

Optimierung thermischer Verhältnisse bei der Bahnplanung  
für das thermische Spritzen mit Industrierobotern

**Dissertation**

zur Erlangung des Grades eines

D o k t o r s d e r I n g e n i e u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund  
an der Fakultät für Informatik

von

Daniel Hegels

Dortmund

2017

Tag der mündlichen Prüfung: 14.07.2017

Dekan: Prof. Dr.-Ing. Gernot A. Fink

Gutachter: Prof. Dr. Heinrich Müller  
Prof. Dr. Dominik Henrich



# Danksagung

An dieser Stelle möchte ich bei allen, die mich bei der Entstehung dieser Arbeit unterstützt haben, ganz herzlich bedanken. Ganz besonders möchte ich mich bei meinem Doktorvater, Herrn Prof. Dr. Heinrich Müller, für seine wertvolle Betreuung, die zahlreichen fachlichen Diskussionen sowie die großen zeitlichen Freiräume bedanken. Des Weiteren bedanke ich mich bei Herrn Prof. Dr. Dominik Henrich für seine Bereitschaft, das Zweitgutachten zu erstellen, sowie bei Herrn Prof. Dr. Dietmar Jannach und Herrn Prof. Dr. Peter Buchholz für deren Mitgliedschaft in der Kommission.

Meinen Kollegen am Lehrstuhl für Graphische Systeme danke ich für das hervorragende Arbeitsklima. Besonders bedanke ich mich bei Adrian Böckenkamp für die vielen fachlichen Diskussionen und seine Hilfsbereitschaft. Bei Thomas Wiederkehr möchte ich mich für die gute Zusammenarbeit im Sonderforschungsbereich 708 sowie an der Simulationsschnittstelle bedanken.

Abschließend möchte ich mich ganz besonders bei meinen Eltern bedanken, die mich während des Studiums und der Promotion stets unterstützt und motiviert haben. Meiner Freundin Leonie danke ich für ihre vielfältige Unterstützung und für ihre Bereitschaft, mir in der Endphase der Arbeit den nötigen zeitlichen Freiraum zur Fertigstellung der Arbeit zu geben.



# Zusammenfassung

Diese Arbeit befasst sich mit der Erzeugung und Optimierung von neuartigen Bahnen für Industrieroboter beim thermischen Spritzen auf komplexen Freiformoberflächen unter besonderer Berücksichtigung der thermischen Verhältnisse in dem Werkstück. Thermisches Spritzen ist ein Produktionsprozess, bei dem eine Werkstückoberfläche mit geschmolzenem Material beschichtet wird, so dass die Oberfläche die gewünschten Oberflächeneigenschaften aufweist. Ein Alleinstellungsmerkmal des präsentierten Systems ist der modulare Aufbau, der vor allem eine in diesem Bereich unübliche Trennung zwischen der Initialbahnplanung und der Bahnoptimierung vorsieht. Die Basis des Gesamtsystems bilden verschiedene Simulationskomponenten, wie die Beschichtungssimulation, die thermische Simulation und die Robotersimulation.

Die Initialbahnplanung erzeugt flächenüberdeckende Bahnen auf einem Werkstück unter Berücksichtigung verschiedener Qualitätsmerkmale. Dazu werden die Bahnen über flexible Bahnstrukturen repräsentiert, darunter neuartige Strukturen, wie die Rand-zu-Rand Bahnen und die Punkt-zu-Punkt Bahnen. Die Qualität der Bahnen wird über verschiedene Zielfunktionen bewertet, die neben der Schichtqualität vor allem die thermischen Varianzen berücksichtigen, welche bisher nur selten in Betracht gezogen wurden, obwohl sie großen Einfluss auf die endgültige Schichtqualität haben. Weitere praxisrelevante Zielkriterien, wie die Roboterachsbeschleunigungen und der Overspray, welcher das Material beschreibt, das nicht auf der funktionalen Fläche abgelagert wird, werden ebenfalls beachtet. Das Problem der Initialbahnplanung wird als mehrkriterielles Optimierungsproblem formuliert und mit Hilfe eines Evolutionären Algorithmus optimiert. Verschiedene Varianten für die Operatoren des Evolutionären Algorithmus werden verwendet und gegeneinander evaluiert. Hieraus wird die Kombination von Operatoren bestimmt, mit der der Algorithmus mit hoher Konvergenzgeschwindigkeit strukturell gute Bahnen für den anschließenden Bahnoptimierungsprozess erzeugt.

Die Bahnoptimierung wird für die Verbesserung vorhandener Bahnen bezüglich der Beschichtungsfehler und der Ausführbarkeit mit Robotern verwendet. Ein neuartiges Konzept zur kombinierten Anwendung des in der Arbeit entwickelten, analytischen Auftragsmodells mit einer externen Blackbox Simulation wird verwendet, um die Bahnen mit Hilfe des Verfahrens der nichtlinearen konjugierten Gradienten zu optimieren. Die Fehler werden hierbei über die externe Simulation und die Gradienten über das analytische

Auftragsmodell bestimmt. Die Verwendung der Bahnoptimierung beschränkt sich nicht nur auf die Optimierung der Bahnen, die von der Initialbahnplanung erstellt worden sind, sondern kann ebenfalls genutzt werden, um bereits erstellte Bahnen an andere Spritzprozesse oder ähnliche Werkstückgeometrien anzupassen. Hierdurch lässt sich der erhebliche Aufwand zur Generierung neuer Bahnen stark reduzieren.

Zum Abschluss der Arbeit wird ein Verfahren vorgestellt, das die bisher unberücksichtigte Roboterdynamik in das System miteinbezieht. Dazu wird eine Dynamikkorrektur präsentiert, die die Bahnen mit Hilfe einer Roboterherstellersoftware in den dynamisch zulässigen Bereich projiziert. Diese Projektion wird in einer weiteren Optimierungsschleife alternierend mit der Bahnoptimierung genutzt, um eine dynamisch zulässige Bahn zu erzeugen, die sehr gute Ergebnisse bezüglich der Qualitätsmaße liefert.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Thermisches Spritzen . . . . .	2
1.3	Problemstellung und Lösungsweg . . . . .	4
1.3.1	Probleme . . . . .	5
1.3.2	Ziele . . . . .	5
1.3.3	Lösung . . . . .	6
1.3.4	Systemarchitektur . . . . .	6
1.3.5	Abgrenzung zu verfügbarer Bahnplanungssoftware . . . . .	8
1.4	Beiträge der Dissertation . . . . .	10
1.5	Eigene Publikationen . . . . .	13
1.6	Struktur der Dissertation . . . . .	14
1.7	Danksagung . . . . .	14
<b>2</b>	<b>Zentrale Begriffe</b>	<b>15</b>
2.1	Bahnen . . . . .	15
2.2	Werkstücke . . . . .	18
2.3	Industrieroboter . . . . .	21
2.3.1	Aufbau . . . . .	21
2.3.2	Kinematik . . . . .	22
2.3.3	Programmierung . . . . .	25
<b>3</b>	<b>Simulation</b>	<b>27</b>
3.1	Einleitung . . . . .	27
3.2	Auftragssimulation . . . . .	28
3.2.1	Einordnung und Abgrenzung zu existierenden Modellen . . . . .	29
3.2.2	Auftragsmodell . . . . .	30
3.2.3	Kalibrierung . . . . .	33
3.3	Thermische Simulation . . . . .	35
3.3.1	Einordnung und Abgrenzung zu existierenden Modellen . . . . .	35
3.3.2	Thermisches Modell . . . . .	35
3.3.3	Gesamtmodell der thermischen Simulation . . . . .	38
3.3.4	Kalibrierung . . . . .	38

3.4	Gesamtmodell . . . . .	40
3.5	Evaluation . . . . .	41
3.5.1	Auftragssimulation . . . . .	41
3.5.2	Thermische Simulation . . . . .	44
3.6	Fazit . . . . .	47
<b>4</b>	<b>Initialbahnplanung</b>	<b>49</b>
4.1	Einleitung . . . . .	50
4.2	Stand der Forschung . . . . .	51
4.2.1	Bahnplanung beim thermischen Spritzen . . . . .	51
4.2.2	Bahnplanung unter Berücksichtigung der Wärme . . . . .	54
4.3	Übersicht . . . . .	55
4.4	Formulierung als Optimierungsproblem . . . . .	56
4.5	Bahnerstellung . . . . .	58
4.5.1	Region of Interest . . . . .	59
4.5.2	Rand-zu-Rand Bahnen . . . . .	59
4.5.3	Punkt-zu-Punkt Bahnen . . . . .	64
4.5.4	Quasi-konturparallele Bahnen . . . . .	65
4.5.5	Vergleich der Bahnstrukturen . . . . .	67
4.5.6	Erstellen der Werkzeugbahn . . . . .	67
4.5.7	Zu- und Abfuhrbahnen . . . . .	68
4.5.8	Roboterplatzierung . . . . .	69
4.6	Zielfunktion . . . . .	71
4.7	Evolutionäre Optimierung . . . . .	74
4.7.1	Komponenten eines Evolutionären Algorithmus . . . . .	74
4.7.2	Zielfunktion, Fitness und mehrkriterielle Optimierung . . . . .	75
4.7.3	Repräsentation . . . . .	76
4.7.4	Initialisierung . . . . .	77
4.7.5	Rekombination . . . . .	77
4.7.6	Mutation . . . . .	77
4.7.7	Elternselektion . . . . .	78
4.7.8	Überlebensselektion . . . . .	78
4.7.9	Bekannte Algorithmen . . . . .	79
4.7.10	Anwendung der Evolutionären Optimierung auf die Bahnstrukturen	79
4.8	Problemstruktur . . . . .	79
4.8.1	Zusammenhang mit dem TSP . . . . .	80
4.8.2	Zusammenhang mit dem Rucksackproblem . . . . .	82
4.8.3	Zusammenhang mit dem TTP . . . . .	83
4.8.4	Zusammenhang mit dem Scheduling . . . . .	84
4.9	Optimierung von Punkt-zu-Punkt Bahnen . . . . .	85
4.9.1	Zielfunktion . . . . .	86
4.9.2	Repräsentation . . . . .	90
4.9.3	Initialisierung . . . . .	90
4.9.4	Rekombination . . . . .	92

