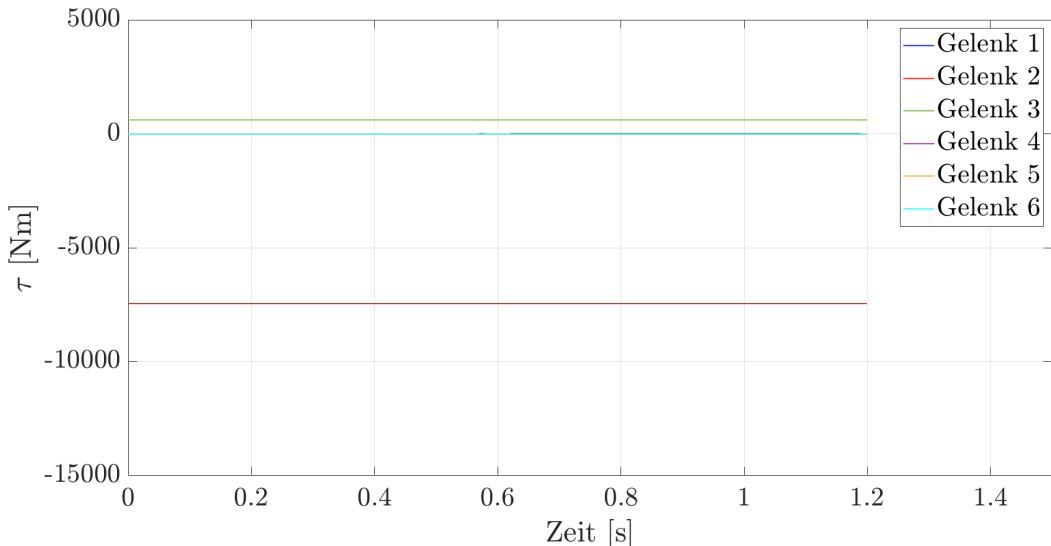


Abbildung 4.1: Simulierter Anteil der Gewichtskraft an den Drehmomenten in der Startposition der untersuchten Bewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Qualitative Auswertung der Drehmomente

Die Gelenke werden nachfolgend einzeln betrachtet. Für das erste Gelenk weist der gemessene Drehmomentverlauf einen steileren Anstieg als die simulierten Werte auf. Dieser Anstieg ist äquivalent zum Verlauf der Winkelbeschleunigung im selben Zeitabschnitt, siehe Abbildung 3.6. Das Maximum der simulierten Werte für das erste Gelenk ist ca. 50 % kleiner als die gemessenen Werte. Ursächlich hierfür ist die Vernachlässigung der

Getriebereibung. Der gemessene Drehmomentverlauf im ersten Gelenk weist außerdem keinen negativen Anteil auf. Es wird angenommen, dass die Getriebemomente in der Signalaufzeichnung mithilfe von messtechnisch erfassten Motorströmen sowie der Drehmomentkonstanten und der Getriebeübersetzungsfaktoren berechnet werden. Unter der Annahme, dass die Bremsenergie nicht zurückgewonnen wird, sondern über Bremswiderstände dissipiert, entsprechen die gemessenen Werte für $t > 0,8$ s dem Abbremsvorgang. Belegt wird dies mit dem übereinstimmenden Verlauf der Winkelgeschwindigkeit, siehe Abbildung 3.4. Dass der Drehmomentverlauf dem Geschwindigkeitsverlauf dabei um ca. 0,02 s nachläuft, wird den Energie speichernden Induktivitäten der Antriebe zugeschrieben. Für den Drehmomentverlauf im zweiten Gelenk sind ebenfalls steilere Anstiege infolge eines höheren Rucks des realen Roboters festzustellen. Im Verlauf ist das negativen Maximum des simulierten Drehmoments im zweiten Gelenk ca. 2000 Nm größer als das aufgezeichnete Drehmoment des realen Systems. Die Differenz wird dem nur näherungsweise modellierten Gewichtsausgleich zugeschrieben. Aufgrund der kinematischen Kopplung wirkt der Einfluss des Gewichtsausgleichs am realen Roboter indirekt auch auf das dritte Gelenk womit der Offset zu Beginn der Bewegung erklärt wird. Dass die Größenordnung des Drehmoments im dritten Gelenk nicht erreicht wird, wird der Vernachlässigung der Stator-Massen für die Motoren vier und fünf, der Rotor-Trägheit des dritten Motors sowie der Steifigkeit und Masse des Schlauchpakets zugeschrieben. Mit einem Gewicht von 65,3 kg ohne elektrische Leitungen steht die Masse des Schlauchpakets, gemäß CAD-Modell, mindestens im Verhältnis 1:5 zur Masse des dritten Verbindungsglieds. Daraus folgt eine zusätzliche Massenträgheit sowie die Verschiebung des Masseschwerpunkts, unter der Annahme, dass die Hardware ausschließlich am dritten Verbindungsglied befestigt ist. Aufgrund fehlender Daten entfällt eine Berücksichtigung des Schlauchpakets.

Auswertung der mechanischen Leistung

Die simulierten Leistungsdaten des ersten Gelenks stimmen in etwa mit den Werten überein, die auf der Grundlage der RSI Daten berechnet wurden. Der simulierte Verlauf der mechanischen Leistung für das zweite Gelenk weist einen ähnlichen Verlauf wie die messtechnisch erfassten Werte auf. Der Nulldurchgang der gemessenen Leistungskurve ist um 1,3 s nach rechts verschoben. Dies lässt sich unter Berücksichtigung der RSI-

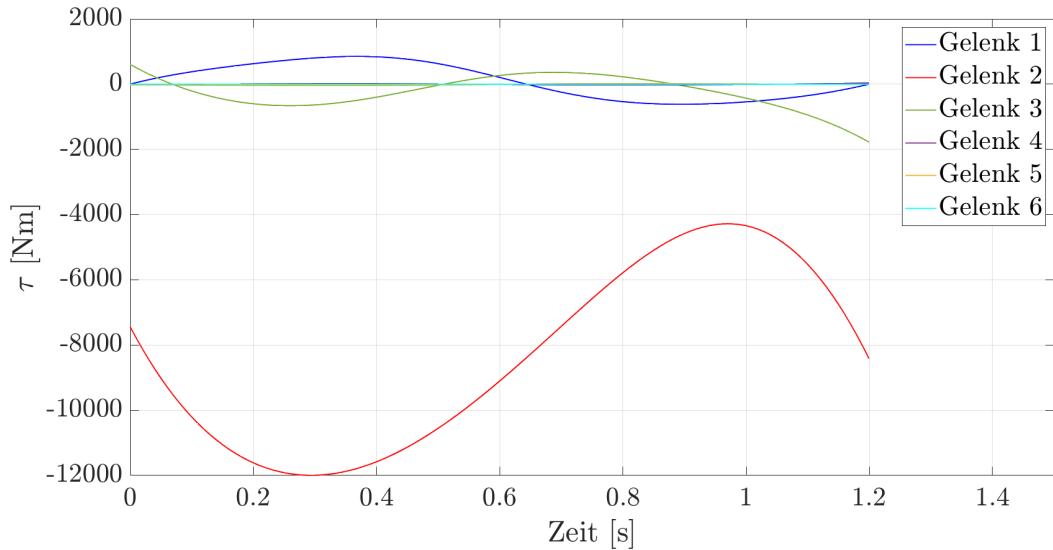


Abbildung 4.2: Simulierter Drehmomentverlauf ohne Gewichtsausgleich für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

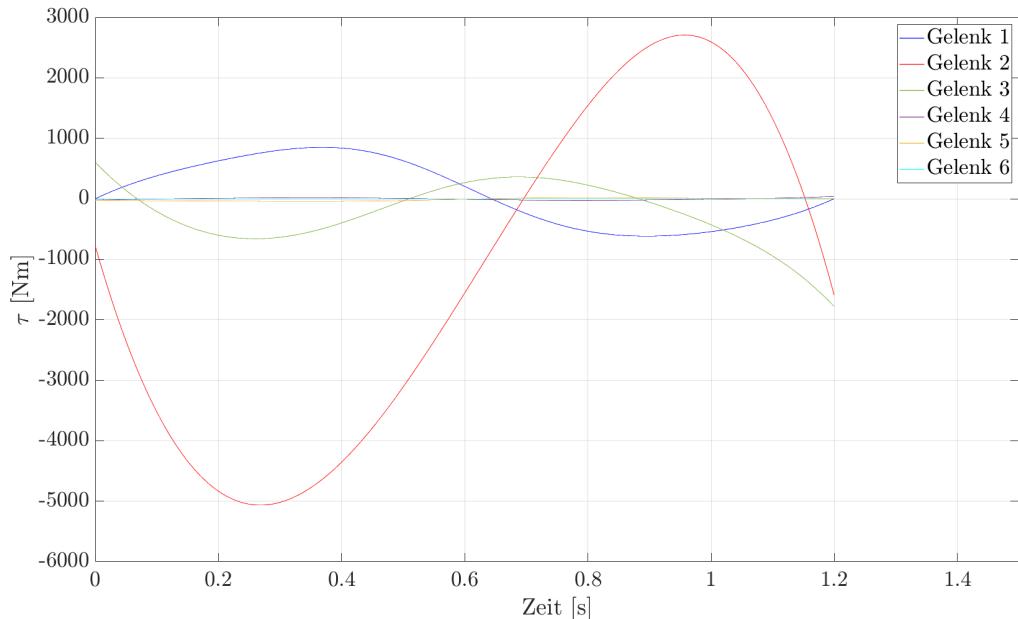


Abbildung 4.3: Simulierter Drehmomentverlauf inkl. näherungsweise modellierten Gewichtsausgleich für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

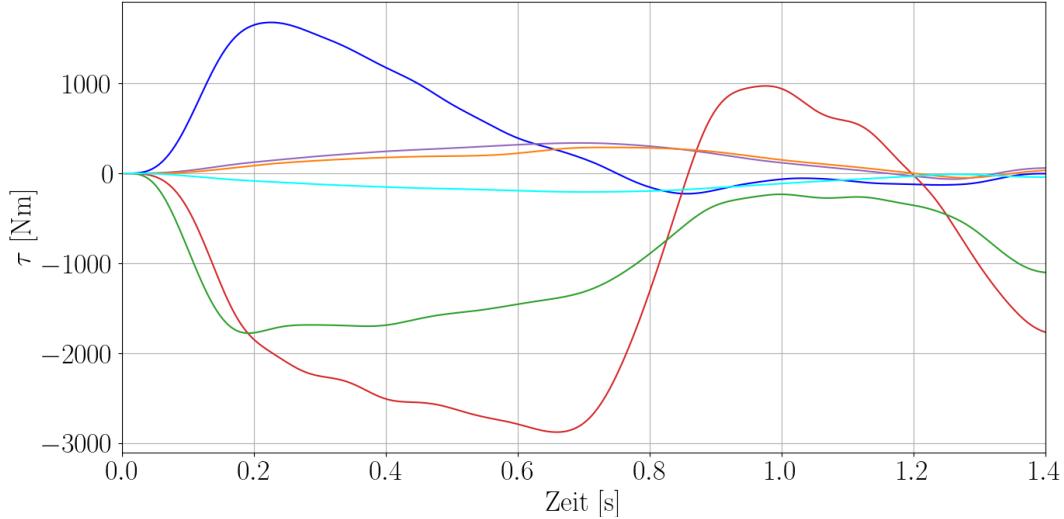


Abbildung 4.4: Gemessener Drehmomentverlauf für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Daten zur Winkelbeschleunigung in Abbildung 3.6 auf die kürzere Abbremsdauer des realen Roboters gegenüber der im Modell implementierten Bahnplanung zurückführen. In der gezeigten Bewegung wird die mechanische Leistung in den anderen Antrieben nicht durch das Modell abgebildet. Infolgedessen liegt der Schwerpunkt der Optimierung auf der Anpassung der Roboterkonfiguration zugunsten des Drehmoments im zweiten Gelenk.

Schlussfolgerung

Es wurde die Qualität eines Modells zur Simulation der mechanischen Leistungsaufnahme des Roboters für eine Bewegung mit der Start- und Zielkonfiguration gemäß Tabelle 3.1 untersucht. Die ausgewählte Bahn entspricht dem Verfahrweg des Roboters vom letzten Prozesspunkt zur Grundstellung im Produktionsprogramm Kleben-Seitenwand. Aufgrund des hohen Anteils der gemessenen sowie simulierten mechanischen Leistung für das zweite Gelenk, im Vergleich zu den anderen Gelenken, wird das größte Optimierungspotenzial im Modell berücksichtigt. Für die untersuchte Bewegung erfüllt das Modell den Zweck der Zielfunktionsberechnung zur Optimierung der ausgewählten Be-

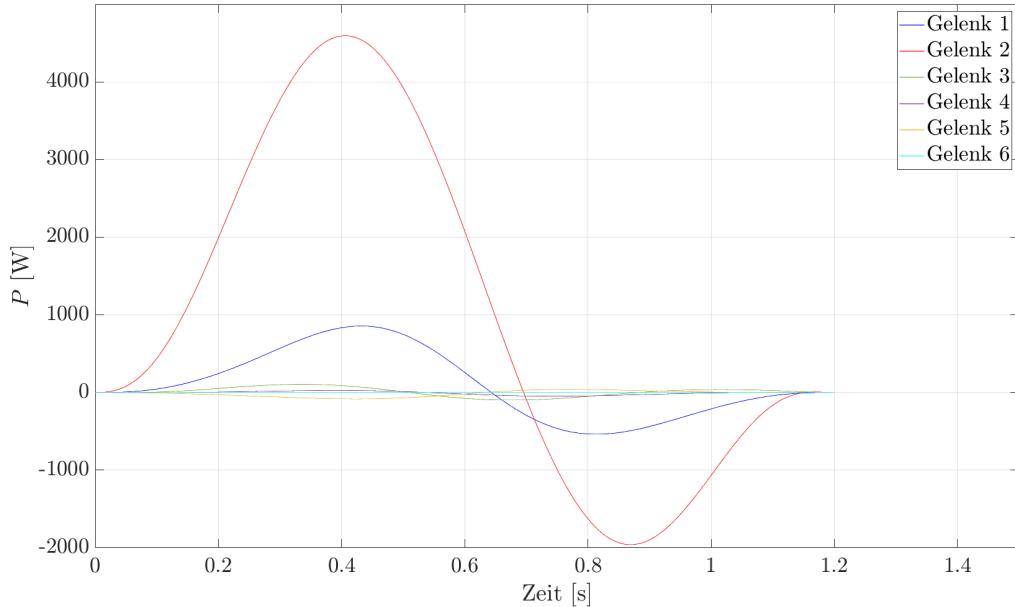


Abbildung 4.5: Simulierte Mechanische Leistung für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

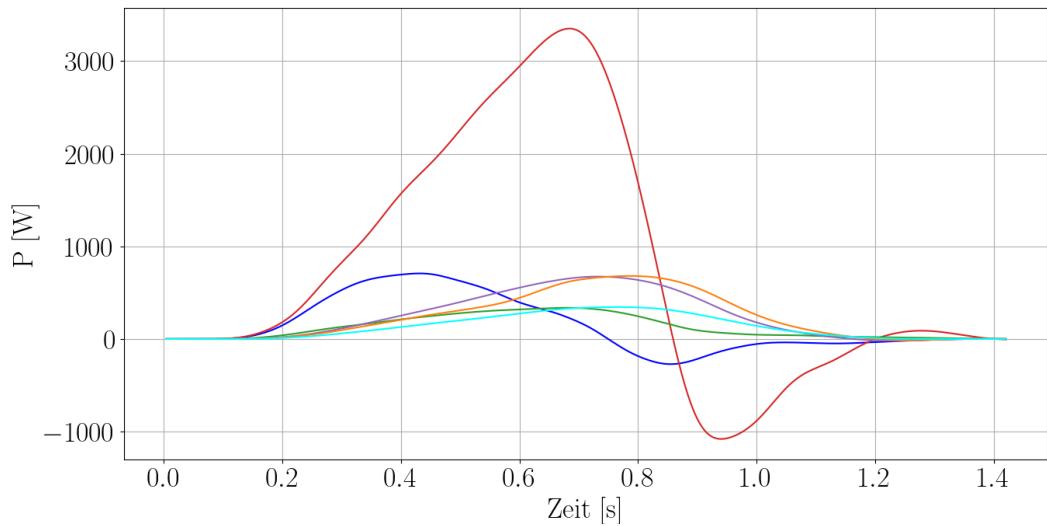


Abbildung 4.6: Gemessene Mechanische Leistung für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

wegungsbahn.

5 Bahnoptimierung

Im Folgenden wird die zuvor im Kapitel 3 eingeführte Bewegungsbahn des Programms Kleben-Seitenwand unter Berücksichtigung der vom Roboter aufgenommenen Energie optimiert. Dazu wird das Optimierungsproblem zunächst definiert, eine passende Zielfunktion aufgestellt und geeignete Nebenbedingungen festgelegt. Anschließend werden der numerische Solver sowie der verwendete Algorithmus beschrieben. Zum Abschluss erfolgt eine Analyse der Optimierungsergebnisse.

