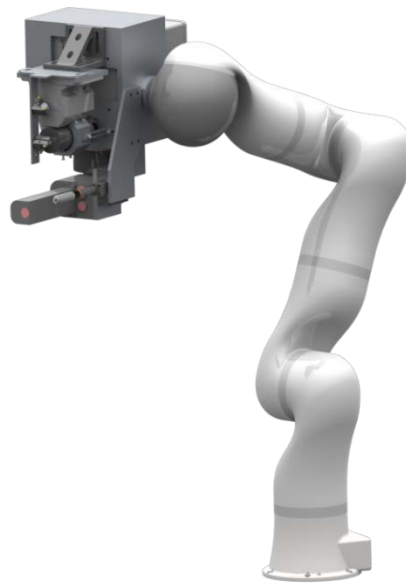


Simulative Optimierung der Bahnplanung mit mehrfacher Redundanz bei der roboterassistierten Laserosteotomie



Bachelorarbeit B-02/13-371

Moritz Schappler

Matrikel-Nr.: 2641920

Hannover, Februar 2013

Erstprüfer Prof. Dr.-Ing. T. Ortmaier

Betreuer Dipl.-Ing. Alexander Fuchs

Simulative Optimierung der Bahnplanung mit mehrfacher
Redundanz bei der roboterassistierten Laserosteotomie

Bachelorarbeit

zur Erlangung des akademischen Grades „Bachelor of Science (B.Sc.)“
im Studiengang Wirtschaftsingenieur der Fakultät für Elektrotechnik und Informatik,
Fakultät für Maschinenbau und der Wirtschaftswissenschaftlichen Fakultät
der Leibniz Universität Hannover

vorgelegt von

Name: Schappler

Vorname: Moritz

Geb. am: 21.11.1989

in: Hannover

Prüfer: Prof. Dr.-Ing. T. Ortmaier

Hannover, den 20.02.2013

Bachelorarbeit

Herr Moritz Schappler, Matrikelnummer: 2641920

Simulative Optimierung der Bahnplanung mit mehrfacher Redundanz bei der roboterassistierten Laserosteotomie

Allgemeines:

Im Rahmen des Forschungsprojektes *Roboterassistierte Laserosteotomie* wird die Möglichkeit einer hochgenauen Knochenbearbeitung durch den Einsatz eines Lasers am Roboter untersucht. Das Versuchssystem besteht aus redundantem Leichtbauroboter und Laser-System. In der Simulation lassen sich Abläufe planen und optimieren. Eine erste Version der Simulationssoftware ermöglicht die Berechnung von Gelenktrajektorien des redundanten Roboters sowie die Darstellung der berechneten Bewegung. Durch die zusätzlichen Freiheitsgrade (FHG) der Laser-Scanner-Einheit wird das Gesamtsystem mehrfach redundant, was jedoch in der Planung bislang unberücksichtigt bleibt.

Aufgabe:

Ziel dieser Arbeit ist daher eine Erweiterung der vorhandenen Simulationssoftware um eine angepasste Bahnplanung für die mehrfache Redundanz. Die 7 FHG des Roboters und die 3 FHG des Laser-Scanner-Systems müssen bei der Berechnung der Trajektorie aus den kartesischen Vorgaben mit einbezogen werden. Verschiedene Kriterien der Redundanzauflösung (kleinste Gelenkbewegung, höchste Genauigkeit, etc.) sind ebenso zu untersuchen wie unterschiedliche Strategien bei der Verwendung der FHG (kontinuierliche Einbeziehung aller FHG oder getrennte Roboter/Scanner-Bewegungen). Nach einer ausführlichen Recherche des Stands der Forschung zu dieser Fragestellung folgt eine Einarbeitung in die bereits bestehende Programmstruktur. Die Umsetzung der erarbeiteten Ansätze erfolgt in der Programmierumgebung MATLAB. Eine Visualisierung der Ergebnisse umfasst die Einbindung der Laser-Scanner-Einheit in das vorhandene Graphical User Interface (GUI) und ist zu implementieren. Abschließend ist anhand von Beispieltrajektorien ein Vergleich der verschiedenen Strategien durchzuführen. Der Programmaufbau soll dabei möglichst modular und allgemeingültig sein und die Bewertung verschiedener Trajektorien ermöglichen.

Ausgabe der Aufgabenstellung: 31.12.2012

Abgabe der Arbeit: 20.02.2013

Betreuer: Dipl.-Ing. Alexander Fuchs

Inhaltsverzeichnis

| | |
|---|-----|
| Abbildungsverzeichnis..... | III |
| Tabellenverzeichnis..... | IV |
| Abkürzungsverzeichnis | V |
| Formelzeichen..... | V |
| 1 Einleitung..... | 1 |
| 2 Stand der Technik..... | 2 |
| 2.1 Robotik | 2 |
| 2.2 Roboterassistierte Laserosteotomie | 5 |
| 2.3 Leistungsmerkmale | 7 |
| 2.4 Optimierungsverfahren | 12 |
| 3 Redundanzauflösung..... | 15 |
| 3.1 Redundanzen im System | 15 |
| 3.2 Nicht-redundante Inverse Kinematik des Roboters | 17 |
| 3.3 Redundante inversen Kinematik des Roboters | 19 |
| 3.4 Nullraumbewegung..... | 23 |
| 3.5 Betrachtung des Gesamtsystems..... | 23 |
| 4 Optimierung der Bahnplanung..... | 24 |
| 4.1 Gemeinsame Optimierung aller Freiheitsgrade | 25 |
| 4.2 Getrennte Optimierung von Roboter und Laserwerkzeug | 26 |
| 5 Implementierung in MATLAB | 35 |
| 5.1 Arbeitsprozess..... | 35 |
| 5.2 Module..... | 39 |
| 6 Evaluation | 40 |
| 6.1 Wahl der Zielfunktion..... | 41 |
| 6.2 Inverse Kinematik..... | 43 |

| | | |
|-----|--|----|
| 6.3 | Bahnoptimierung | 45 |
| 6.4 | Getrennte und gemeinsamer Optimierung der Freiheitsgrade | 47 |
| 7 | Zusammenfassung und Ausblick | 48 |
| 8 | Literaturverzeichnis..... | 50 |
| A | Anhang | 51 |
| A.1 | Unterschiedliche Konfigurationen | 51 |
| A.2 | Austrittswinkel des Laserstrahls | 53 |
| A.3 | Bilder zu den Ergebnissen der Bahnoptimierung..... | 54 |
| A.4 | Dateiformate | 55 |

Abbildungsverzeichnis

| | |
|--|----|
| Bild 2.1: Laserwerkzeug mit integrierter Strahlerzeugung und Kohärenztomographie..... | 6 |
| Bild 2.2: (a) Aufbau des HurryScan 20 Scan-Kopfes; Quelle: [SL20], (b) Kinematische Kette des Laserstrahls, (c) Laser-Arbeitsraum; Quelle (b und c): [Bur10/116] | 6 |
| Bild 2.3: Einordnung der verwendeten Optimierungsverfahren | 12 |
| Bild 3.1: Nullraumbewegung, Redundanz der Grundachsen des LBR; Quelle: KUKA Roboter GmbH | 16 |
| Bild 4.1: Abdeckung der Beispieltrajektorie mit Laserarbeitsräumen..... | 27 |
| Bild 4.2: Verteilung der möglichen Lasermittelpunkte in einem Schritt des genetischen Algorithmus..... | 28 |
| Bild 4.3: Reihenfolge der Lasergrundstellungen nach der Optimierung mit dem Greedy-Algorithmus | 29 |
| Bild 4.4: Reihenfolge der Lasergrundstellungen nach der Optimierung mit dem TSP-Algorithmus..... | 30 |
| Bild 4.5: γE - $q3$ -Lösungsraum der inversen Kinematik für den Punkt $x_E = -450$ mm, $y_E = -19$ mm, $z_E = 358$ mm, $\alpha_E = -166^\circ$, $\beta_E = 0^\circ$ | 31 |
| Bild 4.6: Posen des Roboters für zwei Fälle der Nullraumanalyse des Punktes $x_E = -450$ mm, $y_E = -19$ mm, $z_E = 358$ mm, $\alpha_E = -166^\circ$, $\beta_E = 0^\circ$ | 32 |
| Bild 5.1: Gesamtansicht der MATLAB-GUI..... | 35 |
| Bild 5.2: Linker Teil der GUI; enthält die Eingabe und Analyse der Trajektorie sowie Eingabe und Anzeige der kartesischen Lage und Gelenkwinkel | 37 |
| Bild 5.3: Rechter Teil der GUI; enthält die Anzeige des Roboters und der Trajektorie..... | 38 |
| Bild 5.4: Hierarchieschema der verwendeten MATLAB-Funktionen | 39 |
| Bild 6.1: Beispieltrajektorien im Arbeitsraum des Roboters..... | 40 |
| Bild 6.2: Streudiagramme zur Darstellung der Korrelation der Kennzahlen. Datengrundlage: ca. 5 Mio. Gelenkwinkelkombinationen. Wertebereich: 0 bis 1. | 41 |
| Bild 6.3: Gelenkwinkel-, Laserauslenkungs- und Zielfunktionsverlauf der optimierten Dreieck-Beispieltrajektorie mit dynamischer Optimierung | 46 |
| Bild 6.4: Laserauslenkungsverlauf der optimierten Dreieck-Beispieltrajektorie mit dynamischer Optimierung bei gemeinsamer Optimierung von Roboter und Laser | 47 |

| | |
|--|----|
| Bild A.1: (a) bis (f); Konfigurationen des LBR 4+ | 52 |
| Bild A.2: Vergleich des realen und vereinfachten Austrittswinkels des Laserstrahls | 53 |
| Bild A.3: Ausschnitt aus Gelenkwinkel- , Laserauslenkungs- und Zielfunktionsverlauf bei der Rechteck-Beispieltrajektorie mit Optimierung durch Greedy- Algorithmus ohne Berücksichtigung der Stetigkeit. | 54 |
| Bild A.4: Ausschnitt aus Gelenkwinkel- , Laserauslenkungs- und Zielfunktionsverlauf bei der Rechteck-Beispieltrajektorie mit Optimierung durch Greedy- Algorithmus mit Berücksichtigung der Stetigkeit. | 54 |

Tabellenverzeichnis

| | |
|---|----|
| Tabelle 2.1: Denavit-Hartenberg-Parameter des LBR und eingezeichnete KS | 3 |
| Tabelle 2.2: Abgeschätzte Gelenksteifigkeiten in Verdrehrichtung..... | 8 |
| Tabelle 2.3: Technische Grenzen der Gelenkwinkel des KUKA LBR4+ | 11 |
| Tabelle 4.1: Bedeutung der Merkmale des GA für die Laserarbeitsraum- Positionsoptimierung | 28 |
| Tabelle 4.2: Bedeutung der Merkmale des GA für die Laserarbeitsraum- Reihenfolgeoptimierung..... | 29 |
| Tabelle 4.3: Optimalität der Ergebnisse von Greedy-Algorithmus und dynamischer Optimierung für die Gesamt-Trajektorie | 34 |
| Tabelle 4.4: Belegung der Begriffe der dynamischen Optimierung in der Anwendung auf das Bahnplanungsproblem | 34 |
| Tabelle 6.1: Werte der Leistungsmerkmale bei unterschiedlicher Gewichtung in der Zielfunktion für die Dreieck-Beispieltrajektorie..... | 42 |
| Tabelle 6.2: Vergleich der Lösungsalternativen zur Auflösung der kinematischen Redundanz des Roboters | 44 |
| Tabelle 6.3: Vergleich der gemittelten Zielfunktionen bei unterschiedlicher Gesamtoptimierung | 47 |

Abkürzungsverzeichnis

| Zeichen | Beschreibung | Englischer Fachbegriff |
|---------|------------------------------------|--------------------------------|
| DH | Denavit-Hartenberg | Denavit-Hartenberg |
| DO | Dynamische Optimierung | Dynamic Programming |
| FHG | Freiheitsgrad | Degree of Freedom (DoF) |
| GUI | Graphische Benutzerschnittstelle | Graphical User Interface |
| KS | Koordinatensystem | frame |
| RaLo | Roboterassistierte Laserosteotomie | Robot Assisted Laser Osteotomy |
| TRR | Trust-Region-Reflective | Trust-Region-Reflective |

Zitierweise:

[Aaa99/123] Quellenabkürzung Aaa, Erscheinungsjahr (19)99, Seite 123

Formelzeichen

Keine Hervorhebung: a Skalar

Unterstrichen: \underline{a} Vektor

Fettgedruckt: \mathbf{A} Matrix

| Zeichen | Bedeutung |
|----------------------------------|--|
| $f(\underline{q}_x)$ | Gütefunktion in Abhängigkeit der Optimierungsvariablen \underline{q}_x |
| \mathbf{J} | Jacobi-Matrix |
| K | Kennzahl eines Leistungsmerkmals |
| \underline{q}_l | Auslenkung des Laserstrahls |
| $\underline{q}, \underline{q}_r$ | Gelenkwinkel des Roboters |
| ${}^{(0)}\underline{x}_7$ | Ortsvektor des Ursprungs von KS 7 ausgedrückt im KS 0 |
| ${}^0\mathbf{R}_7$ | Rotationsmatrix vom KS 0 ins KS 7 |
| ${}^0\mathbf{T}_7$ | Homogene Transformation von KS 0 nach KS 7 |
| ${}^{(0)}\underline{x}_E$ | Endeffektorlage in globalen Koordinaten (KS 0) |
| z_K | Zielfunktion, gewichtete Summe von Kennzahlen |

1 Einleitung

Am Institut für Mechatronische Systeme wird ein prototypisches Roboter-Laser-System zur roboterassistierten Laserosteotomie aufgebaut [Fuc12]. Diese Arbeit beschäftigt sich mit der Weiterentwicklung einer Simulationsumgebung in MATLAB um eine Bahnplanung, die die mehrfache kinematische Redundanz des Systems berücksichtigt. Die Redundanz entsteht durch das Zusammenspiel des an sich bereits redundanten Leichtbauroboters mit den zusätzlichen Freiheitsgraden des Laser-Scanner-Systems. Es werden unterschiedliche Ansätze zur Auflösung vorgestellt, in MATLAB implementiert und anschließend bewertet.

Nach der Nennung der Grundlagen in Kapitel 2 wird die Auflösung der Redundanz des Systems mit der inversen Kinematik in Kapitel 3 hergeleitet. Bei der Berechnung der inversen Kinematik für einen Bahnpunkt wird auf der Arbeit von [Ölt10] aufgebaut und Algorithmen mit quadratischer anstatt linearer Konvergenz sowie andere Ansätze verwendet. Des Weiteren wird die Redundanz des Roboters und der Bearbeitungssituation genauer untersucht.

Zur Optimierung des gesamten Bahnverlaufs in Kapitel 4 wird insbesondere das Potential einer getrennten Roboter- und Scanner-Bewegung untersucht und ein Algorithmus vorgestellt. Dabei wird mit der dynamischen Optimierung auf ein Verfahren zurückgegriffen, das hauptsächlich in der wirtschaftlichen Planungsrechnung verwendet wird, und gezeigt, dass eine Anwendung hier sinnvoll ist.

Die in der Arbeit erweiterte graphische Benutzerschnittstelle wird in Kapitel 5 vorgestellt. Es wird dabei auf einen übersichtlichen Arbeitsfluss von der Trajektoriengenerierung zur -Analyse und -Optimierung geachtet, wobei die Ergebnisse visualisiert werden.

Die Bewertung der vorgestellten Algorithmen und Verfahren wird in Kapitel 6 durchgeführt. Die Analyse und Optimierung von Beispieltrajektorien wird hinsichtlich unterschiedlicher Zielkriterien verglichen.

2 Stand der Technik

2.1 Robotik

An dieser Stelle werden grundlegende Begriffe der Robotik kurz vorgestellt, auf die im weiteren Verlauf der Arbeit eingegangen wird. Eine gute Zusammenfassung über das Thema bieten die Vorlesungen Robotik I und II ([Ortm11a], [Ortm12b]) oder die Lehrbücher [Sici08] und [Cork11].

2.1.1 Koordinatentransformation

Bei der Beschreibung von Robotern als mechanische Mehrkörpersysteme wird jedem Segment des Roboters ein eigenes Koordinatensystem (KS) zugewiesen, mit dem die Bewegung berechnet werden kann. Zur Berechnung des Übergangs von einem Koordinatensystem zum nächsten gibt es mehrere Methoden. In dieser Arbeit werden Rotationsmatrizen \mathbf{R} verwendet, die die Drehung eines Koordinatensystems eindeutig beschreiben. Die Rotationsmatrix enthält die Koordinatenachsen des neuen KS ausgedrückt im alten KS gemäß

$${}^{\text{alt}}\mathbf{R}_{\text{neu}} = \begin{bmatrix} ({}^{\text{alt}})\underline{x}_{\text{neu}} & ({}^{\text{alt}})\underline{y}_{\text{neu}} & ({}^{\text{alt}})\underline{z}_{\text{neu}} \end{bmatrix}. \quad (2.1)$$

Um beliebige Drehungen zu beschreiben wird die Winkelkonvention nach Kardan verwendet, bei der das KS nacheinander um die mitgedrehten x-, y- und z-Achsen des kartesischen Anfangs-Koordinatensystems gedreht wird. Die Drehung wird durch die Kardan-Winkel α , β und γ durch

$$\mathbf{R}_{\text{Kardan}} = \mathbf{R}_x(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_z(\gamma) \quad (2.2)$$

beschrieben [Ortm11a/15]. Die Winkel α , β und γ für eine bekannte Rotationsmatrix sind durch Umstellung von Gl. (2.2) bestimmbar.

Um zusätzlich die Translation des KS-Ursprungs \underline{r} zu berücksichtigen werden homogene Transformationsmatrizen \mathbf{T} mit

$${}^{\text{alt}}\mathbf{T}_{\text{neu}} = \begin{bmatrix} {}^{\text{alt}}\mathbf{R}_{\text{neu}} & ({}^{\text{alt}})\underline{r}_{\text{neu}} \\ \underline{0}^T & 1 \end{bmatrix} \quad (2.3)$$

verwendet.

2.1.2 Direkte Kinematik, Denavit-Hartenberg Parameter

Nach [Heim06/179] ist die Kinematik die formale Beschreibung der Zusammenhänge von Lage, Geschwindigkeit und Beschleunigung der einzelnen Körper eines Mehrkörpersystems. Mit der direkten Kinematik wird die Endeffektorlage des Roboters aus seinen Gelenkwinkeln berechnet. Für serielle Roboter führt die Beschreibung der Struktur mit Denavit-Hartenberg-Parametern (DH) zur vereinfachten Berechnung der direkten Kinematik. Ausgehend vom vorherigen KS wird für jedes Glied des Roboters nacheinander eine Rotation und Translation um die z-Achse (Winkel θ , Strecke d) sowie eine Translation und Rotation um die x-Achse (Strecke a und Winkel α) durchgeführt. Die Größen θ_i , d_i , a_i und α_i bilden die DH-Parameter für ein Segment des Roboters. Im Folgenden sind die DH-Parameter für den in dieser Arbeit betrachteten Roboter Kuka LBR 4+ dargestellt.

Tabelle 2.1: Denavit-Hartenberg-Parameter des LBR und eingezeichnete KS

| Gelenk Nr. i | Achse | Gelenkwinkel θ_i | d_i [mm] | a_i [mm] | α_i |
|----------------|-------|-------------------------|------------|------------|-------------|
| 1 | A1 | q_1 | 310 | 0 | 90° |
| 2 | A2 | q_2 | 0 | 0 | -90° |
| 3 | E1 | q_3 | 400 | 0 | -90° |
| 4 | A3 | q_4 | 0 | 0 | 90° |
| 5 | A4 | q_5 | 390 | 0 | 90° |
| 6 | A5 | q_6 | 0 | 0 | -90° |
| 7 | A6 | q_7 | 78 | 0 | 0° |

Im Bild in Tabelle 2.1 ist zu erkennen, dass es mit der DH-Konvention nicht immer möglich ist, die Koordinatensysteme korrekt in die Gelenke zu legen. In dieser Arbeit wird daher auf angepasste Transformationsmatrizen für jedes Segment des Roboters zurückgegriffen, die nicht der DH-Konvention entsprechen. Die DH-Parameter werden für Berechnungen mit der Robotics Toolbox [CorRTb] benutzt.

2.1.3 Inverse Kinematik

Die Berechnung der direkten Kinematik einer seriellen Roboterstruktur ist immer eindeutig möglich. Die inverse Kinematik beschreibt den umgekehrten Rechenweg: Die Berechnung der für eine vorgegebene Endeffektorlage notwendigen Gelenkwinkel. Die Lösung dieses Problems ist nicht immer eindeutig. Die inverse Kinematik wird analytisch, geometrisch oder numerisch gelöst. Die Lösbarkeit hängt neben der Endeffektorlage auch von der Konstruktion des Roboters sowie der Redundanz von Freiheitsgraden ab. Die unterschiedlichen Methoden zur Lösung der inversen Kinematik werden z. B. in [Ortm11a/40], [Cork11/146] und [Sici08/27] erklärt.

2.1.4 Nullraumbewegung und Redundanz

Roboter sind redundant, wenn die Anzahl ihrer Freiheitsgrade \underline{q} größer ist als die Anzahl der Endeffektorfreiheitsgrade \underline{x}_E . Die Ordnung der Redundanz wird mit

$$\dim(\underline{q}) - \dim(\underline{x}_E)$$

angegeben. $\underline{q} = [q_1, \dots, q_N]^T$ sind dabei die abtriebsseitigen N Gelenkwinkel des Roboters und $\underline{x}_E = [x_E, y_E, z_E, \alpha_E, \beta_E, \gamma_E]^T$ die kartesischen Koordinaten sowie die Orientierung des Endeffektors ausgedrückt mit Kardan-Winkeln. Der verwendete Leichtbauroboter besitzt einen redundanten Freiheitsgrad, da $\dim(\underline{q}) = 7$ und allgemein $\dim(\underline{x}_E) = 6$ gilt.

Eine Nullraumbewegung bezeichnet die Bewegung der Achsen des Roboters, während die Endeffektorlage gleich bleibt, [Ortm11a/76]. Eine Nullraumbewegung ist nur mit redundanten Robotern möglich. Durch die Nullraumbewegung sind eine Hindernisvermeidung und die Optimierung der Pose des Roboters möglich. Die Pose beschreibt die Gelenkwinkelstellung des Roboters, die Lage die Gesamtheit aus Position und Orientierung. Bei redundanten Robotern existieren mehrere mögliche Posen für eine Endeffektorlage.

2.1.5 Differentielle Kinematik (Jacobi-Matrix)

Die Jacobi-Matrix J ist die wesentliche Kenngröße zur Beschreibung von Roboterkinematiken. Sie beschreibt die inkrementelle Änderung von Endeffektorgrößen \underline{x}_E infolge von inkrementellen Änderungen der Gelenkgrößen \underline{q} . Die Berechnung der geometrischen Jacobi-Matrix J_g erfolgt spaltenweise [Ortm11a/49]:

$$J_g(\underline{q}) = \frac{\partial {}_{(0)}\underline{x}_E}{\partial \underline{q}} = \begin{bmatrix} {}_{(0)}\underline{z}_{j-1} \times {}_{(0)}\underline{r}_{j-1,E} \\ {}_{(0)}\underline{z}_{j-1} \end{bmatrix}_{j \in [1, \dots, \dim(\underline{q})]} \quad (2.4)$$

Es wird für jedes Gelenk des Roboters die Drehrichtung ${}_{(0)}\underline{z}_{j-1}$ der Achse und der Hebelarm ${}_{(0)}\underline{r}_{j-1,E}$ des Gelenks zum Endeffektor eingesetzt.