---

4.9.5	Mutation . . . . .	95
4.9.6	Elternselektion . . . . .	97
4.9.7	Überlebensselektion . . . . .	99
4.9.8	Auswahlselektion . . . . .	101
4.9.9	Caching . . . . .	101
4.9.10	Parallelisierung und Generationenkonzept . . . . .	102
4.9.11	Parameterkontrolle . . . . .	102
4.9.12	Parameteroptimierung: Rekombination, Mutation und Selektion	103
4.9.13	Analyse der Auflösungen . . . . .	109
4.9.14	Analyse der Algorithmeigenschaften . . . . .	114
4.9.15	Analyse der Laufzeit . . . . .	128
4.10	Optimierung von Rand-zu-Rand Bahnen . . . . .	134
4.11	Optimierung von quasi-konturparallelen Bahnen . . . . .	136
4.12	Evaluation . . . . .	138
4.12.1	Testdaten und Qualitätsmerkmale . . . . .	138
4.12.2	Vergleich der Verfahren . . . . .	139
<b>5</b>	<b>Bahnoptimierung</b>	<b>147</b>
5.1	Einleitung . . . . .	148
5.2	Stand der Forschung . . . . .	149
5.3	Übersicht . . . . .	150
5.4	Zielfunktion . . . . .	151
5.4.1	Bahnbasierte Zielfunktionskomponenten . . . . .	153
5.4.2	Roboterbasierte Zielfunktionskomponenten . . . . .	154
5.4.3	Werkstückbasierte Zielfunktionskomponenten . . . . .	155
5.4.4	Gesamtzielfunktion . . . . .	155
5.4.5	Einschränkungen . . . . .	156
5.5	Nichtlineare konjugierte Gradienten . . . . .	156
5.5.1	Richtungen . . . . .	157
5.5.2	Schrittweite . . . . .	158
5.6	Optimierung . . . . .	159
5.6.1	Bahnbasierte Optimierung . . . . .	160
5.6.2	Roboterbasierte Optimierung . . . . .	160
5.6.3	Werkstückbasierte Optimierung . . . . .	160
5.7	Evaluation . . . . .	160
5.7.1	Testdaten und Qualitätsmerkmale . . . . .	161
5.7.2	Vergleich der Verfahren und Gradientenfunktionen . . . . .	162
5.7.3	Anzahl der Iterationen . . . . .	164
5.7.4	Einfluss der Parameter . . . . .	165
5.7.5	Einfluss des Modells . . . . .	167
5.7.6	Einfluss der Eingabe . . . . .	169
5.7.7	Winkel der Gradienten . . . . .	171
5.7.8	Einfluss der Knotenverschiebung . . . . .	172
5.7.9	Zusammensetzung der Laufzeit . . . . .	173

---

5.7.10 Fallbeispiele . . . . .	174
<b>6 Nutzung optimierter Bahnen</b>	<b>179</b>
6.1 Einleitung . . . . .	180
6.2 Übersicht . . . . .	180
6.3 RobotStudio . . . . .	182
6.4 Iterative Anpassung der Bahnen . . . . .	183
6.5 Evaluation . . . . .	185
6.5.1 Testdaten und Qualitätsmerkmale . . . . .	186
6.5.2 Evaluation auf Werkstück A . . . . .	186
6.5.3 Evaluation auf Werkstück B . . . . .	188
<b>7 Zusammenfassung und Ausblick</b>	<b>193</b>
7.1 Zusammenfassung . . . . .	193
7.2 Ausblick . . . . .	195
<b>A Ableitungen</b>	<b>197</b>
A.1 Bahnbasierte Optimierung . . . . .	197
A.1.1 Ableitungen der Zielfunktion . . . . .	197
A.1.2 Ableitungen der Glättungsterme . . . . .	200
A.2 Roboterbasierte Optimierung . . . . .	204
A.2.1 Ableitungen der Zielfunktion . . . . .	204
A.2.2 Ableitungen der Glättungsterme . . . . .	205
A.3 Werkstückbasierte Optimierung . . . . .	205
A.3.1 Ableitungen der Zielfunktion . . . . .	205
A.3.2 Ableitungen der Glättungsterme . . . . .	206
<b>Mathematische Notation</b>	<b>207</b>
<b>Abbildungsverzeichnis</b>	<b>211</b>
<b>Tabellenverzeichnis</b>	<b>213</b>
<b>Literaturverzeichnis</b>	<b>215</b>



# Einleitung

Diese Arbeit befasst sich mit der Konzeptionierung, Realisierung und Evaluation von neuartigen Bahnplanungsmethoden für das thermische Spritzen mit besonderem Fokus auf die Minimierung thermischer Varianzen. In der folgenden Einleitung wird zunächst das Problem der Bahnplanung im Kontext des thermischen Spritzens erläutert und das Vorgehen motiviert. Anschließend wird eine kurze Einführung in das thermische Spritzen gegeben, die die für das Verständnis der Arbeit notwendigen Grundlagen des Prozesses erläutert. Es folgt die Beschreibung der genauen Problemstellung und des Lösungswegs dieser Arbeit. Anschließend werden die Beiträge der Dissertation präsentiert und es wird eine Übersicht über eigene Publikationen gegeben, bevor abschließend die Struktur der Dissertation dargelegt wird.

## 1.1 Motivation

Thermisches Spritzen ist ein Produktionsprozess, bei dem die Oberfläche eines Werkstücks so behandelt wird, dass sie die für ihren Einsatzzweck benötigten Eigenschaften aufweist. Dazu wird geschmolzenes Material auf die Oberfläche gespritzt, das beim anschließenden Abkühlen erstarrt und eine feste Schicht bildet, siehe Abschnitt 1.2. Um die gewünschten Eigenschaften in der notwendigen Qualität herzustellen, muss die Schicht meist gleichmäßig unter Vorgabe einer genauen Zielschichtdicke aufgetragen werden. Die Qualität der Schicht ist demnach wesentlich von dem Auftragsprozess abhängig. Daher ist es nötig, auf Roboter zurückzugreifen, die, nach entsprechender Programmierung, mit hoher Wiederholgenauigkeit viele Werkstücke mit gleichbleibender Qualität beschichten können. Die Aufgabe besteht dann in der Programmierung des Roboters mit Bahnen, die die gewünschte Schicht auftragen.

Aus den oben genannten Gründen ist die Qualität der Schicht besonders von der Programmierung des Roboters abhängig. In der Praxis wird häufig auf Teaching Verfahren zurückgegriffen [131], bei denen ein erfahrener Experte den Roboter direkt führt, während der

Roboter die Bewegungen speichert, um sie später zu wiederholen. Da dieser Prozess zeit-aufwändig, teuer und nicht ungefährlich ist und die Qualität der Bahn und demnach der Schicht von den Fähigkeiten des Experten abhängt, sind Offline-Bahnplanungsmethoden zu bevorzugen. Hier wird mit Hilfe von Algorithmen und Simulationen eine Bahn erstellt, die später von dem Roboter ausgeführt werden kann.

Bahnplanungsalgorithmen sind speziell im Bereich des thermischen Spritzens wenig untersucht. Häufig werden einfache Bahnstrukturen wie Mäanderbahnen eingesetzt, welche nur für strukturell einfache Werkstückgeometrien geeignet sind, da sie nicht in der Lage sind, die Krümmungen auf komplexeren Oberflächen zu berücksichtigen. Flexiblere Bahnstrukturen, wie sie in dieser Arbeit vorgestellt werden, können sich an die Krümmungen der Oberfläche anpassen und den Beschichtungsfehler an diesen Stellen reduzieren. Des Weiteren wird der Aspekt der thermischen Varianzen in dem Werkstück selten berücksichtigt. Hohe thermische Varianzen sind jedoch qualitätsmindernd, da durch die Temperaturunterschiede im Werkstück sowie der Schicht Spannungen durch unterschiedliche Ausdehnungen in den Materialien auftreten können und die Schicht im schlimmsten Fall Risse bekommen oder von dem Werkstück abplatzen kann. Einige wenige Untersuchungen zur Bahnplanung mit dem Ziel der Minimierung der thermischen Varianzen sind von Jones et al. [94, 54, 95] durchgeführt worden. Die hierin beschriebene Methodik ist jedoch auf planare, rechteckige Oberflächen beschränkt.