5.1 Definition des Optimierungsproblems

Gegenstand der Optimierung ist die Minimierung des Energieverbrauchs eines KR210 R2700-2 Industrieroboters entlang einer Bewegungsbahn. Im Ziel der Arbeit ist festgelegt, den Ansatz möglichst praktikabel für die Automobilproduktion zu gestalten. Der Begriff Bewegungsbahn ist dabei näher einzugrenzen. Zu unterscheiden sind prozessbezogene Bewegungen, z. B. entlang einer Schweißnaht oder Klebebahn, und nichtprozessbezogenen Bewegungen, z. B. das Anfahren einer Vorposition in der Bewegungsart PTP. Im Folgenden werden explizit nicht-prozessrelevante Bewegungen betrachtet. Ziel ist die geometrische Anpassung der Bewegungsbahn durch Optimierung eines Parametervektors \mathbf{q}_v . Dieser umfasst die Gelenkwinkel des nach der Hälfte der Bewegungsdauer definierten Via-Punkts [Zia+17, S 532 f.]

$$\mathbf{q}_v = [q_{v,1}, \dots, q_{v,6}]^T . \quad (5.1)$$

Der Startwert jedes Via-Punkts wird auf der Hälfte der Winkelposition zwischen dem Start- und Zielpunkt definiert.

$$q_{v,i,Start} = \frac{q_{s,i} + q_{e,i}}{2} \quad \forall i \in \{1, \dots, 6\} \quad (5.2)$$

Die Minimierung der Leistungsaufnahme folgt aus der konfigurationsabhängigen Reduktion der Massenträgheitsmomente sodass bei gleichbleibender Winkelbeschleunigung die Drehmomente in den Gelenken reduziert werden [Zia+17, S. 531]. Das Optimierungsproblem besteht darin, den Parametervektor zu identifizieren, für den die minimale Zielfunktion erreicht wird.

$$\arg \min_{\mathbf{q}_v} J(\mathbf{q}_v) = \{\mathbf{q}_v \mid J(\mathbf{q}_v) = \min_{\mathbf{q}_{v,opt}} J(\mathbf{q}_{v,opt})\} \quad (5.3)$$

5.2 Zielfunktion

Für die Definition der Zielfunktion wird angenommen, dass die generatorisch erzeugte Leistung während des Abbremsvorgangs über Bremswiderstände dissipiert [Pel+15, S. 1327]. Zur Identifizierung des Optimums werden die Zielfunktionen 5.4, 5.5 [SRB08, S. 1216] und 5.6 [Egg19, S. 57] aufgestellt. Die Integrationsgrenzen t_s und t_e entsprechen dem Start- bzw. Zielzeitpunkt. Der Ausdruck 5.4 wird in abgewandelter Form in [Han+12, S. 1] beschrieben und erzielt durch Quadrieren den Vorteil, dass hohe Drehmomente stärker gewichtet werden. Dabei ist jedoch nicht ersichtlich, ob Leistung aufgenommen oder abgegeben wird.

$$J_\tau(\mathbf{q}_v) = \int_{t_s}^{t_e} \sum_{i=1}^n \tau_i(t)^2 \, dt \quad (5.4)$$

Von einem Vorschlag der Zielfunktion 5.5 wird ebenso Abstand genommen, da hierbei eine Bewertung der generatorisch umgewandelten Leistung während des Abbremsvorgangs als aufgenommene Leistung der Verbraucher erfolgt.

$$J_{P_{mechges}}(\mathbf{q}_v) = \int_{t_s}^{t_e} \sum_{i=1}^n |\tau_i(t) \dot{q}_i|^2 \, dt \quad (5.5)$$

Die Zielfunktion 5.6 entspricht dem physikalischen Ausdruck der verrichteten mechanischen Arbeit über die zugeführte Leistung. Es wird ausschließlich die motorisch aufgenommene, mechanische Leistung $P_{mech} > 0$ berücksichtigt.

$$J_{P_{mech}}(\mathbf{q}_v) = E_{mech}(\mathbf{q}_v) = \int_{ts}^{te} P_{mech}(\mathbf{q}_v, t) dt \quad (5.6)$$

Die numerische Implementierung der Zielfunktion in MATLAB® entspricht der Gleichung 5.7. Die Schrittweite ist in Anlehnung an die Abtastung der Messeinrichtung mit $\Delta t = 0.004\text{s}$ festgelegt. Die Anzahl der Schritte m wird gemäß $(t_s - t_e)/\Delta t$ berechnet.

$$J_{P_{mech}}(\mathbf{q}_v) = E_{mech}(\mathbf{q}_v) = \sum_{k=1}^m P_{mech}(\mathbf{q}_v, t_k) \Delta t \quad (5.7)$$

5.3 Nebenbedingungen

Die Bewegungsdauer bleibt gegenüber der Initialbahn konstant und wird durch den Zielzeitpunkt t_e vorgegeben. In [Egg19, S. 40] bzw. [Han+12, S. 5] werden die Gelenkwinkel der energieoptimierten Bewegungsbahn in der Robotersteuerung über einen Software-in-the-Loop (SiL) Ansatz berechnet. Dadurch ist die Einhaltung von Drehmoment-, Gelenkwinkel-, Geschwindigkeit- und Beschleunigungsgrenzen entsprechend der Gleichungen 5.8, 5.9, 5.10 und 5.11 gewährleistet.

$$q_{i,min} \leq q_i(t) \leq q_{i,max} \quad \forall i \in \{1, \dots, 6\}, t \in [t_s; t_e] \quad (5.8)$$

$$|\dot{q}_i(t)| \leq |\dot{q}_{i,max}| \quad \forall i \in \{1, \dots, 6\}, t \in [t_s; t_e] \quad (5.9)$$

$$|\ddot{q}_i(t)| \leq |\ddot{q}_{i,max}| \quad \forall i \in \{1, \dots, 6\}, t \in [t_s; t_e] \quad (5.10)$$

$$|\tau_i(t)| \leq |\tau_{i,max}| \quad \forall i \in \{1, \dots, 6\}, t \in [t_s; t_e] \quad (5.11)$$

In der vorliegenden Arbeit sind ausschließlich die Schranken der Gelenkwinkel $q_{v,i}$ im Via-Punkt definiert. Die Formulierung zielt darauf ab, dass die optimierte Bewegungsbahn nicht signifikant von der Initialbahn abweicht [Han+12, S. 5].

$$q_{v,i} \in [q_{i,min}; q_{i,max}] \quad \forall i \in \{1, \dots, 6\} \quad (5.12)$$

Grenzwerte für die Drehmomente, Winkelgeschwindigkeiten und Winkelbeschleunigungen werden nicht explizit formuliert. Im Anschluss an die Optimierung erfolgt eine manuelle Überprüfung der Verläufe.

5.4 Solver- und Optimierungsalgorithmus

Die numerische Umsetzung der Optimierung erfolgt unter Verwendung der MATLAB[®] Optimization ToolboxTM. Für das nichtlineare Optimierungsproblem wird anhand einer Entscheidungstabelle in [o. 23a, S. 80] der Solver fmincon ausgewählt. Im Anhang D.6 ist der implementierte Code aufgeführt. Die Zielfunktion wird mithilfe des Dynamik-Modells in der separaten MATLAB[®]-Funktion calc_objective(qs,qe,qv,te) berechnet, siehe Anhang D.7. Es sind keine Gleichungs- bzw. Ungleichungsnebenbedingungen definiert. Es ist jedoch möglich, die im letzten Abschnitt beschriebenen Nebenbedingungen zu einem späteren Zeitpunkt hinzuzufügen. Für die einzelnen Via-Punkte des Parametervektors \mathbf{q}_v sind jeweils die obere und untere Schranke entsprechend der Gleichung 5.12 festgelegt. Für den Algorithmus wird das Verfahren der sequentiellen quadratischen Programmierung (sequential quadratic programming (SQP)) gewählt. SQP ist eine Verallgemeinerung des Newton-Verfahrens¹ für beschränkte Problemstellungen [PLB15, S. 113]. In [o. 23a, S. 253] wird beschrieben, dass für jeden Iterationsschritt die Einhaltung der Schranken gewährleistet ist. Dies wird als Vorteil aufgrund der Komplexität der dynamischen Zusammenhänge zur Berechnung der Zielfunktion bewertet. Notwendige Bedingungen zur Identifikation eines lokalen Minimums sind die Karush-Kuhn-Tucker (KKT) Konvergenzkriterien [NW06, S. 321]. In jeder Iteration des SQP-Verfahrens wird versucht, die KKT-Bedingungen für das ursprüngliche Problem durch eine quadratische Approximation der Zielfunktion zu erfüllen [RHG13, S. 337 ff.]. Zusätzlich bietet MATLAB[®] für fmincon die Option, Konvergenzkriterien zu definieren. Im Rahmen der ersten KKT Bedingung wird der Gradient der Zielfunktion überprüft. Falls dieser Wert nur annähernd Null erreicht, kann ein Schwellenwert festgelegt werden, bei dessen Unterschreitung die erste KKT-Bedingung erfüllt ist. Es ist außerdem möglich, eine maximale Anzahl von Iterationen festzulegen, bei deren Erreichen der Algorithmus beendet wird.

¹Das Newton-Verfahren ist ein numerischer Ansatz zur Lösung nichtlinearer Gleichungssysteme, bei dem eine iterative Approximation mithilfe der nach ersten Glied abgebrochenen Taylorreihe erfolgt [PLB15, S. 46].

Die Variable '`MaxIterations`' wird mit 25 frei gewählt. Für eine ausführliche Herleitung von SQP-Algorithmen auf der Basis des Newton-Lagrange-Verfahrens wird auf [NW06, S. 529 ff.] verwiesen.

5.5 Durchführung der Optimierung

Die Arbeit verfolgt den Anspruch, dass die identifizierten Einsparungen über das Laborumfeld hinaus in der industriellen Praxis realisiert werden können. Dazu wird eine Bewegung der Bewegungsart PTP betrachtet, die in nahezu jedem Programm den Verfahrweg nach Programmende in die Grundstellung abbildet. Exemplarisch wird die Bewegung vom letzten Prozesspunkt zurück in die Grundstellung im Produktionsprogramm Kleben-Seitenwand optimiert. Diese wurde bereits im Kapitel Trajektorie-Planung 3 eingeführt. Die Start- und Zielkonfiguration ist in der Tabelle 3.1 hinterlegt. Im ersten Schritt wird die Initial-Trajektorie abgefahren. Dabei werden die Bewegungsdaten über die RSI-Signalaufzeichnung erfasst. Anhand der Genlenkwinkelgeschwindigkeitsverläufe, siehe Abbildung 3.4, wird die Gesamtbewegungsdauer mit 1,2 s geschätzt. Für die Startwerte 5.2 des Parametervektors 5.1 wird die Optimierung in MATLAB®, siehe Anhang D.6, ausgeführt. Die Anzahl der Iterationen ist auf 25 beschränkt. In der Abbildung 5.1 wird deutlich, dass die energieoptimierten Gelenkwinkel bereits nach acht Iterationsschritten näherungsweise erreicht sind.

5.6 Auswertung der Optimierungsergebnisse

Die identifizierten, energieoptimierten Via-Punkte sind in der Tabelle 5.1 aufgeführt. Auffällig ist, dass für $q_{v,3}$ und $q_{v,4}$ die obere Schranke sowie für $q_{v,6}$ die untere Schranke aktiv ist. Das bedeutet der Gelenkwinkel im Via-Punkt weist den selben Wert auf wie der Gelenkwinkel im Start- oder Zielpunkt. Für den Gelenkwinkel $q_{v,5}$ wird festgestellt, dass dieser auffällig nah an der unteren Schranke liegt. Daraus folgt, dass die resultierende Trajektorie für die Gelenkwinkel drei, vier und sechs Werte außerhalb des Intervalls $[q_{i,s}; q_{i,e}] \forall i \in \{3, 4, 5, 6\}$ annehmen wird.

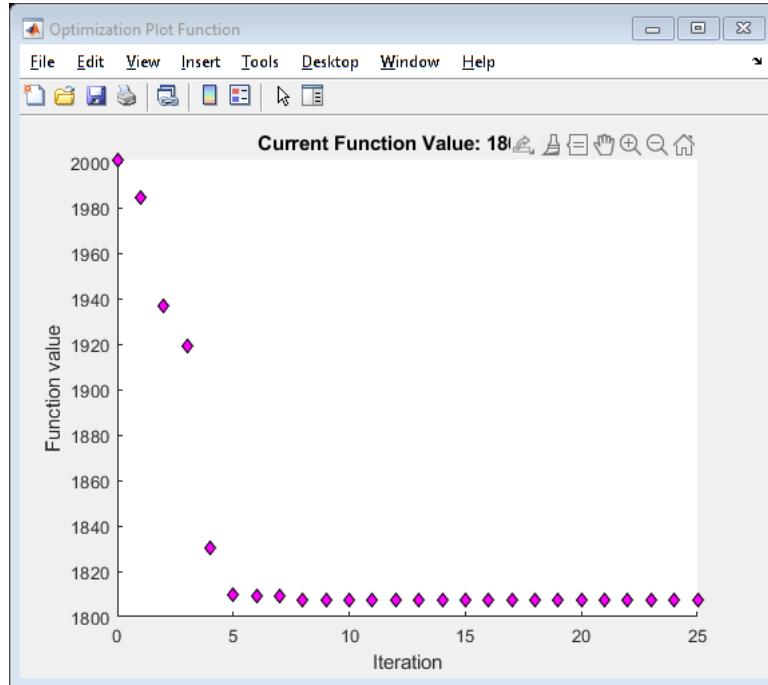


Abbildung 5.1: Zielfunktionswerte der Optimierungsiterationen

Tabelle 5.1: Winkeldefinition der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand inklusive der energieoptimierten, justierten Via-Punkte

Startwert Gelenkwinkel	Zielwert Gelenkwinkel	Via-Punkte
$q_{s,1} = -53,8^\circ$	$q_{e,1} = -7,6^\circ$	$q_{v,1} = -25^\circ$
$q_{s,2} = -70,3^\circ$	$q_{e,2} = -119,3^\circ$	$q_{v,2} = -96,1^\circ$
$q_{s,3} = 98,8^\circ$	$q_{e,3} = 88,5^\circ$	$q_{v,3} = 98,8^\circ$
$q_{s,4} = -69,9^\circ$	$q_{e,4} = 10,3^\circ$	$q_{v,4} = 10,3^\circ$
$q_{s,5} = -58,7^\circ$	$q_{e,5} = 32,4^\circ$	$q_{v,5} = -47,1^\circ$
$q_{s,6} = 55,7^\circ$	$q_{e,6} = -10,2^\circ$	$q_{v,6} = -10,2^\circ$

Abbildung 5.2 zeigt den simulierten Verlauf der mechanischen Leistung für die initiale Bewegungsbahn² sowie für die energieoptimierte Bewegungsbahn³. Wie erwartet, erzielt die Optimierung im Wesentlichen ein Reduzieren der Leistungsaufnahme im zweiten Gelenk. Der simulierte, mechanische Energieverbrauch des Roboters für die Initialbahn beträgt 2001 J. Die optimierte Bewegungsbahn wird mit einem mechanischen Energieverbrauch von 1807 J simuliert. Damit wird eine Energieeinsparung von ca. 9,7 % prognostiziert. Für das erste Gelenk ist qualitativ eine Linksverschiebung des Bewegungsablaufs um 0,04 s erkennbar. Des Weiteren nimmt die maximalen Leistungsaufnahme im ersten Gelenk geringfügig zu. Für die Gelenke drei bis sechs sind minimale Abweichungen gegenüber des Initialbewegungsablaufs erkennbar, die aufgrund ihrer Größenordnung keinen signifikanten Einfluss auf die Energieeinsparung nehmen.

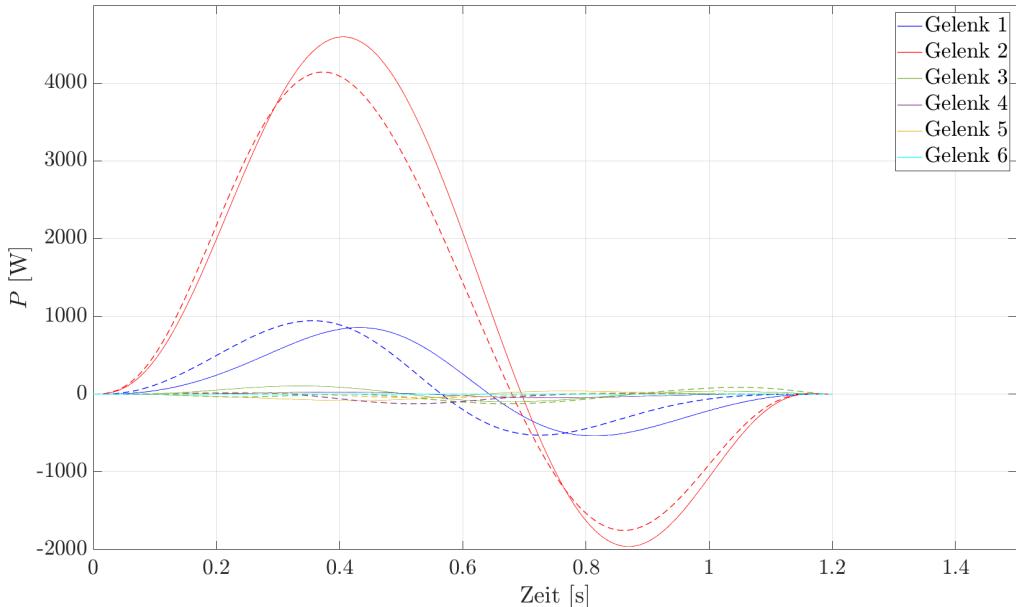


Abbildung 5.2: Vergleich der simulierten Leistungsaufnahme für die Initialbewegung und die optimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Nachfolgend werden der Verlauf der Gelenkwinkel in Abbildung 5.3 sowie die Geschwindigkeitsnebenbedingungen in Abbildung 5.4 untersucht. Für den Verlauf des Winkels im zweiten Gelenk ist keine auffällige Abweichungen gegenüber der Initialbahn zu er-

²ausgefüllte Linien

³gestrichelte Linien

kennen. Die beiden Verläufe der Gelenkwinkel eins und drei zeigen geringe Differenzen im Bereich der Via-Punkte auf. Signifikante Unterschiede sind für die Verläufe der Gelenkwinkel vier, fünf und sechs erkennbar. Bei einer Betrachtung der Abbildung 5.4 fällt auf, dass die Winkelgeschwindigkeiten $\dot{q}_4(t)$, $\dot{q}_5(t)$ und $\dot{q}_6(t)$ erheblich größer ausfallen als die Originalverläufe. Dies ist auf die Nähe der Gelenkwinkel $q_{v,4}$, $q_{v,5}$ und $q_{v,6}$ zu den Grenzwerten in Kombination mit der Größe des zurückgelegten Gelenkwinkels $\Delta q_i = |q_{s,i} + q_{e,i}| \forall i \in \{4, 5, 6\}$ zurückzuführen.