In dieser Arbeit wird die Jacobi-Matrix zur numerischen Berechnung der inversen Kinematik eingesetzt. Durch inkrementelle Änderung der Gelenkwinkel $\Delta \underline{q}$ wird die Soll-Endeffektorlage $\underline{x}_{E,soll}$ mit

$$\Delta \underline{x}_E = J(\underline{q}) \cdot \Delta \underline{q} \Leftrightarrow \Delta \underline{q} = J(\underline{q})^{-1} \cdot \Delta \underline{x}_E \quad (2.5)$$

angenähert. Dabei ist $\Delta \underline{x}_E = \underline{x}_{E,soll} - \underline{x}_{E,ist}$.

Die Einträge der Jacobi-Matrix variieren über den gesamten Arbeitsraum. Es gibt singuläre Konfigurationen, bei denen die Invertierung nicht möglich ist. Bei redundanten Robotern ist die Jacobi-Matrix nicht quadratisch, zur Invertierung wird dann die Moore-Penrose-Pseudo-Inverse gebildet.

2.2 Roboterassistierte Laserosteotomie

Seriellkinematische Roboter werden als Industrieroboter für viele unterschiedliche Bearbeitungs- und Handhabungsaufgaben eingesetzt. Ein weiteres Einsatzgebiet ist die Medizintechnik. Die Ergebnisse dieser Arbeit werden in einem System für roboterassistierte Laserosteotomie (RaLo) eingesetzt. Diese Technik wird in [Bur10] beschrieben. Ziel ist es, Gewebe und Knochen durch Einwirken von Laserpulsen gezielt zu bearbeiten.

2.2.1 Laser-Werkzeug

Als Bearbeitungswerkzeug wird ein Laserstrahl eingesetzt, dessen Brennpunkt durch Spiegel und Linsen verändert wird. Eine Möglichkeit ist, den Strahl durch eine externe Laserquelle zu erzeugen und über eine Spiegelkinematik zum Endeffektor des Roboters zu führen (siehe [Bur10/123]). Die andere Möglichkeit ist, den Strahl in einem Gerät direkt am Werkzeugflansch des Roboters zu erzeugen und zu leiten. Diese Möglichkeit wird im Rahmen des Projekts am imes untersucht. Die prototypische Baugruppe aus Bild 2.1 erzeugt einen Bearbeitungslaser, dessen Brennpunkt durch zwei Drehspiegel und eine translatorisch verschiebbare Linse veränderbar ist. Zur

Überwachung des Ablationsprozesses¹ wird der bearbeitete Punkt mittels einer in der Baugruppe integrierten optischen Kohärenztomographie untersucht. Die Vorgehensweise dabei ist in [Fuc12] erläutert. Der Vorteil dieser integrierten Bauweise ist neben der Beobachtbarkeit der Ablationstiefe des Bearbeitungsprozesses der Verzicht auf eine Kinematik zur Strahlführung. Nachteilig sind das Gewicht und das Volumen des Werkzeugs, das zu Selbstkollision mit dem Roboter führen kann.

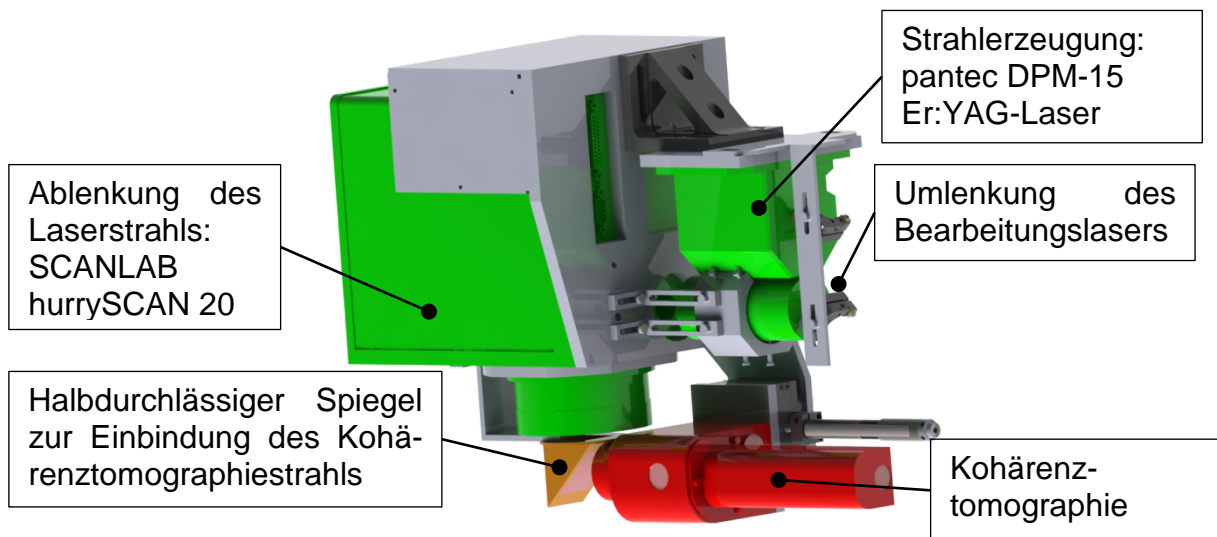


Bild 2.1: Laserwerkzeug mit integrierter Strahlerzeugung und Kohärenztomographie
Die Position des Laserbrennpunktes ist von den drei Freiheitsgraden des Laserwerkzeugs abhängig. Der Strahlengang und die Spiegelanordnung sind in Bild 2.2a gezeigt. Die formale Beschreibung folgt der von seriellen Kinematiken mit rotatorischen und translatorischen Freiheitsgraden (siehe Bild 2.2b). Der erreichbare Arbeitsraum des Brennpunktes ist ein Pyramidenstumpf, wie in Bild 2.2c gezeigt.

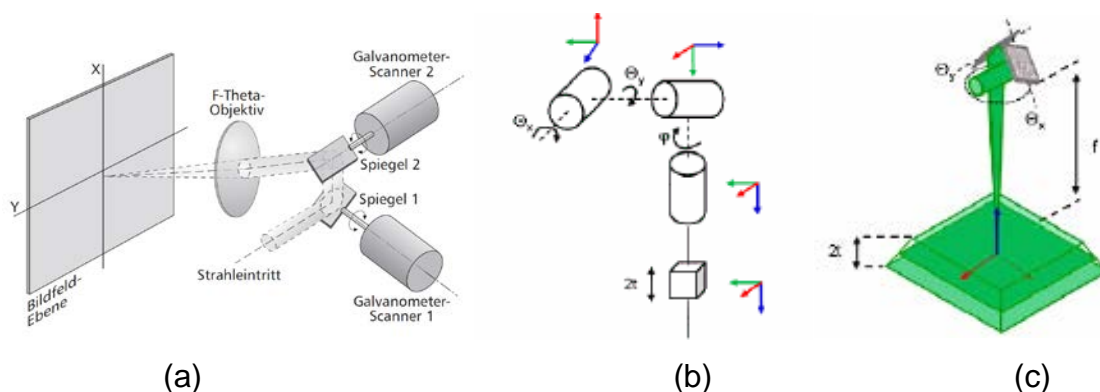


Bild 2.2: (a) Aufbau des HurryScan 20 Scan-Kopfes; Quelle: [SL20], (b) Kinematische Kette des Laserstrahls, (c) Laser-Arbeitsraum; Quelle (b und c): [Bur10/116]

¹ Ablation bezeichnet die Entfernung von Gewebe

2.2.2 Gesamtsystem

Das Gesamtsystem setzt sich aus einem Roboter und dem Laserwerkzeug zusammen. Hier wird der kinematisch redundante serielle Leichtbauroboter Kuka LBR 4+ eingesetzt [KukaLBR4]. Der LBR kann aufgrund seiner Redundanz Nullraumbewegungen durchführen, also z. B. Hindernissen ausweichen ohne dabei die Endeffektorlage zu verändern. Dies ist insbesondere in unbekanntem, wechselnden Umgebungen von Vorteil. Dieser Aspekt wird in Kapitel 3.4 behandelt. Der Roboter verfügt über sieben Achsen, das Laserwerkzeug hat drei Freiheitsgrade (FHG). Das Gesamtsystem verfügt damit über insgesamt zehn unabhängige FHG. Zur Festlegung von Position und Orientierung des Endeffektors sind allgemein sechs FHG notwendig. Es stehen also vier FHG für Optimierungen zur Verfügung. Dies ermöglicht die Verbesserung der im folgenden Kapitel 2.3 genannten Leistungsmerkmale.

2.3 Leistungsmerkmale

Im Folgenden werden Leistungsmerkmale genannt, die die Güte der Pose eines Roboters beschreiben. Aus den Leistungsmerkmalen werden Kennzahlen gebildet, die im weiteren Verlauf der Arbeit benutzt werden, um die Trajektorien des Systems zu optimieren. Zur Vergleichbarkeit werden alle Kennzahlen auf den Bereich von 0 bis 1 skaliert. Der Wert 0 stellt den besten erreichbaren Wert und 1 den schlechtesten Wert dar.

2.3.1 Endeffektornachgiebigkeit

Die Nachgiebigkeit \mathbf{H}_E des Roboters am Endeffektor ist der Kehrwert der Steifigkeit \mathbf{K}_E . Die Nachgiebigkeitsmatrix \mathbf{H}_E enthält die direkte und indirekte Nachgiebigkeit für sechs Freiheitsgrade (drei translatorische und drei rotatorische) und wird durch Multiplikation der Diagonalmatrix der Gelenknachgiebigkeiten \mathbf{H}_q mit der Jacobimatrix \mathbf{J} nach

$$\mathbf{H}_E(\underline{q}) = \mathbf{J}(\underline{q}) \cdot \mathbf{H}_q \cdot \mathbf{J}(\underline{q})^T \quad (2.6)$$

berechnet [Ortm11a/66].

Wenn wie hier ein Bearbeitungsprozess ohne Prozesskraft vorliegt, ist nur die Abdrängung des Roboters infolge der Gewichtskraft des Werkzeugs zu berücksichtigen. Die Kennzahl K_{hzz} ergibt sich durch

$$K_{h_{zz}} = \frac{h_{zz}}{h_{zz,\max}} \quad (2.7)$$

aus der translatorischen Komponente der Nachgiebigkeit in z-Richtung h_{zz} unter Berücksichtigung des Maximalwertes für h_{zz} , $h_{zz,\max}$.

Die größte Nachgiebigkeit $h_{zz,\max}$ bzw. geringste Steifigkeit $k_{zz,\min}$ in z-Richtung treten ein wenn der Arm ganz ausgestreckt ist und ergeben sich zu

$$h_{zz,\max} = 4,4 \frac{\mu\text{m}}{\text{N}} \Leftrightarrow k_{zz,\min} = \frac{1}{h_{zz,\max}} = 0,23 \frac{\text{N}}{\mu\text{m}}.$$

Um den Maximalwert zu berechnen wurde eine Verdrehsteifigkeit in den Gelenken gemäß folgender Tabelle angenommen. Die Werte wurden aus einer am IFW durchgeführten Steifigkeitsuntersuchung eines KUKA-Roboters abgeschätzt [Sch12].

Tabelle 2.2: Abgeschätzte Gelenksteifigkeiten in Verdrehrichtung

| Achse i | 1 (A1) | 2 (A2) | 3 (E1) | 4 (A3) | 5 (A4) | 6 (A5) | 7 (A6) |
|---------------------|--------|--------|--------|--------|--------|--------|--------|
| $k_{q,i}$ [Nm/mrad] | 400 | 400 | 100 | 100 | 100 | 20 | 8 |

2.3.2 Manipulierbarkeit

Die Manipulierbarkeit μ gibt an, wie stark sich eine Bewegung der Gelenkwinkel auf den Endeffektor auswirkt [Ortm11a/79] und berechnet sich zu

$$\mu(\underline{q}) = \left| \det \left(\underline{J}(\underline{q}) \cdot \underline{J}(\underline{q})^T \right) \right| = |\lambda_1 \cdot \dots \cdot \lambda_N|. \quad (2.8)$$

Dabei sind λ_i die Eigenwerte der Matrix $\underline{J} \cdot \underline{J}^T$. Die Berechnung der Kennzahl K_μ erfolgt nach

$$K_\mu = \frac{\mu_{\max} - \mu}{\mu_{\max}}. \quad (2.9)$$

Die Einheit von μ ist bei einer siebenachsigen redundanten Kinematik ein Polynom 14. Grades der Einheiten Meter und Radiant und wird vernachlässigt. Der Maximalwert μ_{\max} für μ wird durch Berechnung der Manipulierbarkeit in möglichst vielen verschiedenen Posen, die unter anderem alle Streck- und Decklagen des Roboters enthalten, abgeschätzt und ergibt sich zu

$$\mu_{\max} = 0,002 \cdot [\mu].$$

2.3.3 Geschicklichkeit

Die Geschicklichkeit η bewertet anstelle von betragsmäßig großen Eigenwerten der Jacobi-Matrix (siehe Manipulierbarkeit) die Gleichförmigkeit der Eigenwerte. Eine gute Geschicklichkeit wird also erreicht, wenn alle Gelenkwinkel gleichförmig den Endeffektor beeinflussen [Ortm12b/44], [Ortm12b/109]. Bei redundanten Robotern wird mit Singulärwerten anstelle von Eigenwerten der Jacobi-Matrix gerechnet, da diese nicht quadratisch ist und somit keine Eigenwerte berechenbar sind. σ_i sind die Singulärwerte von $J(\underline{q})$ und λ_i sind die Eigenwerte von $J(\underline{q}) \cdot J(\underline{q})^T$. Es gilt $\sigma_i^2 = \lambda_i$. Die Geschicklichkeit η berechnet sich als Verhältnis von kleinstem Singulärwert σ_1 zu größtem Singulärwert σ_N zu

$$\eta(J(\underline{q})) = \frac{|\sigma_1|}{|\sigma_N|}. \quad (2.10)$$

Die Geschicklichkeit liegt bereits im Bereich von 0 bis 1, allerdings bedeutet eine Geschicklichkeit von Eins eine isotrope Verstärkung der Gelenkwinkelbewegung zum Endeffektor und stellt den besten erreichbaren Fall dar. Eine Geschicklichkeit von Null bedeutet eine Singularität bei der mindestens eine Achse keinen Einfluss hat. Zur Berechnung der Kennzahl K_η nach

$$K_\eta = \frac{\eta_{\max} - \eta}{\eta_{\max}} \quad (2.11)$$

wird berücksichtigt, dass für den Roboter ein maximaler Wert von

$$\eta_{\max} = 0,025 \cdot [\eta]$$

für η analog zur Vorgehensweise im Teilkapitel 2.3.2 geschätzt wurde.

2.3.4 Genauigkeit

Die Positioniergenauigkeit $\Delta \underline{x}_E$ des Endeffektors ist unter anderem von den in den Gelenken verbauten Winkelencodern abhängig. Die Genauigkeit $\Delta \underline{x}_E$ berechnet aus möglichen Abweichung $\Delta \underline{q}_{\text{Encoder}}$ der Gelenkwinkel durch Genauigkeit der Encoder zu

$$\Delta \underline{x}_E(\underline{q}) = |J(\underline{q})| \cdot \Delta \underline{q}_{\text{Encoder}}.$$

Für $\Delta \underline{q}_{\text{Encoder}}$ wird ein Wert von $0,01^\circ$ für alle Achsen angenommen [Ortm12b/46]. Die Kennzahl K_Δ berechnet sich als

$$K_{\Delta} = \frac{|\Delta \underline{x}_E| - |\Delta \underline{x}_E|_{\min}}{|\Delta \underline{x}_E|_{\max} - |\Delta \underline{x}_E|_{\min}} \quad (2.12)$$

aus dem minimal und maximal erreichbaren Endeffektorfehler $|\Delta \underline{x}_E|_{\min}$ bzw. $|\Delta \underline{x}_E|_{\max}$. Nach der Variation der Gelenkwinkel werden diese Werte zu

$$|\Delta \underline{x}_E|_{\min} = 11,1 \mu\text{m}, \quad |\Delta \underline{x}_E|_{\max} = 468 \mu\text{m}$$

abgeschätzt.

2.3.5 Ellenbogenhöhe

Die Ellenbogenhöhe h_{EB} eignet sich als Kennzahl, da Bearbeitungsobjekte wie OP-Tische oder Werkstückhalter in der Regel unterhalb des Roboters liegen und der Roboter von oben angreift. Sie berechnet sich zu

$$h_{EB}(q_2) = d_1 + d_3 \cdot \cos(q_2).$$

Eine große Ellenbogenhöhe verhindert, dass der Roboter mit dem Werkstück kollidiert. Die Optimierung nach der Kennzahl K_{EB} gem.

$$K_{EB} = \frac{h_{EB,\max} - h_{EB}}{h_{EB,\max} - h_{EB,\min}} \quad (2.13)$$

verhindert Konfigurationen mit dem Ellenbogen unten. Die minimale und maximale Ellenbogenhöhe $h_{EB,\min}$ und $h_{EB,\max}$ berechnen sich aus den technischen Grenzen für Achse A2 zu

$$h_{EB,\min} = h_{EB}(120^\circ) = 110\text{mm},$$

$$h_{EB,\max} = h_{EB}(0^\circ) = 710\text{mm}.$$

2.3.6 Abstand der Gelenkwinkel von technischen Grenzen

Um durchführbare Konfigurationen zu erhalten führt eine Annäherung von Gelenkwinkeln an die technischen Grenzen zu einer Verschlechterung der Kennzahl K_q nach

$$K_q(\underline{q}) = \frac{2}{\sqrt{N}} \sqrt{\sum_{i=1}^N \left(\frac{q_i - q_{i,\text{mid}}}{q_{i,\max} - q_{i,\min}} \right)^2}. \quad (2.14)$$

N gibt die Anzahl der Achsen des Roboters an [Sici08/256]. Die technischen Grenzen für die Gelenkwinkel $q_{i,\min}$ und $q_{i,\max}$ sind dem Datenblatt [KukaLBR4/12] zu entnehmen:

Tabelle 2.3: Technische Grenzen der Gelenkwinkel des KUKA LBR4+

| Achse i | 1 (A1) | 2 (A2) | 3 (E1) | 4 (A3) | 5 (A4) | 6 (A5) | 7 (A6) |
|---------------------------|--------|--------|--------|--------|--------|--------|--------|
| $q_{i,\min}$ [Grad] | -170 | -120 | -170 | -120 | -170 | -120 | -170 |
| $q_{i,\text{mid}}$ [Grad] | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $q_{i,\max}$ [Grad] | 170 | 120 | 170 | 120 | 170 | 120 | 170 |

2.3.7 Eintrittswinkel des Laserstrahls

Während des Bearbeitungsprozesses muss der Laserstrahl möglichst senkrecht auf der Oberfläche stehen. Die genaue Orientierung des Strahls ist allerdings nicht relevant, da der Strahl als rotationssymmetrisch angenommen wird und die Polarisation vernachlässigt wird. Eine Möglichkeit, dies zu implementieren besteht in der Optimierung nach der Kennzahl K_w mit

$$K_w = \frac{1}{2} \left(1 - \cos \left(\angle \left({}_{(0)}\underline{z}_L, {}_{(0)}\underline{n} \right) \right) \right) = \frac{1}{2} - \frac{1}{2} {}_{(0)}\underline{z}_L^T \cdot {}_{(0)}\underline{n}, \quad (2.15)$$

die den Winkel des Laserstrahls zur Oberfläche abbildet. ${}_{(0)}\underline{z}_L$ ist die z-Achse des Endeffektor-KS bzw. die Achse des Laserstrahls. ${}_{(0)}\underline{n}$ bezeichnet die Normale der zu bearbeitenden Oberfläche.

Die geänderte Orientierung des Laserstrahls infolge des Verstellens der Spiegel im Laserwerkzeug wird nicht berücksichtigt. Bei der gegebenen Strahlgeometrie ergibt sich damit ein Fehler von max. 3° bei Vollausslenkung des Laserstrahls (siehe Kapitel A.2 im Anhang).

2.3.8 Änderung der Gelenkwinkel

Große Verfahrswege der Gelenkwinkel führen zu langer Zeitdauer und hohem Energieaufwand. Eine Optimierung mit der Kennzahl $K_{\Delta q}$ nach

$$K_{\Delta q} = \frac{1}{\sqrt{N}} \cdot \sqrt{\sum_{i=1}^N \left(\frac{q_i(t) - q_i(t-1)}{q_{i,\max} - q_{i,\min}} \right)^2} \quad (2.16)$$

minimiert die Verfahrswege aller N Achsen. $q_i(t)$ bezeichnet den Gelenkwinkel der Achse Nr. i für den Zeitschritt t . Durch die Differenzbildung mit den Gelenkwinkeln des vorherigen Zeitschrittes $t-1$ wird der zurückgelegte Weg der Achsen einbezogen. Die technischen Grenzen der Achsen $q_{i,\min}$ und $q_{i,\max}$ sind Tabelle 2.3 zu entnehmen.

2.4 Optimierungsverfahren

Ein wesentlicher Teil dieser Arbeit ist die Anwendung von Optimierungsverfahren auf Probleme der Robotik und der Bahnplanung. Die Grundlagen zu den verwendeten Optimierungsverfahren werden hier kurz dargestellt. Es wird besonders darauf geachtet, Verfahren zu benutzen, die in der Entwicklungsumgebung MATLAB bereits implementiert und bewährt sind. Dies verringert den Programmieraufwand und mögliche Fehlereinflüsse. Die für die numerische Berechnung der inversen Kinematik verwendeten gradientenbasierten Näherungsverfahren werden in [Ortm12b/86] aufgezeigt. Als heuristisches Verfahren wird der genetische Algorithmus (GA) zur globalen Optimierung der Laserarbeitsräume eingesetzt. Hierzu wird auf [Ortm12b/100] verwiesen. Um eine Trajektorie der Gelenkwinkel aus einzelnen Konfigurationen des Roboters zusammensetzen wird die dynamische Optimierung eingesetzt. Dieses Verfahren wird vor allem in der Planungsrechnung eingesetzt, Einzelheiten sind [Dom11/157] zu entnehmen. Die genannten Verfahren werden in Bild 2.3 eingeordnet.