In dieser Arbeit werden neuartige Methoden zur Bahnplanung erarbeitet, die einerseits für komplexe Werkstückgeometrien geeignet sind und andererseits die thermischen Varianzen gezielt minimieren.

## 1.2 Thermisches Spritzen

Unter thermischem Spritzen versteht man eine Menge von Prozessen zur Oberflächenbehandlung von Werkstücken. Das grundlegende Prinzip besteht in dem Verflüssigen eines Beschichtungsmaterials in der Spritzpistole und dem Auftragen des aufgeschmolzenen Materials auf die Werkstückoberfläche mittels eines Gasstroms. In Abbildung 1.1 ist der Lichtbogenspritzprozess abgebildet, bei dem die Spritzpistole auf einen Roboterarm montiert wurde. Die unterschiedlichen Prozesse verwenden jeweils verschiedene Verfahren zum Aufschmelzen und Beschleunigen der Partikel und können unter anderem über die Art des Energieträgers kategorisiert werden, siehe Abbildung 1.2.

In der vorliegenden Arbeit wird der Prozess vorwiegend theoretisch betrachtet und die Algorithmen sind für alle Prozesse gleichermaßen anwendbar. Mit Hilfe einer Kalibrierung der Prozessparameter (Footprint) kann das Verfahren auf einen konkreten Prozess angepasst werden. Um praxisnahe Ergebnisse zu erzielen, werden die genutzten Simulationen mit Hilfe der Daten eines Lichtbogenspritzprozesses parametrisiert. Diese Daten wurden für die genutzte externe Simulation anhand real vermessener Footprintmodelle (siehe Kapitel 3) erhoben und können somit als praxisrelevant angesehen werden. Auch die

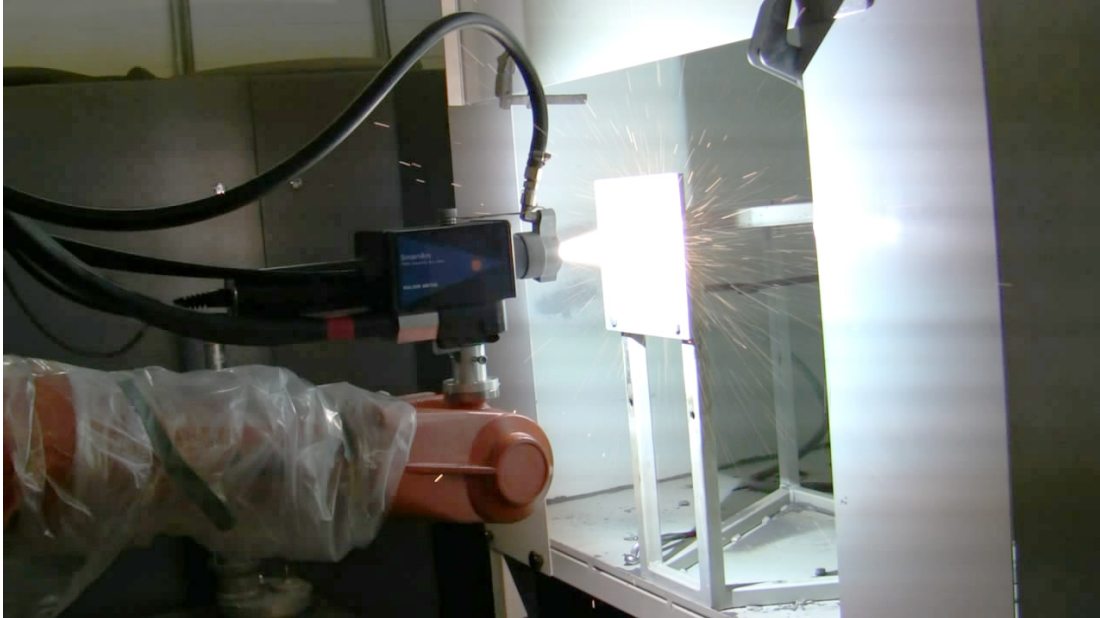


Abbildung 1.1: Industrieroboter beim thermischen Spritzen mit Lichtbogen-Spritzpistole (Zur Verfügung gestellt von Thomas Wiederkehr).

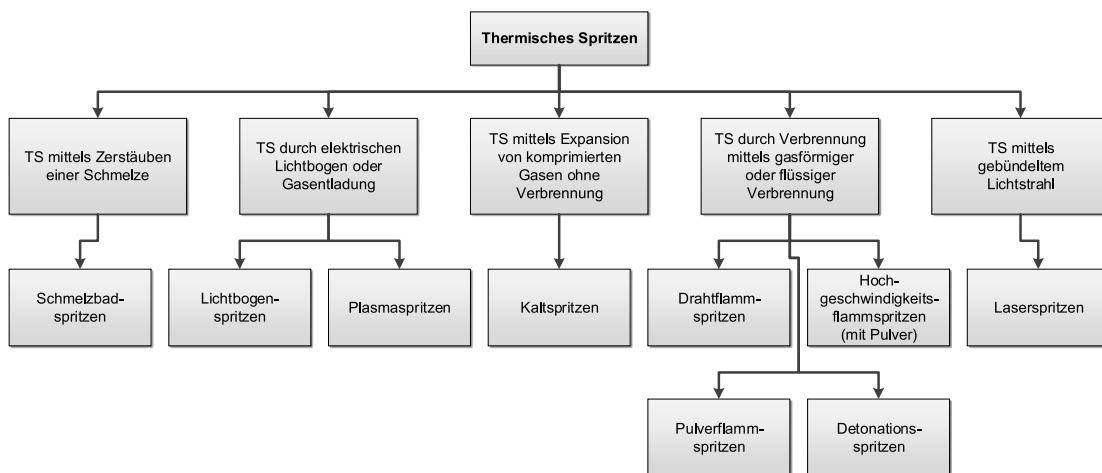


Abbildung 1.2: Gliederung thermischer Spritzmethoden nach Energieträgern (nach [51]).

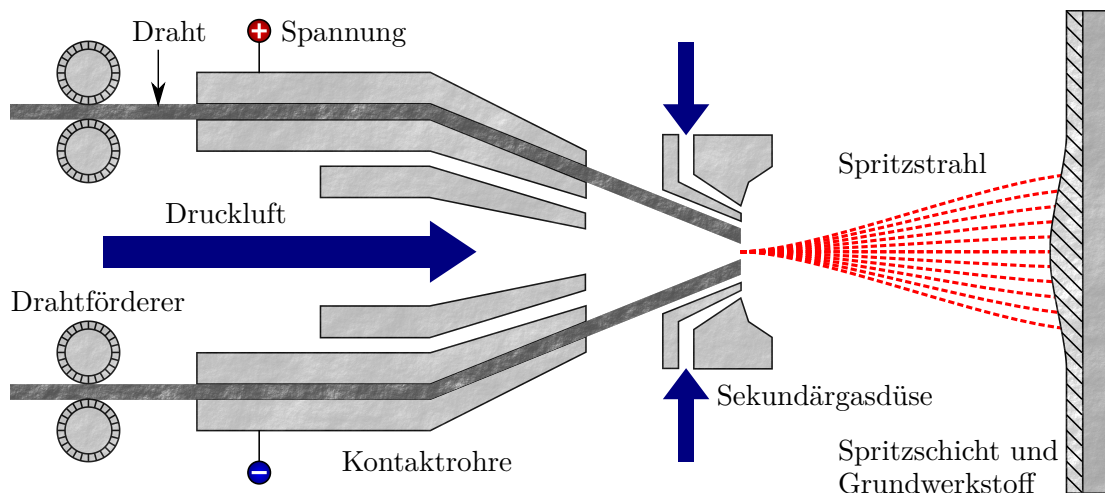


Abbildung 1.3: Schematischer Aufbau einer Lichtbogen-Spritzpistole nach [51, 169].

selbst entwickelte Simulation wird gegenüber der externen Simulation kalibriert, so dass diese implizit auf die gleiche Datenbasis zurückgreift.

Der schematische Aufbau einer Spritzpistole für das Lichtbogenspritzen ist in Abbildung 1.3 dargestellt. Bei dem Lichtbogenspritzen wird das aufzutragende Material in Form von Drähten in die Spritzpistole geführt. Die Drähte werden so an eine Spannungsversorgung angeschlossen, dass sich zwischen ihnen ein Potential ausbildet. Wenn die Drahtenden in der Spritzpistole nahe aneinandergeführt werden, entsteht ein Lichtbogen, der die Drahtenden aufschmilzt und die Materialpartikel freisetzt. Über eine Primärgasversorgung wird Druckluft von hinten in die Spritzpistole geleitet, so dass die Partikel in dem Gasstrom in Richtung der Oberfläche mitgerissen werden. Über eine Sekundärgasdüse an der Austrittsdüse der Pistole kann der Spritzstrahl noch einmal fokussiert werden.