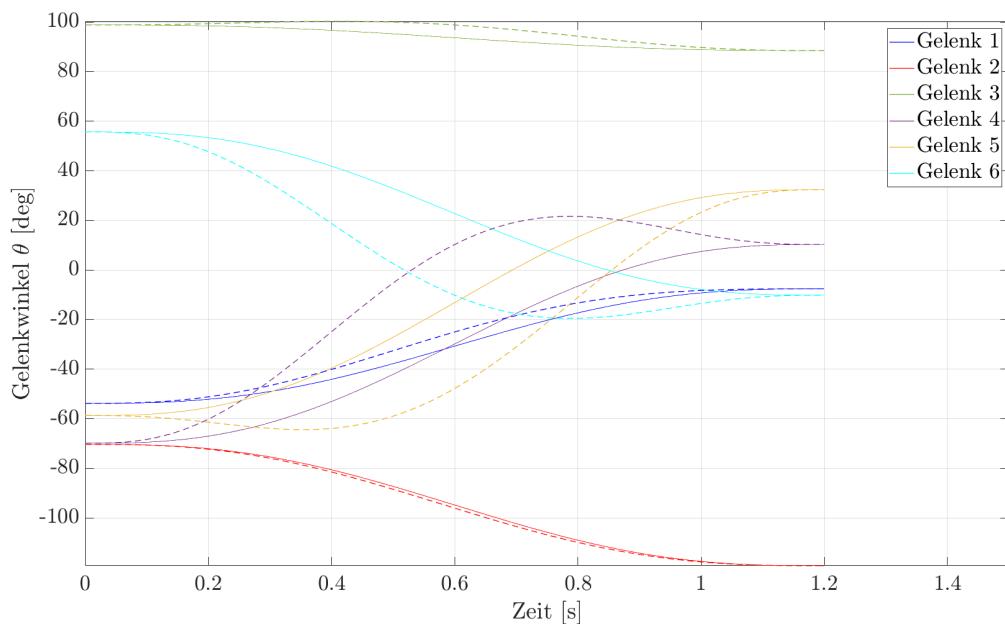


Abbildung 5.3: Vergleich der simulierten Gelenkwinkel für die Initialbewegung und die optimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Infolgedessen werden die Via-Punkte $q_{v,4}$, $q_{v,5}$ und $q_{v,6}$ manuell auf den Mittelwert zwischen der initialen Definition und den energieoptimierten Werten, siehe Gleichung 5.13, nachjustiert.

$$q_{v,i,justiert} = \frac{q_{v,i,opt} + \frac{q_{s,i} + q_{e,i}}{2}}{2} \quad \forall i \in \{4, 5, 6\} \quad (5.13)$$

Die resultierenden Winkelgeschwindigkeiten in Abbildung 5.5 werden damit auf ein akzeptables Maß reduziert. Die energieoptimierten Verläufe sind weiterhin gestrichelt dargestellt. Die Kurven der justierten Via-Punkte werden punktiert abgebildet.

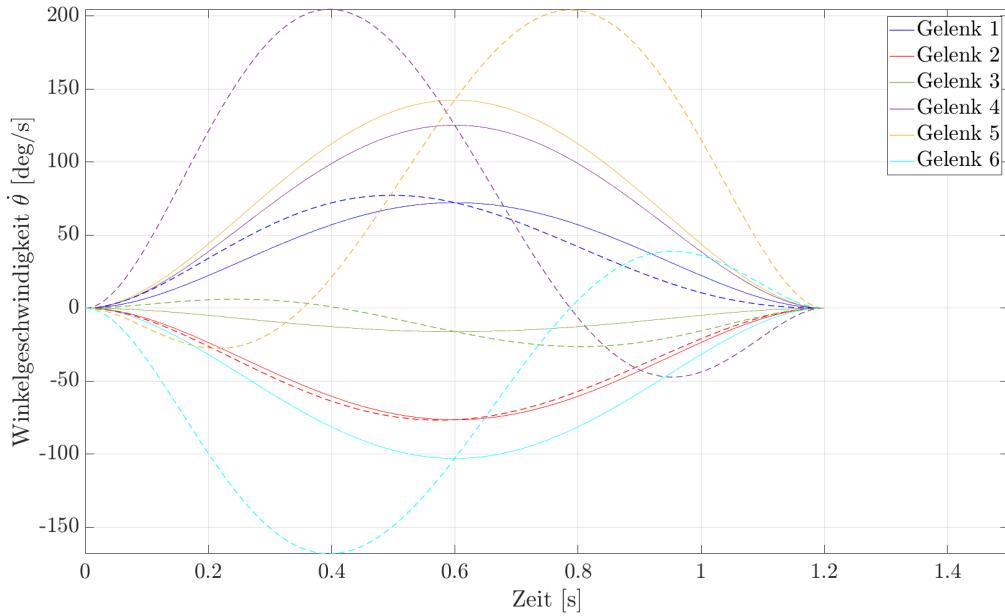


Abbildung 5.4: Vergleich der simulierten Winkelgeschwindigkeit für die Initialbewegung und die optimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Für den justierten Parametervektor beträgt der Energieverbrauch 1858 J. Damit wird eine Energieeinsparung von 7,15 % gegenüber der Initialbahn erzielt. Der Leistungsverlauf für den justierten Parametervektor wird dem der Initial-Trajektorie in Abbildung 5.6 gegenüber gestellt. Der Verlauf entspricht qualitativ dem der energieoptimierten Bewegungsbahn. Es wird infolge der Anpassung eine leichte Zunahme der Leistungsaufnahme im zweiten Gelenk notiert. Die Gegenüberstellungen der Gelenkwinkel-, Winkelgeschwindigkeits- Beschleunigungs- und Drehmomentverläufe der initialen- und justierten-Trajektorie sind im Anhang E.1 aufgeführt.

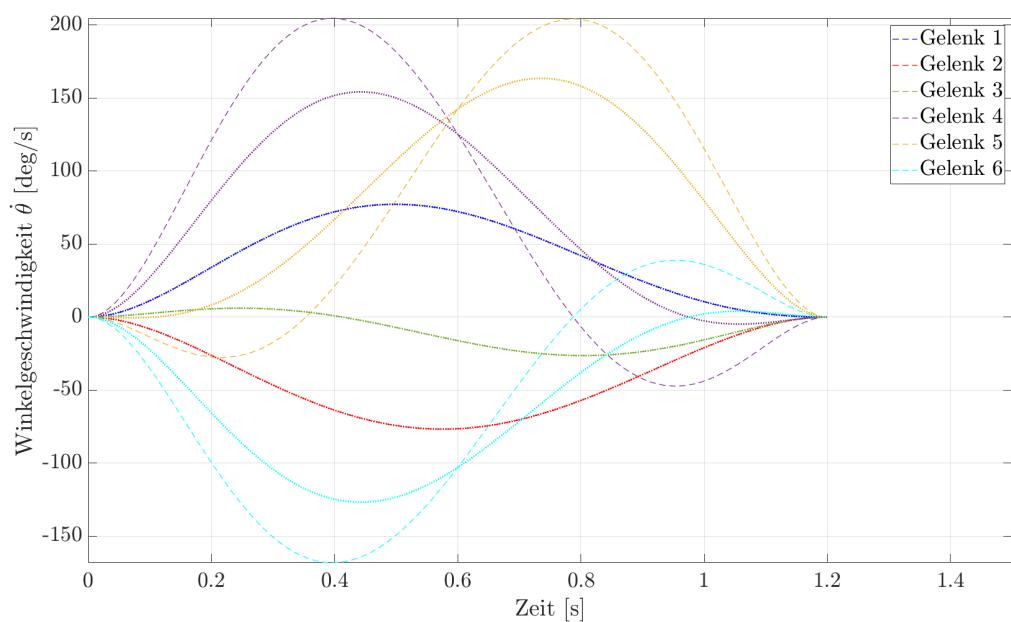


Abbildung 5.5: Vergleich der simulierten Winkelgeschwindigkeit für die energieoptimierte Bewegungsbahn und die justierte, energieoptimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

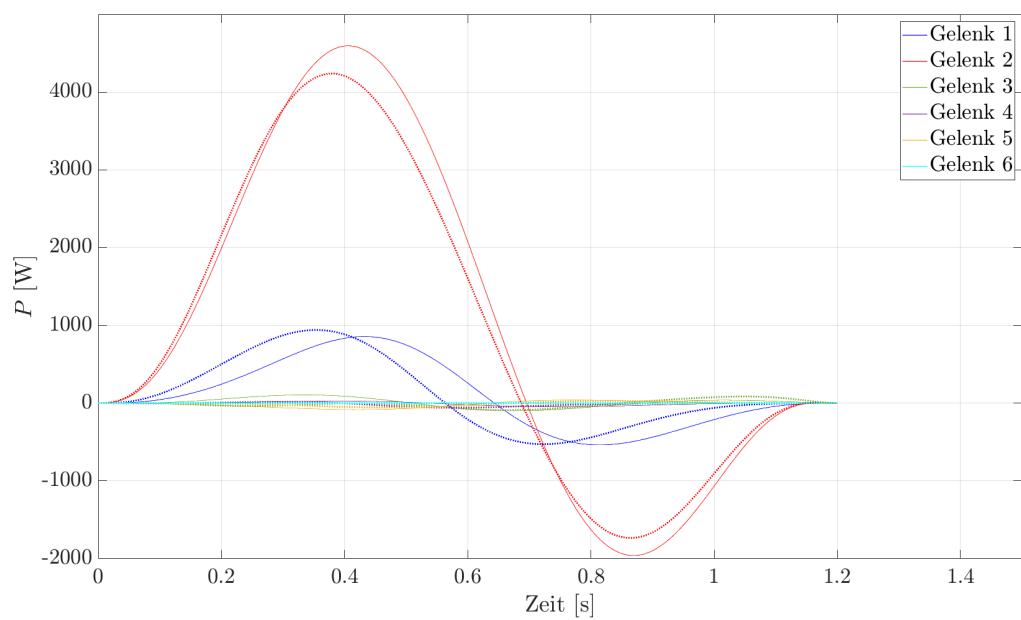


Abbildung 5.6: Vergleich der simulierten Leistungsaufnahme für die energieoptimierte Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

6 Validierung der Optimierungsergebnisse

Das Ziel der Validierung besteht darin, die in der Simulation berechneten Energieeinsparungen am realen System mithilfe des optimierten Parametervektors nachzuweisen. Der Optimierer wurde gegenwärtig nur an einem einzigen Pfad für den Roboter getestet. Von einer Abbildung der Validierung auf weitere Optimierungsversuche wird abgesehen. Diese sind im Einzelfall zu überprüfen. Der Messaufbau ist identisch zu der Beschreibung im Kapitel 4 Validierung des Roboterdynamik-Modells.

6.1 Durchführung

Roboterprogramme werden auf der KR C5 in der Programmiersprache KRL geschrieben. Ein KRL-Programm ist dabei aus einem source-file (src) und einem data-file (dat) aufgebaut. Der justierte-energieoptimierte Parametervektor \mathbf{q}_v wird auf der KR C5 zwischen dem Start- und Zielpunkt der letzten Bewegungsbahn des Programms Kleben-Seitenwand als Via-Punkt ViaJustOptUp im src-file angelegt.

```
;FOLD PTP ViaJustOptUp Vel=100 % PDAT8 Tool [1] Base [0] ;%{PE}
```

Ohne einen expliziten Befehl fährt der Roboter den Via-Punkt millimetergenau an und reduziert seine Geschwindigkeit bei Erreichen des Punkts auf Null. In der Problemstellung der Arbeit wurde eine signifikante Abweichung von der originalen Bewegungsdauer als unzulässig definiert. Infolgedessen wird der Implementierte Via-Punkt im src-file um den KRL Überschleifbefehl CONT erweitert.

```
;FOLD PTP ViaJustOptUp CONT Vel=100 % PDAT8 Tool[1] Base[0] ;%{PE}
```

Zusätzlich muss der Robotersteuerung mitgeteilt werden, in welchem Modus der Via-Punkt zu überschleifen ist und welcher Überschleifradius angewendet werden soll. Dies erfolgt im dat-file über die Parameterdefinition PDAT, welche dem Via-Punkt src-file zugewiesen ist. Die Approximationsmethode APO_MODE wird mit CDIS festgelegt. Das bedeutet ein Überschleifen beginnt frühestens, wenn die Entfernung zum Punkt den Wert von APO_DIST = 500 mm unterschreitet [o. 23b, S. 578].

```
DECL PDAT PPDAT8={VEL 100.0000,ACC 100.000,APO_DIST 500.000,APO_MODE #  
CDIS,GEAR_JERK 100.000,EXAX_IGN 0}
```

Die maximale Überschleifdistanz wird auf 500 mm festgelegt. Von der Nutzung des maximal möglichen Überschleifradius 1000 mm wird Abstand genommen, um das Risiko einer Kollision des Roboters mit peripheren Bauteilen zu begrenzen. Die neu programmierte Bewegung wird zunächst auf Kollisionsfreiheit durch manuelles Abfahren mit reduzierter Geschwindigkeit in der KUKA Betriebsart T1 getestet. Nach einer Verifizierung der Überschleifbewegung für eine programmierte Geschwindigkeit von 50 % in der Betriebsart T2 erfolgt die Signalaufzeichnung der justierten-energieoptimierten Trajektorie mit einer programmierten Geschwindigkeit von 100 %.

6.2 Auswertung

Die über alle sechs Gelenke summierte Leistungsaufnahme des Roboters wird in 6.1 abgebildet.

Dabei ist eine Zunahme der Bewegungsdauer für die justierte-energieoptimierte Trajektorie um ca. 0,2 s gegenüber der Initial-Trajektorie festzustellen. Die exakte Dauer kann nur näherungsweise angegeben werden. Dies liegt an der im Rahmen der Arbeit entwickelten Datenverarbeitung. Die über den RSI aufgezeichneten und am Server empfangenen Signale werden nachträglich separiert und der einzelnen Bewegungen zugewiesen. Eine neue Bewegung wird beim Durchlaufen einer niedrig gewählten Leistungsschwelle

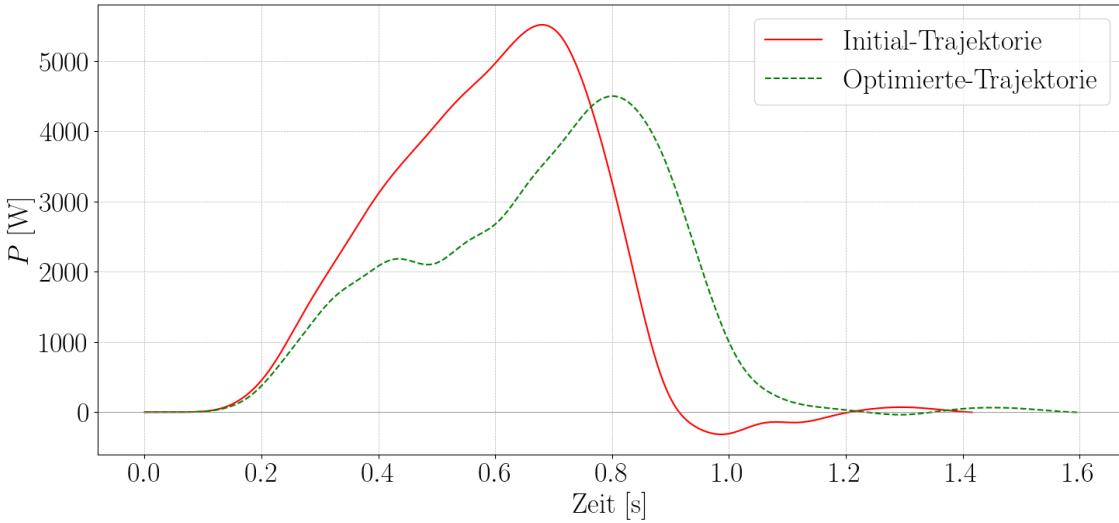


Abbildung 6.1: Vergleich der summierten Leistungsaufnahme für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

indiziert. Die leichte Zunahme der Bewegungsdauer wird dem zusätzlichen Programmschritt durch die Definition des Via-Punkts mit CDIS = 500 mm angerechnet. In der PTP-Bewegungsart führt der Roboter den TCP entlang der schnellsten Bahn zum Zielpunkt [o. 23b, S. 429]. Der Via-Punkt hat zur Folge, dass der TCP nicht mehr auf der schnellsten Bahn geführt wird. Die maximal aufgenommene Momentanleistung konnte um ca. 1 kW gesenkt werden. Gegenüber der stetig ansteigenden Leistungsaufnahme der Initial-Trajektorie weist die Leistungskurve der Optimierten-Trajektorie einen Sattel auf. Der Verlauf der simulierten Leistung, siehe Abbildung 5.6, sowie die im Anhang hinterlegten simulierten Bewegungsdaten weisen dagegen keine Unstetigkeit auf. Die dem Sattel zugrunde liegende temporäre Reduktion der Geschwindigkeit setzt auf der Hälfte der Bewegungsdauer zwischen dem Start und dem Via-Punkt ein. Die Ursache dafür liegt in der Begrenzung der Geschwindigkeit des sechsten Gelenks. In der Konsequenz ist der Optimierer um die Geschwindigkeitsnebenbedienung 5.10, sowie die im Kapitel 5.3 genannten und in [Han+12, S. 5] beschriebenen Formulierungen zur Begrenzung der maximalen Beschleunigung bzw. des maximalen Drehmoments zu erweitern. Die über den optimierten Via-Punkt erzielte Energieeinsparung wird in der Grafik 6.2 dargestellt. Die Prognose der Simulation, dass die höchste Einsparung für das zweite Gelenk zu ver-

6 Validierung der Optimierungsergebnisse

zeichnen ist, wird erfüllt. Wie erwartet ist für den mechanischen Energieverbrauch im ersten Gelenk eine leichte Zunahme festzustellen. Die Optimierungsergebnisse werden in der Konsequenz als plausibel eingestuft.