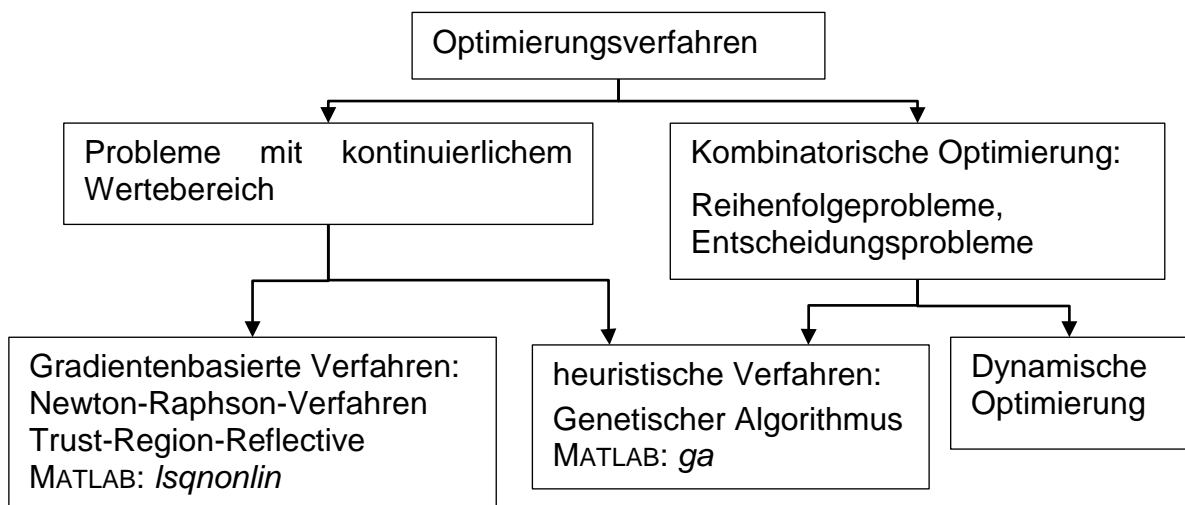


Bild 2.3: Einordnung der verwendeten Optimierungsverfahren

2.4.1 Line-Search

Das Line-Search-Verfahren ist ein lokal konvergentes, gradientenbasiertes Näherungsverfahren zur Lösung von vektorwertigen Funktionen in Abhängigkeit mehrerer Variablen. Ausgehend von einem Startwert \underline{q}_0 wird iterativ die Minimumstelle \underline{q}_{\min} einer Funktion $\underline{g}(\underline{q})$ mit der Vorschrift

$$\underline{q}_{i+1} = \underline{q}_i + \Delta \underline{q}_i \quad (2.17)$$

und

$$\Delta \underline{q}_i = - \left[\frac{\partial \underline{g}(\underline{q})}{\partial \underline{q}} \right]^{-1} \Big|_{\underline{q}=\underline{q}_i} \cdot \underline{g}(\underline{q}_i) \quad (2.18)$$

gesucht [Ortm12b/84]. Mit diesem Verfahren wird die inverse Kinematik eines Roboters iterativ berechnet, wenn die Funktion $\underline{g}(\underline{q})$ die Differenz zwischen einer Soll- und Ist- Endeffektorlage beinhaltet und \underline{q} die Gelenkwinkel des Roboters bezeichnet [Ortm11a/44]. Ist die Jacobi-Matrix nicht symmetrisch, so wird zur Invertierung die Moore-Penrose-Pseudoinverse verwendet. Dadurch wird die quadratische Summe der Gelenkwinkelgeschwindigkeit zusätzlich minimiert [Ortm11a/72].

2.4.2 Newton-Raphson-Verfahren

Das Newton-Raphson-Verfahren basiert auf der quadratischen Approximation einer skalaren Zielfunktion. Im Gegensatz zum Line-Search-Verfahren werden in der Zielfunktion alle Komponenten (Positions- und Orientierungsfehler sowie Optimierungskriterien) gewichtet aufaddiert. Die Bestimmung des nächsten Iterationsschrittes \underline{q}_{i+1} erfolgt nach

$$\Delta \underline{q}_i = - \left[\nabla^2 g(\underline{q}_{i-1}) \right]^{-1} \cdot \nabla g(\underline{q}_{i-1}). \quad (2.19)$$

Der Nabla-Operator ∇ berechnet den Gradienten bzw. ∇^2 die Hesse-Matrix der Zielfunktion g [Ortm12b/86].

2.4.3 Trust-Region-Reflective

Der Trust-Region-Reflective-Algorithmus ist das Standard-Verfahren in der MATLAB-Funktion *lsqnonlin* zur Optimierung nichtlinearer Funktionen. Ähnlich wie beim Newton-Raphson-Verfahren wird die Funktion quadratisch angenähert. Zusätzlich wird die Güte dieser Approximation bewertet und die Schrittweite daran angepasst. Bei Überschreitung von gegebenen Nebenbedingungen für die Variablen reflektiert die Suchrichtung an den Randwerten. Diese Eigenschaften führen zu einer verbesserten Konvergenz und Robustheit des Verfahrens [Ortm12b/91].

2.4.4 Genetischer Algorithmus

Die bisher vorgestellten Verfahren sind nur lokal konvergent und können ein lokales Minimum nicht verlassen. Der genetische Algorithmus (GA) ist ein heuristischer Ansatz nach dem Prinzip der Evolution, bei dem durch zufällige Veränderung und Austausch von einzelnen Parametersätzen ein globales Optimum gefunden werden kann [Ortm12b/100].

2.4.5 Dynamische Optimierung

Mit der dynamischen Optimierung² werden Probleme gelöst, die aus mehreren Stufen bestehen, auf denen Entscheidungen zwischen Zuständen getroffen werden. Durch rekursives Vorgehen von der letzten Stufe bis zum Anfang werden für jede Stufe die Entscheidungen ausgewählt, die bis zum Ende eine Zielfunktion optimieren. Das Ergebnis der dynamischen Optimierung ist eine optimale Teilpolitik für jede Stufe sowie eine optimale Gesamtpolitik. Als Politik werden die Menge der getroffenen Entscheidungen bezeichnet. Aufgrund der allgemeinen Form des Algorithmus gibt es keine MATLAB-Implementierung wie für die oben genannten Optimierungsverfahren, [Dom11/157 ff.].

Um die dynamische Optimierung auf technische Probleme mit kontinuierlichen Entscheidungsvariablen anzuwenden ist es notwendig, die Lösungsmenge zu diskretisieren, um eine endliche Zahl möglicher Entscheidungen auf jeder Stufe zu haben. Der Vorteil liegt darin, dass aus den gegebenen Entscheidungsmöglichkeiten die optimale ausgewählt wird. Die Berechnung der gesamten Lösung ist je nach Problemstruktur schneller als eine globale Lösung mit kontinuierlichen Entscheidungsvariablen auf jeder Stufe.

² Die dynamische Optimierung wird in der Literatur auch als „dynamische Programmierung“ bzw. engl. „dynamic programming“ bezeichnet.

3 Redundanzauflösung

Redundanz bezeichnet das Vorhandensein von mehr Freiheitsgraden des Systems als für die Erfüllung der Aufgabe notwendig sind. Dadurch ergeben sich Möglichkeiten zur Optimierung von Nebenbedingungen und zur Kollisionsvermeidung. Redundanz von seriellen Roboterkinematiken wird ausführlich in [Sici08/245 ff.] behandelt.

Das untersuchte Gesamtsystem, bestehend aus Roboter und Laserscanner, zeichnet sich durch mehrere Redundanzen aus. Die Eigenschaften und Zuordnung dieser Redundanzen werden in Kapitel 3.1 gezeigt. Unterschiedliche Methoden, die Redundanz bei der Berechnung der inversen Kinematik aufzulösen, werden in den darauf folgenden Kapiteln 3.2 bis 3.5 dargestellt.

3.1 Redundanzen im System

3.1.1 Kinematische Redundanz des Roboters

Die Lage des Roboter-Endeffektors im Raum besteht aus der Position und der Orientierung und wird durch sechs unabhängige Koordinaten $\underline{x}_E = [x_E, y_E, z_E, \alpha_E, \beta_E, \gamma_E]^T$ beschrieben. Die Orientierung kann auf mehrere Arten ausgedrückt werden. In dieser Arbeit werden Kardan-Winkel verwendet. Bei einem siebenachsigen Roboter mit $\underline{q} = [q_1, q_2, q_3, q_4, q_5, q_6, q_7]^T$ liegt somit Redundanz in einem Freiheitsgrad gemäß

$$\dim(\underline{q}) - \dim(\underline{x}_E) = 7 - 6 = 1$$

vor. Der hier verwendete Roboter Kuka LBR 4+ ist wie die meisten Industrieroboter so konstruiert, dass sich die letzten drei Achsen (Handachsen q_5, q_6, q_7) der kinematischen Kette in einem Punkt schneiden. Diese Konstruktionsweise wird als Zentralhand bezeichnet. Damit ist mit der Vorgabe einer Endeffektorlage die Position der Zentralhand definiert. Von den sieben Freiheitsgraden des Roboters sind nur die ersten vier Achsen (Grundachsen q_1, q_2, q_3, q_4) redundant. Die Handachsen sind deterministisch vorgegeben. Diese Eigenschaft ist in Bild 3.1 erkennbar, da die Handachsen während der dargestellten Nullraumbewegung gleich bleiben.



Bild 3.1: Nullraumbewegung, Redundanz der Grundachsen des LBR; Quelle: KUKA Roboter GmbH

3.1.2 Zusätzliche Redundanz durch Freiheitsgrade des Laserwerkzeugs

Die rotatorischen Spiegel-Freiheitsgrade und der Arbeitsraum des Lasers in der Form eines Pyramidenstumpfes werden in Kapitel 2.2.1 beschrieben.

Für diese Arbeit werden die Freiheitsgrade \underline{q}_1 des Laserwerkzeugs als rein translato-
risch angenommen. Die möglichen Brennpunkte des Lasers bilden mit dieser Vereinfachung einen würfelförmigen Arbeitsraum mit der Kantenlänge $a = 10\text{mm}$. Das Werkzeug wird durch eine homogene Transformationsmatrix mit

$$\mathbf{T}_{\text{LWZ}}(\underline{q}_1) = {}^7\mathbf{T}_E = \begin{bmatrix} 0 & 0 & 1 & 12,85 \text{ mm} + q_{11} \\ 0 & 1 & 0 & 228,93 \text{ mm} + q_{12} \\ -1 & 0 & 0 & 91,35 \text{ mm} + q_{13} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

beschrieben. Darin ist die Rotation und Translation vom Werkzeugflansch des Roboters bis zum Brennpunkt des Lasers enthalten. Die Werte wurden mit dem CAD-Modell des Laserwerkzeugs berechnet. Die technischen Grenzen für die Laserfreiheitsgrade sind zu

$$-5 \text{ mm} \leq q_{11}, q_{12}, q_{13} \leq 5 \text{ mm}$$

festgelegt.

3.1.3 Redundanz aus der Bearbeitungsaufgabe

Ähnlich wie Werkzeuge beim Fräsen und Bohren ist der Laserstrahl bei der Laserbearbeitung rotationssymmetrisch. Damit genügen zur vollständigen Beschreibung der Lage des Werkzeugs fünf unabhängige Koordinaten und zur Bearbeitung ist eine fünfsichtige Maschine ausreichend. Bei der Darstellung der Endeffektorlage in Kar-dan-Winkelkonvention wird die letzte Drehung mit dem Winkel γ um die z-Achse (Symmetrieachse) vernachlässigt. Damit ergeben sich die Endeffektorkoordinaten zu

$$\underline{x}_{E,rsym} = [x_E, y_E, z_E, \alpha_E, \beta_E]^T \quad (3.2)$$

und der Roboter ist mit

$$\dim(\underline{q}) - \dim(\underline{x}_{E,rsym}) = 7 - 5 = 2$$

in zwei FHG redundant. Das Gesamtsystem aus Roboter und Laser ist in fünf FHG redundant.

3.2 Nicht-redundante Inverse Kinematik des Roboters

Bevor eine Betrachtung des Gesamtsystems durchgeführt wird, wird nur der Roboter für sich betrachtet. Die analytisch-geometrische Berechnung der inversen Kinematik des Roboters ist nur möglich, wenn auf einen Freiheitsgrad verzichtet wird und somit ein bestimmtes nichtlineares Gleichungssystem aus sechs Soll-Endeffektorfreiheitsgraden und sechs unbekanntem Gelenkfreiheitsgraden entsteht. Dazu wird die zusätzliche Achse E1 zu Null gesetzt.

Der Nachteil dieser Methode ist, dass die Redundanz des Roboters nicht genutzt wird. Vorteilhaft ist, dass die Berechnung in Echtzeit durchführbar ist und das Ergebnis unabhängig von den Startwerten mit vernachlässigbarem Fehler vorliegt, falls die Konfiguration technisch möglich ist. Insgesamt gibt es acht unterschiedliche Konfigurationen der Gelenkwinkel für jede Endeffektorlage. Diese Möglichkeiten sind in Bild A.1 im Anhang dargestellt (Ellenbogen oben/ unten, Hand innen/ außen, Roboter hinten/ vorne).

3.2.1 Berechnung der Position der Zentralhand

Zunächst wird die Transformationsmatrix zum Werkzeugflansch 0T_7 mit

$${}^0T_7 = {}^0T_E \cdot {}^7T_E^{-1} = \begin{bmatrix} {}^0R_7 & ({}^0)z_7 \\ 0 & 1 \end{bmatrix} \quad (3.3)$$

berechnet. 7T_E ist die Transformation des Werkzeugs aus Gl. (3.1). 0T_E ist die Transformation vom Basis-KS zum Endeffektor-KS. Die Position der Zentralhand ${}_{(0)}\underline{x}_H$ mit Berücksichtigung der DH-Parameter aus Tabelle 2.1 wird mit

$${}_{(0)}\underline{x}_H = {}_{(0)}\underline{x}_6 = {}_{(0)}\underline{x}_7 - {}^0R_7 \cdot \begin{bmatrix} 0 \\ 0 \\ -d_7 \end{bmatrix} = \begin{bmatrix} x_H \\ y_H \\ z_H \end{bmatrix} \quad (3.4)$$

bestimmt. Dabei bezeichnen die Indizes „7“ den Werkzeugflansch, „E“ den Endeffektor (Brennpunkt des Laserstrahls) und „6“ bzw. „H“ das siebte Gelenk, das im Mittelpunkt der Zentralhand liegt.

3.2.2 Berechnung der Grundachsen

Unter der Annahme, dass die Zusatzachse E1 zu Null gesetzt ist ($q_3 = 0$), berechnen sich die verbliebenen Grundachsen durch Sinus- und Kosinus-Satz gemäß der Gleichungen

$$q_1 = \text{atan2}(y_H, x_H), \quad (3.5)$$

$$q_4 = \mp \arccos\left(\frac{d_3^2 + d_5^2 - c^2}{-2 \cdot d_3 \cdot d_5}\right) = \begin{cases} q_{4,o} \\ q_{4,u} \end{cases} \quad (3.6)$$

$$q_2 = \pm \left| \frac{\pi}{2} - \arcsin\left(\frac{z_H - d_1}{c}\right) - \arcsin\left(\frac{d_5}{c} \cdot \sin|q_4|\right) \right| = \begin{cases} q_{2,o} \\ q_{2,u} \end{cases} \quad (3.7)$$

und mit

$$c^2 = x_H^2 + y_H^2 + (z_H - d_1)^2.$$

Die Funktion atan2 erweitert die arctan-Funktion um die Betrachtung der Vorzeichen gemäß

$$\text{atan2}(y, x) = \begin{cases} \arctan(y/x) & x > 0 \\ \arctan(y/x) + \pi & x < 0 \end{cases}$$

Aus den beiden möglichen Fällen in Gl. (3.6) ergeben sich die Ellenbogenkonfigurationen „unten“ und „oben“, die im Vergleich von Bild A.1a mit Bild A.1c erkennbar sind. Zusätzlich wird noch der Fall definiert, dass sich der Roboter nach dem Ergebnis aus Gl. (3.5) weiter nach hinten dreht als die Gelenkwinkelgrenzen es zulassen. Dafür wird zwischen den Konfigurationen „hinten“ und „vorne“ unterschieden, die im Unterschied Bild A.1c mit Bild A.1e deutlich werden. In diesem Fall wechselt das Vorzeichen des zweiten arcsin-Terms in Gl. (3.7) und der erste Gelenkwinkel wird um 180° gedreht.

3.2.3 Berechnung der Handachsen

Nachdem die Position und Orientierung der Zentralhand durch die Grundachsen festgelegt sind, werden die Winkel der Handachsen durch die folgenden Gleichungen eindeutig bestimmt. Die Herleitung erfolgt über das Gleichsetzen der Matrixeinträge der Rotationsmatrix von der Zentralhand bis zum Werkzeugflansch 4R_7 für die unbekannten Gelenkwinkel mit den Einträgen der bekannten Rotationsmatrix ${}^4R_{7,\text{soll}}$ nach

$${}^4R_{7,\text{soll}} = [{}^0R_{4,\text{ist}}]^T \cdot {}^0R_{7,\text{soll}} = \begin{bmatrix} N_x & O_x & A_x \\ N_y & O_y & A_y \\ N_z & O_z & A_z \end{bmatrix}. \quad (3.8)$$

Die Vorgabe von ${}^0R_{7,\text{soll}}$ erfolgt aus der Soll-Endeffektorlage aus Gl. (3.3). ${}^0R_{4,\text{ist}}$ ergibt sich aus den oben berechneten Winkeln der Grundachsen. Der zweite Winkel der Zentralhand q_6 ergibt sich zu

$$q_6 = \begin{cases} -\arccos(A_z) = q_{6,i} \\ +\arccos(A_z) = q_{6,a} \end{cases}. \quad (3.9)$$

Es ergeben sich zwei mögliche Konfigurationen für die Stellung der Zentralhand: Nach „innen“ oder nach „außen“ gedreht, die im Unterschied Bild A.1c zu Bild A.1d zu erkennen sind. Daraus folgen unterschiedliche Winkel für die beiden anderen Achsen der Zentralhand q_5 und q_7 nach

$$q_5 = \text{atan2}\left(\frac{-A_y}{\sin(q_6)}, \frac{-A_x}{\sin(q_6)}\right), \quad (3.10)$$

$$q_7 = \text{atan2}\left(\frac{-O_z}{\sin(q_6)}, \frac{N_z}{\sin(q_6)}\right). \quad (3.11)$$

3.3 Redundante inversen Kinematik des Roboters

In diesem Kapitel werden die unterschiedlichen numerischen Methoden zur Lösung der inversen Kinematik beschrieben und deren Voraussetzungen gezeigt. Im vorherigen Kapitel wurde gezeigt, dass die Redundanz des Roboters nur in den Grundachsen vorliegt. Die dazu notwendige vorgegebene Orientierung des Endeffektors und damit der Zentralhand führt nun zu einem Zielkonflikt mit der Redundanz, die sich aus dem rotationssymmetrischen Werkzeug ergibt. Wird die Rotationssymmetrie des Laserstrahls berücksichtigt, ist die Position der Zentralhand nicht mehr konstant vorgegeben. In den Unterkapiteln 3.3.1 bis 3.3.4 werden Ansätze vorgestellt, mit denen unter den gegebenen Voraussetzungen eine Lösung möglich ist.

Gemeinsam ist allen Ansätzen, dass eine Gütefunktion mit dem Trust-Region-Reflective-Algorithmus (TRRA) gelöst wird. Nur die Wahl der Gütefunktion und der Variablen wird geändert. Damit ist die Konvergenz und Rechenzeit der Verfahren von der jeweiligen Startposition und der Lage von Singularitäten abhängig. Die Verfahren sind somit nicht echtzeitfähig und eignen sich nur für die Offline-Optimierung von Trajektorien.

Die gezeigten Gütefunktionen sind nicht auf den TRRA beschränkt. Die Verwendung eines heuristischen Verfahrens wie des genetischen Algorithmus (GA) anstelle eines gradientenbasierten Verfahrens hat neben der globalen Konvergenz den Vorteil, dass keine Jacobi-Matrizen berechnet werden müssen und somit in der gleichen Zeit mehr Funktionsauswertungen möglich sind.

Die Größe des möglichen Wertebereiches und damit die Rechenzeit bis zur Konvergenz eines heuristischen Verfahrens steigen exponentiell mit der Anzahl der zu optimierenden Variablen. Das Verfahren der alleinigen Optimierung der Grundachsen in Kapitel 3.3.3 ist zur Anwendung mit dem GA geeignet, da nur vier Variablen optimiert werden.

Im Folgenden werden die Begriffe Leistungsmerkmal, Kennzahl, Zielfunktion und Gütefunktion für diese Arbeit definiert:

- Leistungsmerkmal: Quantitative Beschreibung einer Eigenschaft des Roboters nach Kapitel 2.3,
- Kennzahl: Umwandlung eines Leistungsmerkmals in den Bereich von 0 (optimaler Wert) bis 1 (schlechtester eintretender Wert),
- Zielfunktion: Gewichtete Zusammenfassung mehrerer Leistungsmerkmale,
- Gütefunktion: Funktion, die in einem Optimierungsverfahren minimiert wird und sowohl Zielfunktionswerte als auch Lagefehler des Roboters enthält.

3.3.1 Optimierung aller Achsen, sechs Endeffektorfreiheitsgrade

Bei der Optimierung aller sieben Achsen des Roboters mit vorgegebener Endeffektorlage wird ein Freiheitsgrad für die Optimierung von Zielfunktionen benutzt. Drei Freiheitsgrade werden für die translatorische und drei für die rotatorische Endeffektorlage eingesetzt. Damit ergeben sich Gütefunktion \underline{f}_1 und Optimierungsvariablen \underline{q}_x nach

$$\underline{f}_1(\underline{q}_x) = \begin{bmatrix} \Delta \underline{t} \\ \Delta \underline{r} \\ z_K \end{bmatrix}, \quad \underline{q}_x = [q_1, q_2, q_3, q_4, q_5, q_6, q_7]^T. \quad (3.12)$$

Für den Translationsfehler $\Delta \underline{t}(\underline{q}_x)$ wird die Differenz aus Ist- und Soll-Position gebildet. Für den Rotationsfehler $\Delta \underline{r}(\underline{q}_x)$ können nicht die Kardan-Winkel direkt verglichen werden, da es sich um Winkel in mitgedrehten KS handelt [Ortm11a/162]. Stattdessen wird der Rotationsfehler durch Hin- und Rückrotation von Soll- und Ist-Rotationsmatrix berechnet, [Cork11/53]. Die Berechnung erfolgt mit der Funktion *tr2delta* der Robotics Toolbox [CorRTb]. Die zusätzlichen skalaren Zielfunktionskriterien $z_K(\underline{q}_x)$ werden durch die Gewichtung der Kennzahlen K der Leistungsmerkmale mit dem Vektor $\underline{\gamma}$ zu

$$z_K = \Gamma \cdot [K_{hzz}, K_\mu, K_\eta, K_\Delta, K_{EB}, K_q] \cdot \underline{\gamma} \quad (3.13)$$

zusammengefasst. Im TRR-Algorithmus wird gefordert, dass sowohl der Lagefehler $[\Delta \underline{t}, \Delta \underline{r}]$, als auch die Zielfunktion z_K minimiert werden sollen. Um das zu erreichen muss die Größenordnung der Werte ungefähr gleich sein. Da die in z_K enthaltenen Kennzahlen im Bereich 0 bis 1 skaliert sind, wird ein Vorfaktor Γ so gewählt, dass z_K und $[\Delta \underline{t}, \Delta \underline{r}]$ in der gleichen Größenordnung sind.