Als Material kommen beim thermischen Spritzen viele verschiedene Stoffe in Frage. Beim Lichtbogenspritzen ist eine grundlegende Bedingung die elektrische Leitfähigkeit des Materials. Die Auswahl des entsprechenden Materials ergibt sich aus den gewünschten Eigenschaften, die auf der Oberfläche erreicht werden sollen. Häufig werden Materialien verwendet, die dem Verschleißschutz oder der Wärmedämmung dienen. Auch die zu erzeugende Schichtdicke, die von wenigen Mikrometern bis zu einigen Millimetern reichen kann, wird von dem Anwendungszweck und dem verwendeten Material bestimmt.

### 1.3 Problemstellung und Lösungsweg

In diesem Abschnitt wird die Problemstellung beim roboterbasierten thermischen Spritzen und der in der Arbeit verfolgte Lösungsweg beschrieben. Hierzu werden zunächst die Probleme erläutert, die sich im Zusammenhang mit dem Verfahren ergeben. Hieraus

lassen sich die Zielsetzungen für diese Arbeit ableiten, aus denen wiederum der Lösungsansatz hergeleitet wird. Es folgt eine Beschreibung der Systemarchitektur, mit der der Lösungsansatz realisiert wird. Schließlich wird noch eine Abgrenzung zu bereits vorhandenen, kommerziellen Softwarelösungen präsentiert.

### 1.3.1 Probleme

Für die Bahnplanung beim roboterbasierten thermischen Spritzen ergeben sich neben den allgemeinen, für robotergeführte Prozesse üblichen Problemen, auch prozessspezifische Fragestellungen. Daher müssen insbesondere folgende Aspekte betrachtet werden:

- **Auftrag:** Die Schichtdicke muss an allen Stellen der Oberfläche möglichst genau mit dem geforderten Soll übereinstimmen.
- **Wärme:** Die Wärmeverteilung auf der Oberfläche sowie in dem Werkstück soll während des Prozesses möglichst geringe Varianzen aufweisen.
- **Kinematik:** Die Kinematik des Roboters muss beachtet werden, so dass die Bahn von dem Roboter problemlos abgefahren werden kann.
- **Dynamik:** Die Dynamik muss beachtet werden, welche die Bewegung des Roboters als Resultat von Kräften betrachtet, die zum Beispiel von den Gelenkmotoren verursacht werden. Die benötigten Kräfte sollen reduziert werden.
- **Platzierung:** Der Roboter muss so platziert werden, dass die Bahn kinematisch problemlos abgefahren werden kann und die Kräfte, die durch die Gelenkmotoren erzeugt werden, möglichst gering sind.
- **Overspray:** Der Overspray, welcher die Menge an Material beschreibt, das nicht auf der funktionalen Oberfläche abgelagert wird, sollte möglichst gering sein.

### 1.3.2 Ziele

Aus der oben genannten Problemstellung lassen sich die Ziele für diese Arbeit definieren:

- Reduktion des Beschichtungsfehlers,
- Reduktion der Wärmevarianz,
- Verbesserung der Roboterkinematik bei gegebener Platzierung des Roboters,
- Verminderung des Oversprays.

In dieser Arbeit stehen die Punkte Materialauftrag, Wärme, Kinematik und Overspray im Fokus. Die Platzierung des Roboters wird über ein einfaches Verfahren realisiert. Für komplexere Methoden wird auf die Literatur verwiesen. Die Roboterdynamik wird durch eine externe Simulationssoftware in das System integriert, siehe Kapitel 6.

### 1.3.3 Lösung

Um die Ziele der Arbeit zu erreichen, wird ein neuartiges Lösungskonzept präsentiert. Dieses beruht auf der Trennung der Bahnplanung in zwei Phasen, die Initialbahnplanung (Kapitel 4) und die Bahnoptimierung (Kapitel 5). In der Initialbahnplanung wird eine Bahn basierend auf der Form der Oberfläche bestimmt, so dass der Auftrag einigermaßen dem Soll entspricht, der Overspray minimiert wird, die Roboterkinematik berücksichtigt wird und die geometrische Struktur der Bahn geringe thermische Varianzen verursacht. Hierbei wird auf flexible Bahnstrukturen und nicht wie bisher auf unflexible Mäanderbahnen zurückgegriffen. In der Bahnoptimierung wird die durch die Initialbahnplanung bestimmte geometrische Form der Bahn lokal so verändert, dass der Auftrag und die Roboterkinematik optimiert werden ohne die thermischen Varianzen bedeutend zu verschlechtern. Gestützt wird das Konzept durch die Nutzung verschiedener Simulationen für den Auftrag (Abschnitt 3.2), die Wärme (Abschnitt 3.3) sowie das Roboterverhalten (Abschnitt 2.3).

### 1.3.4 Systemarchitektur

In dieser Arbeit wird die Bahnplanung für das thermische Spritzen in verschiedene, untereinander vernetzte Komponenten zerlegt. Die Eingabe ist ein Werkstück, für das die Bahnplanung ausgeführt werden soll. Die Ausgabe ist eine Bahn, die die vorgegebenen Ziele erfüllt. Optional kann während der Bahnoptimierung auch die Geometrie des Eingabewerkstücks leicht verändert werden, um die Zielkontur nach der Beschichtung möglichst exakt zu erreichen.

Abbildung 1.4 zeigt alle Komponenten und deren Zusammenhang in der Gesamtarchitektur. Die Optimierungskomponenten bilden hierbei auf horizontaler Ebene eine Art Pipeline, die den normalen Workflow beschreibt. Dieser besteht zunächst aus der Bestimmung einer geeigneten Roboterposition mit Hilfe des gegebenen Werkstücks, danach aus der Initialbahnplanung und als Letztes aus der Bahnoptimierung. Der genaue Ablauf wird durch die durchgezogenen Pfeile dargestellt. Die Zwischenergebnisse der einzelnen Module werden von einem zum nächsten Modul weitergegeben. Die Simulations- und externen Module stehen in jedem Schritt zur Verfügung. Die gestrichelten Pfeile stellen die Kommunikation und den Datenaustausch der einzelnen Komponenten mit diesen Modulen dar.

Ein erster Schritt ist die Roboterplatzierung. Dieser Schritt ist optional und muss nur dann ausgeführt werden, wenn eine roboterbasierte Bahnoptimierung ausgeführt werden soll, siehe Kapitel 5. Die Roboterplatzierung nutzt ein Grid-Search Verfahren, das den Roboter bezüglich des gegebenen Werkstücks so positioniert, dass alle Stellen des Werkstücks, welche beschichtet werden sollen, erreicht werden können. Dabei wird die Position so gewählt, dass möglichst keine Achse nahe an ihre Extremstellungen gerät.

Der nächste Schritt ist die Initialbahnplanung. Diese ist für das Erstellen von initialen Bahnen anhand der Werkstückgeometrie verantwortlich, siehe Kapitel 4. Verschiedene

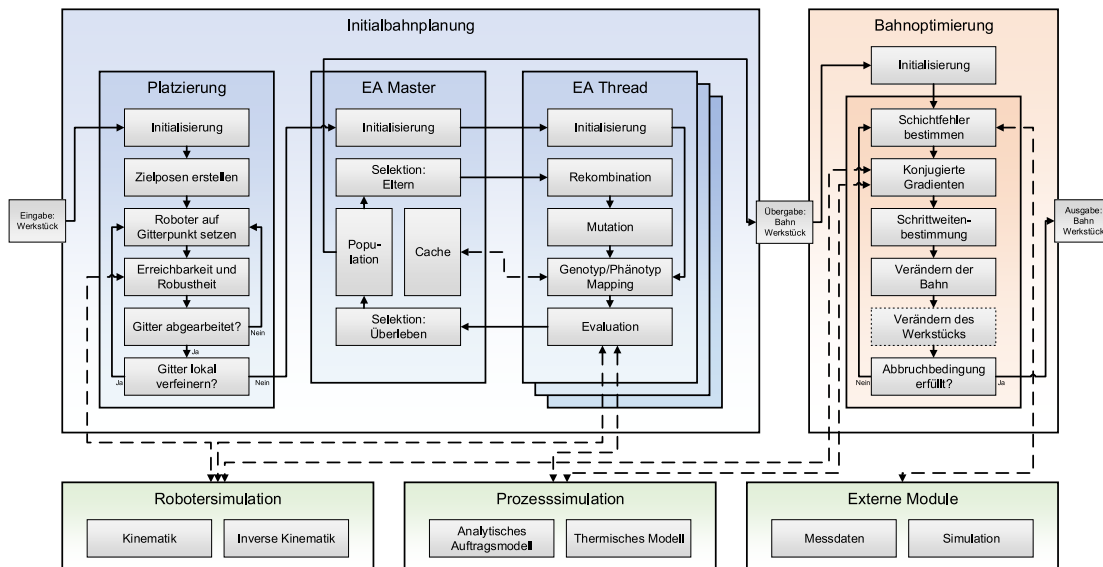


Abbildung 1.4: Systemarchitektur der Bahnplanung beim thermischen Spritzen. Das System besteht aus der Initialbahnplanung zur Erzeugung von neuen Bahnen auf einem Werkstück inklusive der Roboterplatzierung (Kapitel 4), der Bahnoptimierung zur Verbesserung der Bahn (Kapitel 5), der Prozesssimulation (Kapitel 3), der Robotersimulation (Abschnitt 2.3) und einigen externen Modulen. Die Komponenten des Systems sind durch graue Rechtecke dargestellt. Optionale Komponenten haben einen gepunkteten Rand. Die Kontrollflüsse sind durch die durchgezogenen Pfeile gekennzeichnet, die Datenkommunikation über gestrichelte Pfeile.