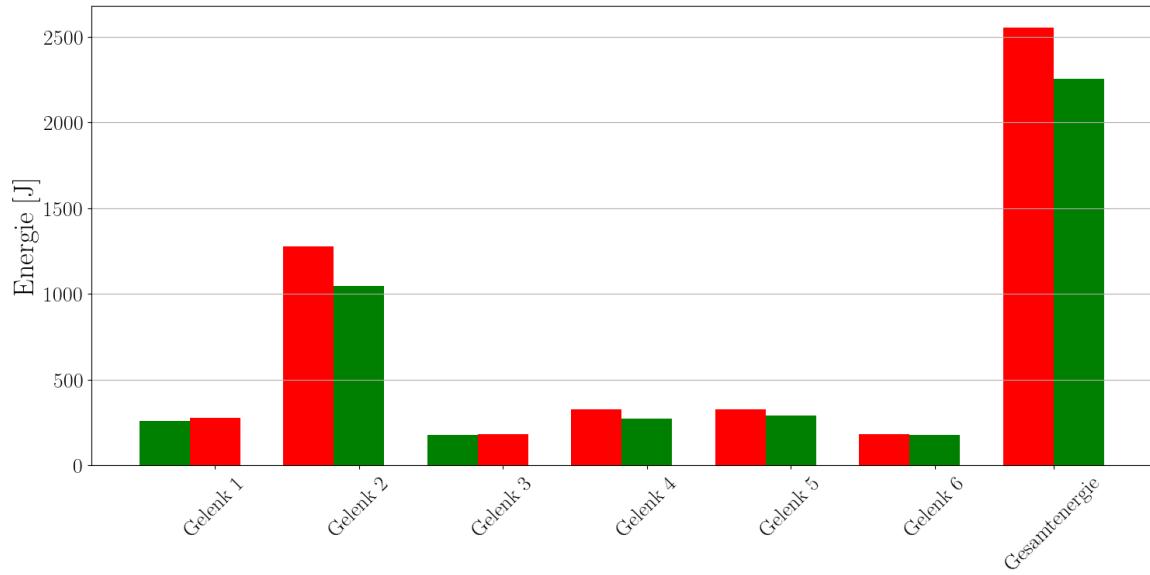


Abbildung 6.2: Vergleich des mechanischen Energieverbrauchs für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

7 Bewertung der Optimierungsergebnisse

Der Arbeit liegt die Forschungsfrage zugrunde, ob ein für die Produktion entwickeltes Roboterprogramm durch Hinzufügen und Verschieben von Via-Punkten energetisch optimiert werden kann, ohne die Bewegungsdauer signifikant zu erhöhen. Hierfür wurde die Bewegung vom letzten Prozesspunkt zurück in die Grundstellung im Produktionsprogramm Kleben-Seitenwand um einen Via-Punkt erweitert. Die Bewegung zeichnet sich dadurch aus, dass der TCP im kartesischen Raum nach oben bewegt wird und der Roboter somit gegen die Schwerkraft arbeitet. Der Via-Punkt wurde auf Basis eines Modells hinsichtlich des Energieverbrauchs des Roboters optimiert. Eine quantitative Gegenüberstellung der Ergebnisse aus Messung und Simulation erfolgt in den Tabellen 7.1 und 7.2. Die gemessene relative Einsparung beträgt 11,7 %. Der gemessene Energieverbrauch ist insgesamt etwa 25 % höher als in der Simulation prognostiziert. Dies lässt sich darauf zurückführen, dass der Verbrauch in den Gelenken drei bis sechs im Modell deutlich niedriger berechnet wird. Die Zunahme der Bewegungsdauer beträgt ca. 15 % und ist im Kontext vor- und nachgelagerter Bearbeitungsschritte bzw. unter Berücksichtigung des langsamsten Prozessschrittes für einen in der Linienfertigung eingesetzten Roboter zu bewerten. Für das isoliert betrachtete Szenario wird die Erhöhung der Bewegungsdauer um 0,2 s als akzeptabel beurteilt. Der Anteil der Energieeinsparung, der auf die Erhöhung der Bewegungsdauer zurückzuführen ist, wird als marginal eingeschätzt. Aus dem Leistungsverlauf in Abbildung 6.1 geht hervor, dass sowohl die Initial-Trajektorie als auch die optimierte Trajektorie zum Zeitpunkt $t = 1,2$ s einen annähernd gleichen Wert nahe Null erreicht. Die Zielsetzung dieser Arbeit definiert die Durchführung einer Bahnoptimierung für einen Industrieroboter mit serieller Kinematik. Die Prämisse der Arbeit, den Via-Punkt auf Basis eines Modells zu optimieren, wurde erreicht. Für den untersuchten Fall wird die Forschungsfrage bestätigt.

Tabelle 7.1: Simulierter, mechanischer Energieverbrauch für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Trajektorie	Größe	Einheit	Wert
Initial	Energieverbrauch	[kJ]	2.001
Justiert-energieoptimiert	Energieverbrauch	[kJ]	1.858
	Energieeinsparung	[kJ]	0.143
	Energieeinsparung	[%]	7,15

Tabelle 7.2: Gemessener, mechanischer Energieverbrauch für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

Trajektorie	Größe	Einheit	Wert
Initial	Energieverbrauch	[kJ]	2.553
Justiert-energieoptimiert	Energieverbrauch	[kJ]	2.254
	Energieeinsparung	[kJ]	0.299
	Energieeinsparung	[%]	11,7

Kritisch wird dabei der Testumfang zur Beurteilung des Optimierers bewertet. Infolgedessen beschränkt sich die Bewertung des Optimierers ausschließlich auf die untersuchte Trajektorie. Für den ausgewählten SQP-Algorithmus weist der fmincon-Solver aus der MATLAB® Optimization Toolbox™ eine hohe Effizienz auf. Bereits nach acht Iterations-schritten wird der energieoptimierte Parametervektor näherungsweise identifiziert. Die Zielfunktion des Optimierungsproblems ist nicht konvex. Infolgedessen garantiert der angewandte SQP-Algorithmus nicht, ein globales Optimum zu finden [Nocedal 2006]. Eine Möglichkeit zur Lösung des Problems besteht darin, die Startdefinition des Parametervektors \mathbf{q}_v zu variieren. Alternativ kann durch die Anwendung von stochastischen Verfahren, wie beispielsweise Evolutionären Algorithmen [PLB15, S.], die Wahrscheinlichkeit erhöht werden, ein globales anstelle eines lokalen Minimums zu finden. Hierzu wird auf die Arbeit [Non+22] verwiesen. Abschließend wird die absolute Energieeinsparung gemäß der Berechnung 7.1 bewertet.

$$0,299 \text{ kJ} \cdot \frac{1}{3600} \frac{\text{Wh}}{\text{J}} = 8,306 \cdot 10^{-5} \text{ kWh} \quad (7.1)$$

Unter der Annahme, dass das Programm Kleben-Seitenwand pro Fahrzeugseite einmal ausgeführt wird, täglich 2.400 Einheiten produziert werden und die Produktion an 250 Tagen im Jahr erfolgt, ergibt sich eine elektrische Gesamtenergieeinsparung von ca. 100

kWh. Hierbei wurden die Wirkungsgrade der Antriebe nicht berücksichtigt. Des Weiteren wurde ausschließlich aufgenommene Energie betrachtet und von einer Dissipation der Bremsenergie angenommen. Die Berechnung führt zu dem Schluss, dass die Modellbildung erst dann rentabel ist, wenn die Optimierung auf eine große Anzahl baugleicher Roboter skaliert werden kann. Alternativ sind Lösungsansätze zu verfolgen, die wie in [Egg19, S. 40] beschreiben ein SiL-Verfahren zur Bahnplanung und Zielfunktionsberechnung nutzen. Eine weitere Möglichkeit ist die Verwendung von KUKA.RCS zur Simulation der Roboterdynamik in einer Softwareumgebung von Siemens oder Dassault Systemes [o. 19].

8 Zusammenfassung und Ausblick

Die Einleitung Kapitel 1 ordnet die Arbeit in den betrieblichen Kontext ein. Es ist hervorzuheben, dass die Steigerung der Energieeffizienz von Industrierobotern einen Anteil zur Erreichung des produktions- und planungsrelevanten Umweltziels „Energieeinsparung“ leistet. Aus der Literaturanalyse ergibt sich für die Arbeit im wissenschaftlich-technischen Kontext die Rolle einer praktischen Umsetzung bereits erforschter Ansätze zur Energieeinsparung durch Via-Punkt basierte Bahnoptimierung.

Das Kapitel 2 Mechanische Modellbildung beschreibt die Vorwärtsskinematik gemäß der Denavit-Hartenberg Konvention und beschreibt die Implementierung des rekursiven Newton-Euler Algorithmus zur Berechnung der im Gelenk auftretenden Getriebemomente entlang einer Bewegungsbahn. Basierend auf der Implementierung des Modells in Matlab kann der Energieverbrauch des Roboters simuliert werden, ohne dass das reale System angesteuert werden muss. Mit dem Modell können die Auswirkungen von Geschwindigkeit, Massenträgheit und der Einfluss des Gewichtsausgleichs analysiert werden. Dadurch kann die in der Ausgangssituation beschriebene Lücke des DOE zur Identifikation der Faktoren für einen energetisch optimierten Via-Punkt geschlossen werden. Das Kapitel zur Modellbildung schließt mit einer Beschreibung der Annahmen und Vernachlässigungen bezüglich des dynamischen Verhaltens. Es wird ausdrücklich angemerkt, dass nur die mechanische Leistung des Roboters beschrieben ist und auf eine Modellierung von Reibungseinflüssen verzichtet wurde. In der Literaturübersicht wird auf Veröffentlichungen verwiesen, die diesen Bereich abdecken.

Im Kapitel 3 wird die Trajektorie-Planung für die Gelenkwinkel des Roboters eingeführt. Hierbei wird ein Polynom sechster Ordnung verwendet. Zur Validierung werden die berechneten Gelenkwinkel, Winkelgeschwindigkeiten und Winkelbeschleunigungen mit den Bewegungsdaten verglichen, die mithilfe einer RSI-Signalaufzeichnung an der Robotersteuerung erfasst werden.

8 Zusammenfassung und Ausblick

Im Kapitel 4 Validierung des Roboterdynamik-Modells wird zunächst das RSI Technologiepaket zur Aufzeichnung von Messwerten und Bewegungsdaten an der Robotersteuerung vorgestellt. Unter Berücksichtigung der getroffenen Annahmen wird untersucht, ob das implementierte Modell entlang der berechneten Trajektorie die Roboterdynamik hinreichend genau simuliert. Auf Basis dieser Untersuchung wurde eine Anpassung der RNEA-Implementierung vorgenommen, um den Gewichtsausgleichszylinder am zweiten Gelenk zu berücksichtigen. Anhand eines qualitativen und quantitativen Vergleichs der simulierten und gemessenen Drehmomente sowie der berechneten Leistungsaufnahme konnte das Modell für die im Rahmen der Optimierung untersuchte Bewegungsbahn des Produktionsprogramms Kleben-Seitenwand validiert werden.

Kapitel 5 setzt die Optimierung der Bewegungsbahn um. Hierfür werden zunächst das Optimierungsproblem definiert, eine geeignete Zielfunktion aufgestellt und Nebenbedingungen festgelegt. Zudem werden der numerische Solver und der verwendete Algorithmus beschrieben. Abschließend werden die Ergebnisse der Optimierung analysiert. Es konnte eine Reduktion des Energieverbrauchs für die untersuchte Bewegungsbahn um etwa 7 % erzielt werden. Des Weiteren wird gezeigt, dass der Optimierer für eine Verwendung, die über den Rahmen der Bachelorarbeit hinausgeht, die Definition zusätzlicher Nebenbedingungen benötigt, um sicherzustellen, dass die kinematischen und dynamischen Grenzen des Roboters in der Optimierung berücksichtigt werden.

In Kapitel 6 Validierung der Optimierungsergebnisse werden die in der Simulation berechneten Energieeinsparungen am realen System mithilfe des optimierten Parametervektors überprüft. Die Ergebnisse der Optimierung konnten im Versuch plausibilisiert werden, wobei eine Energieeinsparung von 11,7 % erzielt wurde.

Die Arbeit schließt mit einer Bewertung der Ergebnisse in Kapitel 7 ab. Für die optimierte Bewegungsbahn wird eine Erhöhung der Bewegungsdauer um 0,2 s festgestellt. Diese ist im Kontext einer Produktion unter Berücksichtigung vor- und nachgelagerter Bearbeitungsschritte zu bewerten. Für den isoliert betrachteten Roboter wird die Zunahme der Bewegungsdauer als tolerierbar eingestuft. Die Zielsetzung der Arbeit, einen Via-Punkt modellbasiert zu optimieren, wurde erreicht. Die Forschungsfrage, ob ein für die Produktion entwickeltes Roboterprogramm durch Hinzufügen und Verschieben von Via-Punkten energetisch optimiert werden kann, ohne die Bewegungsdauer signifikant zu erhöhen, konnte bestätigt werden. Die prognostizierte Energieeinsparung von etwa

8 Zusammenfassung und Ausblick

10 %, die in der Ausgangssituation mithilfe des DOE ermittelt wurde, konnte sowohl im Modell als auch in einem praktischen Versuch nachgewiesen werden.

In der Arbeit wurde eine Stromwandlung aus Bewegungen mit der Schwerkraft über die Funktion der Antriebe als Generator nicht betrachtet. Im Rahmen eines Literaturüberblicks erfolgt der Verweis auf Forschungsarbeiten, die diese Lücke schließen. Für weitere Untersuchungen zur Optimierung der Roboterbahn sollte zunächst eine Potenzialanalyse durchgeführt werden, in der die Skalierbarkeit des Ansatzes ermittelt wird. Um das eingangs beschriebene Ziel der kontinuierlichen Effizienzsteigerung zu erreichen, ist darüber hinaus die Festlegung der Verantwortlichkeiten für die Umsetzung der Optimierung von Bedeutung. Anzustreben ist die Berücksichtigung der Energieeffizienz neben der Taktzeit bereits bei der Erstprogrammierung der Bewegungsbahn bzw. vor der Inbetriebnahme eines neuen Programmablaufs. Empfehlenswert ist hier eine Anwendung in der Bewegungsabläufe manuell markiert werden, auf deren Basis anschließend eine automatisierte Erweiterung um Via-Punkte und deren Optimierung erfolgt. Ein wichtiges Kriterium ist dabei die Gewährleistung von Kollisionsfreiheit. Wurden im Optimierer bereits dynamische, z. B. kinetische Randbedingungen berücksichtigt, ist im Anschluss an die Optimierung lediglich eine geometrische Überprüfung der Kollisionsfreiheit in einer Simulationsssoftware erforderlich. Ein zweiter Ansatz ist die Sicherstellung der Kollisionsfreiheit vor der Optimierung. Vorab könnte ein kartesischer Raum definiert werden, in dem Kollisionsfreiheit vorliegt. Mit dem Ziel, dies möglichst automatisiert ablaufen zu lassen, ist z. B. der Einsatz von Bilderkennungsverfahren zur Bestimmung von EE-Geometrien geeignet, für die keine CAD-Daten vorliegen.

Literatur

- [Bjo+13] Staffan Bjorkenstam u. a. „Energy efficient and collision free motion of industrial robots using optimal control“. In: *2013 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2013, S. 510–515. ISBN: 978-1-4799-1515-6. DOI: [10.1109/CoASE.2013.6654025](https://doi.org/10.1109/CoASE.2013.6654025).
- [CWV17] Giovanni Carabin, Erich Wehrle und Renato Vidoni. „A Review on Energy-Saving Optimization Methods for Robotic and Automatic Systems“. In: *Robotics* 6.4 (2017), S. 39. DOI: [10.3390/robotics6040039](https://doi.org/10.3390/robotics6040039). URL: <https://www.mdpi.com/2218-6581/6/4/39>.
- [Egg+06] Kai Eggers u. a. „Verfahren zur echtzeitüberwachung des leistungsverlaufs und/oder energieflusses einer antriebseinrichtung und antriebseinrichtung sowie computerprogramm hierzu: Europäische Patentanmeldung“. Pat. EP 3 255 513 A1. 9.06.2016.
- [Egg19] Kai Benjamin Eggers. „Energieeffizienter Betrieb von Industrierobotern“. Dissertation. Hannover: Gottfried Wilhelm Leibniz Universität, 2019. DOI: [10.15488/5332](https://doi.org/10.15488/5332). URL: <https://www.repo.uni-hannover.de/handle/123456789/5379>.
- [Eng08] Robert Engelke. „Modellierung und Optimierung von Robotern mit einseitigen Bindungen und lokalen Verspannungen“. Dissertation. München: Technische Universität München, 2008.
- [GRN13] Hubert Gattringer, Roland Riepl und Matthias Neubauer. „Optimizing Industrial Robots for Accurate High-Speed Applications“. In: *Journal of Industrial Engineering* 2013 (2013), S. 1–12. ISSN: 2314-4882. DOI: [10.1155/2013/625638](https://doi.org/10.1155/2013/625638).