3.3.2 Optimierung aller Achsen, fünf Endeffektorfreiheitsgrade

Um die Symmetrie des Werkzeugs zu berücksichtigen wird anstelle des Rotationsfehlers mit drei Komponenten $\Delta \underline{r}(\underline{q}_x)$ nur der Winkelfehler $K_w(\underline{q}_x)$ zwischen Laserstrahl und Oberflächennormale benutzt (siehe Kap. 2.3.7). Die Gütefunktion \underline{f}_2 berechnet sich zu

$$\underline{f}_2(\underline{q}_x) = \begin{bmatrix} \Delta \underline{t} \\ K_w \\ z_K \end{bmatrix} \quad \text{mit} \quad \underline{q}_x = [q_1, q_2, q_3, q_4, q_5, q_6, q_7]^T. \quad (3.14)$$

3.3.3 Optimierung der Grundachsen, sechs Endeffektorfreiheitsgrade

Wird berücksichtigt, dass der Roboter nur Redundanz in den Grundachsen und nicht in allen Achsen aufweist, wird die inverse Kinematik wie in Abschnitt 3.2 schrittweise berechnet. Zuerst wird die Soll-Position der Zentralhand aus der Soll-Endeffektorposition gemäß Abschnitt 3.2.1 berechnet. Anschließend erfolgt die Optimierung der Grundachsen mit der Gütefunktion f_3 gem.

$$\underline{f}_3(\underline{q}_x) = \begin{bmatrix} \Delta \underline{t}_H \\ z_K \end{bmatrix}, \quad \underline{q}_x = [q_1, q_2, q_3, q_4]^T. \quad (3.15)$$

Es wird nun der Translationsfehler der Zentralhand $\Delta \underline{t}_H$ mit

$$\Delta \underline{t}_H = {}_{(0)}\underline{r}_{H,ist} - {}_{(0)}\underline{r}_{H,soll}$$

optimiert. Um die Erfüllung der Zielfunktion z_K bestimmen zu können, werden bereits innerhalb der Gütefunktion f_3 mit den Gl. (3.8) bis (3.11) aus Kapitel 3.2.3 die Winkel der Handachsen berechnet. Der Vorteil dieses Verfahrens gegenüber Abschnitt 3.3.1 liegt darin, dass nur vier Variablen optimiert werden müssen und dass kein Rotationsfehler des Endeffektors auftritt, da die Handachsen ausgehend von einer Übereinstimmung der Orientierung analytisch berechnet werden.

3.3.4 Optimierung der Grundachsen, fünf Endeffektorfreiheitsgrade

Wie bereits in der Einleitung zu Kap. 3.3 ausgeführt, ist die Optimierung der Grundachsen von der bekannten Endeffektororientierung abhängig, was die Freigabe eines Rotationsfreiheitsgrades nicht erlaubt. Um die sich daraus ergebende Redundanz zu nutzen wird der Kardanwinkel γ_E auf einen festen Wert gesetzt und dann mit dem Verfahren aus Kapitel 3.3.3 die inverse Kinematik gelöst. Durch die Variation des Wertes γ_E in einer äußeren Schleife wird ein optimaler Wert für das System mit den Freiheitsgraden $[q_1, q_2, q_3, q_4, \gamma_E]$ gefunden. In dieser Arbeit werden in der äußeren Schleife zunächst $\gamma_{E,i}$ mit linearem Abstand vorgegeben. Alternativen hierzu sind die Verwendung von gradientenbasierten oder heuristischen Ansätzen, die eine Gütefunktion nur in Abhängigkeit von γ_E benötigen.

3.4 Nullraumbewegung

Aufgrund der nur lokalen Konvergenz der numerischen Berechnung der inversen Kinematik ist es teilweise notwendig, Anfangswerte mit dem geometrisch-analytischen nicht-redundanten Ansatz aus Kapitel 3.2 zu bestimmen. Wird mit diesen Anfangswerten eine weitergehende Optimierung nach Kapitel 3.3 durchgeführt, verlässt der Winkel für die Zusatzachse E1 den Startwert von Null nur langsam. Diese Beschränkung wird durch die Nullraumanalyse aufgehoben.

Für die Nullraumanalyse werden die oben definierten Gütefunktionen mit einem konstanten Wert $q_{i, NR}$ für einen Gelenkwinkel q_i definiert. Die restlichen Gelenkwinkel werden so berechnet, dass die Endeffektorlage erreicht wird. Nachdem $q_{i, NR}$ in einer äußeren Schleife variiert wurde, liegt die komplette Nullraumbewegung für die Endeffektorlage vor und jede berechnete Pose erfüllt die inverse Kinematik.

3.5 Betrachtung des Gesamtsystems

Der Schwerpunkt der obigen Kapitel liegt auf dem Roboter und der Auflösung seiner Redundanz. In diesem Abschnitt wird das Gesamtsystem aus Laser und Roboter betrachtet. Die Freiheitsgrade des Roboters zeichnen sich dadurch aus, dass ihr Einfluss auf die Endeffektorlage im Vergleich zu den Laserfreiheitsgraden sehr groß ist. Der Hebelarm der Robotergelenkwinkel beträgt in den Grundachsen bis zu 500 mm und in den Handachsen durch das relativ große Werkzeug noch bis zu 150 mm. Der Laserscanner kann dagegen nur einen Weg von 10 mm verfahren. Es ist möglich, Roboter und Laser in einer Gütefunktion gemeinsam zu optimieren. Aufgrund der stark unterschiedlichen Steuerbarkeit der Endeffektorlage durch die Freiheitsgrade wird die Optimierung nacheinander durchgeführt.

Vor der Optimierung des Roboters wird geprüft, ob mit der Verstellung des Lasers allein die Soll-Endeffektorlage von der Startlage aus erreicht werden kann. Ist dies der Fall, muss keine Änderung der Robotergelenkwinkel mehr durchgeführt werden. Reicht die Verstellung des Lasers nicht aus, wird der Laser an den Rand seines Arbeitsbereiches verfahren und der restliche Weg mit den Robotergelenken berechnet.

4 Optimierung der Bahnplanung

Eine Bahn oder Trajektorie bezeichnet den zeitlichen Verlauf eines Weges im kartesischen Raum oder im Achsraum des Roboters. In diesem Kapitel werden Verfahren zur Bestimmung der Gelenkwinkeltrajektorie entwickelt. Die Bahn der kartesischen Endeffektorlage wird bereits im Vorfeld durch die zu bearbeitende Geometrie festgelegt. Zeitvorgaben für einzelne Bahnpunkte ergeben sich aus der Geschwindigkeit und Beschleunigung des Endeffektors sowie der Bearbeitungsdauer und werden zunächst nur durch die Forderung der Gelenkwinkelstetigkeit berücksichtigt. Dies wird durch Gelenkwinkelbegrenzungen und Hindernisse erschwert. Gleichzeitig wird versucht, vorgegebene Zielfunktionen zu optimieren. Die hier vorgestellten Algorithmen führen die in [Ölt10] begonnene diskrete Trajektorienoptimierung weiter, indem insbesondere der stetige Verlauf im Achsraum und die zusätzliche aufgabenbezogene Redundanz der Endeffektororientierung bei der Laserbearbeitung berücksichtigt werden.

Zunächst wird in Kapitel 4.1 die kontinuierliche Optimierung aller Freiheitsgrade des Systems untersucht. Die gleichzeitige Bewegung von Laser und Roboter ist bei Laserbearbeitung mit kontinuierlichem Strahlverlauf sinnvoll. Bei gepulstem Laserstrahl eignet sich eine getrennte Bewegung von Laser und Roboter, die in Kapitel 4.2 behandelt wird. Bei der getrennten Bewegung wird der Roboter in eine Pose gefahren und alle von dort erreichbaren Bahnpunkte werden mit Laserpulsen bearbeitet. Durch das Anhalten des Roboters wird die Gesamtdauer der Bewegung erhöht, dafür ist eine genauere Überprüfung der Roboterposition z. B. mit externer Bilderkennung möglich. Unterschiedliche Latenzzeiten bei der Ansteuerung des Roboters und des Lasers haben bei getrennter Bewegung keinen Einfluss. Bei gemeinsamer Bewegung entsteht durch unterschiedliche Latenzzeiten eine Bahnabweichung des Lasers.

Die Berechnung der getrennten Bewegung ist nicht in Echtzeit bei der Eingabe einer Bahn in den Roboter, sondern nur offline möglich. Zur Echtzeitoptimierung eignen sich einige der Algorithmen aus Kapitel 3.3 bei gemeinsamer Bewegung von Roboter und Laser.

4.1 Gemeinsame Optimierung aller Freiheitsgrade

Bei der gemeinsamen Optimierung der Freiheitsgrade von Roboter und Laser erfolgt die Berechnung von Roboter gelenkwinkeln und Laserauslenkung für jeden Bahnpunkt wie in Kapitel 3.5 vorgestellt: Zuerst wird die Durchführbarkeit nur mit Umstellung des Lasers geprüft. Liegt der Zielpunkt außerhalb des Laserarbeitsraums, erfolgt die Berechnung der Roboter gelenke mit den Methoden aus Kapitel 3. Die Bewegung des Lasers wird aufgrund der größeren Genauigkeit, geringerer Hysterese und niedrigerem Energieverbrauch günstiger als die Roboterbewegung bewertet. Als Ergebnis wird der Laser bis an den Rand seines Arbeitsraumes ausgelenkt und anschließend nur noch der Roboter bewegt

Je nach gewähltem Berechnungsverfahren für die inverse Kinematik werden alle Möglichkeiten zur Bestimmung der Gelenkwinkelkonfiguration für einen Bahnpunkt mit drei verschachtelten Schleifen berechnet:

1. Acht mögliche Anfangskonfigurationen, die die Vorzeichen der Roboter gelenkwinkel beinhalten (siehe Bild A.1),
2. unterschiedliche Werte für den redundanten Roboter gelenkwinkel (Nullraum-analyse),
3. Drehung des Endeffektor-KS um die z-Achse mit dem Winkel γ_E .

Die aus diesen Kombinationen entstehenden Lösungen für die Gelenkwinkel werden mit den oben vorgestellten Zielfunktionen bewertet und es wird für jeden Bahnpunkt die beste Lösung gewählt. Der wesentliche Nachteil der gemeinsamen Optimierung der FHG ist, dass die globale Optimierung über alle Bahnpunkte im Nachhinein nur schlecht möglich ist. Die Ursache ist, dass für die Positionierung des Lasers die vorherige Lage des Roboters bekannt sein muss. Bei einer globalen Optimierung, wie in Kapitel 4.2.4 vorgestellt, muss dies zum Zeitpunkt der Berechnung der inversen Kinematik nicht bekannt sein. Ein weiterer Nachteil ist, dass die drei zusätzlichen Freiheitsgrade des Lasers kaum genutzt werden. Vorteile sind der geringere Rechenaufwand und die Möglichkeit, die inverse Kinematik online zu berechnen.

4.2 Getrennte Optimierung von Roboter und Laserwerkzeug

Die getrennte Betrachtung von Roboter- und Laserfreiheitsgraden ermöglicht die Vermeidung der oben genannten Nachteile der gemeinsamen Optimierung. Es wird ausgenutzt, dass der durch Spiegelverdrehung und Linsenverschiebung erreichbare Laserarbeitsraum von $10 \times 10 \times 10$ mm mehrere Bahnpunkte beinhalten kann. Dies ist bei der Laserablation der Fall. Die Zuordnung der Bahnpunkte zu einem Laserarbeitsraum wird in Kapitel 4.2.1 anhand einer Beispieltrajektorie gezeigt. Für die Optimierung der Roboterbewegung in einem Bahnpunkt wird die Vorgehensweise in Abschnitt 4.2.2 verdeutlicht. Die eigentliche Berechnung der Laserauslenkung für den einzelnen Bahnpunkt wird in Kapitel 4.2.3 vorgestellt. Die Optimierung der gesamten Bahn wird schließlich in Kapitel 4.2.4 dargestellt.

4.2.1 Optimierung der Laser-Arbeitsräume

Die Optimierung der Laserbewegung besteht darin, die Laser-Arbeitsräume so im Raum zu verteilen, dass alle zu bearbeitenden Bahnpunkte enthalten sind und gleichzeitig der zurückzulegende Weg des Roboters minimal ist. Dieses Mengenüberdeckungsproblem gehört der Klasse der Nicht-Polynomial-vollständigen³ Probleme an und ist nicht analytisch lösbar. Erschwerend kommt hier hinzu, dass die Lösungsmenge unendlich ist, da der Mittelpunkt eines Laserarbeitsraumes eine stetige Variable ist. Um eine gute Lösung bei geringem Rechenaufwand zu erhalten, wird ein gieriger Algorithmus (Greedy-Algorithmus) angewendet [Bur10/114]. In Bild 4.1 ist das Ergebnis des Algorithmus für eine Beispieltrajektorie nach sechs Zwischenschritten abgebildet. Die als blaue Kreise dargestellten zu bearbeitenden Bahnpunkte bilden ein Dreieck mit den Eckpunkten $[-400, -50, 350]$, $[-450, -50, 350]$ und $[-450, 50, 375]$ mm. In jedem Schritt des Algorithmus wird ein würfelförmiger Laser-Arbeitsraum platziert. Dabei werden die x -, y - und z -Koordinaten des Mittelpunktes des Würfels hinsichtlich im Folgenden beschriebener Kriterien optimiert.

³ NP-vollständige Probleme sind in der Anzahl der Rechenschritte zur optimalen Lösung nicht durch ein Polynom begrenzt. Bei steigender Problemgröße wächst die Rechendauer exponentiell.

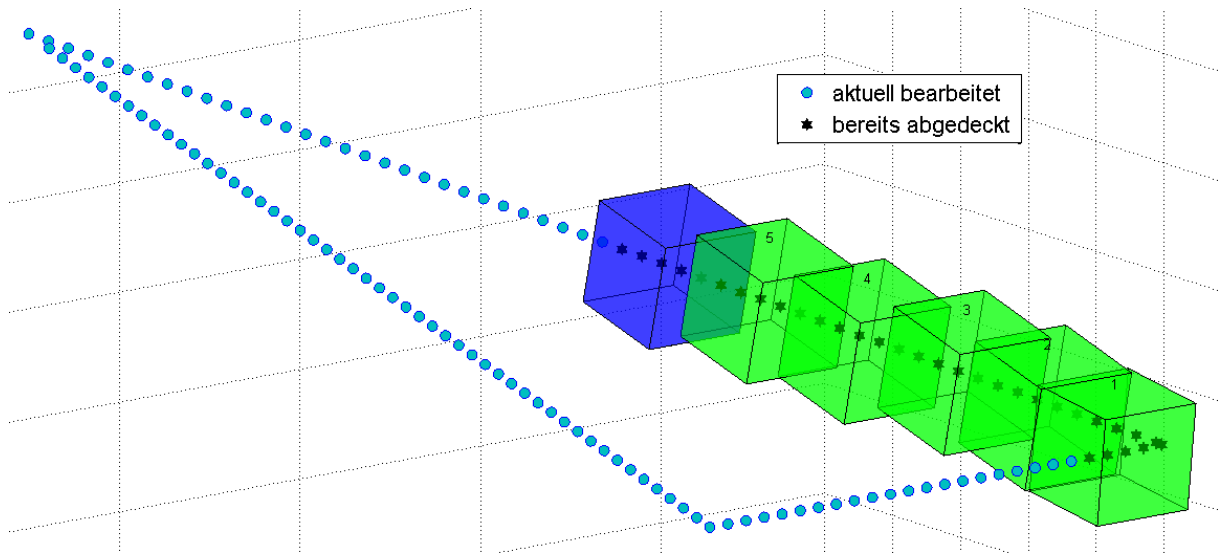


Bild 4.1: Abdeckung der Beispieltrajektorie mit Laserarbeitsräumen

Die Optimierung des Mittelpunktes eines Laserarbeitsraumes erfolgt mit dem genetischen Algorithmus. Die x-, y- und z-Koordinaten mehrerer möglicher neuer Mittelpunkte werden solange leicht verändert und untereinander ausgetauscht, bis ein Punkt ausgewählt wird, der die vorgegebene skalare Zielfunktion z_{GA} nach

$$z_{GA} = -N_I + d_v + 0,9 \cdot N_N \quad (4.1)$$

minimiert. Dabei ist

- N_I die Anzahl der Bahnpunkte, die im Laserarbeitsraum enthalten sind,
- d_v der Abstand des Lasermittelpunktes zum vorherigen Lasermittelpunkt und
- N_N die Anzahl der Bahnpunkte, die nicht im Laserarbeitsraum enthalten sind, aber näher am Lasermittelpunkt sind als der am weitesten entfernte, innerhalb liegende Bahnpunkt.

Durch die Wahl der Zielfunktion sollen zum einen so viele Bahnpunkte wie möglich in einem Arbeitsraum liegen und zum anderen die gesamte Strecke klein bleiben. Die Verschlechterung der Zielfunktion durch nicht enthaltene nahe Punkte soll verhindern, dass diese erst ganz am Schluss mit einem Umweg aufgenommen werden.

Zur einfacheren Berechnung der Zugehörigkeit von Bahnpunkten zum Arbeitsraum wird dieser als Kugel mit Durchmesser 10 mm und nicht als Würfel mit Kantenlänge 10 mm angenommen. Dadurch wird bei der Laserarbeitsraumoptimierung keine Orientierung des Roboter-Endeffektors vorgegeben und zur Optimierung sind nur drei Variablen notwendig. Die Kenndaten des genetischen Algorithmus für dieses Problem sind in folgender Tabelle 4.1 noch einmal zusammengefasst.

Tabelle 4.1: Bedeutung der Merkmale des GA für die Laserarbeitsraum-Positionsoptimierung

| GA-Parameter | Bedeutung |
|--------------------------------|---|
| Merkmal (Optimierungsvariable) | x-, y- und z-Koordinaten der Lasermittelpunkte |
| Population | Gesamtheit der verglichenen Lasermittelpunkte |
| Anfangspopulation | Bereich um den letzten Lasermittelpunkt herum sowie alle übrigen Bahnpunkte |
| Zielfunktion, Fitnessfunktion | Abstand zum vorherigen Punkt, Anzahl der enthaltenen Punkte |

Bild 4.2 zeigt die Verteilung der möglichen Lasermittelpunkte für einen Schritt bei der Optimierung der Beispieltrajektorie. Die Bahnpunkte sind in Bild 4.2 blau dargestellt, mögliche Lasermittelpunkte je nach Höhe der Zielfunktion schwarz, orange oder grün. Der vorherige Arbeitsraum liegt in der unteren rechten Ecke des Dreiecks. Aufgrund der kurzen Entfernung ist dort eine Anhäufung der als gut bewerteten Mittelpunkte zu sehen.

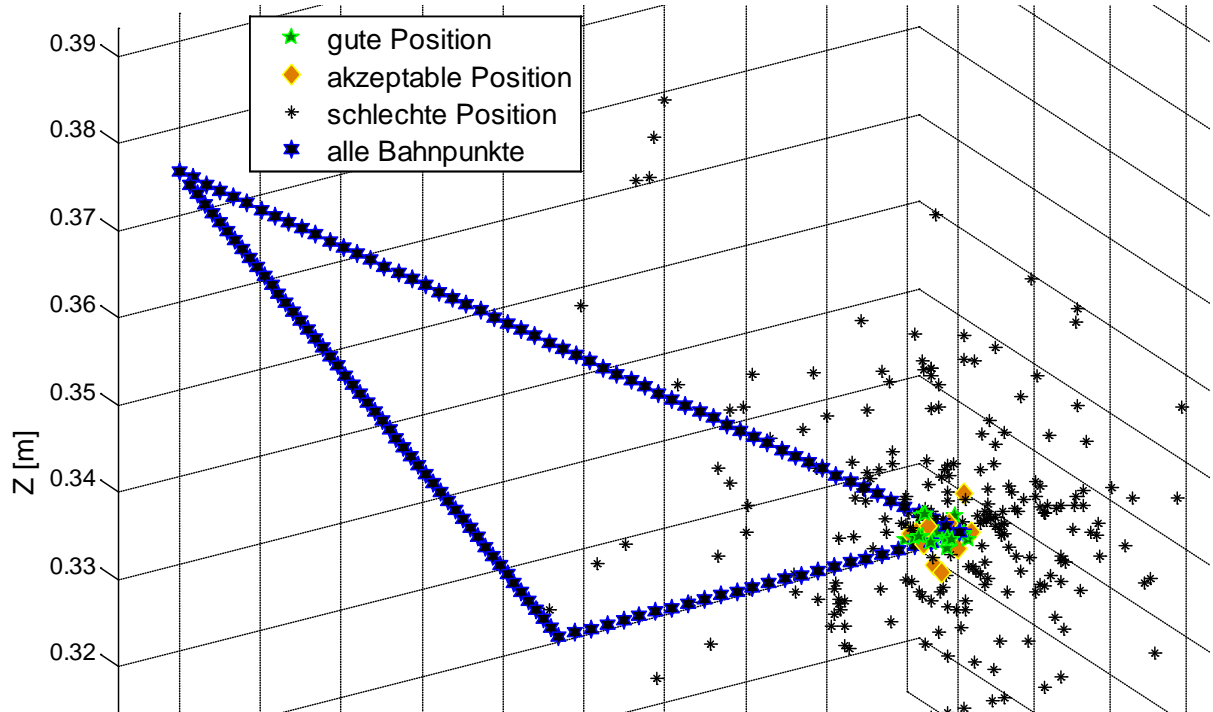


Bild 4.2: Verteilung der möglichen Lasermittelpunkte in einem Schritt des genetischen Algorithmus

Der resultierende Weg des Lasermittelpunktes ist in Bild 4.3 in blau eingezeichnet. Der Greedy-Algorithmus führt in diesem Fall nicht zum optimalen Ergebnis, da die Zielfunktion hier dazu neigt, Arbeitsräumen mit vielen Punkten zu folgen und so einzelne Punkte übrig bleiben, die zum Schluss nur mit einem langen Weg erreicht werden können.

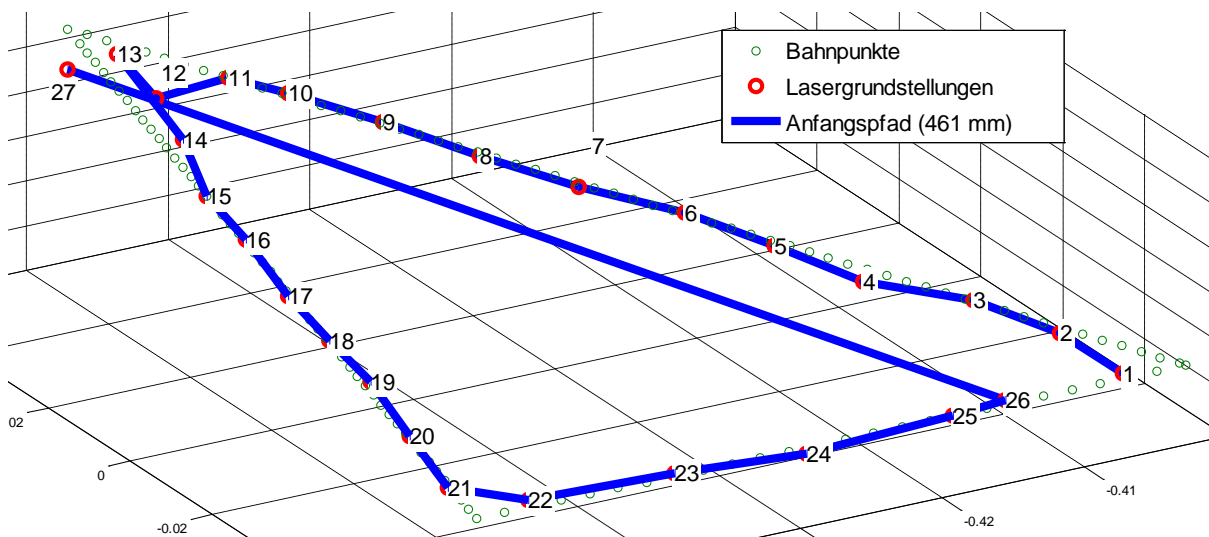


Bild 4.3: Reihenfolge der Lasergrundstellungen nach der Optimierung mit dem Greedy-Algorithmus

Zur Verbesserung dieses Ergebnisses wird die Reihenfolge der Punkte optimiert. Diese Art von Reihenfolgeproblemen ist als Problem des Handlungsreisenden (Traveling Salesman-Problem) bekannt und ebenfalls NP-vollständig. Zur Lösung wird ein genetischer Algorithmus eingesetzt [MLTSP13]. Mehrere unterschiedliche Reihenfolgen der Lasermittelpunkte werden zufällig vertauscht, bis die Reihenfolge mit dem kürzesten Gesamtweg als Ergebnis ausgesucht wird. Die Eigenschaften des genetischen Algorithmus sind in Tabelle 4.2 zusammengefasst.