Bahnstrukturen können verwendet werden, welche auf unterschiedlichen Konzepten wie klassischen Parallelbahnen oder netzbasierten Techniken basieren. Das Ziel der Initialbahnplanung ist die Anpassung der durch die Bahnstruktur gegebenen Freiheitsgrade, um eine Gesamtbahn zu erzeugen, die die gewünschten Kriterien wie die Einhaltung der Schichtdicke oder die Reduktion der thermischen Varianz möglichst gut erfüllt. Zur Optimierung der Freiheitsgrade wird in diesem Schritt ein Evolutionärer Algorithmus genutzt, da dieser in der Lage ist, den extrem großen Suchraum bestmöglich abzusuchen und dadurch einen Bereich des Lösungsraums mit guten Lösungen zu finden.

Das auf die Initialbahnplanung folgende Modul ist die Bahnoptimierung (siehe Kapitel 5), die ausgehend von einer gegebenen Initialbahn diese bezüglich der Zielkriterien optimiert. Die Zielkriterien sind die Minimierung der Abweichungen der Schicht von der Zielschichtdicke sowie die Ausführbarkeit der Bahn durch einen Roboter. Um die Ziele zu erreichen, wird eine externe Simulation zur exakten Vorhersage von Schichtdicken genutzt sowie optional eine Simulation der Roboterbewegung. Die Optimierung geschieht in diesem Modul mittels eines nichtlinearen konjugierten Gradientenverfahrens und verändert die Bahn bezüglich aller zur Verfügung stehenden Bahnparameter wie Position, Spritzrichtung und Geschwindigkeit. Im Gegensatz zu der Optimierung von Parametern einer gegebenen Bahnstruktur, wie zum Beispiel dem Bahnabstand bei Mäanderbah-



nen, bietet diese Art der Optimierung mehr Möglichkeiten, die Bahn bezüglich der Optimierungsziele zu verbessern.

Im Bereich der Simulationskomponenten stehen eine Prozess- (Kapitel 3) und eine Robotersimulation (Abschnitt 2.3) zur Verfügung. Die Prozesssimulation befasst sich mit der Simulation des Auftrags und der Wärme, während sich die Robotersimulation mit der Kinematik und inversen Kinematik des Roboters beschäftigt.

Außerdem werden für die Bahnoptimierung einige externe Module verwendet. Hierbei handelt es sich um eine externe Simulation zur Schichtdickenvorhersage sowie um eventuell vorhandene Messdaten der Schichtdicke, die in der Optimierung genutzt werden können.

### 1.3.5 Abgrenzung zu verfügbarer Bahnplanungssoftware

Es existieren nicht besonders viele kommerzielle Softwaresysteme zur Offline-Bahnplanung für industrielle Einsatzzwecke, die thermische Spritzverfahren direkt unterstützen oder einfach auf diesen Prozess angepasst werden können. Die möglichen Bahnen, die von diesen Softwaresystemen generiert werden können, basieren häufig auf einfachen Bahnstrukturen, die nur auf geringe Weise an den eigentlichen Prozess angepasst werden können. Simulationsunterstützte Bahnoptimierung wird nach Kenntnis des Autors in keiner Software genutzt. Im Folgenden wird eine kurze Übersicht über einige wichtige Softwaresysteme sowie deren Eigenschaften gegeben.

Im Bereich der digitalen Fertigung werden umfangreiche Product Lifecycle Management Systeme (PLM) angeboten, in denen unter anderem Bahnplanungs- und Simulationskomponenten enthalten sind. Zwei dieser Systeme sind Robcad<sup>1</sup> und Delmia<sup>2</sup>, welche in Tecnomatix von Siemens PLM und in Delmia von Dassault Systèmes integriert sind. Die PLM Systeme unterstützen den gesamten Lebenszyklus von der Planung über die Fertigung bis hin zur Wartung. Kernkompetenzen in der Planungsphase, in der unter anderem die Bahnplanung und Simulation stattfinden, sind das Design von Roboterzellen und ganzer Fertigungsstraßen sowie die Simulation der Roboterbewegung unter Berücksichtigung von Dynamikeffekten. Die Simulationspakete großer PLM Softwaresysteme decken viele Prozesse wie Punkt- oder Bahnschweißen, Bohren, Nieten, Schneiden, Polieren oder Lackieren ab, häufig jedoch nicht das thermische Spritzen. Dieser Prozess kann in Delmia über die Drittanbietersoftware FastSurf<sup>3</sup> integriert werden. Die Simulationssoftware in Robcad Paint für Lackierprozesse bietet auch eine Unterstützung für thermisches Plasmaspritzen, obwohl dieser Prozess deutliche Unterschiede zum Lackieren beinhaltet. In beiden PLM Systemen dient die Simulation jedoch primär zur Visualisierung und Verifikation der Schichtdicken, nicht jedoch zur automatischen Anpassung der Bahnen, die von der Bahnplanung erzeugt wird. Die Bahnplanung selber wird in den PLM

<sup>1</sup>[http://www.plm.automation.siemens.com/de\\_de/products/tecnomatix/robotics\\_automation/robcad/](http://www.plm.automation.siemens.com/de_de/products/tecnomatix/robotics_automation/robcad/) (07.02.2017)

<sup>2</sup><https://www.3ds.com/de/produkte-und-services/delmia/> (07.02.2017)

<sup>3</sup>[http://www.cenit.com/de\\_DE/plm/digitale-fabrik/software/fastsurf.html](http://www.cenit.com/de_DE/plm/digitale-fabrik/software/fastsurf.html) (07.02.2017)



Systemen häufig durch parametrisierbare, jedoch recht einfache Bahnstrukturen wie Mäanderbahnen basierend auf den CAD Daten des Werkstücks erzeugt. Die Nutzung dieser einfachen Bahnstrukturen ist wesentlich unflexibler als die in dieser Arbeit verfolgte Lösung, die deutlich komplexere Bahnstrukturen zulässt und die Simulationsergebnisse in der Bahnplanung berücksichtigt.

Weitere Software zur Bahnplanung für Industrieroboter sind RobotWorks<sup>4</sup>, Eureka<sup>5</sup> und RobotMaster<sup>6</sup>. In diesen Softwareumgebungen werden (teil-)automatisierte oder interaktive Bahnplanungsmethoden angeboten. Die Möglichkeiten beschränken sich jedoch wieder auf die Parametrisierung der Bahnparameter wie den Bahnoffset bei Mäanderbahnen, aber es werden keine simulationsbezogenen Daten zur Bahnoptimierung genutzt.

Die Software 3D-ProSim<sup>7</sup> der Gfai ist zur Bahnplanung im Bereich der Keramikglasur geeignet. Die Software bietet interaktive Werkzeuge zum virtuellen Teachen der Roboterbahn. Dies wird durch den prozessbedingt recht großen Eingriffsbereich, der durch einen Sprühkegel mit sehr großem Öffnungswinkel entsteht, und der dadurch entstehenden kurzen Bahn ermöglicht. Für das thermische Spritzen, dessen Eingriffsbereich wesentlich kleiner ist und dadurch wesentlich längere Bahnen erfordert, ist dieses Verfahren jedoch nicht geeignet.

Das Thermal Spray Toolkit [48, 47, 27] bietet Methoden zur Offline-Generierung von Bahnen anhand der gegebenen Werkstückgeometrie an. Dabei liegt der Fokus auf der Berücksichtigung der Roboterkinematik, die dadurch verbessert wird, dass im Bereich großer Krümmungen der Anstellwinkel der Spritzpistole schon vorzeitig angepasst wird. Zur Simulation des Prozesses wird ein vereinfachtes Modell einer univariaten Gaußverteilung genutzt, so dass eine korrekte Vorhersage auf komplexen Werkstückoberflächen nicht erwartet werden kann.

Insgesamt bietet jede der vorgestellten Softwaresysteme eigene Vor- und Nachteile. Jedoch ist allen Systemen gemeinsam, dass sie einerseits nur herkömmliche Bahnstrukturen nutzen und andererseits, trotz zur Verfügung stehender Simulation des Prozesses, die Ergebnisse der Simulation nicht zur Optimierung der Bahn nutzen. Vor allem prozessspezifische Eigenschaften, wie die thermische Situation auf dem Werkstück beim thermischen Spritzen, bleibt unberücksichtigt. Die Nutzung von komplexen Bahnstrukturen und die Berücksichtigung von Simulationsergebnissen sind die Kernkomponenten der vorliegenden Arbeit und damit eine wesentliche Abgrenzung zu der verfügbaren Software.