Literatur

- [Han+12] Christian Hansen u. a. „Enhanced approach for energy-efficient trajectory generation of industrial robots“. In: *2012 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2012, S. 1–7. ISBN: 978-1-4673-0430-6. DOI: 10.1109/CoASE.2012.6386343.
- [KK] Wolfgang Kemmetmueller und Andreas Kugi. *Modellbildung: Vorlesung und Übung SS 2023*.
- [Lin+18] Jianjie Lin u. a. „An Efficient and Time-Optimal Trajectory Generation Approach for Waypoints Under Kinematic Constraints and Error Bounds“. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, S. 5869–5876. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8593577.
- [LY21] David G. Luenberger und Yinyu Ye. *Linear and Nonlinear Programming*. Bd. 228. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-85449-2. DOI: 10.1007/978-3-030-85450-8.
- [Non+22] Kazuki Nonoyama u. a. „Energy-Efficient Robot Configuration and Motion Planning Using Genetic Algorithm and Particle Swarm Optimization“. In: *Energies* 15.6 (2022), S. 2074. DOI: 10.3390/en15062074.
- [NW06] Jorge Nocedal und Stephen J. Wright. *Numerical optimization*. Second edition. Springer series in operation research and financial engineering. New York, NY: Springer, 2006. ISBN: 978-0387-30303-1. URL: <http://www.loc.gov/catdir/enhancements/fy0818/2006923897-d.html>.
- [o. 19] o. V. *KUKA.RCS Module 8.6: Installations- und Bedienanleitung*. Augsburg, 2019. URL: https://xpert.kuka.com/service-express/portal/project1_p/document/kuka-project1_p-common_PB13326_de?context=%7B%22filter%22%3A%7B%7D,%22text%22%3A%22KUKA.RobotSensorInterface%205.0%22,%22page%22%3A1,%22useExpertQuery%22%3A0%7D.
- [o. 20] o. V. *KUKA.RobotSensorInterface 5.0: Für KUKA System Software 8.7*. Augsburg, 2020. URL: https://xpert.kuka.com/service-express/portal/project1_p/document/kuka-project1_p-common_PB13326_de?context=%7B%22filter%22%3A%7B%7D,%22text%22%3A%22KUKA.RobotSensorInterface%205.0%22,%22page%22%3A1,%22useExpertQuery%22%3A0%7D.

Literatur

- [o. 22] o. V. *Aktualisierte Umwelterklärung 2022 Mercedes-Benz AG Standort Sindelfingen*. Hrsg. von Christoph Stapmanns und Karolina Schuhl. 2022. URL: <https://group.mercedes-benz.com/nachhaltigkeit/betrieblicher-umweltschutz/umwelterklaerungen/>.
- [o. 23a] o. A. *Optimization Toolbox: User's Guide*. 2023. URL: https://de.mathworks.com/help/pdf_doc/optim/index.html.
- [o. 23b] o. V. *KUKA System Software 8.7: Bedien- und Programmieranleitung für Systemintegratoren*. Augsburg, 2023.
- [Par+15] Paryanto u. a. „Reducing the energy consumption of industrial robots in manufacturing systems“. In: *The International Journal of Advanced Manufacturing Technology* 78.5-8 (2015), S. 1315–1328. ISSN: 0268-3768. DOI: 10.1007/s00170-014-6737-z.
- [Pel+11] Marcello Pellicciari u. a. „A minimal touch approach for optimizing energy efficiency in pick-and-place manipulators“. In: *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, S. 100–105. ISBN: 978-1-4577-1159-6. DOI: 10.1109/ICAR.2011.6088620.
- [Pel+15] M. Pellicciari u. a. „AREUS — Innovative hardware and software for sustainable industrial robotics“. In: *2015 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2015, S. 1325–1332. ISBN: 978-1-4673-8183-3. DOI: 10.1109/CoASE.2015.7294282.
- [PLB15] Markos Papageorgiou, Marion Leibold und Martin Buss. *Optimierung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. ISBN: 978-3-662-46935-4. DOI: 10.1007/978-3-662-46936-1.
- [RHG13] Rüdiger Reinhardt, Armin Hoffmann und Tobias Gerlach. *Nichtlineare Optimierung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. ISBN: 978-3-8274-2948-3. DOI: 10.1007/978-3-8274-2949-0.
- [Rie22] Jochen M. Rieber. *Einführung in die Robotik: Vorlesung im Studium Elektrotechnik / Vorlesung im Studium Elektrotechnik / Automation Kapitel 4b*. Ravensburg/Friedrichshafen, 2022.
- [RS16] Stefan Roth und Achim Stahl. *Mechanik und Wärmelehre*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. ISBN: 978-3-662-45303-2. DOI: 10.1007/978-3-662-45304-9.

Literatur

- [SHV20] Mark W. Spong, Seth Hutchinson und Mathukumalli Vidyasagar. *Robot modeling and control*. Second edition. Hoboken, NJ: Wiley, 2020. ISBN: 978-1119523994.
- [Sic+09] Bruno Siciliano u. a. *Robotics: Modelling, Planning and Control*. London: Springer London, 2009. ISBN: 978-1-84628-641-4. DOI: 10.1007/978-1-84628-642-1.
- [SRB08] R. Saravanan, S. Ramabalan und C. Balamurugan. „Evolutionary optimal trajectory planning for industrial robot with payload constraints“. In: *The International Journal of Advanced Manufacturing Technology* 38.11-12 (2008), S. 1213–1226. ISSN: 0268-3768. DOI: 10.1007/s00170-007-1169-7.
- [SSP02] Eleftheria S. Sergaki, George S. Stavrakakis und Anastasios D. Pouliezos. „Optimal Robot Speed Trajectory by Minimization of the Actuator Motor Electromechanical Losses Eleftheria S. Sergaki, George S. Stavrakakis & Anastasios D. Pouliezos“. In: *Journal of Intelligent and Robotic Systems* 33.2 (2002), S. 187–207. ISSN: 09210296. DOI: 10.1023/A:1014643401778.
- [Whi+06] Steven White u. a. *Windows Sockets 2*. 13.06.2023. URL: <https://learn.microsoft.com/de-de/windows/win32/winsock/windows-sockets-start-page-2>.
- [Zia+17] Zygimantas Ziaukas u. a. „Optimizing PTP Motions of Industrial Robots through Addition of Via-points“. In: *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics*. SCITEPRESS - Science and Technology Publications, 2017, S. 527–538. ISBN: 978-989-758-263-9.

Abbildungsverzeichnis

2.1	KUKA KR210 R2700-2	8
2.3	DH-Konvention - Festlegung der Koordinatensysteme	9
2.4	Generisches Verbindungsglied [Sic+09, S. 283]	13
2.5	Ersatzschaltbild der aufgenommenen Netzleistung [Egg19, S. 20]	17
3.1	Simulierte Gelenkwinkel der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	21
3.2	Gemessene Gelenkwinkel der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	21
3.3	Simulierte Winkelgeschwindigkeit der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	22
3.4	Gemessene Winkelgeschwindigkeit der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	22
3.5	Simulierte Winkelbeschleunigung der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	23
3.6	Gemessene Winkelbeschleunigung der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	23
4.1	Simulierter Anteil der Gewichtskraft an den Drehmomenten in der Startposition der untersuchten Bewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	27
4.2	Simulierter Drehmomentverlauf ohne Gewichtsausgleich für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	29
4.3	Simulierter Drehmomentverlauf inkl. näherungsweise modellierten Gewichtsausgleich für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	29

Abbildungsverzeichnis

4.4	Gemessener Drehmomentverlauf für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	30
4.5	Simulierte Mechanische Leistung für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	31
4.6	Gemessene Mechanische Leistung für die Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	31
5.1	Zielfunktionswerte der Optimierungsiterationen	38
5.2	Vergleich der simulierten Leistungsaufnahme für die Initialbewegung und die optimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	39
5.3	Vergleich der simulierten Gelenkwinkel für die Initialbewegung und die optimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	40
5.4	Vergleich der simulierten Winkelgeschwindigkeit für die Initialbewegung und die optimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	41
5.5	Vergleich der simulierten Winkelgeschwindigkeit für die energieoptimierte Bewegungsbahn und die justierte, energieoptimierte Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	42
5.6	Vergleich der simulierten Leistungsaufnahme für die energieoptimierte Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	43
6.1	Vergleich der summierten Leistungsaufnahme für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	46
6.2	Vergleich des mechanischen Energieverbrauchs für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	47
E.1	Simulierte Gelenkwinkel der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	97

Abbildungsverzeichnis

E.2	Simulierte Winkelgeschwindigkeit der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	97
E.3	Simulierte Winkelbeschleunigung der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	98
E.4	Simulierte Drehmomente der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	98

Tabellenverzeichnis

2.1	Denavit-Hartenberg Parameter	10
3.1	Winkeldefinition der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand	19
5.1	Winkeldefinition der Initialbewegung vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand inklusive der energieoptimierten, justierten Via-Punkte	38
7.1	Simulierter, mechanischer Energieverbrauch für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand .	49
7.2	Gemessener, mechanischer Energieverbrauch für die initiale Bewegungsbahn und die justierte energieoptimierte, Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand .	49
C.1	Massen der Verbindungsglieder in [kg]	70
C.2	Trägheitstensoren in $\left[\frac{\text{kg}}{\text{m}^2} \right]$	70
C.3	Getriebeübersetzung	70

Anhang A - Planung

Planung Bachelorarbeit

Aa Name	↗ Blocked by	↗ Blocking	📅 Datum	⌚ Status
<u>Literaturreview</u>		<u>DH-Konvention</u>	@July 10, 2023 → July 12, 2023	Abgeschlossen
<u>DH-Konvention</u>	<u>Literaturreview</u>	<u>Inverse Dynamik in Matlab implementieren</u>	@July 13, 2023 → July 14, 2023	Abgeschlossen
<u>Inverse Dynamik in Matlab implementieren</u>	<u>DH-Konvention</u>	<u>Trajektorie-Planung</u>	@July 17, 2023 → August 1, 2023	Abgeschlossen
<u>Trajektorie-Planung</u>	<u>Inverse Dynamik in Matlab implementieren</u>	<u>Modell validieren, ggf. nachbessern</u>	@August 2, 2023 → August 4, 2023	Abgeschlossen
<u>Modell validieren..ggf. nachbessern</u>	<u>Trajektorie-Planung</u>	<u>Zielfunktion, Optimierungsproblem, Nebenbedingungen definieren</u>	@August 7, 2023 → August 11, 2023	Abgeschlossen
<u>Zielfunktion, Optimierungsproblem, Nebenbedingungen definieren</u>	<u>Modell validieren..ggf. nachbessern</u>	<u>Solver und Algorithmus selektieren</u>	@August 14, 2023 → August 18, 2023	Abgeschlossen
<u>Solver und Algorithmus selektieren</u>	<u>Zielfunktion, Optimierungsproblem, Nebenbedingungen definieren</u>	<u>Optimierung durchführen</u>	@August 21, 2023 → August 23, 2023	Abgeschlossen
<u>Optimierung durchführen</u>	<u>Solver und Algorithmus selektieren</u>	<u>Messsystem und Steuerungsarchitektur vorbereiten, Server aufsetzen</u>	@August 24, 2023 → August 25, 2023	Abgeschlossen
<u>Messsystem und Steuerungsarchitektur vorbereiten, Server aufsetzen</u>	<u>Optimierung durchführen</u>	<u>Energieoptimierung validieren</u>	@August 28, 2023 → September 1, 2023	Abgeschlossen
<u>Energieoptimierung validieren</u>	<u>Messsystem und Steuerungsarchitektur vorbereiten, Server aufsetzen</u>	<u>Bewertung der Ergebnisse</u>	@September 4, 2023 → September 5, 2023	Abgeschlossen
<u>Bewertung der Ergebnisse</u>	<u>Energieoptimierung validieren</u>		@September 6, 2023 → September 8, 2023	Abgeschlossen
<u>Blocker</u>			@September 11, 2023 → September 22, 2023	

Anhang B - Theoretische Grundlagen

B.1 Kinematische Kette

Als kinematische Kette wird die Zusammensetzung eines Roboterarms aus $n+1$ Gliedern und n Gelenken definiert. Das letzte Glied der kinematischen Kette wird als Endeffektor bezeichnet. Die Glieder werden von der festen Basis (Glied 0) bis n durchnummieriert. Mit jedem Glied $i - 1$ wird das Gelenk i starr verbunden. Des Weiteren wird jedem Glied i ein fest verbundenes Koordinatensystem $o_i x_i y_i z_i$ zugeordnet. Eine Aktuierung des Gelenks i führt zu einer Bewegung von Glied i und Koordinatensystem $KS\{i\}$. Die Gelenke weisen einen Freiheitsgrad (degree-of-freedom (DOF)) q_i auf und können als Dreh- oder Schubgelenke ausgeführt werden. Die Gelenkvariable q_i entspricht dem Rotationswinkel θ_i bzw. der Translation d_i . [SHV20, S. 75]

B.2 Euklidische Gruppe

$$SE(3) = \left\{ \mathbf{T} \mid \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{o} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \right\}, \quad \mathbf{R} \in R^{3 \times 3}, \quad \mathbf{r} \in R^3 \quad (B.1)$$

[SHV20, S. 534]

B.3 Geschwindigkeits-Kinematik

Die Herleitung der Zusammenhänge ist [KK, S. 79 f.] und [SHV20, S. 106] entnommen. Die Rotation eines Starrkörpers um eine feste Drehachse wird über die Beziehung $\boldsymbol{\omega} = \dot{\theta}\mathbf{k}$ ausgedrückt. Dabei sind $\dot{\theta}$ die zeitliche Ableitung des Gelenkwinkels θ und \mathbf{k} der Einheitsvektor der Drehachse. Die lineare Geschwindigkeit eines Punktes P, welcher um die Drehachse rotiert, ergibt sich zu $\mathbf{v} = \boldsymbol{\omega} \times \mathbf{r}$. Wobei der Vektor \mathbf{r} die Position des Punktes P orthogonal auf der Drehachse \mathbf{k} angibt. [SHV20, S. 102]

B.3.1 Schiefsymmetrische Matrizen

Die Berechnung der Geschwindigkeits-Kinematik lässt sich durch Verwendung Schiefsymmetrischer Matrizen vereinfachen. Eine $n \times n$ Matrix \mathbf{S} ist schief symmetrisch, wenn

$$\mathbf{S}^T + \mathbf{S} = 0. \quad (\text{B.2})$$

Damit hat jede schiefsymmetrische 3×3 Matrix die Form

$$\mathbf{S} = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix}. \quad (\text{B.3})$$

Für weitere Eigenschaften schiefsymmetrischer Matrizen wird auf die referenzierte Literatur [SHV20, S. 104] verwiesen. Für die Drehmatrix $\mathbf{R} = \mathbf{R}(\theta) \in SO(3)$, deren Elemente Funktionen der Drehwinkel $\theta(t)$ sind, gilt

$$\mathbf{R}(\theta)\mathbf{R}(\theta)^T = \mathbf{E}. \quad (\text{B.4})$$

B.3.2 Drehwinkelgeschwindigkeit

Die Ableitung beider Terme der Gleichung B.4 ergibt

$$\frac{d}{d\theta} [\mathbf{R}(\theta)\mathbf{R}(\theta)^T] = \left[\frac{d}{d\theta} \mathbf{R}(\theta) \right] \mathbf{R}(\theta)^T + \mathbf{R}(\theta) \left[\frac{d}{d\theta} \mathbf{R}(\theta)^T \right] = 0 \quad (\text{B.5})$$

Unter Berücksichtigung der Eigenschaft schiefsymmetrischer Matrizen, siehe Gleichung B.2 folgt

$$\mathbf{S} = \left[\frac{d}{d\theta} \mathbf{R}(\theta) \right] \mathbf{R}(\theta)^T = -\mathbf{R}(\theta) \left[\frac{d}{d\theta} \mathbf{R}(\theta)^T \right]. \quad (\text{B.6})$$

Daraus folgt

$$\frac{d}{d\theta} \mathbf{R}(\theta) = \mathbf{S} \mathbf{R}(\theta) \quad (\text{B.7})$$

Die zeitliche Ableitung der Rotationsmatrix $\mathbf{R} = \mathbf{R}(t) \in SO(3)$ lautet

$$\frac{d}{dt} \mathbf{R}(t) = \mathbf{S}(\boldsymbol{\omega}(t)) \mathbf{R}(t) \quad (\text{B.8})$$

Für Verkettete Rotationen gilt

$$\frac{d}{dt} \mathbf{R}_n^0(t) = \mathbf{S}(\boldsymbol{\omega}_{0,n}^0(t)) \mathbf{R}_n^0(t) \quad (\text{B.9})$$

$$\boldsymbol{\omega}_{0,n}^0 = \boldsymbol{\omega}_{0,1}^0 + \mathbf{R}_1^0 \boldsymbol{\omega}_{1,2}^1 + \mathbf{R}_2^0 \boldsymbol{\omega}_{2,3}^2 + \dots + \mathbf{R}_{n-1}^0 \boldsymbol{\omega}_{n-1,n}^{n-1} \quad (\text{B.10})$$

B.3.3 Lineare Geschwindigkeit

Für die Lineare Geschwindigkeit eines Punktes werden nachfolgend zwei Szenarien unterschieden. In beiden Fällen ist der Punkt P fest mit dem KS*{i}* verbunden. Im ersten Fall wird eine Rotation des KS*{i}* relativ zum KS*{0}* betrachtet.

$$\mathbf{p}^0 = \mathbf{R}_i^0(t) \mathbf{p}^i \quad (\text{B.11})$$

$$\dot{\mathbf{p}}^0 = \boldsymbol{\omega}_{0,i}^0 \times \mathbf{p}^0 \quad (\text{B.12})$$

Im zweiten Fall bewegt sich KS*{i}* rotatorisch und translatorisch relativ zu KS*{0}*.

$$\mathbf{p}^0 = \mathbf{R}_i^0(t) \mathbf{p}^i + \mathbf{o}_i^0 \quad (\text{B.13})$$

$$\dot{\mathbf{p}}^0 = \boldsymbol{\omega}_{0,i}^0 \times \mathbf{p}^0 + \dot{\mathbf{o}}_i^0 \quad (\text{B.14})$$

Auf eine Beschreibung des Szenarios, dass sich P gegenüber KS*{i}* bewegt, wird verzichtet.