Tabelle 4.2: Bedeutung der Merkmale des GA für die Laserarbeitsraum-Reihenfolgeoptimierung

| GA-Parameter | Bedeutung |
|--------------------------------|--|
| Merkmal (Optimierungsvariable) | Reihenfolge der Lasermittelpunkte |
| Population | Verschiedene Reihenfolgen |
| Anfangspopulation | Zufällig generierte Reihenfolgen sowie ursprüngliche Reihenfolge |
| Zielfunktion, Fitnessfunktion | Gesamtstrecke zwischen Mittelpunkten |

Das Ergebnis der Optimierung ist Bild 4.4 zu entnehmen. Durch die Kombination der beiden Algorithmen kann für beliebige Trajektorien eine gute Anordnung der Laserarbeitsräume berechnet werden. Ein optimales Ergebnis kann durch die heuristischen Algorithmen und die Komplexität der Probleme jedoch nicht erreicht werden.

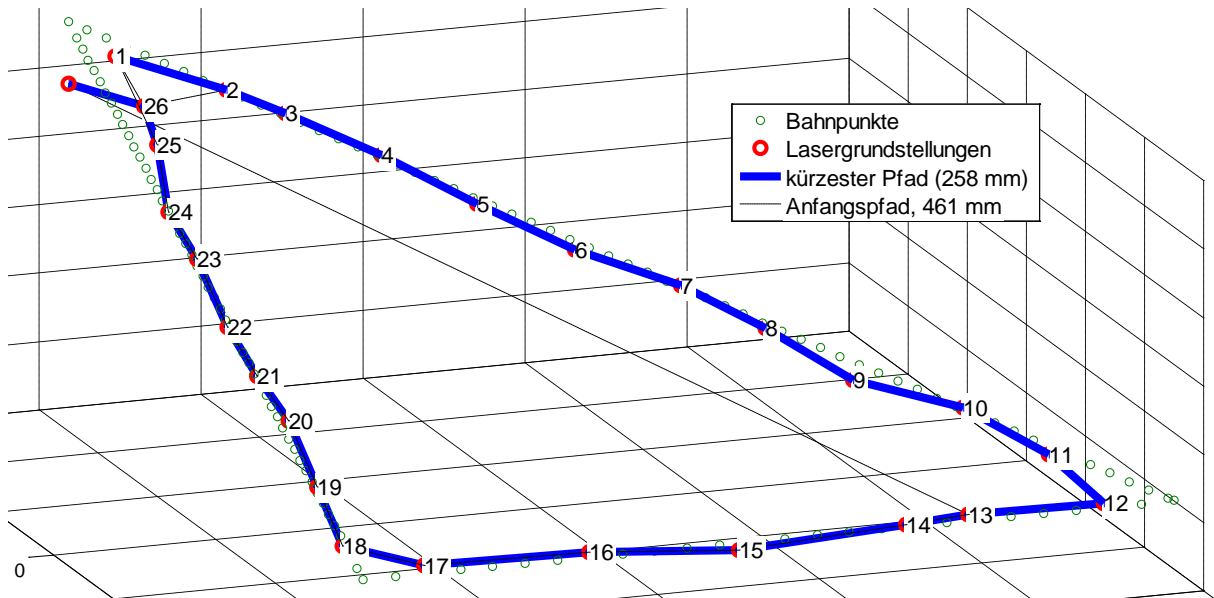


Bild 4.4: Reihenfolge der Lasergrundstellungen nach der Optimierung mit dem TSP-Algorithmus

4.2.2 Optimierung der Roboterbewegung

In Kapitel 3 werden unterschiedliche Methoden zur Lösung der inversen Kinematik gezeigt, mit denen die Redundanz des Roboters aufgelöst wird. Aufbauend auf diesen Verfahren wird die Gelenkstellung des Roboters für jeden Laserarbeitsraum berechnet. Aufgrund der Redundanz gibt es eine Vielzahl unterschiedlicher Gelenkwinkel, die die Positionsvorgabe im kartesischen Arbeitsraum erfüllen. Die Auswahl erfolgt nach der Berechnung aller Möglichkeiten des Roboters für die gesamte Bahn. Die Redundanz liegt in drei Stufen vor, die verschachtelt berechnet werden.

Erste Stufe: Konfiguration des Roboters

Unabhängig von der redundanten Zusatzachse hat der Roboter acht kinematisch redundante Konfigurationen, die die gleiche Endeffektorlage zur Folge haben. Die Konfigurationen „Roboter hinten/ vorne“, „Ellenbogen oben/ unten“ und „Hand innen/ außen“ sind von den Vorzeichen der Achsen A2, A4 und A6 abhängig (siehe Bild A.1 im Anhang). Da die Konvergenz der vorgestellten Optimierungsverfahren von ihren Anfangswerten abhängig ist, werden die Verfahren mit acht unterschiedlichen An-

fangswerten entsprechend der Kombinationen der Konfigurationen berechnet. Die nur lokale Konvergenz der Optimierung entlang eines Gradienten verhindert ansonsten das Wechseln von einer Konfiguration zu einer anderen.

Zweite Stufe: Redundanz der Zusatzachse

Durch die Vorgabe diskreter Werte für die Zusatzachse E1 wird eine Nullraumanalyse durchgeführt. Der Vorteil gegenüber der Optimierung aller sieben Achsen des Roboters gleichzeitig liegt darin, dass für einen Schritt nicht optimale Lösungen nicht verworfen werden, die für die Gesamttrajektorie möglicherweise optimal wären.

Dritte Stufe: Redundanz aus Symmetrie

Die Symmetrie des Laserstrahls wird je nach verwendetem Optimierungsverfahren unterschiedlich implementiert. Beim Verfahren nach Kapitel 3.3.4 wird die Orientierung in einem diskreten Wertebereich vorgegeben. Damit liegen auch alle Ergebnisse zur nachträglichen Optimierung vor. Wird nach Kapitel 3.3.2 vorgegangen, ist dies nicht der Fall.

Ergebnis: Nullraumanalyse

Am Ende der Berechnung liegt für jeden Bahnpunkt eine Vielzahl von Möglichkeiten zur Lösung der inversen Kinematik vor. Die Werte der Zielfunktion z_K aus Gl. (3.13) der Lösungen für einen Punkt der Beispieltrajektorie werden in einem Konturdiagramm in Bild 4.5 dargestellt.

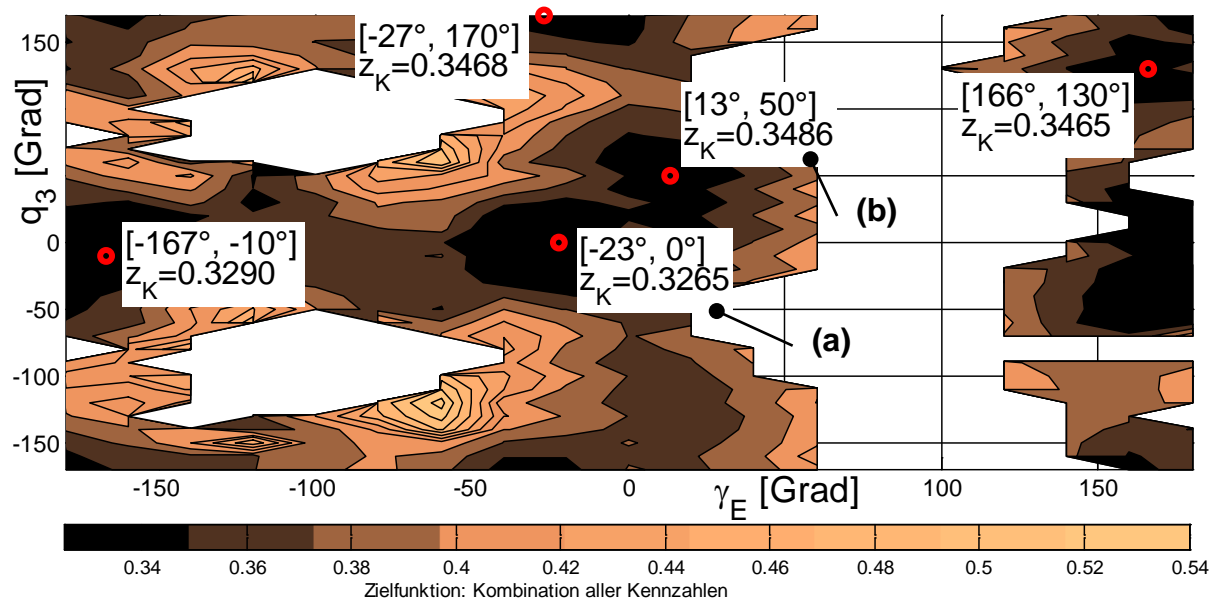
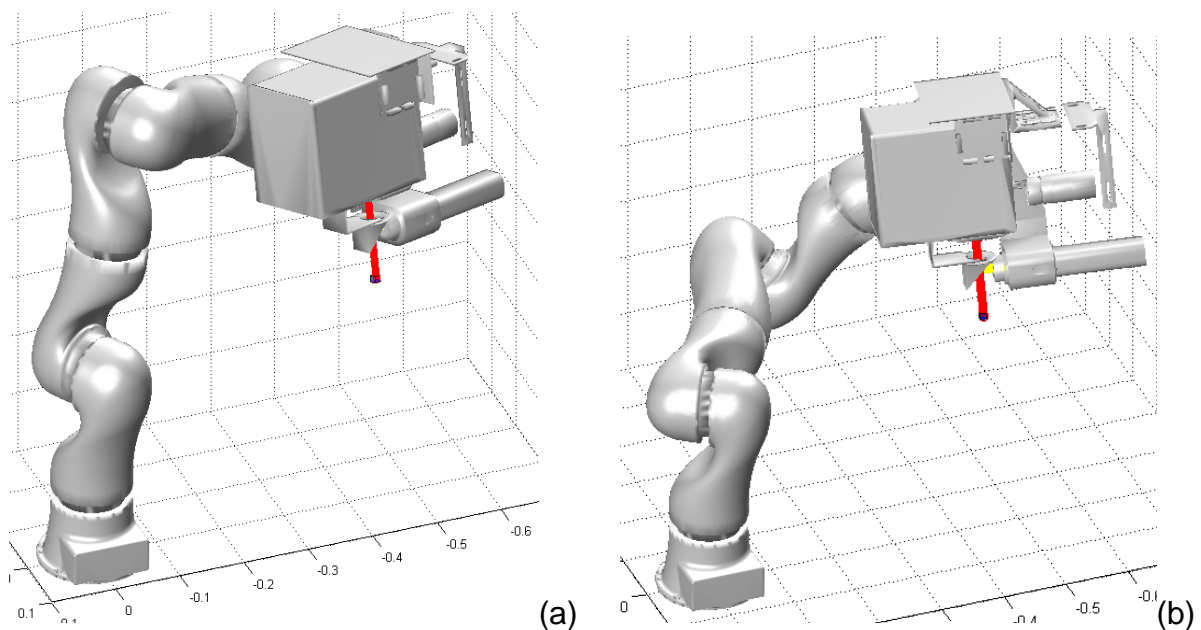


Bild 4.5: γ_E - q_3 -Lösungsraum der inversen Kinematik für den Punkt $x_E = -450$ mm, $y_E = -19$ mm, $z_E = 358$ mm, $\alpha_E = -166^\circ$, $\beta_E = 0^\circ$

Die gezeigte Fläche enthält eine Auswahl aus allen durchführbaren Kombinationen aus Laserstrahlrotation γ_E und Gelenkwinkel der Zusatzachse (q_3). Die Farbe des Diagramms zeigt den Wert der Zielfunktion z_K für die gegebene Kombination $[\gamma_E, q_3]$ an. Nicht eingefärbte Bereiche entsprechen nicht existenten Lösungen. Im Bild sind lokale Minima von z_K markiert, die ähnlich gute Erfüllung der Zielfunktion bieten.

Die Gelenkwinkelstellung zu den beiden in Bild 4.5 beispielhaft markierten Fällen (a) und (b) sind in Bild 4.6 dargestellt. Daraus ist zu erkennen, dass durch die mehrfache Redundanz komplett unterschiedliche Roboterstellungen zur gewünschten Lage des Laserstrahls führen.



$$\underline{q}_a = [31^\circ, 12^\circ, 0^\circ, -103^\circ, 81^\circ, -102^\circ, 17^\circ]^T \quad \underline{q}_b = [61^\circ, 93^\circ, 50^\circ, 85^\circ, 1^\circ, -76^\circ, -35^\circ]^T$$

Bild 4.6: Posen des Roboters für zwei Fälle der Nullraumanalyse des Punktes $x_E = -450 \text{ mm}$, $y_E = -19 \text{ mm}$, $z_E = 358 \text{ mm}$, $\alpha_E = -166^\circ$, $\beta_E = 0^\circ$

4.2.3 Optimierung der Laserbewegung

Zu jedem Mittelpunkt des Laserarbeitsraums liegt nun mindestens eine mögliche Stellung des Roboters vor. Die Auslenkung \underline{q}_l des Lasers wird für jede Roboterkonfiguration \underline{q}_r mit

$$\underline{q}_l = {}^0\mathbf{R}_{\text{LMP}}^T \cdot ({}_{(0)}\underline{r}_E - {}_{(0)}\underline{r}_{\text{LMP}}) \quad (4.2)$$

berechnet. Die Transformationsmatrix bis zum Mittelpunkt des Laserarbeitsraums ${}^0\mathbf{T}_{\text{LMP}}$ wird aus den zu diesem Arbeitsschritt bekannten Robotergelenkwinkeln \underline{q}_r mit der direkten Kinematik nach

$${}^0\mathbf{T}_{LMP}(\underline{q}_r) = {}^0\mathbf{T}_1(q_1) \cdot {}^1\mathbf{T}_2(q_2) \cdot \dots \cdot {}^6\mathbf{T}_7(q_7) \cdot \mathbf{T}_{LWZ}(\underline{0}) = \begin{bmatrix} {}^0\mathbf{R}_{LMP} & ({}^0)\underline{r}_{LMP} \\ \underline{0} & 1 \end{bmatrix}$$

bestimmt. Da im Vorfeld bereits die Anordnung der Laserarbeitsräume optimiert wurde, ist bekannt, dass jede Endeffektorposition $({}^0)\underline{r}_E$ innerhalb eines Arbeitsraums mit dem Mittelpunkt $({}^0)\underline{r}_{LMP}$ liegt.

Bei der numerischen Berechnung der inversen Kinematik des Roboters bleibt ein Positionsfehler, da die Lösung nur angenähert wird. Durch die anschließende exakte geometrische Berechnung der Laserstellung werden diese Fehler ausgeglichen.

4.2.4 Dynamische Optimierung der Gesamt-Trajektorie

Werden die Bewegungen von Roboter und Laser getrennt optimiert, gibt es unterschiedliche Ansätze um die Gesamt-Trajektorie zusammzusetzen. Die Möglichkeit, für jeden Bahnpunkt eine Roboterstellung zu finden, die die Zielfunktion optimiert, wurde bereits in [Ölt10/17] gezeigt. Der Nachteil dieser gradientenbasierten Methode ist, dass nur jeder Bahnpunkt für sich und nicht die gesamte Bahn betrachtet wird. Mit Bild 4.5 und Bild 4.6 wird deutlich, dass komplett unterschiedliche Konfigurationen des redundanten Roboters eine ähnlich gute Erfüllung der Zielfunktion liefern. Ein Aneinanderreihen dieser unterschiedlichen Konfigurationen zur gesamten Trajektorie hat große Sprünge im Gelenkwinkelverlauf und damit hohe Gelenkwinkelgeschwindigkeiten zur Folge. Dieses Problem tritt besonders bei großen Trajektorien auf, da diese nicht durchgängig mit einer Roboterkonfiguration, also der Hand innen oder außen bzw. dem Ellenbogen unten oder oben, durchgeführt werden können. Zur Vermeidung dieses Problems wird die Zielfunktion z_K um die Kennzahl der Gelenkwinkelstetigkeit $K_{\Delta q}$ zu

$$z'_K(\underline{q}_j, \underline{q}_{j-1}) = z_K(\underline{q}_j) + K_{\Delta q}(\underline{q}_j, \underline{q}_{j-1}) \quad (4.3)$$

erweitert. Die neue Zielfunktion z'_K ist damit nicht nur von den Gelenkwinkel des aktuellen Bahnpunktes (Index j , \underline{q}_j), sondern auch von den Gelenkwinkeln des vorherigen Bahnpunktes (\underline{q}_{j-1}) abhängig.

Das Ergebnis des bisher verwendeten gierigen Algorithmus ist die optimale Summe Z_G der Zielfunktion z_K über alle N_B Bahnpunkte:

$$Z_G = \sum_{j=1}^{N_B} z_K(\underline{q}_j). \quad (4.4)$$

Das Ergebnis des gierigen Algorithmus für die Summe Z_{GS} der neuen Zielfunktion z'_K nach

$$Z_{GS} = \sum_{j=1}^{N_B} z'_K(\underline{q}_j, \underline{q}_{j-1}). \quad (4.5)$$

ist nicht optimal, da der Algorithmus sich zu Beginn der Bahn für bestimmte Gelenkwinkel entscheidet, die nachfolgende Möglichkeiten ausschließen. Der umgekehrte Ansatz wird von der dynamischen Optimierung (DO) verfolgt: Ausgehend von der letzten Stufe des Problems, also dem letzten Bahnpunkt, wird durch rekursives Vorgehen bis zum Anfang der Bahn die Folge von Gelenkwinkeln ausgewählt, die die Summe der gewählten Kosten- oder Zielfunktion bis zum Ende optimiert. Dadurch ist die dynamische Optimierung sowohl bei der Optimierung nach Z_G als auch nach Z_{GS} optimal, was in folgender Tabelle 4.3 zusammengefasst wird.

Tabelle 4.3: Optimalität der Ergebnisse von Greedy-Algorithmus und dynamischer Optimierung für die Gesamt-Trajektorie

| Optimierung nach | Greedy-Algorithmus | Dynamische Optimierung |
|------------------|--------------------|------------------------|
| Z_G | Optimal | Optimal |
| Z_{GS} | Nicht optimal | Optimal |

Die Zusammenhänge der in Kapitel 2.4.5 beschriebenen allgemeinen Form der dynamischen Optimierung mit dem Bahnplanungsproblem werden in der folgenden Tabelle 4.4 zusammengefasst. Die Ergebnisse der Trajektorienoptimierung für unterschiedliche Verfahren werden in Kapitel 6 verglichen.

Tabelle 4.4: Belegung der Begriffe der dynamischen Optimierung in der Anwendung auf das Bahnplanungsproblem

| DO-Begriff | Bedeutung |
|-----------------------|---|
| Stufe | Bahnpunkt der Gesamt-Trajektorie |
| Zustandsmenge | Unterschiedliche Gelenkwinkelstellungen als Ergebnis der Nullraumanalyse (entspricht Bild 4.5). |
| Kostenfunktion | Höhe der Zielfunktion z_K (reihenfolgeunabhängig) und der Stetigkeit der Gelenkwinkel $K_{\Delta q}$ aus Gl. (2.16) (Abhängig von den gewählten Zuständen der Stufen) |
| Entscheidungsvariable | Gewählte Gelenkwinkel des Roboters für eine Stufe |

5 Implementierung in MATLAB

Die Algorithmen zur Simulation und Optimierung von Trajektorien für das Roboter-Scanner-System werden in der Programmierumgebung MATLAB umgesetzt. Zur Visualisierung wird eine bestehende graphische Benutzerschnittstelle (GUI) um die notwendigen Schalt- und Ausgabeflächen erweitert, siehe Bild 5.1. In Kapitel 5.1 wird der typische Arbeitsprozess zur Optimierung einer Trajektorie in der GUI dargestellt. In Kapitel 5.2 werden Besonderheiten bei der Programmierung benannt und die Eigenschaften der wichtigsten erstellten Module gezeigt.

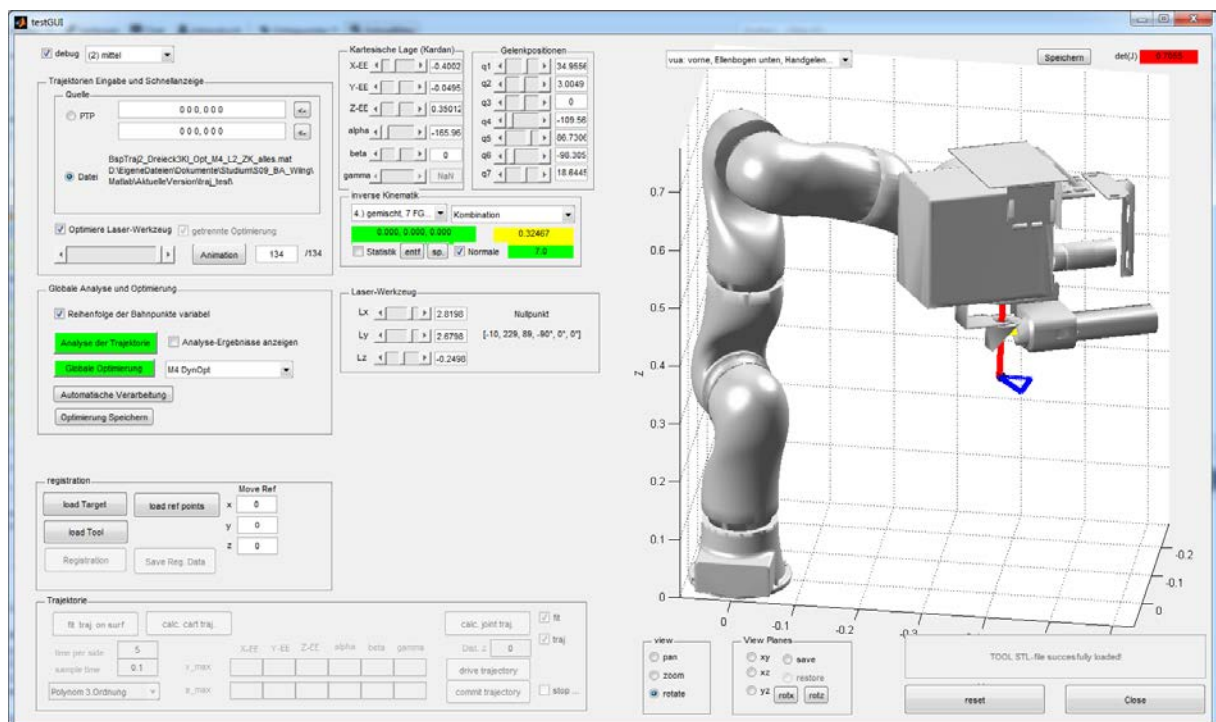


Bild 5.1: Gesamtansicht der MATLAB-GUI

5.1 Arbeitsprozess

Parameter zur Einstellung der Optimierung werden in den linken Teil der GUI eingegeben, der in Bild 5.2 dargestellt ist. Im Bild sind Eingabebereiche mit Buchstaben gekennzeichnet. Bildbereich (a) dient zum Laden einer Trajektorie und zur Steuerung der Anzeige einzelner Bahnpunkte. In Teil (b) wird die Analyse und Optimierung der Trajektorie durchgeführt. Das Modell und die Koordinaten des Werkzeugs werden durch die Schaltfläche in Teil (c) geladen. Teil (d) dient zur Anzeige der Gelenkwinkel und der Kartesischen Lage des Roboters. Eingaben führen zur Berechnung der di-

rekten bzw. inversen Kinematik. In Teil (e) wird die Methode zur Lösung der inversen Kinematik eingestellt sowie der entstehende Fehler angezeigt. Die Anzeige und Einstellung des Laser-Werkzeugs erfolgt in Bereich (f).