---

<sup>4</sup><http://www.robotworks-eu.com/> (07.02.2017)

<sup>5</sup><http://www.roboris.it/en> (07.02.2017)

<sup>6</sup><http://www.robotmaster.com/> (07.02.2017)

<sup>7</sup><http://www.gfai.de/deutsch/projekte/bildverarbeitung-industrielle-anwendungen-projekte/3d-prosim-robotik.html> (07.02.2017)

## 1.4 Beiträge der Dissertation

Die wesentlichen neuen wissenschaftlichen Beiträge der Dissertation lassen sich wie folgt zusammenfassen:

### **Zweischrittige Bahnplanung**

Die Bahnplanung ist besonders für Werkstücke mit freigeformten Oberflächen ein aufwändiges Problem. Daher wird hier ein Ansatz präsentiert, der eine Vereinfachung der Bahnplanung ermöglicht. Dazu werden zunächst einigermaßen geeignete „Initialbahnen“ mit global günstiger Struktur erstellt, die schließlich in einem zweiten Schritt durch die Bahnoptimierung zu „finalen Bahnen“ angepasst werden. Für die Initialbahnplanung ergibt sich die Vereinfachung insbesondere durch

- die Verwendbarkeit von effizient berechenbaren Ersatzmodellen für den Fertigungsprozess, hier das thermische Spritzen, sowie für das Roboterverhalten. Diese Vereinfachung ermöglicht insbesondere den praktikablen Einsatz globaler Suchverfahren.
- die Nutzbarkeit von vorhandenen Bahnen für schon einmal beschichtete ähnliche Werkstücke als Initialbahnen.
- die Einsatzmöglichkeit von Bahnen als Initialbahnen, die mit einem nicht beschichtungsspezifischen Bahnplanungsverfahren erstellt wurden.

Die mit der Vereinfachung üblicherweise verbundenen Defizite einer nur approximativen Lösung des Problems sollen mit Hilfe der Bahnoptimierung behoben werden. Die Bahnoptimierung wird wiederum durch das Vorhandensein einer geeigneten Startlösung für lokale Suchverfahren in Form der Initialbahn vereinfacht.

### **Initialbahnplanung mit Punkt-zu-Punkt Bahnen**

In der Initialbahnplanung wird eine neue Bahnstruktur präsentiert, die sich durch eine wesentlich höhere Flexibilität als die konventionellen Bahnstrukturen, wie etwa mäanderförmige Bahnen, auszeichnet. Auf diese Weise werden Krümmungseigenschaften der Werkstückoberfläche sowie die Anforderungen des werkzeugführenden Roboters für komplexe Freiformoberflächen besser berücksichtigt. Die Verwendung konventioneller Bahnstrukturen auch auf komplexen Werkstücken in der Praxis des thermischen Spritzens ist primär durch deren Verfügbarkeit in kommerziellen, manuellen oder semi-automatischen Bahnplanungswerkzeugen bedingt, die meist für andere Prozesse, wie das Fräsen, entwickelt wurden. Ansätze zu allgemeineren Bahnstrukturen existieren bereits in Form der Billard-Bahnen [95] sowie deren Verallgemeinerung zu Rand-zu-Rand Bahnen [82, 84]. Die Punkt-zu-Punkt Bahnen beheben das Defizit der nur globalen Kontrollierbarkeit des Bahninneren bei Rand-zu-Rand Bahnen, das bei stärker gekrümmten Flächen problematisch ist.

Die oben genannten, allgemeinen Bahnstrukturen schließen in der Regel die manuelle oder analytische Lösung des Bahnplanungsproblems aus (eine Ausnahme sind die Billardbahnen für den sehr speziellen Fall von ebenen, rechteckigen Gebieten, die analytisch berechnet werden können). Für Punkt-zu-Punkt Bahnen erfolgt die Bahnplanung durch das Lösen eines mehrkriteriellen Optimierungsproblems, das als Zielkriterien die Minimierung des Auftragsfehlers, der thermischen Varianzen, der Roboterachsbeschleunigungen und des Oversprays, das heißt das Spritzen außerhalb der Werkstückoberfläche, verfolgt. Die Optimierung geschieht durch einen evolutionären Algorithmus. Dieser zeichnet sich durch verschiedene Maßnahmen zur Effizienzsteigerung aus, vor allem durch

- die Verwendung heuristischer Ersatzmodelle für die Auftrags- sowie die thermische Simulation. Besonders hervorzuheben ist hierbei das thermische Modell, das sich an Modellen der Literatur orientiert, diese jedoch erweitert und unter anderem folgende Besonderheiten aufweist:
  - Die Berücksichtigung von häufig nicht betrachteten physikalischen Prozessen wie die Konvektion, Diffusion und Strahlung.
  - Die besonders schnelle Berechnung der thermischen Varianzen ohne zu großen Qualitätsverlust gegenüber physikalisch genaueren Berechnungsmethoden.
- die Verwendung von Operatoren und Parametern, die mittels Metaparameteroptimierung so bestimmt werden, dass der Gesamtprozess eine besonders hohe Konvergenz zeigt. Bei Versuchen auf Werkstück A sind die *Desirability Werte*, welche als Zielfunktion verwendet werden (siehe Abschnitt 4.9.1), bei Nutzung optimierter Parameter im Mittel um ca. 25 % besser als bei Nutzung manuell ausgewählter, plausibler Werte. Dies liegt insbesondere daran, dass die manuelle Auswahl von insgesamt 16 zum Teil voneinander abhängigen Parametern sehr schwierig ist, so dass die automatische Bestimmung guter Parameter besonders erstrebenswert ist.
- die Nutzung eines parallelen Master-Slave Multi-Threading Ansatzes zur Ausnutzung der zur Verfügung stehenden Rechenleistung. Hierdurch lässt sich eine nahezu lineare Reduktion der Laufzeit bezüglich der Anzahl der genutzten Prozessorkerne erreichen, da die parallele Erzeugung und Evaluation neuer Individuen unabhängig von anderen Individuen ablaufen kann.

#### **Effiziente simulationsbasierte Optimierung für die Bahnoptimierung**

Die Bahnoptimierung beruht auf einer simulationsbasierten Optimierung einer einkriteriellen Zielfunktion, die sich aus der gewichteten Summe mehrerer Einzelkriterien ergibt. Zur Bestimmung der Fehlerwerte wird auf eine genaue, jedoch aufwändige Beschichtungssimulation zurückgegriffen [169]. Die Lösung des Optimierungsproblems geschieht durch die Methode der nichtlinearen konjugierten Gradienten. Besonders hervorzuheben ist die effiziente Realisierung der Optimierung durch eine Verminderung der Anzahl der Simulationsaufrufe. Dies wird durch die näherungsweise Gradientenbestimmung mit

Hilfe eines analytisch differenzierbaren Ersatzmodells für den Schichtauftrag erreicht. Dadurch kann der Aufwand zur numerischen Ermittlung der Gradienten durch vielfaches Aufrufen der zeitaufwändigen, genauen Simulation umgangen werden.

### **Berücksichtigung der Roboterdynamik durch iterierte Bahnprojektion**

Die Nutzbarkeit der Bahnen in der Praxis wird durch die explizite Berücksichtigung der Roboterdynamik sichergestellt. Hierzu wird ein iteratives Verfahren präsentiert, das die Bahnoptimierung mit einem Verfahren zur Bahnprojektion, d. h. der Ersetzung von nicht abfahrbaren Bahnen durch abfahrbare, also für den Roboter zulässige Bahnen, kombiniert. Als Grundlage der Bahnprojektion dient ein existierendes Simulationssystem für Industrieroboter, das die Dynamik des Roboters berücksichtigt. Die Vorgehensweise besteht darin, eine bezüglich der Beschichtung optimierte, möglicherweise jedoch nicht zulässige Bahn zunächst durch Projektion in eine dynamisch zulässige Bahn umzuwandeln. Die resultierende Bahn weist eventuell ein verschlechtertes Beschichtungsergebnis auf. Sollte die Qualität der Beschichtung nicht ausreichen, wird die Bahn erneut bezüglich der Beschichtung optimiert und das Verfahren iteriert.