Jacobi-Matrizen

Die lineare Geschwindigkeit \mathbf{v}_n^0 des Endeffektors, sowie seine Winkelgeschwindigkeit $\boldsymbol{\omega}_{0,n}^0$ ausgedrückt in $\text{KS}\{0\}$ lassen sich über die $3 \times n$ -Jacobi-Matrizen \mathbf{J}_v und \mathbf{J}_ω berechnen.

$$\mathbf{v}_n^0 = \mathbf{J}_v \dot{\mathbf{q}} = [\mathbf{J}_{v_1} \ \mathbf{J}_{v_2} \ \dots \ \mathbf{J}_{v_n}] \dot{\mathbf{q}} \quad (\text{B.15})$$

$$\boldsymbol{\omega}_{0,i}^0 = \mathbf{J}_\omega \dot{\mathbf{q}} = [\mathbf{J}_{\omega_1} \ \mathbf{J}_{\omega_2} \ \dots \ \mathbf{J}_{\omega_n}] \dot{\mathbf{q}} \quad (\text{B.16})$$

$$\mathbf{J}_{vi} = \begin{cases} \mathbf{z}_{i-1}^0 \times (\mathbf{o}_n^0 - \mathbf{o}_{i-1}^0) & \text{falls Gelenk } i \text{ vom Typ R} \\ \mathbf{z}_{i-1}^0 & \text{falls Gelenk } i \text{ vom Typ T} \end{cases} \quad (\text{B.17})$$

$$\mathbf{J}_{\omega i} = \begin{cases} \mathbf{z}_{i-1}^0 = \mathbf{R}_{i-1}^0 \mathbf{e}_3 & \text{falls Gelenk } i \text{ vom Typ R} \\ \mathbf{0}_{3 \times 1} & \text{falls Gelenk } i \text{ vom Typ T} \end{cases} \quad (\text{B.18})$$

- Ein Gelenk vom Typ R ist ein Rotationsgelenk.
- Ein Gelenk vom Typ T ist ein Translationsgelenk.

[Rie22]

Anhang B - Roboterdaten

C.1 KR210 2700-2 Datenblatt

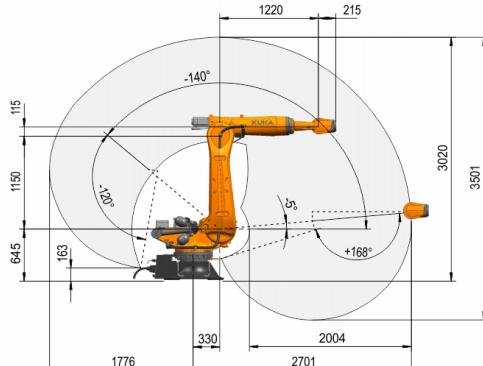


KR 210 R2700-2



Arbeitsraumgrafik

Dimensions: mm



Technische Daten

Maximale Reichweite	2701 mm
Nenn-Traglast	210 kg
Maximale Traglast	275 kg
Maximale Zusatzlast Karussell / Schwinge / Arm	300 kg / 130 kg / 150 kg
Positionswiederholgenauigkeit (ISO 9283)	± 0,05 mm
Anzahl Achsen	6
Einbaulage	Boden
Aufstellfläche	754 mm x 754 mm
Gewicht	ca. 1077 kg

Achssdaten

Bewegungsbereich	
A1	±185 °
A2	-140 ° / -5 °
A3	-120 ° / 168 °
A4	±350 °
A5	±125 °
A6	±350 °
Geschwindigkeit bei Nenn-Traglast	
A1	120 °/s
A2	115 °/s
A3	112 °/s
A4	179 °/s
A5	172 °/s
A6	220 °/s

Betriebsbedingungen

Umgebungstemperatur bei Betrieb 0 °C bis 55 °C (273 K bis 328 K)

Schutzart

Schutzart (IEC 60529)	IP65
Schutzart Roboterhand (IEC 60529)	IP65 / IP67

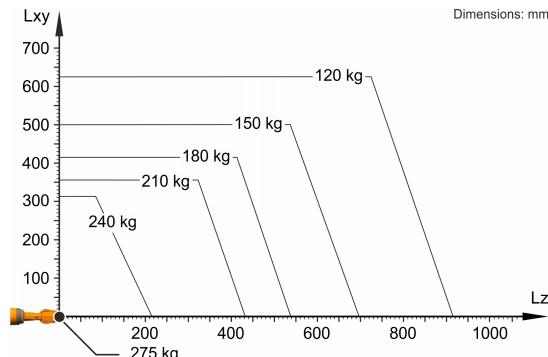
Steuerung

Steuerung	KR C5; KR C4
-----------	-----------------

Zertifikate

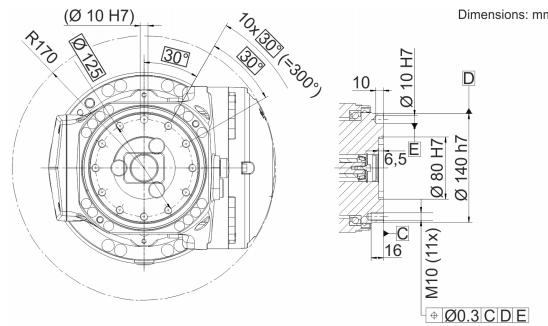
ESD-Anforderungen IEC61340-5-1; ANSI/ESD S20.20

Traglastdiagramm



Der KR 210 R2700-2 ist ausgelegt für eine Nenn-Traglast von 210 kg, um die Performance und Dynamik des Roboters optimal zu nutzen. Nur wenn die Lage des Massenschwerpunkts 0 mm und eine für den Lastfall optimierte Zusatzlast angebracht sind, gilt die maximale Traglast von 275 kg. Der spezifische Lastfall muss mit KUKA.Load oder KUKA Compose überprüft werden. Für weitere Beratung steht der KUKA Support zur Verfügung.

Anbauflansch



C.2 Systemparameter

Die Massen m_i der Starrkörper i , siehe Tabelle C.1 sind dem CAD-Modell des Herstellers entnommen. Hierbei wird angenommen, dass der Werkstoff korrekt zugeordnet ist.

Tabelle C.1: Massen der Verbindungsglieder in [kg]

$m_1 = 535$	$m_2 = 696,3$	$m_3 = 361,6$	$m_4 = 39,676$	$m_5 = 53,619$	$m_6 = 4,528$
-------------	---------------	---------------	----------------	----------------	---------------

Die Trägheitstensoren, werden vom CAD-Modell, bezogen auf das KS{0} vorgegeben. Gleichung 2.10 zeigt, wie diese einmalig über eine Ähnlichkeitstransformation auf das Körperfeste Koordinatensystem KS{i} transformiert werden. In der Tabelle C.2 sind die Trägheitstensoren bezogen auf das KS{0} angegeben.

Tabelle C.2: Trägheitstensoren in $\left[\frac{\text{kg}}{\text{m}^2} \right]$

$I_{1xx} = 17,3$	$I_{2xx} = 138,14$	$I_{3xx} = 3,03$	$I_{4xx} = 0,1$	$I_{5xx} = 0,48$	$I_{6xx} = 0,01$
$I_{1xy} = -2,71$	$I_{2xy} = -0,01$	$I_{3xy} = -1,39$	$I_{4xy} = -0,03$	$I_{5xy} = 0,12$	$I_{6xy} = 0$
$I_{1xz} = 1,67$	$I_{2xz} = 0,39$	$I_{3xz} = -0,05$	$I_{4xz} = 0$	$I_{5xz} = 0$	$I_{6xz} = 0$
$I_{1yy} = 32,67$	$I_{2yy} = 136,66$	$I_{3yy} = 30,03$	$I_{4yy} = 0,7$	$I_{5yy} = 0,42$	$I_{6yy} = 0,01$
$I_{1yz} = -0,49$	$I_{2yz} = 9,23$	$I_{3yz} = -0,21$	$I_{4yz} = 0$	$I_{5yz} = 0$	$I_{6yz} = 0$
$I_{1zz} = 29,45$	$I_{2zz} = 11,84$	$I_{3zz} = 29,69$	$I_{4zz} = 0,7$	$I_{5zz} = 0,68$	$I_{6zz} = 0,01$

Nachfolgend ist die Lage der Massenschwerpunkte C_i^0 im KS{0} angegeben.

$$\begin{aligned} r_{0,C_1}^0 &= [-30, 103, 441, 452, 213] \text{ mm} \\ r_{0,C_2}^0 &= [330, 979, -222, 585, 1094, 239] \text{ mm} \\ r_{0,C_3}^0 &= [705, 894, -8, 126, 1908, 436] \text{ mm} \\ r_{0,C_4}^0 &= [1383, 185, 4, 492, 1909, 983] \text{ mm} \\ r_{0,C_5}^0 &= [1601, 767, 33, 833, 1909, 955] \text{ mm} \\ r_{0,C_6}^0 &= [1743, 222, -0, 007, 1910051] \text{ mm} \end{aligned}$$

Die Getriebeübersetzung in Tabelle C.3 ist der Datei Machine Data (MADA) der Robotersteuerung entnommen.

Tabelle C.3: Getriebeübersetzung

$i_1 = -\frac{7}{1798}$	$i_2 = -\frac{17}{4576}$	$i_3 = \frac{3}{754}$	$i_4 = -\frac{55}{10387}$	$i_5 = -\frac{483}{91834}$	$i_6 = \frac{49400}{6485103}$
-------------------------	--------------------------	-----------------------	---------------------------	----------------------------	-------------------------------

Anhang C - MATLAB-Implementierung

D.1 DH-Transformation

```
function [T,R,R0i] = dhtrafo(q1, q2, q3, q4, q5, q6)
%Vorwärtskinematik
% Denavit–Hartenberg–Konvention

theta1 = -q1;
theta2 = q2+deg2rad(-90);
theta3 = q3;
theta4 = q4;
theta5 = q5;
theta6 = q6+deg2rad(90);

alpha1 = deg2rad(-90);
alpha2 = deg2rad(0);
alpha3 = deg2rad(90);
alpha4 = deg2rad(-90);
alpha5 = deg2rad(-90);
alpha6 = deg2rad(0);

a1 = 330/1000;
a2 = 1150/1000;
a3 = 115/1000;
a4 = 0;
a5 = 0;
a6 = 0;
```

```

d1 = 645/1000;
d2 = 0;
d3 = 0;
d4 = -1220/1000;
d5 = 0;
d6 = 215/1000;

T01 = [ cos(theta1), -sin(theta1)*cos(alpha1), sin(theta1)*sin(alpha1), a1
    *cos(theta1);
        sin(theta1), cos(theta1)*cos(alpha1), -cos(theta1)*sin(alpha1), a1*sin
    (theta1);
        0, sin(alpha1), cos(alpha1), d1;
        0, 0, 0, 1];

T12 = [ cos(theta2), -sin(theta2)*cos(alpha2), sin(theta2)*sin(alpha2), a2
    *cos(theta2);
        sin(theta2), cos(theta2)*cos(alpha2), -cos(theta2)*sin(alpha2), a2*sin
    (theta2);
        0, sin(alpha2), cos(alpha2), d2;
        0, 0, 0, 1];

T23 = [ cos(theta3), -sin(theta3)*cos(alpha3), sin(theta3)*sin(alpha3), a3
    *cos(theta3);
        sin(theta3), cos(theta3)*cos(alpha3), -cos(theta3)*sin(alpha3), a3*sin
    (theta3);
        0, sin(alpha3), cos(alpha3), d3;
        0, 0, 0, 1];

T34 = [ cos(theta4), -sin(theta4)*cos(alpha4), sin(theta4)*sin(alpha4), a4
    *cos(theta4);
        sin(theta4), cos(theta4)*cos(alpha4), -cos(theta4)*sin(alpha4), a4*sin
    (theta4);
        0, sin(alpha4),
        cos(alpha4), d4;
        0, 0, 0, 1];

T45 = [ cos(theta5), -sin(theta5)*cos(alpha5), sin(theta5)*sin(alpha5), a5
    *cos(theta5);

```

```

sin(theta5), cos(theta5)*cos(alpha5), -cos(theta5)*sin(alpha5), a5*sin(theta5);
0, sin(alpha5), cos(alpha5), d5;
0, 0, 0, 1];

T56 = [ cos(theta6), -sin(theta6)*cos(alpha6), sin(theta6)*sin(alpha6), a6*cos(theta6);
sin(theta6), cos(theta6)*cos(alpha6), -cos(theta6)*sin(alpha6), a6*sin(theta6);
0, sin(alpha6), cos(alpha6), d6;
0, 0, 0, 1];

T = cat(3, T01, T12, T23, T34, T45, T56);

R01 = T01(1:3,1:3);
R12 = T12(1:3,1:3);
R23 = T23(1:3,1:3);
R34 = T34(1:3,1:3);
R45 = T45(1:3,1:3);
R56 = T56(1:3,1:3);

T02 = T01*T12;
T03 = T02*T23;
T04 = T03*T34;
T05 = T04*T45;
T06 = T05*T56;

R02 = T02(1:3,1:3);
R03 = T03(1:3,1:3);
R04 = T04(1:3,1:3);
R05 = T05(1:3,1:3);
R06 = T06(1:3,1:3);

R = cat(3, R01, R12, R23, R34, R45, R56);
R0i = cat(3, R01, R02, R03, R04, R05, R06);

end

```

D.2 Parameter

```
function [I , rec , rae ,m,G] = parameter()
% Berechnung konstante Parameter
```

D.2.1 Vorwärtsskinematik für die Ausgangsstellung

```
[T,~,~] = dhtrafo(0, 0, 0, 0, 0, 0);
```

```
T01 = T(:,:,1);
T12 = T(:,:,2);
T23 = T(:,:,3);
T34 = T(:,:,4);
T45 = T(:,:,5);
T56 = T(:,:,6);
T06 = T01*T12*T23*T34*T45*T56;
T05 = T01*T12*T23*T34*T45;
T04 = T01*T12*T23*T34;
T03 = T01*T12*T23;
T02 = T01*T12;
```

```
R01 = T01(1:3,1:3);
R02 = T02(1:3,1:3);
R03 = T03(1:3,1:3);
R04 = T04(1:3,1:3);
R05 = T05(1:3,1:3);
R06 = T06(1:3,1:3);
```

D.2.2 Trägheitsmoment I_i^0 am Massezentrum in $\frac{\text{kg}}{\text{m}^2}$

```
I1xx = 17.298;
I1xy = -2.711;
```

```
I1xz = 1.672;
I1yx = -2.711;
I1yy = 32.671;
I1yz = -0.493;
I1zx = 1.672;
I1zy = -0.493;
I1zz = 29.447;

I2xx = 138.135;
I2xy = -0.012;
I2xz = 0.387;
I2yx = -0.012;
I2yy = 136.664;
I2yz = 9.231;
I2zx = 0.387;
I2zy = 9.231;
I2zz = 11.838;

I3xx = 3.031;
I3xy = -1.386;
I3xz = -0.045;
I3yx = -1.386;
I3yy = 30.03;
I3yz = -0.21;
I3zx = -0.045;
I3zy = -0.21;
I3zz = 29.693;

I4xx = 0.099;
I4xy = -0.032;
I4xz = -0.001;
I4yx = -0.032;
I4yy = 0.701;
I4yz = -9.367E-06;
I4zx = -0.001;
I4zy = -9.367E-06;
I4zz = 0.698;

I5xx = 0.481;
I5xy = 0.118;
```

```
I5xz = 0.00;
I5yx = 0.118;
I5yy = 0.424;
I5yz = 0.00;
I5zx = 0.00;
I5zy = 0.00;
I5zz = 0.675;

I6xx = 0.011;
I6xy = 5.143E-07;
I6xz = -2.003E-06;
I6yx = 5.143E-07;
I6yy = 0.006;
I6yz = 1.220E-07;
I6zx = -2.003E-06;
I6zy = 1.220E-07;
I6zz = 0.006;

% Ähnlichkeitstransformation zur Umrechnung in die Körperfesten
% Koordinatensysteme

I1base = [ I1xx , I1xy , I1xz ;
           I1yx , I1yy , I1yz ;
           I1zx , I1zy , I1zz ];

I1 = R01'*I1base*R01;

I2base = [ I2xx , I2xy , I2xz ;
           I2yx , I2yy , I2yz ;
           I2zx , I2zy , I2zz ];

I2 = R02'*I2base*R02;

I3base = [ I3xx , I3xy , I3xz ;
           I3yx , I3yy , I3yz ;
           I3zx , I3zy , I3zz ];

I3 = R03'*I3base*R03;

I4base = [ I4xx , I4xy , I4xz ;
```

```

I4yx , I4yy , I4yz ;
I4zx , I4zy , I4zz ];

I4 = R04'*I4base*R04;

I5base = [ I5xx , I5xy , I5xz ;
I5yx , I5yy , I5yz ;
I5zx , I5zy , I5zz ];

I5 = R05'*I5base*R05;

I6base = [ I6xx , I6xy , I6xz ;
I6yx , I6yy , I6yz ;
I6zx , I6zy , I6zz ];

I6 = R06'*I6base*R06;

I = cat(3,I1,I2,I3,I4,I5,I6);

```

D.2.3 Lage der Massenschwerpunkte im Intertialkoordinatensystem KS{0}

```

com1 = [-30.103; 4.41; 452.213]/1000;
com2 = [330.979; -222.585; 1094.239]/1000;
com3 = [705.894; -8.126; 1908.436]/1000;
com4 = [1383.185; 4.492; 1909.983]/1000;
com5 = [1601.767; 33.833; 1909.955]/1000;
com6 = [1743.222; -0.007; 1910.051]/1000;