Die Reihenfolge des Arbeitsprozesses von der Erstellung bis zur Optimierung einer Trajektorie ist mit Zahlen dargestellt: Zuerst wird mit den Schaltflächen „1“ und „2“ das Werkzeug sowie die zu optimierende Trajektorie geladen. Die Trajektorie wird vorher erstellt und in einem im Anhang A.4.1 definierten Übergabeformat gespeichert. Mit den Menüs bei „3“ wird die Methode und Zielfunktion der inversen Kinematik eingestellt. Die Methoden aus Kapitel 3 stehen hier zur Auswahl. Mit der Schaltfläche „4“ wird die Trajektorie analysiert, indem die Methode aus den Kapiteln 4.2.1 bis 4.2.3 durchgeführt werden. Mit Schaltfläche „5“ wird die Optimierung der Gesamt-Trajektorie nach den Methoden aus Kapitel 4.2.4 durchgeführt. Diese beiden Funktionen werden durch die Schaltfläche „automatische Verarbeitung“ darunter zusammengefasst. Die Analyse und die Optimierung der Trajektorie werden iterativ mehrfach durchgeführt, wobei die vorgegebenen Werte des Endeffektorwinkels γ_E und der Zusatzachse q_3 der optimalen Lösung immer feiner angenähert werden.

Die Ergebnisse dieser Optimierung werden durch Aktivierung des Feldes „6“ durch Diagramme dargestellt. Mit diesen Diagrammen lässt sich ein guter Überblick über die Stetigkeit der Gelenkwinkel und die Zielfunktionswerte bei unterschiedlich optimierten Gelenkwinkeltrajektorien gewinnen. Mit dem Menü „7“ kann eine optimierte Gelenkwinkeltrajektorie gewählt werden. Der gewählte Gelenkwinkelverlauf wird mit dem Schieber und Animationsknopf bei „8“ im rechten Teil der GUI mit einem CAD-Modell animiert dargestellt. Das Ergebnis wird mit der Schaltfläche „9“ als .mat-Datei im Format nach Anhang A.4.2 und A.4.3 gespeichert, um später ohne neue Berechnungen direkt darauf zugreifen zu können.



Bild 5.2: Linker Teil der GUI; enthält die Eingabe und Analyse der Trajektorie sowie Eingabe und Anzeige der kartesischen Lage und Gelenkwinkel

Die graphische Ausgabe des Roboters mit Werkzeug und Trajektorie erfolgt im rechten Teil der GUI, der in Bild 5.3 gezeigt wird. Mit den Schaltflächen bei „1“ wird die Ansicht des Robotermodells geändert. Bei „2“ sind die bearbeitete Trajektorie (blau), der Bearbeitungslaser (rot) und der Kohärenztomographielaser (gelb) schematisch dargestellt. Bei „3“ wird die Konfiguration des Roboters angezeigt. Durch Auswahl ist hier auch eine Vorgabe der Konfiguration in der inversen Kinematik möglich. Bei „4“ wird permanent die Kennzahl der Manipulierbarkeit nach Gl. (2.9) dargestellt, da sie ein wichtiger Indikator für die numerische Lösbarkeit der inversen Kinematik ist. Mit der Schaltfläche „5“ wird das gezeigte Bild als Datei gespeichert.

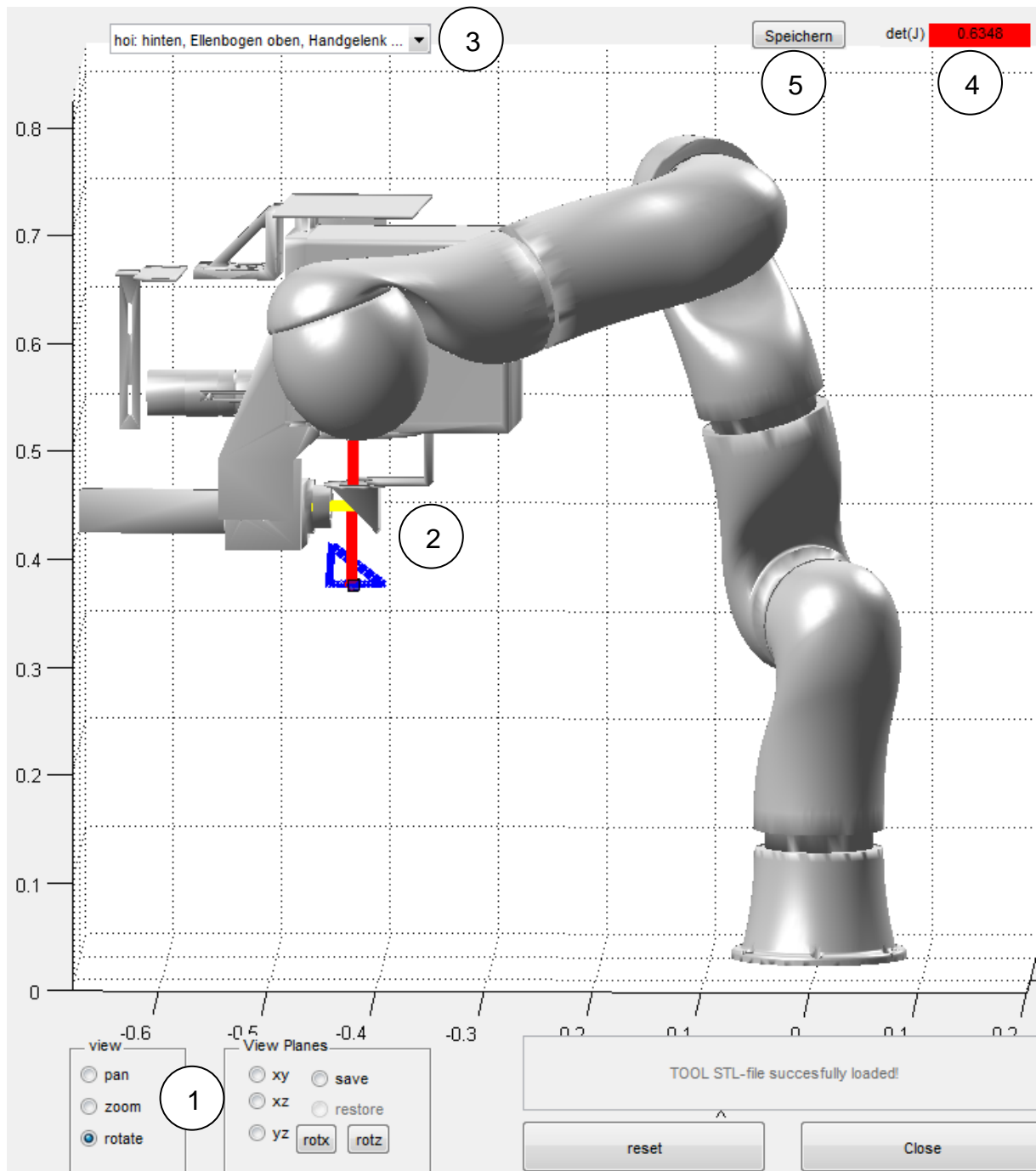


Bild 5.3: Rechter Teil der GUI; enthält die Anzeige des Roboters und der Trajektorie

5.2 Module

Das gesamte Programm ist modular aufgebaut und in zwei Schichten getrennt, wie in Bild 5.4 gezeigt. Zur ersten Schicht gehören Funktionen, die Eingaben in die GUI aufnehmen und verarbeiten bzw. Ausgaben in die GUI schreiben. Diese Funktionen sind in der Hauptdatei `testGUI.m` sowie in weiteren Funktionsdateien enthalten. Die Eingabedaten werden über Zeiger (handles) auf die Elemente der GUI ausgelesen. Die erste Funktionsschicht ist dadurch nicht allgemein verwendbar, sondern nur zusammen mit der GUI ausführbar.

Die in dieser Arbeit vorgestellten Algorithmen sind in Funktionen der zweiten Schicht implementiert und sind unabhängig lauffähig. Soweit möglich wurden bereits vorhandene und getestete Funktionsbausteine aus anderen Quellen benutzt. Hier sind besonders die MATLAB-Implementierungen des genetischen Algorithmus (*ga*) und des Trust-Region-Reflective-Algorithmus (*lsqnonlin*) aus der MATLAB-Optimization Toolbox und Funktionen zur Koordinatentransformation und Roboterkinematik aus der Robotics Toolbox [CorRTb] sowie die am imes erstellte Grundstruktur der GUI zu nennen.

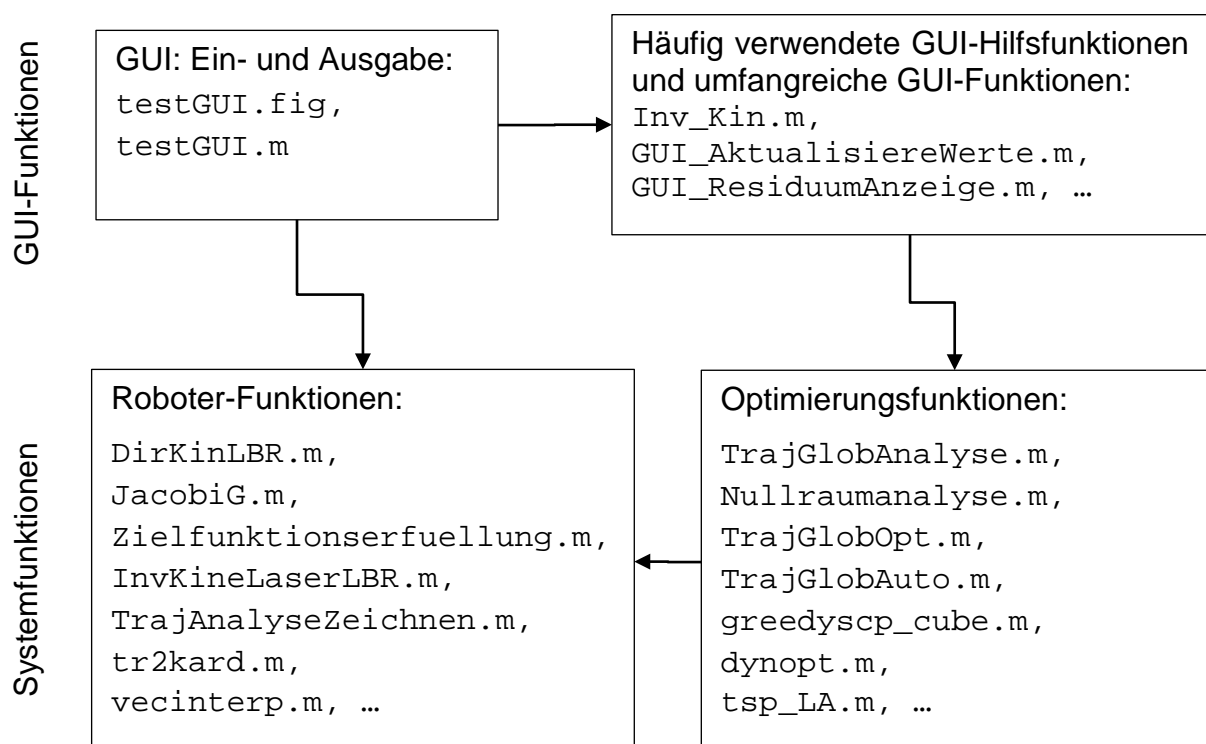


Bild 5.4: Hierarchischeschema der verwendeten MATLAB-Funktionen

6 Evaluation

In diesem abschließendem Kapitel werden die zuvor vorgestellten Algorithmen verglichen und bewertet. Zunächst erfolgt in Kapitel 6.1 die Darstellung der Wahlmöglichkeiten bei der Zielfunktion. Die Gegenüberstellung der Vor- und Nachteile der in Kapitel 3 vorgestellten Methoden zur Lösung der inversen Kinematik findet in Kapitel 6.2 statt. Anschließend werden in Kapitel 6.3 die Alternativen der Bahnoptimierung aus Kapitel 4.2 näher untersucht. Der Vergleich der getrennten mit der gemeinsamen Optimierung aller Freiheitsgrade wird in Kapitel 6.4 durchgeführt. Es werden die beiden in Bild 6.1 dargestellten Beispieltrajektorien untersucht: Die erste (Dreieck mit Flächeninhalt 26 cm^2) ist komplett mit unterschiedlichen Konfigurationen des Roboters durchführbar. Die zweite Trajektorie (Rechteck, Flächeninhalt $0,13 \text{ m}^2$) ist aufgrund ihrer großen Abmessungen nicht ohne Konfigurationswechsel durchführbar.

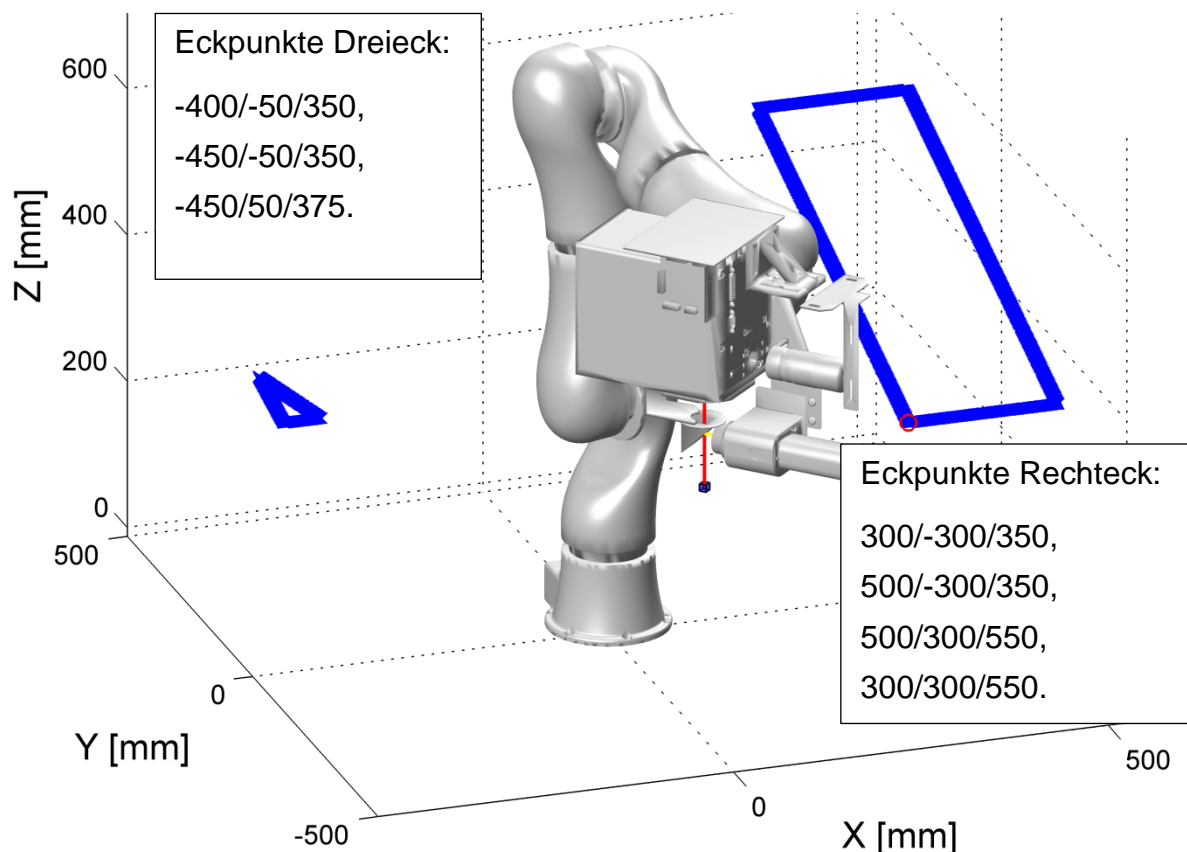


Bild 6.1: Beispieltrajektorien im Arbeitsraum des Roboters

6.1 Wahl der Zielfunktion

Vor der Berechnung der inversen Kinematik wird die Zielfunktion ausgewählt, die minimiert wird. Es stehen die Kennzahlen Genauigkeit (K_{Δ}), Nachgiebigkeit (K_{hzz}), Manipulierbarkeit (K_{μ}), Geschicklichkeit (K_{η}), Ellenbogenhöhe (K_{EB}) und Gelenkwinkelwertebereich (K_q) aus Kapitel 2.3 zur Auswahl. Die Gewichtung dieser Kennzahlen hängt von der genauen Aufgabenstellung ab. Bild 6.2 zeigt Korrelationsdiagramme für die Abhängigkeit aller Kennzahlen untereinander. Darauf ist zu erkennen, dass keine Kennzahl mit einer anderen korreliert und damit überflüssig wäre. Die Kennzahlen im Bild sind im Wertebereich 0 bis 1 dargestellt, die Achsenbeschriftungen sind für alle Spalten und Zeilen am Bildrand angegeben.

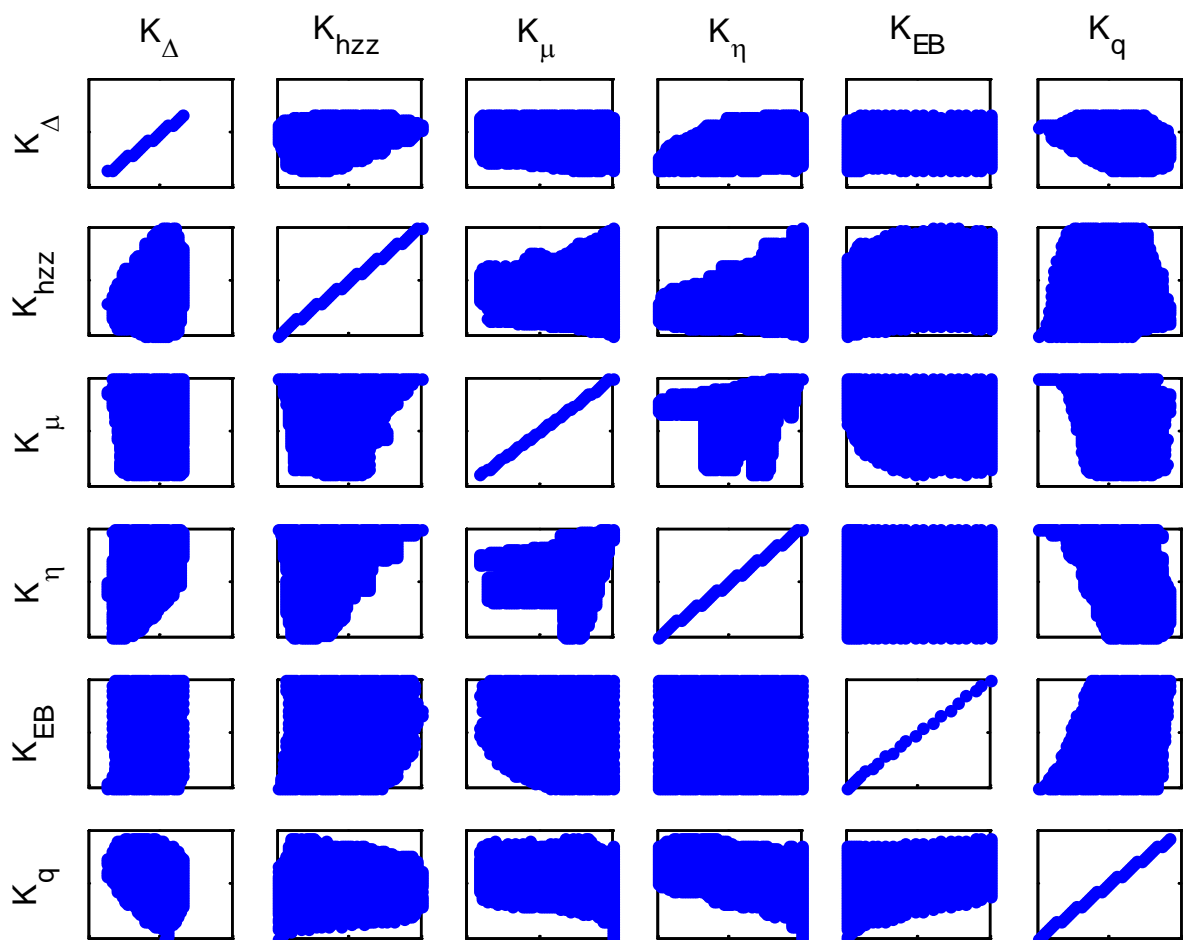


Bild 6.2: Streudiagramme zur Darstellung der Korrelation der Kennzahlen. Datengrundlage: ca. 5 Mio. Gelenkwinkelkombinationen. Wertebereich: 0 bis 1.

Des Weiteren besteht ein Zielkonflikt zwischen den Kennzahlen: Die Verbesserung eines Leistungsmerkmals ist im Allgemeinen nur erreichbar, wenn andere Leistungsmerkmale verschlechtert werden (Pareto-Optimum). Tabelle 6.1 zeigt die Ausprägung aller Kennzahlen, wenn die Zielfunktion sich nur aus dem Wert der linken Spalte zusammensetzt. Die dargestellten Zahlenwerte sind die Mittelwerte über die gesamte Dreiecks-Beispieltrajektorie. Der beste und schlechteste erreichte Wert auf der Trajektorie für jede Kennzahl ist in der entsprechenden Spalte grün und rot markiert. Die Kennzahlen erreichen ihren vollen Wertebereich von 0 bis 1 nur über den gesamten Arbeitsraum.