### **Experimenteller Nachweis der prinzipiellen Nutzbarkeit**

In den Evaluationen wird das gesamte Verfahren der Bahnplanung bezüglich der Praktikabilität analysiert. Hierzu werden durchgehend zwei verschiedene Werkstücke mit unterschiedlich komplexen Oberflächen genutzt, wobei eines der Werkstücke auf Grund der sehr ausgeprägten Krümmungen besonders herausfordernd ist („Werkstück B“, siehe Abbildung 4.13). Die Analysen zeigen insbesondere

- die hohe Qualität der dynamisch zulässigen Bahnen nach der Bahnplanung verglichen mit herkömmlich erzeugten Mäanderbahnen auf komplexen Werkstücken. So liegen z. B. der mittlere und maximale relative Auftragsfehler (Gleichungen (4.3) und (4.10)) einer herkömmlich generierten Mäanderbahnen auf Werkstück B bei ca. 0,124 und 3,257, während die entsprechenden Werte einer dynamisch zulässigen, optimierten Punkt-zu-Punkt Bahn ca. 0,001 und 0,068 betragen. Die maximale thermische Varianz nach Gleichung (4.11) liegt für die Mäanderbahn bei 1,02 und für die Punkt-zu-Punkt Bahn bei 0,14.
- für die Praxis akzeptable Rechenzeiten von wenigen Stunden ( $\leq 5$  h) für den gesamten Prozess der Bahnplanung auf durchschnittlich großen Werkstücken (Werkstück B, Größe ca.  $340 \times 160 \times 130$  mm, Region of Interest: ca. 750 Knoten grob aufgelöst und ca. 5050 Knoten feinaufgelöst) mit einem herkömmlichen PC. Die Rechenzeit setzt sich hierbei wie folgt zusammen, wobei relativ viele Iterationen durchgeführt wurden:
  - Initialbahnplanung: 20–30 min (20000 Iterationen, Libo-Footprint, 6 Threads, Zielschichtdicke: 0,5 mm, grob aufgelöstes Werkstück)
  - Bahnoptimierung: 20–30 min (100 Iterationen, feinaufgelöstes Werkstück)
  - Dynamik: ca. 1,5 min (ca. 8860 Zielposen)

- Gesamt: ca. 290 min (1× Initialbahnplanung + 10× Bahnoptimierung + 10× Dynamik)

## 1.5 Eigene Publikationen

Die in dieser Arbeit vorgestellten Methoden sind zu großen Teilen in verschiedenen Veröffentlichungen präsentiert worden. Im Folgenden werden diese kurz zusammengefasst.

Ein im Tagungsband der *gecco 2013* erschienener Beitrag mit Titel „Evolutionary Path Generation for Reduction of Thermal Variations in Thermal Spray Coating“ [84] beinhaltet die Arbeiten zu einer frühen Version der Initialbahnplanung basierend auf der Masterarbeit „Heuristische Wegfindung zur Reduktion thermischer Varianz beim roboterbasierten thermischen Spritzen“ [82]. Die in der Masterarbeit entwickelte Initialbahnplanung nutzt auf planare Oberflächen beschränkte Rand-zu-Rand Bahnstrukturen, ein wesentlich vereinfachtes Auftrags- sowie thermisches Modell und einen weniger angepassten Evolutionären Algorithmus, so dass die hier entwickelten Punkt-zu-Punkt Bahnstrukturen, die Simulationsmodelle sowie die wesentlichen Komponenten des Evolutionären Algorithmus neue Beiträge darstellen. Eine weitere Veröffentlichung hierzu ist in *5. Öffentliches Kolloquium - SFB 708* mit dem Titel „Heuristische Wegfindung zur Reduktion thermischer Varianz beim roboterbasierten thermischen Spritzen“ [83] erschienen. Die Beiträge des Koautors und Betreuers der Arbeit Prof. Dr. Heinrich Müller in den genannten Veröffentlichungen besteht in der technischen Diskussion zu Einzelaspekten sowie Hinweisen zur Präsentation der Ergebnisse.

Der Artikel „Simulation based iterative post-optimization of paths of robot guided thermal spraying“ [86] ist im Journal *Robotics and Computer-Integrated Manufacturing* erschienen. In dieser Arbeit sind wesentliche Teile der Bahnoptimierung (Kapitel 5) beschrieben. Der Koautor Thomas Wiederkehr hat die Abschnitte zur Auftragssimulation beigetragen und Prof. Dr. Heinrich Müller stand mit technischen Diskussionen und Ratschlägen zur Seite.

Die Bahnoptimierung ist in den Artikeln „Optimization of the process chain for the efficient manufacturing of forming tools with complex 3D surfaces“ (*6. Öffentliches Kolloquium - SFB 708*, [166]), „Adaption of the path strategy to produce near net shape coatings manufactured by HVOF spraying“ (*Proceedings of International Thermal Spray Conference & Exposition 2013*, [155]) und „Fast Coating Deposition Simulation for Path Planning and Iterative Net-Shape Optimization on Complex Workpieces“ (*International Thermal Spray Conference and Exposition (ITSC)*, [172]) verwendet worden. Der Autor dieser Arbeit hat optimierte Bahnen zur Verfügung gestellt, die mit den Methoden dieser Arbeit erstellt wurden.

Des Weiteren ist ein Konzept zur Integration der Bahnoptimierung in ein komplexeres Framework in „A Simulation Based Path Optimization System for Robot Guided Thermal Spray“ (*6. Öffentliches Kolloquium - SFB 708*, [85]) veröffentlicht worden, wobei der Koautor Prof. Dr. Heinrich Müller, wie in den oben genannten Veröffentlichungen,

beratend zur Seite stand. In dieser sowie der oben genannten Veröffentlichung [155] wird auch das in Kapitel 6 präsentierte Konzept der Integration der Roboterdynamik in das Gesamtsystem vorgestellt.

## 1.6 Struktur der Dissertation

Die vorliegende Dissertation ist wie folgt gegliedert. In Kapitel 2 werden einige grundlegende Begriffe und Konzepte vorgestellt. Hierzu gehören insbesondere die Begriffe *Bahn*, *Werkstück* und *Roboter*. Anschließend wird in Kapitel 3 die hier verwendete Simulation vorgestellt. Diese ist in der Lage, die Schichtbildung und die Temperaturen auf dem Werkstück zu simulieren. In Kapitel 4 wird die erste Methode, die Initialbahnplanung, vorgestellt und in Kapitel 5 wird die Bahnoptimierung beschrieben. Diese beiden Kapitel stellen den methodischen Schwerpunkt der Arbeit dar. In Kapitel 6 wird auf die Nutzung der in dieser Arbeit erstellten Bahnen in der Praxis eingegangen. Kapitel 7 schließt die Arbeit mit einer Zusammenfassung und einem Ausblick auf weitere Forschungsmöglichkeiten ab.

## 1.7 Danksagung

Teile dieser Dissertation sind im Rahmen des Sonderforschungsbereichs 708 „3D-Surface Engineering für Werkzeugsysteme der Blechformteilefertigung – Erzeugung, Modellierung, Bearbeitung“ entstanden. Der Autor dankt der Deutschen Forschungsgemeinschaft für die finanzielle Förderung.

# Zentrale Begriffe

Dieses Kapitel stellt die grundlegenden Begriffe und Datenstrukturen vor, die in den Algorithmen der nachfolgenden Kapitel übergreifend verwendet werden. Hierzu zählt als Erstes der Begriff *Bahn*, auf den in Abschnitt 2.1 eingegangen wird. Hier werden die verschiedenen Bahntypen vorgestellt sowie die unterschiedlichen Möglichkeiten der Bahnbeschreibung. Anschließend wird in Abschnitt 2.2 auf die Repräsentationsmöglichkeiten von *Werkstücken* mit Hilfe von Dreiecksnetzen eingegangen, die in dieser Arbeit Verwendung finden. Abschnitt 2.3 beschreibt den Aufbau von *Industrierobotern* und stellt kurz die kinematischen Eigenschaften vor, die beim Bahnplanungsprozess beachtet werden sollten, um die Bahn auf einem realen Roboter nutzen zu können.

## 2.1 Bahnen

Der zentrale Begriff der Bahnplanung für industrielle Einsatzzwecke ist die Trajektorie, im folgenden kurz „Bahn“, welche den zeitlichen und geometrischen Verlauf des Werkzeugs bzw. des Roboters im Raum beschreibt.

In dieser Arbeit wird zwischen der *Eingriffsbahn*, der *TCP-Werkzeughahn*, der *Werkzeughahn* und der *Roboterbahn* unterschieden, siehe auch [103]. Die TCP-Werkzeughahn beschreibt eine Kurve im Raum, die von dem Tool Center Point (TCP) des Werkzeugs abgefahren wird. Der TCP beschreibt dabei einen eindeutigen Punkt des Werkzeugs, der meist in einem logischen Zusammenhang mit dem Eingriff an dem Werkstück steht. Da jedoch der Eingriff des Werkzeugs bei vielen Prozessen nicht punktförmig ist, gibt es einigen Spielraum bei der Auswahl eines geeigneten Punkts als TCP. Beim Fräsen kann dieser Punkt zum Beispiel im Mittelpunkt des Kugelfräskopfes [146] liegen, auch wenn der Eingriff tatsächlich am Rand des Fräskopfes stattfindet. Beim thermischen Spritzen wird der TCP normalerweise in einer vorgegebenen, meist idealen Spritzentfernung vor der Spitze der Spritzpistole entlang der zentralen Spritzachse angegeben [47]. Der eigentliche Eingriff geschieht dann in einem relativ großen Bereich rund um diesen Punkt.