```

D.2.4 r_{i,C_i}^i

```

r1e_c1 = (T01)\[com1;1]; r1e_c1 = r1e_c1(1:3,1);
r2e_c2 = (T02)\[com2;1]; r2e_c2 = r2e_c2(1:3,1);
r3e_c3 = (T03)\[com3;1]; r3e_c3 = r3e_c3(1:3,1);

```

```
r4e_c4 = (T04)\[com4;1]; r4e_c4 = r4e_c4(1:3,1);
r5e_c5 = (T05)\[com5;1]; r5e_c5 = r5e_c5(1:3,1);
r6e_c6 = (T06)\[com6;1]; r6e_c6 = r6e_c6(1:3,1);
rec= cat(3, r1e_c1, r2e_c2, r3e_c3, r4e_c4, r5e_c5, r6e_c6);
```

D.2.5 $r_{i-1,i}^i$

```
r1a_e = -inv(T01); r1a_e = r1a_e(1:3,4);
r2a_e = -inv(T12); r2a_e = r2a_e(1:3,4);
r3a_e = -inv(T23); r3a_e = r3a_e(1:3,4);
r4a_e = -inv(T34); r4a_e = r4a_e(1:3,4);
r5a_e = -inv(T45); r5a_e = r5a_e(1:3,4);
r6a_e = -inv(T56); r6a_e = r6a_e(1:3,4);
rae = cat(3, r1a_e, r2a_e, r3a_e, r4a_e, r5a_e, r6a_e);
```

D.2.6 Masse m_i

```
m1 = 535;
m2 = 696.3;
m3 = 361.6; % +65.3 kg für die Schlauchpaket-Halterung +50 kg für die
             Antriebe 4 und 5;
m4 = 39.676;
m5 = 53.619;
m6 = 4.528;
m = [m1,m2,m3,m4,m5,m6];
```

D.2.7 Getriebe-Übersetzung i_i

```
i1 = -7/1798;
i2 = -17/4576;
i3 = 3/754;
```

```
i4 = -55/10387;
i5 = -483/91834;
i6 = 49400/6485103;
G = [i1, i2, i3, i4, i5, i6];
```

D.3 RNEA

```
function [tau, P_mech] = rnea(q1, q2, q3, q4, q5, q6, qd1, qd2, qd3, qd4,
qd5, qd6, qdd1, qdd2, qdd3, qdd4, qdd5, qdd6)
```

D.3.1 Initialisierung

```
qd = [qd1, qd2, qd3, qd4, qd5, qd6];
qdd = [qdd1, qdd2, qdd3, qdd4, qdd5, qdd6];
omega = zeros(3,1,6);
omega_d = zeros(3,1,6);
a = zeros(3,1,6);
a_com = zeros(3,1,6);
f = zeros(3,1,6);
mu = zeros(3,1,6);
tau = zeros(1,1,6);
P_mech = zeros(1,1,6);
```

D.3.2 Vorwärtsskinematik

```
[~, R, R0i] = dhtrafo(q1, q2, q3, q4, q5, q6);
```

D.3.3 Parameter in der Reihenfolge: $I_i^i, r_{i,C_i}^i, r_{i-1,i}^i, m_i, i_i$

```
[I , rec , rae ,m,G] = parameter () ;
```

D.3.4 Winkelgeschwindigkeit ω_i^i

```
for i=1:6
if i == 1
    omega(:,:,i) = R(:,:,i)' * ([0;0;0] + qd(i)*[0;0;1]) ;
else
    omega(:,:,i) = R(:,:,i)' * (omega(:,:,i-1) + qd(i)*[0;0;1]) ;
end
end
```

D.3.5 Winkelbeschleunigung $\dot{\omega}_i^i$

```
for i=1:1:6
if i == 1
    omega_d(:,:,i) = R(:,:,i)' * ([0;0;0] + qdd(i)*[0;0;1]+ cross(qd(i)
*[0;0;0], [0;0;1])) ;
else
    omega_d(:,:,i) = R(:,:,i)' * (omega_d(:,:,i-1) + qdd(i)*[0;0;1]+ cross(qd
(i)*omega(:,:,i-1), [0;0;1])) ;
end
end
```

D.3.6 Lineare Beschleunigung \ddot{p}_i^i (Siciliano) a_e (Spong)

```
for i=1:6
if i == 1
    a(:,:,i) = R(:,:,i)' * [0;0;-9.81] + cross(omega_d(:,:,i),rae(:,:,i)) +
cross(omega(:,:,i), (cross(omega(:,:,i), rae(:,:,i)))) ;
```

```

else
    a(:,:,i) = R(:,:,i)'*a(:,:,i-1) + cross(omega_d(:,:,i),rae(:,:,i)) +
    cross(omega(:,:,i),(cross(omega(:,:,i), rae(:,:,i)))); 
end
end

```

D.3.7 Lineare Beschleunigung $\ddot{p}_{C_i}^i$ (Siciliano) a_c (Spong)

```

for i=1:1:6
    a_com(:,:,i) = a(:,:,i) + cross(omega_d(:,:,i),rec(:,:,i)) + cross(omega(:,:,i),(cross(omega(:,:,i), rec(:,:,i))));
end

```

D.3.8 Gewichtsausgleich

```

F_gravity = zeros(3, 1, 6); % Vektor für die Gewichtskraft
for i = 1:6
    F_gravity(:,:,i) = m(i) * [0; 0; -9.81]; % Gewichtskraft in
    Weltkoordinaten
    F_gravity(:,:,i) = R0i(:,:,i)' * F_gravity(:,:,i); % Gewichtskraft ins K
    örpersystem transformieren
end

for i = 3:-1:2
    if i == 3
        fgrav(:,:,i) = eye(3)*F_gravity(:,:,i);
        mugrav(:,:,i) = cross(-fgrav(:,:,i),(rae(:,:,i)+rec(:,:,i)));
    else
        fgrav(:,:,i) = R(:,:,i+1)*fgrav(:,:,i+1) + eye(3)*F_gravity(:,:,i);
        mugrav(:,:,i) = cross(-fgrav(:,:,i),(rae(:,:,i)+rec(:,:,i))) + R(:,:,i
        +1)*mugrav(:,:,i+1) + R(:,:,i+1)*cross(fgrav(:,:,i+1), rec(:,:,i));
    end
end

```

D.3.9 Kraft f_i^i

```

for i = 6:-1:1
    if i == 6
        f (:,:,i) = eye(3)*[0;0;0] + m(i)*a_com (:,:,i);
    elseif i == 2
        f (:,:,i) = R (:,:,i+1)*f (:,:,i+1) + m(i)*a_com (:,:,i);
    else
        f (:,:,i) = R (:,:,i+1)*f (:,:,i+1) + m(i)*a_com (:,:,i);
    end
end

```

D.3.10 Drehmoment μ_i^i

Übertragung des Drehmoments von der zweiten auf die ersten Achse ist zu Null gesetzt

```

for i = 6:-1:1
    if i == 6
        mu (:,:,i) = cross(-f (:,:,i),(rae (:,:,i)+rec (:,:,i))) + eye(3)*[0;0;0]
        + eye(3)*cross([0;0;0], rec (:,:,i)) + I (:,:,i)*omega_d (:,:,i) + cross(omega
        (:,:,i),(I (:,:,i)*omega (:,:,i)));
    elseif i == 2
        mu (:,:,i) = cross(-f (:,:,i),(rae (:,:,i)+rec (:,:,i))) + R (:,:,i+1)*mu
        (:,:,i+1) + R (:,:,i+1)*cross(f (:,:,i+1), rec (:,:,i)) + I (:,:,i)*omega_d
        (:,:,i) + cross(omega (:,:,i),(I (:,:,i)*omega (:,:,i)))- mugrav (:,:,i);
    elseif i == 1
        mu (:,:,i) = cross(-f (:,:,i),(rae (:,:,i)+rec (:,:,i))) + R (:,:,i+1)*
        cross(f (:,:,i+1), rec (:,:,i)) + I (:,:,i)*omega_d (:,:,i) + cross(omega
        (:,:,i),(I (:,:,i)*omega (:,:,i)));
    else
        mu (:,:,i) = cross(-f (:,:,i),(rae (:,:,i)+rec (:,:,i))) + R (:,:,i+1)*mu
        (:,:,i+1) + R (:,:,i+1)*cross(f (:,:,i+1), rec (:,:,i)) + I (:,:,i)*omega_d
        (:,:,i) + cross(omega (:,:,i),(I (:,:,i)*omega (:,:,i)));
    end
end

```

D.3.11 Drehmoment τ_i im KS{i-1}

```

for i = 1:1:6
    tau(i) = transpose(mu(:, :, i)) * transpose(R(:, :, i)) * [0; 0; 1]; %* G(i)
end

for i = 1:1:6
    P_mech(i) = tau(i) * qd(i); %/G(i);
end
end

```

D.4 Bahnhplanung

```

function [q, qd, qdd, t] = trajectorie_planning_sixth_order(ts, te, tv,
    stepsize, qs, qds, qdds, qv, qe, qde, qdde)
% Bahnplanung über ein Polynom 6-ter Ordnung inkl. Gelenkwinkel-Definition
des Via-P\dfra

```

D.4.1 Initialisierung

```

t = ts:stepsize:te;
q = zeros(1,1)*(length(t));
qd = zeros(1,1)*(length(t));
qdd = zeros(1,1)*(length(t));

```

D.4.2 Aufstellen der Polynome

```

T = [1, ts, ts^2, ts^3, ts^4, ts^5, ts^6;
      0, 1, 2*ts, 3*ts^2, 4*ts^3, 5*ts^4, 6*ts^5];

```

```

0, 0,    2,      6*ts,   12*ts^2,  20*ts^3,  30*ts^4;
1, tv,   tv^2,   tv^3,   tv^4,    tv^5,    tv^6;
1, te,   te^2,   te^3,   te^4,    te^5,    te^6;
0, 1,    2*te,   3*te^2,  4*te^3,  5*te^4,  6*te^5;
0, 0,    2,      6*te,   12*te^2, 20*te^3, 30*te^4];

```

D.4.3 Definition der Nebenbedingungen

$p = [qs, qds, qdds, qv, qe, qde, qdde];$

D.4.4 Berechnung der Koeffizienten a_0, \dots, a_6

$a = [a0, a1, a2, a3, a4, a5, a6]'$

$a = T \setminus p';$

D.4.5 Berechnung der Trajektorie

```

for index = 1:length(t)
q(index) = a(1,1) + a(2,1)*t(index) + a(3,1)*(t(index))^2 + a(4,1)*(t(index))^3 + a(5,1)*(t(index))^4 + a(6,1)*(t(index))^5 + a(7,1)*(t(index))^6;
qd(index) = a(2,1) + 2*a(3,1)*(t(index)) + 3*a(4,1)*(t(index))^2 + 4*a(5,1)*(t(index))^3 + 5*a(6,1)*(t(index))^4 + 6*a(7,1)*(t(index))^5;
qdd(index) = 2*a(3,1) + 6*a(4,1)*(t(index)) + 12*a(5,1)*(t(index))^2 + 20*a(6,1)*(t(index))^3 + 30*a(7,1)*(t(index))^4;
end

```

D.5 Testsimulation Bahnplanung

D.5.1 Trajektorie Kleben-Seitenwand

```
te = 1.2; % Bewegungsdauer
```

D.5.2 Bewegung 1 (home -> Vorposition)

```
% Startwert Gelenkwinkel qs1 = deg2rad(-7.61); qs2 = deg2rad(-119.27); qs3 = deg2rad(88.49-  
90); qs4 = deg2rad(10.27); qs5 = deg2rad(32.41); qs6 = deg2rad(-10.19);
```

```
% Zielwert Gelenkwinkel qe1 = deg2rad(-14.83); qe2 = deg2rad(-105.81); qe3 = deg2rad(136.16-  
90); qe4 = deg2rad(-27.67); qe5 = deg2rad(-33.44); qe6 = deg2rad(22.89);
```

```
% optimierte Via-Punkte  
% qvd = [];  
% qv1 = deg2rad(qvd(1));  
% qv2 = deg2rad(qvd(2));  
% qv3 = deg2rad(qvd(3));  
% qv4 = deg2rad(qvd(4));  
% qv5 = (deg2rad(qvd(5))+(qs5+qe5)*1/2)*1/2; % Für den Fall, dass der Wert  
% zu dicht an den Grenzen liegt, was eine hohe Beschleunigung zur Folge  
% hat, wird das Mittel aus dem optimierten und dem initialen Via-Punkt  
% gebildet  
% qv6 = (deg2rad(qvd(6))+(qs6+qe6)*1/2)*1/2;
```

D.5.3 Bewegung 3 (letzter Punkt der Trajektorie Kleben Seitenwand -> home)

Startwert Gelenkwinkel

```
qs1 = deg2rad(-53.8);
qs2 = deg2rad(-70.34);
qs3 = deg2rad(98.82-90);
qs4 = deg2rad(-69.87);
qs5 = deg2rad(-58.7);
qs6 = deg2rad(55.7);

% Zielwert Gelenkwinkel
qe1 = deg2rad(-7.61);
qe2 = deg2rad(-119.27);
qe3 = deg2rad(88.49-90);
qe4 = deg2rad(10.27);
qe5 = deg2rad(32.41);
qe6 = deg2rad(-10.19);

%% Startwert Via-Punkte
% qv1 = (qs1+qe1)*1/2;
% qv2 = (qs2+qe2)*1/2;
% qv3 = (qs3+qe3)*1/2;
% qv4 = (qs4+qe4)*1/2;
% qv5 = (qs5+qe5)*1/2;
% qv6 = (qs6+qe6)*1/2;

% optimierte Via-Punkte
qv = [-25.0007 -96.0711 8.8200 10.2700 -47.7822 -10.1900];
qv1 = deg2rad(qv(1));
qv2 = deg2rad(qv(2));
qv3 = deg2rad(qv(3));
qv4 = (deg2rad(qv(4))+(qs4+qe4)*1/2)*1/2;
qv5 = (deg2rad(qv(5))+(qs5+qe5)*1/2)*1/2;
qv6 = (deg2rad(qv(6))+(qs6+qe6)*1/2)*1/2;
```

D.5.4 Bahnplanung

```
stepsize = 0.004;
ts = 0; % Startzeit
```

```

tv = (te-ts)/2; % Via-Punkt Zeitpunkt

% Start- und Endgeschwindigkeiten
qds1 = 0; qde1 = 0;
qds2 = 0; qde2 = 0;
qds3 = 0; qde3 = 0;
qds4 = 0; qde4 = 0;
qds5 = 0; qde5 = 0;
qds6 = 0; qde6 = 0;

% Start- und Endbeschleunigungen
qdds1 = 0; qdde1 = 0;
qdds2 = 0; qdde2 = 0;
qdds3 = 0; qdde3 = 0;
qdds4 = 0; qdde4 = 0;
qdds5 = 0; qdde5 = 0;
qdds6 = 0; qdde6 = 0;

qs = [qs1,qs2,qs3,qs4,qs5,qs6];
qe = [qe1,qe2,qe3,qe4,qe5,qe6];
qv = [qv1,qv2,qv3,qv4,qv5,qv6];
qds = [qds1,qds2,qds3,qds4,qds5,qds6];
qdds = [qdds1,qdds2,qdds3,qdds4,qdds5,qdds6];
qde = [qde1,qde2,qde3,qde4,qde5,qde6];
qdde = [qdde1,qdde2,qdde3,qdde4,qdde5,qdde6];

% Bahnplanung Polynom 6-Ordnung
[q1,qd1,qdd1,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(1), qds(1), qdds(1), qv(1), qe(1), qde(1), qdde(1));
[q2,qd2,qdd2,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(2), qds(2), qdds(2), qv(2), qe(2), qde(2), qdde(2));
[q3,qd3,qdd3,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(3), qds(3), qdds(3), qv(3), qe(3), qde(3), qdde(3));
[q4,qd4,qdd4,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(4), qds(4), qdds(4), qv(4), qe(4), qde(4), qdde(4));
[q5,qd5,qdd5,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(5), qds(5), qdds(5), qv(5), qe(5), qde(5), qdde(5));
[q6,qd6,qdd6,t] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(6), qds(6), qdds(6), qv(6), qe(6), qde(6), qdde(6));

```

D.5.5 Modellberechnung der Drehmomente, Leistungsaufnahme

```
% Initialisierung
tau = zeros(length(t));
Pmech = zeros(length(t));
Pmech_diss = zeros(length(t));
Pmech_sink = zeros(length(t));

% RNEA
for index = 1:1:(length(t))
    [ret_tau, ret_Pmech] = rnea(q1(index), q2(index), q3(index), q4(index), q5(
        index), q6(index), qd1(index), qd2(index), qd3(index), qd4(index), qd5(
        index), qd6(index), qdd1(index), qdd2(index), qdd3(index), qdd4(index),
        qdd5(index), qdd6(index));
    tau(index,1) = ret_tau(1);
    tau(index,2) = ret_tau(2);
    tau(index,3) = ret_tau(3);
    tau(index,4) = ret_tau(4);
    tau(index,5) = ret_tau(5);
    tau(index,6) = ret_tau(6);
    Pmech(index,1) = ret_Pmech(1);
    Pmech(index,2) = ret_Pmech(2);
    Pmech(index,3) = ret_Pmech(3);
    Pmech(index,4) = ret_Pmech(4);
    Pmech(index,5) = ret_Pmech(5);
    Pmech(index,6) = ret_Pmech(6);
end
```

D.5.6 Daten Vorverarbeitung

```
for index = 1:1:(length(t))
    for var = 1:6
        if Pmech(index,var) > 0
            Pmech_sink(index,var) = Pmech(index,var);
        else
            Pmech_diss(index,var) = Pmech(index,var);
```

```
Pmech_sink(index ,var) = 0;  
end  
end  
end
```