Tabelle 6.1: Werte der Leistungsmerkmale bei unterschiedlicher Gewichtung in der Zielfunktion für die Dreieck-Beispieltrajektorie

| | | Werte für ... | | | | | | |
|----------------------|--------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | | K_{Δ} | K_{hzz} | K_{μ} | K_{η} | K_{EB} | K_q | Gl. (6.1) |
| Optimierung nach ... | K_{Δ} | 0,2370 | 0,1953 | 0,3576 | 0,3980 | 0,7726 | 0,6287 | 0,3699 |
| | K_{hzz} | 0,2905 | 0,1446 | 0,6534 | 0,0603 | 0,8570 | 0,6267 | 0,3761 |
| | K_{μ} | 0,3259 | 0,2640 | 0,1130 | 0,4284 | 0,8033 | 0,7173 | 0,3789 |
| | K_{η} | 0,2987 | 0,1815 | 0,6689 | 0,0512 | 0,7729 | 0,6871 | 0,3801 |
| | K_{EB} | 0,4133 | 0,4266 | 0,7383 | 0,2726 | 0,0000 | 0,6000 | 0,3501 |
| | K_q | 0,4205 | 0,5189 | 0,9142 | 0,6848 | 0,0121 | 0,4345 | 0,4264 |
| | Gl. (6.1) | 0,3835 | 0,4355 | 0,5830 | 0,3990 | 0,0086 | 0,4582 | 0,3240 |

Bei den weiteren Berechnungen gehen alle Kennzahlen zu gleichen Teilen in die Zielfunktion z_K mit

$$z_K = \frac{1}{6} \cdot (K_{\Delta} + K_{hzz} + K_{\mu} + K_{\eta} + K_{EB} + K_q) \quad (6.1)$$

ein. Diese Zielfunktion liegt für die Beispieltrajektorien im Bereich von 0,3 bis 0,4. Bessere Werte sind nicht erreichbar, da die Verbesserung einer Kennzahl im berechneten Optimum nur unter Verschlechterung der gewichteten Summe möglich ist. Der beste Wert ist nicht mit allen Kennzahlen gleichzeitig erreichbar, sondern nur einzeln. Der bestmögliche Wert Null kann nicht an allen Punkten des Roboter-Arbeitsraums erreicht werden.

Die um die Stetigkeit erweiterte Zielfunktion z'_K wird durch Versuche zu

$$z'_K = z_K + 100 \cdot (K_{\Delta q})^3 \quad (6.2)$$

ermittelt. Die Potenzierung der Kennzahl der Gelenkwinkeländerung ($K_{\Delta q}$) führt dazu, dass große Änderungen der Gelenkwinkel zu einer stärkeren Verschlechterung von z'_K führen als mehrere verteilte kleine Änderungen. Da die Werte für $K_{\Delta q}$ kleiner als eins liegen, führt die Potenzierung zu einer Verkleinerung der Zahlenwerte. Dies wird durch den Vorfaktor ausgeglichen, sodass z_K und der Stetigkeitsterm in der gleichen Größenordnung liegen. Die Stetigkeit wird damit stärker gewichtet als einzelne Leistungsmerkmale. Mit geringerer Gewichtung der Stetigkeit liegen in der resultierenden Gelenkwinkeltrajektorie zu große Sprünge vor.

6.2 Inverse Kinematik

An dieser Stelle werden die Methoden zur Redundanzauflösung anhand der Ergebnisse bei der Berechnung von Beispieltrajektorien verglichen. Es wird zum einen der Ansatz aus Kapitel 3.3.1. bewertet, bei dem alle Achsen des Roboters während der Optimierung verändert werden. Die Berechnung der Roboter-Grundachsen nach Kapitel 3.3.3 wird sowohl mit dem genetischen Algorithmus (GA), als auch mit dem gradientenbasierten Trust-Region-Reflective-Algorithmus (TRRA) betrachtet.

Aufgrund der geringeren Anzahl an Unbekannten und der exakten Berechnung der Orientierung des Endeffektors ist der Grundachsen-Ansatz allgemein der Optimierung aller Achsen vorzuziehen. Letzteres hat lediglich dann einen Vorteil, wenn die Symmetrie des Werkzeuges berücksichtigt wird. In diesem Fall müssen für die Lösung nicht mehrere Endeffektororientierungen mit Variation von γ_E vorgegeben werden, was die online-Berechnung der inversen Kinematik ermöglicht. Die Unterschiede zwischen den Ansätzen sind Tabelle 6.2 zu entnehmen.

Die Lösungsalgorithmen GA und TRRA unterscheiden sich durch die unterschiedlichen Ansätze bezüglich globaler und lokaler Konvergenz sowie der Rechenzeit. Der TRRA konvergiert schnell gegen die optimale Lösung, außer wenn zwischen Startkonfiguration des Roboters und dem Ziel ein lokales Minimum oder eine Singularität liegt. Der GA konvergiert langsam, da keine genaue Suchrichtung im Gelenkraum vorliegt und deshalb sehr viele Kombinationen der Gelenkwinkel untersucht werden müssen. Der Vorteil des GA liegt darin, dass eine Singularität verlassen werden

kann. Die Unterschiede zwischen GA und TRRA sind ebenfalls in Tabelle 6.2 dargestellt.

Tabelle 6.2: Vergleich der Lösungsalternativen zur Auflösung der kinematischen Redundanz des Roboters

| Kinematischer Ansatz | Optimierung der Grundachsen | | Alle Achsen |
|--|---|--|-----------------------------------|
| Optimierungsverfahren | Genetisch (GA) | Gradient (TRRA) | Gradient (TRRA) |
| Anzahl Variablen | 4 | 4 | 7 |
| Globale Konvergenz | Ja, wenn Lösung existiert | Nein, lokales Optimum kann nicht verlassen werden. | |
| Lokale Konvergenz: Translationsfehler [m] | Ca. 10^{-5} nach 50 Generationen | $< 10^{-7}$ nach 4 bis 9 Iterationen | |
| Lokale Konvergenz: Rotationsfehler [Grad] | Aufgrund der analytischen Berechnung der Handachsen immer Null. | | $< 10^{-10}$ |
| Behandlung der Werkzeugsymmetrie (WZS) | Mehrfache Ausführung, alle Lösungen liegen für Nullraumanalyse vor. | | Eine Ausführung, nur eine Lösung. |
| Rechenzeit⁴ | Ca. 10.000 | Ca. 4 bis 8 | Ca. 5 bis 12 |

Um einen guten Kompromiss aus Rechenzeit und Konvergenz der Verfahren zu erreichen, wird zuerst die Optimierung der Grundachsen mit dem TRRA durchgeführt. Ist damit keine zulässige Lösung erreichbar, wird mit dem geometrisch-analytischen Ansatz aus Kapitel 3.2 ohne Berücksichtigung der Redundanz gerechnet. Dieser Ansatz ist nicht von Anfangswerten abhängig. Ist hiermit ebenfalls keine Lösung verfügbar, ist anzunehmen, dass die Endeffektorlage nicht im Arbeitsraum des Roboters enthalten ist. Diese Reihenfolge der Verfahren wird in den folgenden Teilkapiteln zur Evaluation der Bahnoptimierung eingesetzt.

Der GA ist für einen Einsatz in der GUI ungeeignet, da die Zeit zwischen Eingabe und Anzeige des Ergebnisses zu lang ist. Während die gemessene Dauer der Berechnung beim TRRA unter einer Sekunde liegt, beträgt sie beim GA mehrere Minuten. Die Rechenzeit beim GA vervielfacht sich mit der Vorgabe von Endeffektorwin-

⁴ Die Rechenzeit wird im Verhältnis zur kompilierten geometrisch-analytischen Berechnung angegeben. Diese beträgt ca. 0,04 s ohne WZS und ca. 0,12 s mit WZS. Die Zeit bezieht sich auf die Berechnung einer Punkt-zu-Punkt-Bewegung. Es werden unterschiedliche Punkte und Anfangslagen getestet.

keln bei Werkzeugsymmetrie zusätzlich. Der Einsatz des GA lohnt sich daher nur, wenn eine weit von der Anfangslage entfernte kartesische Lage erreicht werden soll, bei der bekannt ist, dass die inverse Kinematik existiert.

Durch die analytische Berechnung der Laserauslenkung nach der Berechnung der Roboterelenkwinkel werden Abweichungen von der Soll-Position bei allen Optimierungsverfahren ausgeglichen. Dadurch ist eine lokale Konvergenz bereits bei einem Translationsfehler von bis zu 1 mm möglich und die notwendige Anzahl der Rechenschritte zur Konvergenz sinkt von ca. 10 auf 2 bis 3.

6.3 Bahnoptimierung

In diesem Abschnitt werden unterschiedliche Ansätze zur globalen Optimierung der gesamten Bahn verglichen. Zunächst erfolgt die Betrachtung der dynamischen Optimierung aus Kapitel 4.2.4, anschließend wird das bisherige Standardverfahren, der Greedy-Algorithmus, mit und ohne Berücksichtigung der Gelenkwinkelstetigkeit betrachtet.

In Bild 6.3 ist der Verlauf der Systemfreiheitsgrade für den Anfang der Dreieck-Beispieltrajektorie nach der Durchführung der dynamischen Optimierung für die stetige Zielfunktion nach Gl. (6.2) angegeben. Auf der waagerechten Achse erhöht sich die laufende Nummer der Bahnpunkte; dies entspricht einer Zeitachse. Auf den senkrechten Achsen sind die Auslenkungen der Freiheitsgrade des Roboters (A_1, \dots, A_6) und des Lasers (L_1, L_2, L_3) sowie die Werte der Zielfunktion für jeden Bahnpunkt und ihr Mittelwert aufgetragen.

Aufgrund der im Vergleich zum gesamten Arbeitsraum des Roboters kleinen Trajektorie (siehe Bild 6.1) und der hohen Gewichtung der Gelenkwinkelstetigkeit findet nur eine sehr geringe Bewegung des Roboters statt (Bild 6.3a und b). Da die Laserbewegung getrennt von der Roboterbewegung optimiert wurde, wird der Laser über seinen gesamten Wertebereich verwendet (Bild 6.3c). Das gezeigte Verhalten von Roboter und Laser erfüllt die Forderung nach geringer Roboterbewegung und guten Zielfunktionswerten (Bild 6.3d) optimal.

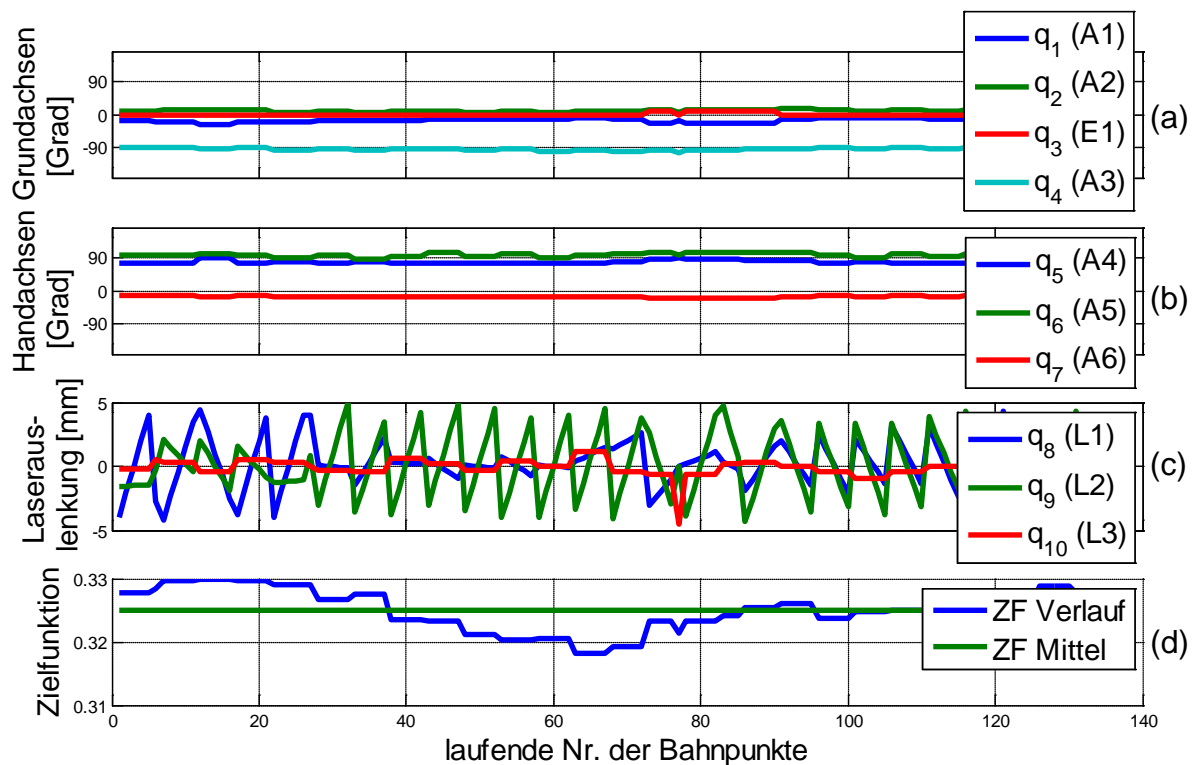


Bild 6.3: Gelenkwinkel-, Laserauslenkungs- und Zielfunktionsverlauf der optimierten Dreieck-Beispieltrajektorie mit dynamischer Optimierung

Zum Vergleich dazu wird in Bild A.3 im Anhang das Verhalten des Roboters bei der Rechteck-Trajektorie gezeigt, wenn die Bahn mit dem Greedy-Algorithmus und der Zielfunktion ohne Stetigkeit nach Gl. (6.1) optimiert wird. Durch die Sprünge in den Robotergelenkwinkeln ist diese Trajektorie nicht durchführbar. Wird beim Greedy-Algorithmus die Stetigkeit der Gelenkwinkel mit Gl. (6.2) berücksichtigt, ergibt sich die durchführbare Gelenkwinkeltrajektorie nach Bild A.4.

Die Zielfunktionswerte der unterschiedlichen Verfahren sind in Tabelle 6.3 zusammengefasst. Die drei betrachteten Verfahren unterscheiden sich kaum in der Größe der Zielfunktionswerte. Bei den beiden Beispieltrajektorien ist sowohl mit dynamischer Optimierung (DO) als auch mit dem Greedy-Algorithmus eine ähnlich gute, stetige Bahn das Ergebnis. Die dynamische Optimierung ist dem Greedy-Algorithmus vorzuziehen, da die gewichtete Summe aus Stetigkeitskennzahl und Zielfunktionswert minimal wird. Der Rechenaufwand⁵ der DO ist höher als der des Greedy-

⁵ Der Rechenaufwand wird durch die Zeitdauer der Berechnung ermittelt. Für die Analyse der Beispieltrajektorie wurde mit einer 6-Kern CPU bei 2,8 GHz eine Dauer von ca. 3,3 Stunden gemessen. Für die Dauer der Optimierung wurden ca. 0,15 Stunden gemessen.

Algorithmus, beträgt aber nur ca. 5% des Aufwands für die Berechnung der inversen Kinematik in der Analyse der Trajektorie.

Tabelle 6.3: Vergleich der gemittelten Zielfunktionen bei unterschiedlicher Gesamtop-
timierung

| mittlerer Zielfunktionswert pro Bahnpunkt | Dreieck | Rechteck |
|--|------------------------|------------------------|
| Greedy, nicht stetig | 0,324 | 0,325 (Bild A.3 unten) |
| Greedy, stetig | 0,325 | 0,340 (Bild A.4 unten) |
| DO, stetig | 0,325 (Bild 6.3 unten) | 0,333 |

6.4 Getrennte und gemeinsamer Optimierung der Freiheitsgrade

Die Ergebnisse des vorherigen Abschnittes 6.3 basieren auf einer getrennten Optimierung von Roboter- und Laserfreiheitsgraden. Aufgrund der nur geringen Gesamtauslenkung des Lasers hat die Wahl zwischen getrennter und gemeinsamer Optimierung der FHG jedoch kaum Einfluss auf die Roboterstellung. Der Verlauf der Laser-
auslenkung bei der gemeinsamen Optimierung der FHG für die Dreieck-
Beispieltrajektorie ist in Bild 6.4 dargestellt. Daraus ist zu entnehmen, dass die Laserfreiheitsgrade kaum genutzt werden und der Laser am Rand seines Arbeitsraumes steht. Für weitere Anwendungen ist also die getrennte Optimierung der FHG vorteilhaft.

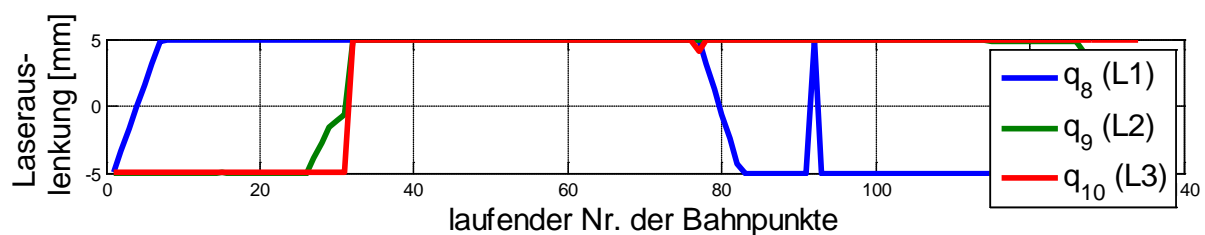


Bild 6.4: Laserauslenkungsverlauf der optimierten Dreieck-Beispieltrajektorie mit dynamischer Optimierung bei gemeinsamer Optimierung von Roboter und Laser

7 Zusammenfassung und Ausblick

In dieser Arbeit wurde eine graphische Benutzerschnittstelle zur Simulation von Trajektorien für die roboterassistierte Laserosteotomie weiterentwickelt. Dabei wurden neue Algorithmen implementiert, die die Eigenschaften des Systems aus redundantem Roboter und Laserkinematik berücksichtigen.

Zunächst wurden Leistungsmerkmale des Roboters wie Nachgiebigkeit, Manipulierbarkeit und Geschicklichkeit zu Kennzahlen für die Bewertung von Trajektorien zusammengefasst. Die Verbesserung einzelner Merkmale verschlechtert dabei andere, sodass zunächst eine Kombination der Kennzahlen zur Optimierung benutzt wurde.

In der inversen Kinematik des Roboters wurde zum einen berücksichtigt, dass nur die vier Grundachsen des Roboters redundant sind und die drei Handachsen durch die Vorgabe einer Endeffektorlage festgelegt sind. Die Grundachsen wurden numerisch mit dem Trust-Region-Reflective-Algorithmus und die Handachsen analytisch berechnet. Zum anderen wurde die fünfachsige Bearbeitungssituation durch den Laser einbezogen, was die Ordnung der für Optimierungen nutzbaren kinematischen Redundanz des Systems erhöht.

Bei der Betrachtung des Gesamtsystems aus Roboter und Laserscanner ist die getrennte Behandlung der Teilsysteme im Anwendungsgebiet der Laserosteotomie besser geeignet. Der Roboter steht bei der Laserpulsbearbeitung still, während alle im Laserarbeitsraum liegenden Punkte nacheinander bearbeitet werden. Die Anordnung der Laserarbeitsräume über die gesamte Trajektorie wird schrittweise mit genetischen Algorithmen durchgeführt.

Zur Optimierung der Roboterbewegung über die gesamte Trajektorie werden die Ergebnisse der Nullraumanalyse für jeden Bahnpunkt vom Ende zum Anfang durchgegangen und so die optimale Trajektorie gefunden. Es werden sowohl die Stetigkeit des Gelenkwinkelverlaufs als auch die gewählte Zielfunktion beachtet. Dieses Verfahren liefert immer bessere Ergebnisse als der bisherige Ansatz, die Bahnpunkte vom Anfang zum Ende durchzugehen, der Vorteil bei den betrachteten Beispieltrajektorien war jedoch gering.

Der nächste durchzuführende Arbeitsschritt ist, die optimierten Trajektorien am Versuchsstand mit dem Roboter zu fahren. Hier ist insbesondere zu untersuchen, wie groß der Einfluss der Masse des Werkzeugs auf die Bahngenauigkeit ist. Aufgrund der im Vergleich zur Nenn-Traglast und zur strukturellen Steifigkeit des Roboters großen Masse und Abmessung des Werkzeugs ist eine Abdrängung durch das Eigengewicht zu erwarten. Diese ist mit externer Sensorik bzw. durch Bearbeitungsversuche zu untersuchen und in den vorgegebenen Robotergelenkwinkeln oder im Brennpunkt des Lasers zu korrigieren, falls keine Korrektur durch die Robotersteuerung in Verbindung mit den eingebauten Drehmomentsensoren erfolgt.

Vor der Verwendung der berechneten Auslenkungen des Laserwerkzeugs ist die Orientierung des in dieser Arbeit festgelegten Werkzeug-Koordinatensystems mit dem in der Steuerung des Laserscanners hinterlegtem KS zu vergleichen.

Die Beispieltrajektorien wurden bisher idealisiert als exakte Vorgabe im kartesischen Arbeitsraum angenommen. Vor der Bearbeitung muss eine Registrierung der Lage des Werkstücks relativ zum Roboter durchgeführt werden aus der diese Vorgabe gewonnen wird. Während der Bearbeitung kann die Überprüfung der relativen Lage durch die Kohärenztomographie und die Verwendung externer Kamerasysteme erfolgen. Die Erkennung wird dadurch erleichtert, dass der Roboter sich während der Bearbeitung nicht bewegt.

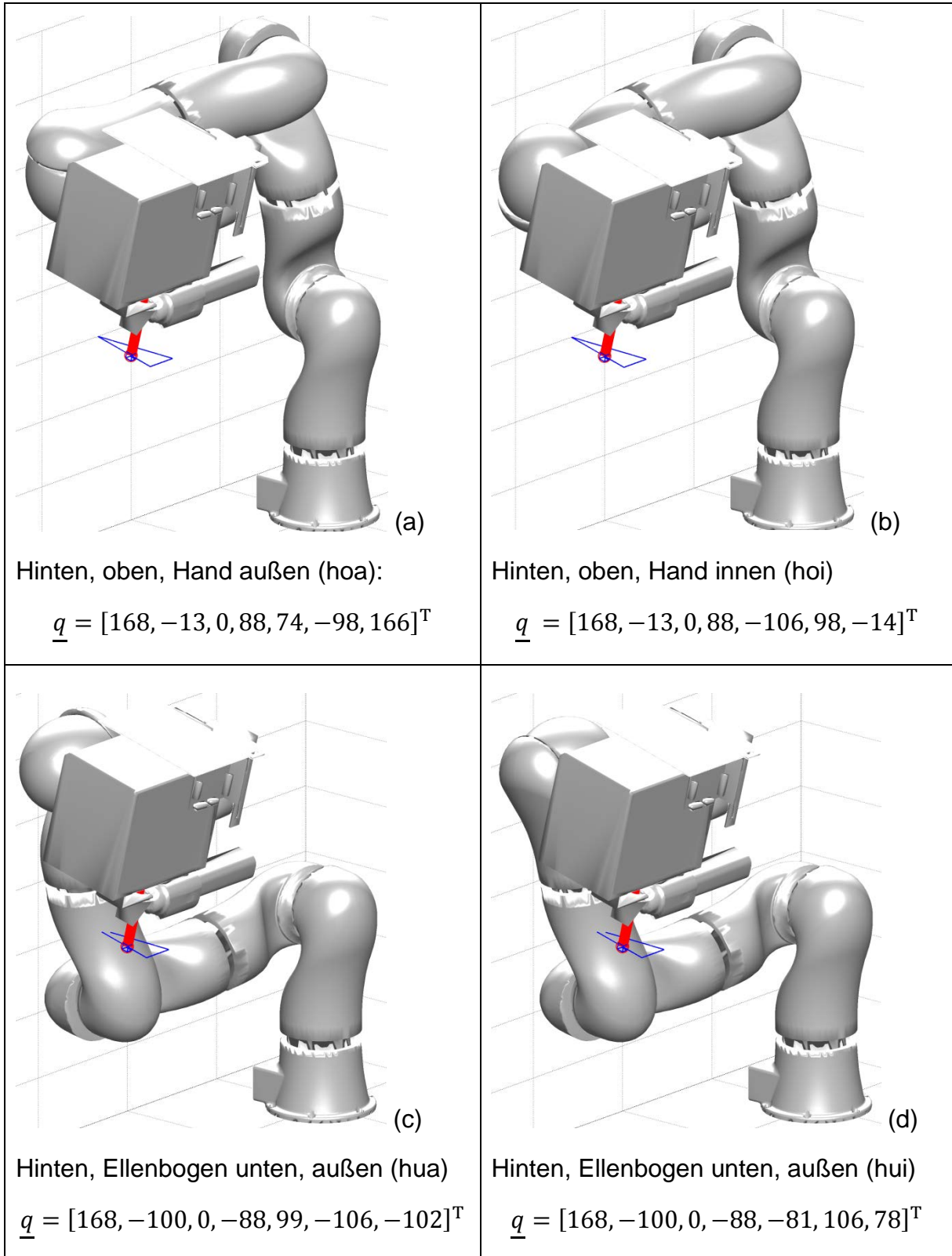
Eine Abweichung des Werkstücks von seiner in der Optimierung vorausgesetzten Lage während der Laserpulsbearbeitung erfordert die sofortige online-Neuberechnung der Robotergelenkwinkel. Zur medizintechnischen Anwendung ist ein Regelkreis zu entwerfen, der ausgehend von der Lageabweichung des Werkstücks die Robotergelenkwinkel und Laserauslenkung in Echtzeit neu berechnet. Kleine Abweichungen unter ca. 1 mm können dabei mit dem Laserscanner ausgeglichen werden. Für größere Abweichungen kann die inverse Kinematik nach den vorgestellten Methoden ausgehend von den bereits im Vorfeld optimierten Gelenkwinkeln neu berechnet werden.

8 Literaturverzeichnis

- [Bur10]** Burgner, J.: Robot Assisted Laser Osteotomy, Dissertation, Karlsruher Institut für Technologie, 2010
- [Cork11]** Corke, P.: Robotics, Vision and Control, Springer-Verlag Berlin Heidelberg, Queensland, Australien, 2011
- [CorRTb]** Corke, P.: Robotics Toolbox for Matlab (2011) [http://petercorke.com/Robotics_Toolbox.html] (Zugriff: 13.09.2012).
- [Dom11]** Domschke, W.; Drexl, A.: Einführung in Operations Research, , 2011
- [Fuc12]** Fuchs, A.; Kundrat, D.; Schultz, M.; Krüger, A.; Ortmaier, T.: Automatisierte Überwachung der Laserosteotomie mittels optischer Kohärenztomographie, Proceedings der 11. CURAC Jahrestagung, 2012
- [Heim06]** Heimann; Gerth; Popp: Mechatronik, Fachbuchverlag Leipzig im Carl Hanser Verlag, Hannover, 2006
- [KukaLBR4]** KUKA Roboter GmbH: Betriebsanleitung Leichtbauroboter 4+, 2010
- [MLTSP13]** MathWorks Inc.: Custom Data Type Optimization Using the Genetic Algorithm - MATLAB & Simulink Example (2012) [<http://www.mathworks.de/de/help/gads/examples/custom-data-type-optimization-using-the-genetic-algorithm.html>] (Zugriff: 29.01.2013).
- [Ölt10]** Öltjen, J.: Aufgabenorientierte Bewegungsplanung einer redundanten seriellen Kinematik hinsichtlich Kriterien der Kraftregelung, Diplomarbeit, Universität Hannover, 2010
- [Ortm11a]** Ortmaier, T.: Robotik I, Vorlesungsskript, Universität Hannover, 2011
- [Ortm12b]** Ortmaier, T.: Robotik II, Vorlesungsskript, Universität Hannover, 2012
- [Sch12]** Schappler, M.: Entwicklung und Parametrierung eines Robotermodells mittels messtechnischer Untersuchungen, Bachelorarbeit (Betreuer: Marcus Schönherr), IFW, Universität Hannover, 2012
- [Sici08]** Siciliano, B.; Khatib, O.: Springer Handbook of Robotics, Springer-Verlag Berlin Heidelberg, Berlin, 2008
- [SL20]** ScanLAB AG: Bedienungsanleitung hurrySCAN 20, 2011
-

A Anhang

A.1 Unterschiedliche Konfigurationen



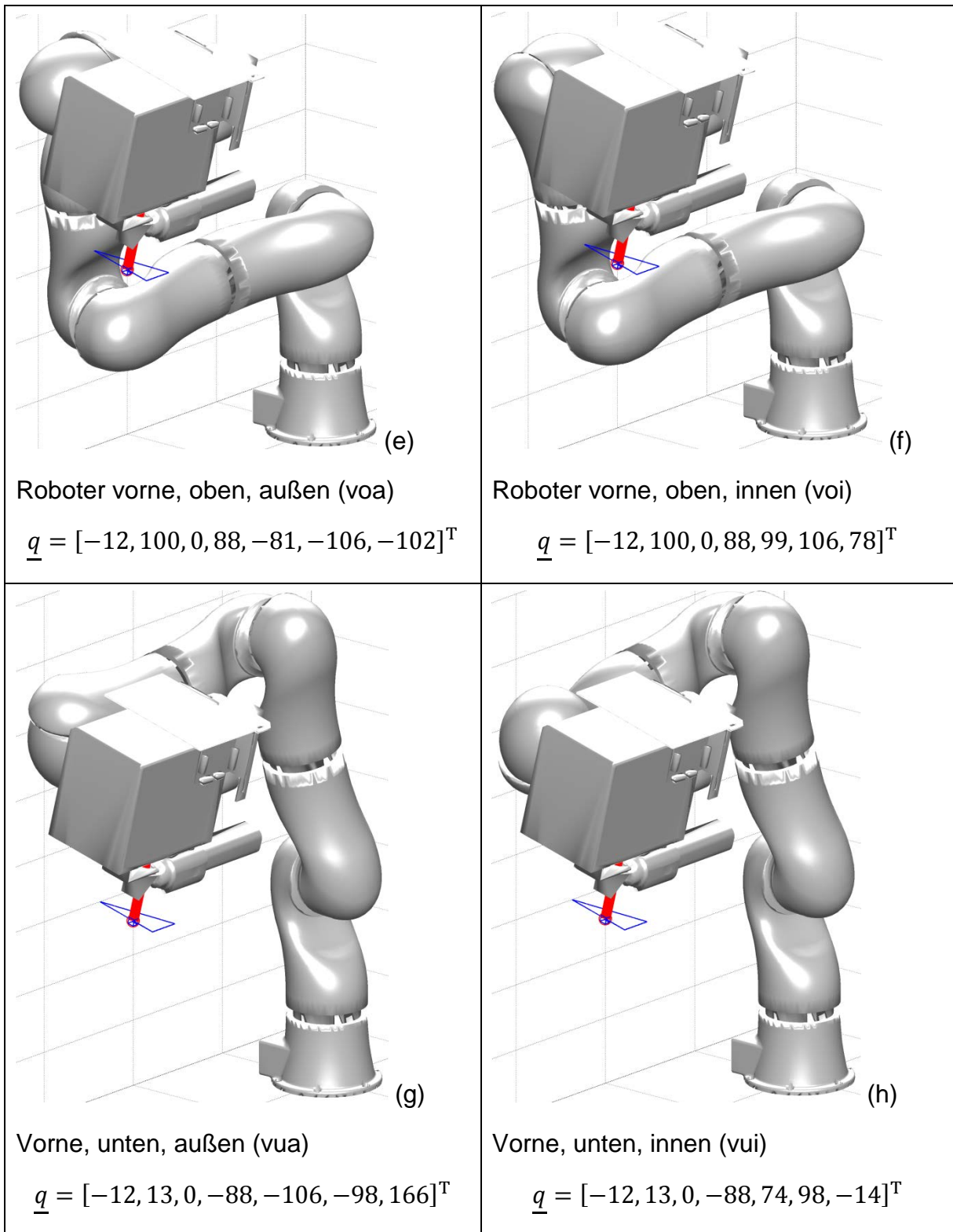


Bild A.1: (a) bis (f); Konfigurationen des LBR 4+

A.2 Austrittswinkel des Laserstrahls

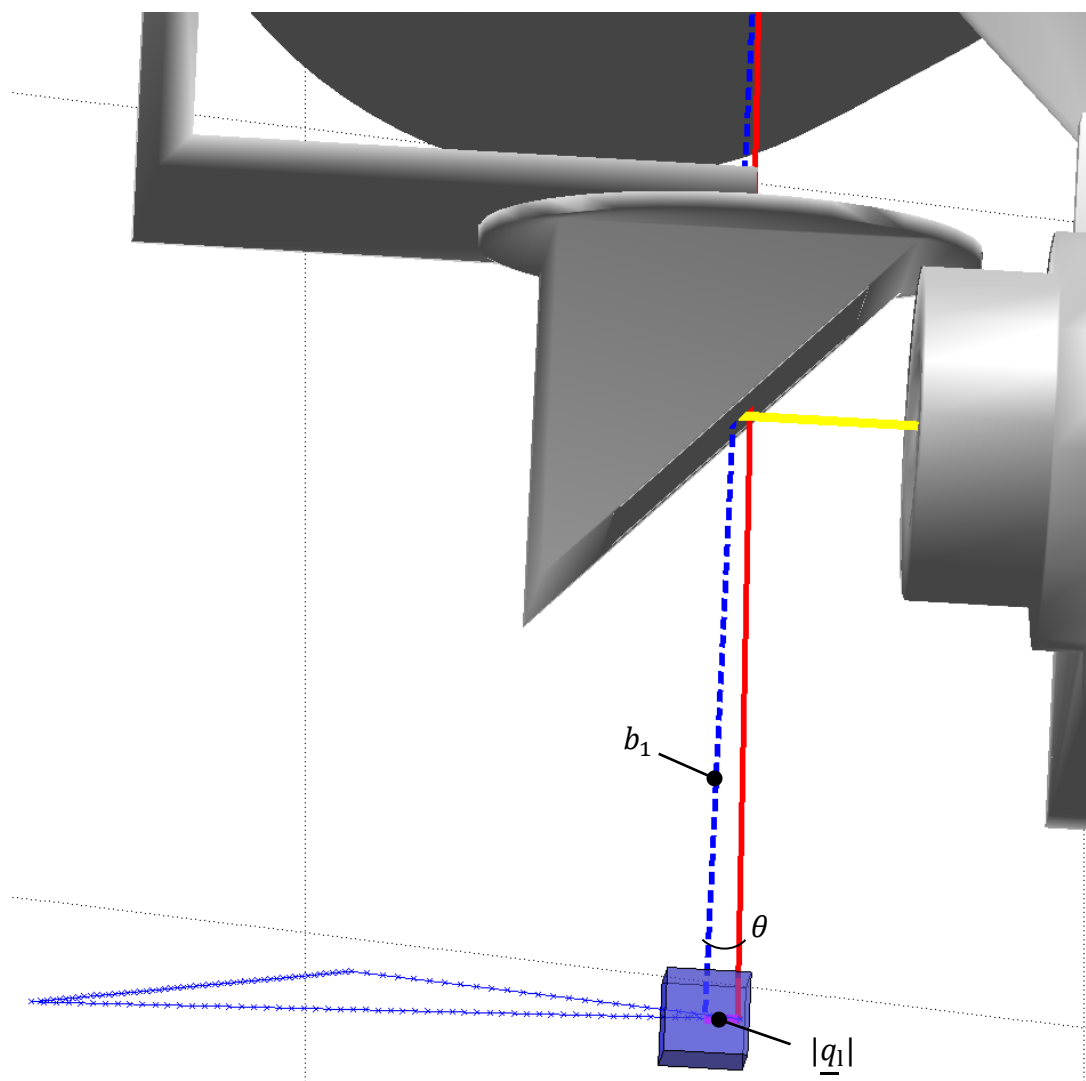


Bild A.2: Vergleich des realen und vereinfachten Austrittswinkels des Laserstrahls

Zur Vereinfachung der Berechnungen wird der Austrittswinkel θ des Laserstrahls, der durch die Ablenkung des Strahls durch bewegliche Spiegel gesteuert wird, vernachlässigt. Der Dabei entstehende Fehler θ_{\max} ergibt sich zu

$$\theta_{\max} = \arctan\left(\frac{|q_1|_{\max}}{b_1}\right) = \arctan\left(\frac{5 \text{ mm}}{100 \text{ mm}}\right) = 0,05 \text{ rad} = 2,86^\circ. \quad (\text{A.1})$$

Die Länge b_1 des freien Strahls und die maximale Auslenkung des Lasers $|q_1|_{\max}$ sind dem Datenblatt des Laserscanners [SL20] zu entnehmen.

A.3 Bilder zu den Ergebnissen der Bahnoptimierung

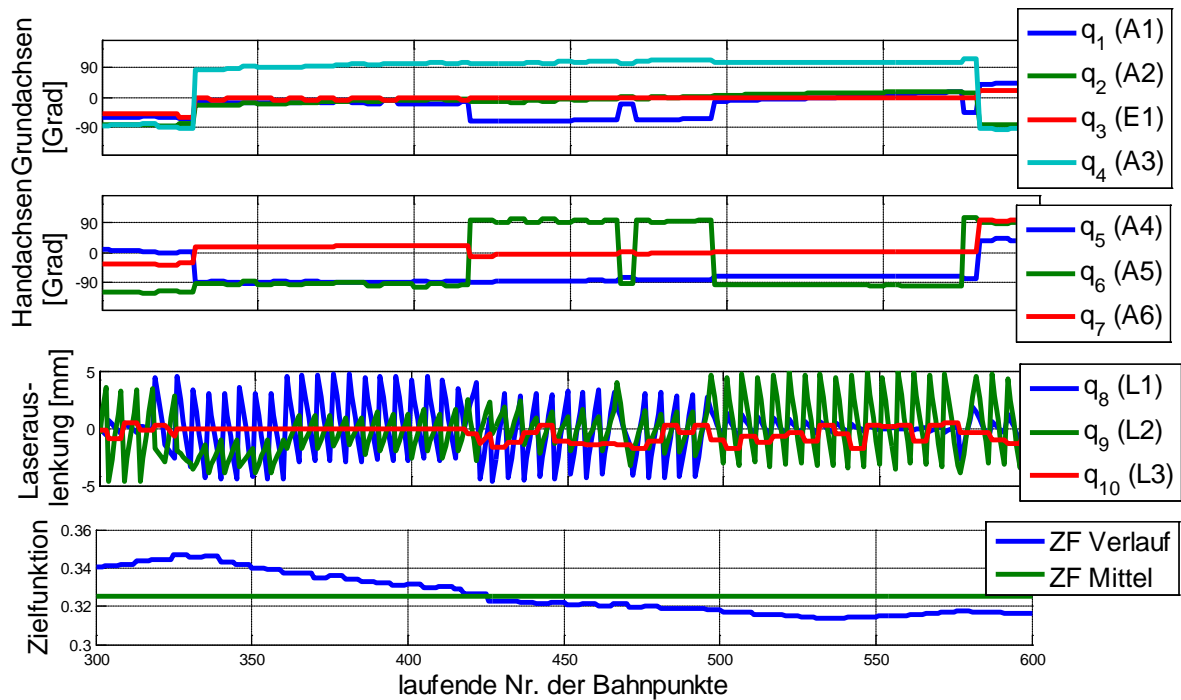


Bild A.3: Ausschnitt aus Gelenkwinkel-, Laserauslenkungs- und Zielfunktionsverlauf bei der Rechteck-Beispieltrajektorie mit Optimierung durch Greedy- Algorithmus ohne Berücksichtigung der Stetigkeit.

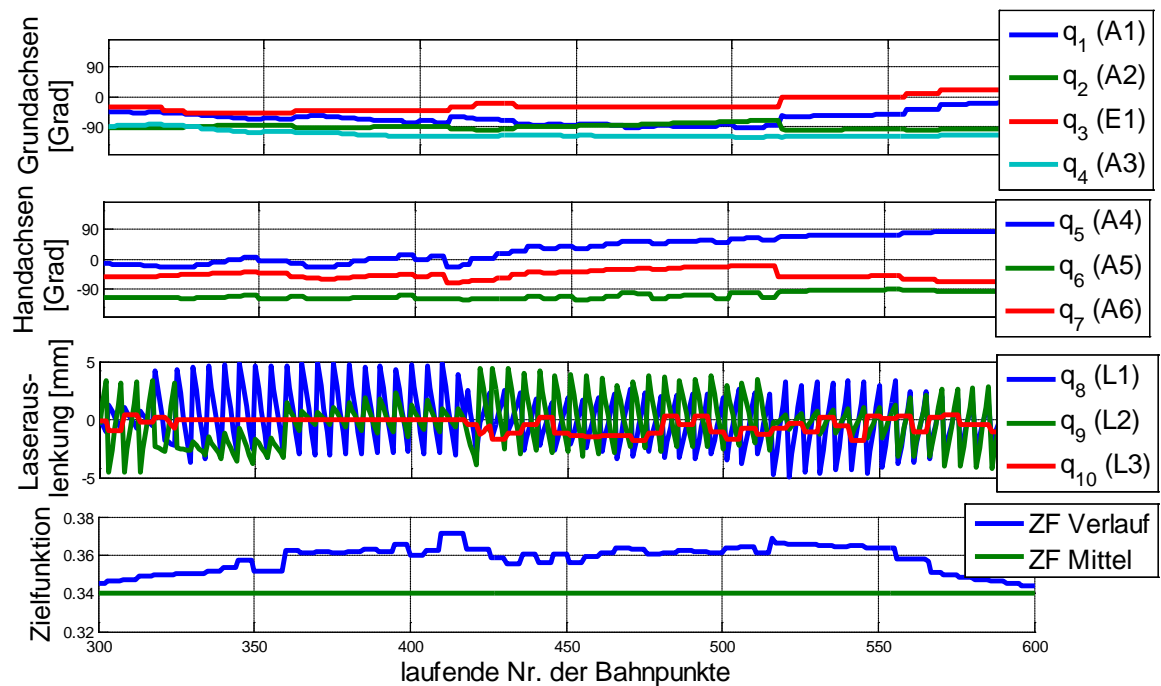


Bild A.4: Ausschnitt aus Gelenkwinkel-, Laserauslenkungs- und Zielfunktionsverlauf bei der Rechteck-Beispieltrajektorie mit Optimierung durch Greedy- Algorithmus mit Berücksichtigung der Stetigkeit.

A.4 Dateiformate

A.4.1 Trajektorie (Traj)

Trajektorien werden in MATLAB als Variable des Typs Struktur mit dem Namen „Traj“ angenommen. Die Einträge der Struktur sind der folgenden Auflistung zu entnehmen.

Traj.BahnEE: Enthält zeilenweise die Position der Bahnpunkte und ihre Orientierung in Kardan-Winkelkonvention. Durch das Setzen des letzten Kardanwinkels γ_E zu NaN (Not a Number) wird die Symmetrie des Werkzeugs angenommen.

Traj.Name: Name der Trajektorie zur Verwendung als Dateiname und zur Beschriftung

Traj.q: Gelenkwinkel des Roboters und Auslenkung des Lasers für jeden Bahnpunkt. Wird durch die Optimierung der Bahn erzeugt.

Traj.qName: Name des Optimierungsverfahrens, mit dem die Gelenkwinkel in Traj.q entstanden sind.

Traj.BahnLWz: Zeilenweise Bahn des Mittelpunktes des Laserwerkzeugs mit kartesischer Position und Orientierung in Kardan-Konvention.

Traj.PunktZuordnung:

Cell-Array mit Eintrag für jeden Punkt des Laserwerkzeugs aus Traj.BahnLWz. Zu jedem Laserarbeitsraum werden hier die Indizes der enthaltenen Bahnpunkt angegeben.

Traj.LwzZuordnung:

Vektor mit einem Eintrag für jeden Bahnpunkt aus Traj.BahnEE mit dem Index des Laserarbeitsraums, in dem der Bahnpunkt enthalten ist.

A.4.2 Trajektorienanalyse (TrajAnalyse)

Nach der Analyse der Trajektorie werden alle Gelenkwinkel der Nullraumanalyse für alle Bahnpunkte in der Struktur „TrajAnalyse“ gespeichert. Die Einträge sind der folgenden Liste zu entnehmen:

TrajAnalyse.KS_tool:

Koordinatentransformation vom Werkzeugflansch zum Mittelpunkt des Laserarbeitsraums in Kardan-Konvention

TrajAnalyse.q_Start:

Start-Roboter-gelenkwinkel, vor Ausführung der Analyse

TrajAnalyse.OptArtLaserWz:

Gibt an, wie der Laser berücksichtigt wird. 0 bedeutet keine Optimierung des Lasers. 1 bedeutet getrennte und 2 gemeinsame Optimierung von Roboter und Laser.

TrajAnalyse.Gewichtung:

Gewichtung der Leistungsmerkmale zur Zielfunktion.

TrajAnalyse.QGesamt:

Cell-Array mit Eintrag für jeden Bahnpunkt aus Traj.BahnEE, der alle zulässigen Gelenkwinkel enthält, die berechnet wurden.

TrajAnalyse.zfGesamt:

Cell-Array, mit den Zielfunktionswerten für die Gelenkwinkel aus TrajAnalyse.QGesamt.

A.4.3 Trajektorienoptimierung (TrajGlobOptErg)

Die Ergebnisse der globalen Bahnoptimierung sind in der Struktur „TrajGlobOptErg“ enthalten.

TrajGlobOptErg.AnzErf

Anzahl der erfolgreich berechneten optimierten Gelenkwinkeltrajektorien mit unterschiedlichen Verfahren.

TrajGlobOptErg.Q_Ges:

Cell-Array mit Gelenkwinkeln für jeden Bahnpunkt aus Traj.BahnEE für jede Gelenkwinkeltrajektorie

TrajGlobOptErg.zf_Verlauf:

Matrix mit dem Verlauf der gewichteten Gesamt-Zielfunktion für alle optimierten Gelenkwinkeltrajektorien

TrajGlobOptErg.zf_VerlaufAlle

Tensor mit dem Verlauf aller Leistungsmerkmale für die optimierten Gelenkwinkeltrajektorien

TrajGlobOptErg.zf_mittel

Vektor mit dem Mittelwert der gewichteten Zielfunktion für alle Gelenkwinkeltrajektorien

TrajGlobOptErg.AltNamen

Namen der alternativen optimierten Gelenkwinkeltrajektorien

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Ort, Datum

Unterschrift