D.5.7 Anzeige der Daten

```
disp( 'qv1' ); disp( rad2deg(qv1));  
disp( 'qv2' ); disp( rad2deg(qv2));  
disp( 'qv3' ); disp( rad2deg(qv3));  
disp( 'qv4' ); disp( rad2deg(qv4));  
disp( 'qv5' ); disp( rad2deg(qv5));  
disp( 'qv6' ); disp( rad2deg(qv6));  
torque = sum(sum(abs(tau)))/length(t); disp( 'Mittelwert der Summe des  
Betrag der Getriebe-Drehmomente in Nm'); disp(torque);  
P_mech = sum(sum((Pmech_sink))/length(t)); disp( 'Mittelwert der  
aufgenommenen mechanischen Leistung in W'); disp(P_mech);  
P_mech_diss = sum(sum((Pmech_diss))/length(t))*(te-ts); disp( 'Dissipierte  
Energie in J'); disp(P_mech_diss);  
E_mech = sum(sum(Pmech_sink)/length(t))*(te-ts); disp( 'Aufgenommene  
Energie in J'); disp(E_mech);
```

D.5.8 Erzeuge die Abbildungen

```
show_graphics(q1, q2, q3, q4, q5, q6, qd1, qd2, qd3, qd4, qd5, qd6, qdd1,  
qdd2, qdd3, qdd4, qdd5, qdd6, t, tau, Pmech)
```

D.6 Optimierung

D.6.1 KlebenSeitenwand

D.6.2 Bewegung 1 (**home -> Vorposition**)

```
% Startwert Gelenkwinkel qs1 = deg2rad(-7.61); qs2 = deg2rad(-119.27); qs3 = deg2rad(88.49-90); qs4 = deg2rad(10.27); qs5 = deg2rad(32.41); qs6 = deg2rad(-10.19);
```

```
% Zielwert Gelenkwinkel qe1 = deg2rad(-14.83); qe2 = deg2rad(-105.81); qe3 = deg2rad(136.16-90); qe4 = deg2rad(-27.67); qe5 = deg2rad(-33.44); qe6 = deg2rad(22.89);
```

D.6.3 Bewegung 3 (**letzter Punkt der Trajektorie Kleben Seitenwand -> home**)

Startwert Gelenkwinkel

```
qs1 = deg2rad(-53.8);  
qs2 = deg2rad(-70.34);  
qs3 = deg2rad(98.82-90);  
qs4 = deg2rad(-69.87);  
qs5 = deg2rad(-58.7);  
qs6 = deg2rad(55.7);
```

```
% Zielwert Gelenkwinkel  
qe1 = deg2rad(-7.61);  
qe2 = deg2rad(-119.27);  
qe3 = deg2rad(88.49-90);  
qe4 = deg2rad(10.27);  
qe5 = deg2rad(32.41);  
qe6 = deg2rad(-10.19);
```

D.6.4 Startwert Via-Punkte

```
denum = 1; % Testen von Startwerten , die von Mittelpunkt abweichen  
qv1 = (qs1+qe1)*1/2;%-(abs(qs1-qe1)/2)/denum;  
qv2 = (qs2+qe2)*1/2;%-(abs(qs1-qe1)/2)/denum;  
qv3 = (qs3+qe3)*1/2;%-(abs(qs1-qe1)/2)/denum;  
qv4 = (qs4+qe4)*1/2;%-(abs(qs1-qe1)/2)/denum;  
qv5 = (qs5+qe5)*1/2;%-(abs(qs1-qe1)/2)/denum;  
qv6 = (qs6+qe6)*1/2;%-(abs(qs1-qe1)/2)/denum;
```

D.6.5 Initial-Trajektorie-Definition

```
te = 1.2;  
qs = [ qs1 , qs2 , qs3 , qs4 , qs5 , qs6 ];  
qe = [ qe1 , qe2 , qe3 , qe4 , qe5 , qe6 ];  
qv = [ qv1 , qv2 , qv3 , qv4 , qv5 , qv6 ];
```

D.6.6 Definition Optimierer

Zielfunktion

```
objective = @(qv) calc_objective(qs,qe,qv,te);  
% Startwert  
x0 = qv;  
% Anzeige des Energieverbrauchs beim Start der Optimierung  
disp(['initial Objective: ' num2str(objective(x0))]);  
% Gleichungsnebenbedingungen  
A = [];  
b = [];  
% Ungleichungssnebenbedingungen  
Aeq = [];  
beq = [];
```

```
% Grenzen des Parametervektors
lb = (qv-(abs(qs-qe))/2);
ub = (qv+(abs(qs-qe))/2);
% Nichtlineare Nebenbedingungen
nonlcon = [];
% Auswahl des Solvers fmincon
% Algorithmus sequentielle quadratische Programmierung
% max. Anzahl der Iterationen = 25
% und Ausgabe des Zielfunktionswertes mit jeder Iteration
options = optimoptions(@fmincon, 'Algorithm', 'sqp', 'MaxIterations', 25,
    'PlotFcn', @optimplotfval);
% Ausführen der Optimierung
[x, fval, ef, output, lambda] = fmincon(objective, x0, A, b, Aeq, beq, lb, ub, nonlcon,
    options);
% Anzeige des energieoptimierten Parametervektors
disp(x*180/pi)

% Startwert
E in J
2.0013e+03

% Zielwert
E in J
1.8074e+03

% identifizierter Parametervektor
-24.8857 -96.0716 8.8200 10.2700 -47.5666 -10.1900
```

D.7 Berechnung der Zielfunktion

```
function [E_mech] = calc_objective(qs, qe, qv, te)

% Berechnung der Zielfunktion (Energieverbrauch) über die nicht
% dissipierte mechanische Leistung
```

D.7.1 Bahnhplanung

```

stepsize = 0.004;
ts = 0;
tv = (te-ts)/2;

qds1 = 0; qde1 = 0;
qds2 = 0; qde2 = 0;
qds3 = 0; qde3 = 0;
qds4 = 0; qde4 = 0;
qds5 = 0; qde5 = 0;
qds6 = 0; qde6 = 0;

qdds1 = 0; qdde1 = 0;
qdds2 = 0; qdde2 = 0;
qdds3 = 0; qdde3 = 0;
qdds4 = 0; qdde4 = 0;
qdds5 = 0; qdde5 = 0;
qdds6 = 0; qdde6 = 0;

qds = [qds1,qds2,qds3,qds4,qds5,qds6];
qdds = [qdds1,qdds2,qdds3,qdds4,qdds5,qdds6];
qde = [qde1,qde2,qde3,qde4,qde5,qde6];
qdde = [qdde1,qdde2,qdde3,qdde4,qdde5,qdde6];

[q1,qd1,qdd1,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(1), qds(1), qdds(1), qv(1), qe(1), qde(1), qdde(1));
[q2,qd2,qdd2,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(2), qds(2), qdds(2), qv(2), qe(2), qde(2), qdde(2));
[q3,qd3,qdd3,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(3), qds(3), qdds(3), qv(3), qe(3), qde(3), qdde(3));
[q4,qd4,qdd4,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(4), qds(4), qdds(4), qv(4), qe(4), qde(4), qdde(4));
[q5,qd5,qdd5,~] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(5), qds(5), qdds(5), qv(5), qe(5), qde(5), qdde(5));
[q6,qd6,qdd6,t] = trajectorie_planning_sixth_order(ts, te, tv, stepsize,
    qs(6), qds(6), qdds(6), qv(6), qe(6), qde(6), qdde(6));

```

D.7.2 Modellberechnung der Drehmomente, Leistungsaufnahme

```
% Initialisierung
tau = zeros(length(t));
Pmech = zeros(length(t));
Pmech_diss = zeros(length(t));
Pmech_sink = zeros(length(t));

for index = 1:1:(length(t))
[ret_tau,ret_Pmech] = rnea(q1(index), q2(index), q3(index), q4(index), q5(index),
    q6(index), qd1(index), qd2(index), qd3(index), qd4(index), qd5(index),
    qd6(index), qdd1(index), qdd2(index), qdd3(index), qdd4(index),
    qdd5(index), qdd6(index));
tau(index,1) = ret_tau(1);
tau(index,2) = ret_tau(2);
tau(index,3) = ret_tau(3);
tau(index,4) = ret_tau(4);
tau(index,5) = ret_tau(5);
tau(index,6) = ret_tau(6);
Pmech(index,1) = ret_Pmech(1);
Pmech(index,2) = ret_Pmech(2);
Pmech(index,3) = ret_Pmech(3);
Pmech(index,4) = ret_Pmech(4);
Pmech(index,5) = ret_Pmech(5);
Pmech(index,6) = ret_Pmech(6);
end
```

D.7.3 Daten Vorverarbeitung

```
for index = 1:1:(length(t))
for var = 1:6
if Pmech(index,var) > 0
Pmech_sink(index,var) = Pmech(index,var);
else
Pmech_diss(index,var) = Pmech(index,var);
Pmech_sink(index,var) = 0;
```

```
end  
end  
end
```

D.7.4 Anzeige der Daten

```
torque = sum(sum((tau).^2))/length(t);  
E_mech = sum(sum((Pmech_sink))/length(t))*(te-ts); disp('E in J'); disp(  
E_mech);
```

Anhang D - Messdaten

E.1 Vergleich der simulierten Verläufe für den initialen und justierten-energieoptimierten Parametervektor

in Abbildung E.1 wird ersichtlich, dass die Gelenkwinkel $q_{i,justiert}(t)$ innerhalb der Start- und Zielwinkel liegen.

$$q_{i,justiert}(t) \in [q_{i,s}; q_{i,e}] \quad \forall i \in \{1, \dots, 6\} \quad (\text{E.1})$$

Die erreichten Winkelgeschwindigkeiten liegen unterhalb der, in C.1 notierten, Winkelgeschwindigkeiten bei Nenn-Traglast.

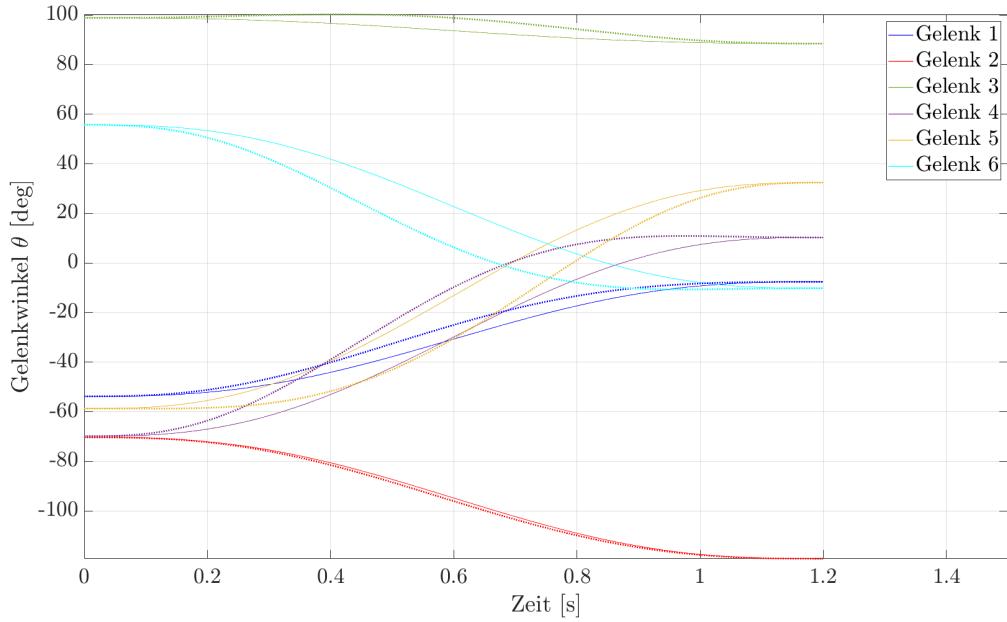


Abbildung E.1: Simulierte Gelenkwinkel der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

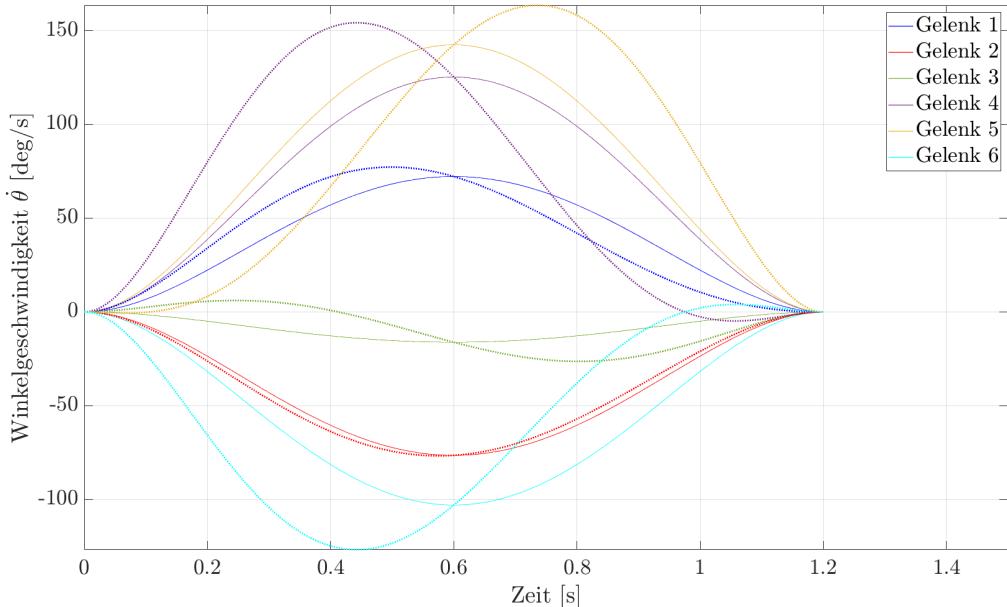


Abbildung E.2: Simulierte Winkelgeschwindigkeit der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

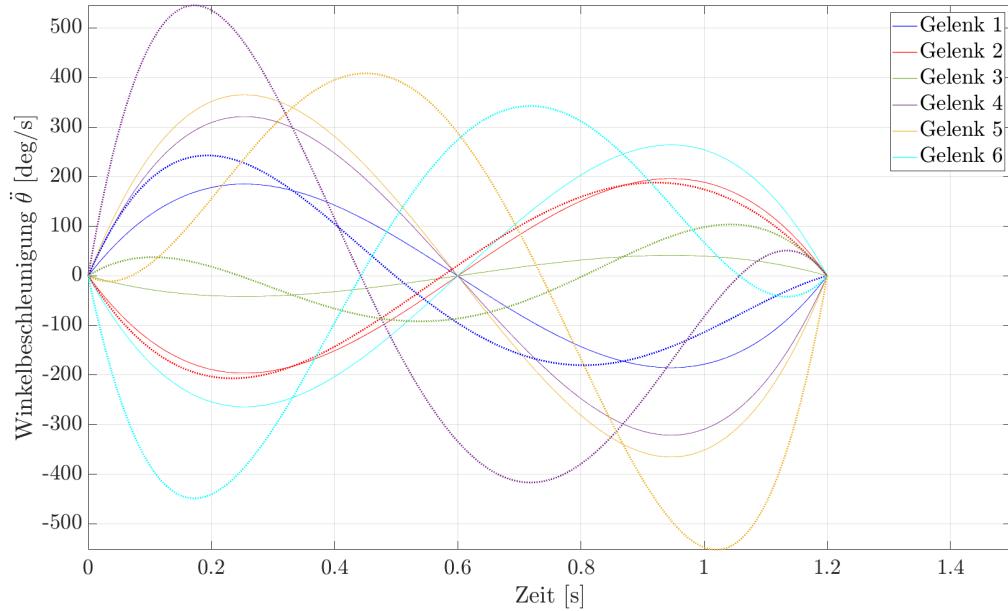


Abbildung E.3: Simulierte Winkelbeschleunigung der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand

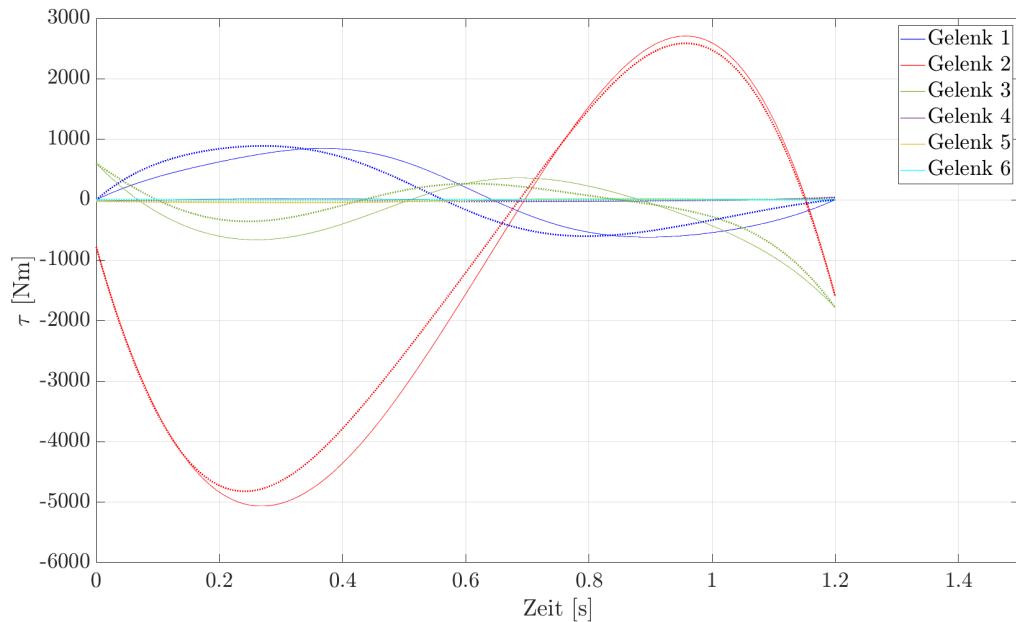


Abbildung E.4: Simulierte Drehmomente der Initialbewegung und justierten, energieoptimierten Bewegungsbahn vom letzten Prozesspunkt auf die Home Position im Programm Kleben-Seitenwand