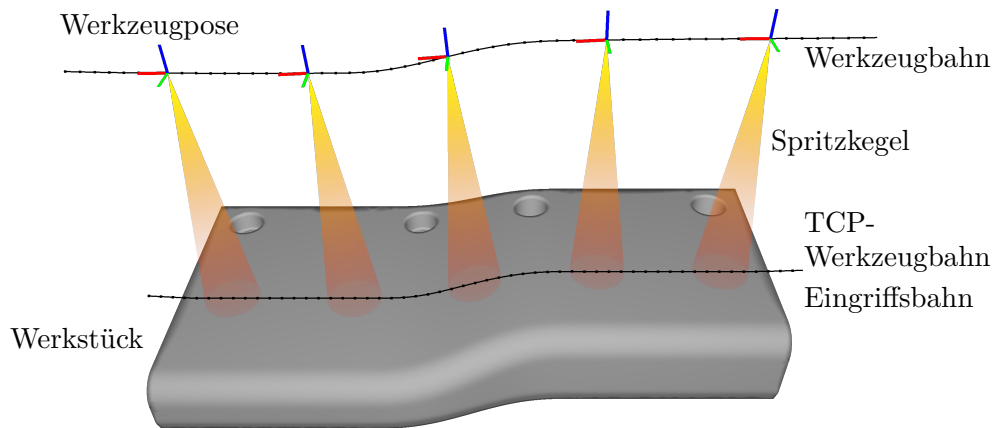


Abbildung 2.1: Visualisierung von verschiedenen Bahntypen. Die über einen Punkt an der Spitze der Spritzpistole definierte Werkzeughahn verläuft über dem Werkstück. Einige Werkzeugposen, die die Ausrichtung der Spritzpistole bestimmen, sowie die dort anliegenden Spritzkegel sind ebenfalls eingezeichnet. Die TCP-Werkzeughahn und die Eingriffsbahn (hier identisch) liegen idealerweise genau auf der Oberfläche des Werkstücks.

Alternativ kann auch ein Punkt an der Spitze der Spritzpistole genutzt werden, über den die Werkzeughahn definiert wird.

Die Eingriffsbahn beschreibt ebenso wie die Werkzeughahn eine Kurve im Raum. Diese Kurve soll den tatsächlichen Eingriff an dem Werkstück verdeutlichen. Beim Fräsen könnte dazu ein Kontaktpunkt des Fräskopfes mit dem Werkstück dienen. Beim thermischen Spritzen wird eine Bahn auf der Oberfläche bestimmt, die durch den Schnitt der zentralen Spritzachse mit dem Werkstück entsteht. In Abbildung 2.1 sind die beiden Bahntypen für den Fall des thermischen Spritzens veranschaulicht. Hier sind die Eingriffsbahn und die TCP-Werkzeughahn identisch und liegen direkt auf der Oberfläche des Werkstücks, während die Werkzeughahn, die über einen Punkt an der Spitze der Spritzpistole definiert wird, über dem Werkstück liegt.

Die Roboterbahn beschreibt genau wie die anderen beiden Bahntypen den Verlauf des Werkzeugs bzw. des TCPs. Die Repräsentation basiert allerdings nicht auf der Angabe der geometrischen Lage des Werkzeugs, sondern auf der Pose des Roboters, der das Werkzeug führt, siehe auch Abschnitt 2.3. Die Pose wird dabei durch die Position und Ausrichtung der Roboterbasis sowie die aktuelle Stellung der zur Verfügung stehenden Achsen in Gelenkkoordinaten bestimmt. Diese Repräsentation ist unter anderem dann von Bedeutung, wenn die Bahn auf Ausführbarkeit getestet werden soll, da der Roboter kinematischen und dynamischen Grenzen unterliegt, deren Einhaltung allein durch Betrachtung der anderen Bahnrepräsentationen nicht ersichtlich ist.

Im Grunde sind die oben genannten Bahnbeschreibungen (bis auf Mehrdeutigkeiten der Achsstellungen des Roboters) äquivalent, da der geometrische Zusammenhang zwi-



schen dem Werkzeug, dem TCP und dem Roboter bekannt ist. Die Wahl der Repräsentationsform sollte daher durch den Anwendungszweck bestimmt werden. Beispielsweise basieren viele Bahngenerierungsalgorithmen darauf, parallele oder konturparallele Linien auf der Oberfläche zu erzeugen und diese zu einer Bahn zu verbinden [27, 28, 29, 59, 66, 103, 100, 164]. Diese Bahn entspricht dann meist direkt der Eingriffsbahn. Auf der anderen Seite kann es sinnvoll sein, die Werkzeugbahn zu modifizieren, um die Bahn zu optimieren, wie es beispielsweise in [86] getan wird. Dies ist dann sinnvoll, wenn der geometrische Zusammenhang zwischen der Werkstückoberfläche und der Werkzeugbahn zur Optimierung genutzt werden kann. Die Roboterbahn wird benötigt, wenn die Bahn mit einem Roboter ausgeführt werden soll. In diesem Fall wird die Roboterbahn zur Programmierung direkt genutzt oder, falls die Bahn zum Beispiel als Werkzeugbahn vorliegt, mit Hilfe der inversen Kinematik in eine Roboterbahn konvertiert. Die Nutzung der Roboterbahn kann allerdings auch für die Analyse der Ausführbarkeit oder Güte der Bahn genutzt werden.

Für die Repräsentation und Speicherung einer im Allgemeinen kontinuierlichen Bahn mit einem Rechner können verschiedene Repräsentationsformen genutzt werden. Zwei Möglichkeiten hierfür sind die Repräsentation mittels einer Menge diskreter Abtaststellen oder auf Basis von Spline Funktionen. Vorteile von Spline-Repräsentationen sind die in der Regel geringe Zahl an Parametern, um eine Bahn zu beschreiben, sowie der häufig durch diese Repräsentation entstehende glatte Bahnverlauf. Ein Nachteil ist jedoch die indirekte Steuerung der Bahn bei einer Bahnmodifikation, die nur über die Stützstellen der Spline-Repräsentation ausgeführt werden kann. Für die Algorithmen in dieser Arbeit wird daher auf die Repräsentation mittels Abtaststellen zurückgegriffen. Die Bahn besteht bei dieser Repräsentation aus einer Menge von indizierten Punkten, an denen die Bahnparameter wie z. B. Position, Ausrichtung und Geschwindigkeit gespeichert werden. Um die kontinuierliche Bahn aus den Punkten zu rekonstruieren, wird zwischen zwei aufeinanderfolgenden Bahnpunkten linear interpoliert. Die Güte der Diskretisierung wird hauptsächlich durch die Dichte der Abtastpunkte bestimmt. Bei zu geringer Abtastrate kann die ursprüngliche Bahn durch die Anwendung der Interpolation stark verfälscht werden. Ein Nachteil bei zu vielen Abtaststellen ist jedoch der steigende Rechenzeitbedarf bei vielen auf dieser Repräsentation basierenden Algorithmen, so dass eine geeignete Abtastung gewählt werden muss.

Formal kann eine Bahn  $\mathcal{X} = (\chi_0, \dots, \chi_n)^\top$  wie oben beschrieben durch eine Sequenz von Bahnpunkten  $\chi_j = (\rho_j, \hat{t}_j)^\top$ ,  $j = 0, \dots, n$ , dargestellt werden. Die Werkzeugpose  $\rho_j$  entspricht dem Ort und der Orientierung des Werkzeugs und kann auf zwei unterschiedliche Arten ausgedrückt werden. Die direkte Darstellung der Pose verwendet je einen Orts- und einen Orientierungsvektor, also  $\rho_j = (\mathbf{x}_j, \phi_j)^\top$ . Hierbei entspricht  $\mathbf{x}_j = (x_j, y_j, z_j)^\top$  der Position des Bahnpunkts bezüglich des Weltkoordinatensystems und  $\phi_j = (\phi_{j,1}, \phi_{j,2}, \phi_{j,3})^\top$  beschreibt die Orientierung des Werkzeugs mit Hilfe dreier Winkel, die für drei hintereinander ausgeführte Rotationen um die Weltkoordinatenachsen  $\mathbf{z}$ ,  $\mathbf{x}$  und  $\mathbf{z}$  stehen. Die alternative Darstellung der Pose wird über die Achsstellungen eines Roboters mit  $\rho_j = \Phi_j$  angegeben. Der Vektor  $\Phi_j = (\Phi_{j,1}, \Phi_{j,2}, \Phi_{j,3}, \Phi_{j,4}, \Phi_{j,5}, \Phi_{j,6})^\top$

beschreibt die sechs Achswinkel eines sechsachsigen Roboters. Für diese Repräsentation muss zusätzlich die Initialtransformation  $\mathbf{T}^r$  des Roboters gegeben sein, die die Position und Ausrichtung der Roboterbasis beschreibt. Über den Parameter  $\hat{t}_j$  wird die Zeit beschrieben, die das Werkzeug benötigt, um die Hälfte des zu dem Bahnpunkt hinführenden sowie von dem Bahnpunkt wegführenden Bahnabschnittes zu absolvieren. Diese Zeit wird als „Verweildauer“ an dem repräsentierenden Bahnpunkt bezeichnet. Auf diese Weise kann ein Bahnpunkt in der gegebenen Repräsentationsform als Repräsentant des oben beschriebenen Bahnabschnittes stehen, analog zum Konzept des Massenschwerpunkts, der die Masse eines Körpers in einem Punkt konzentriert. Mit Hilfe dieser Daten kann auch die Geschwindigkeit des Roboters auf dem Bahnabschnitt bestimmt werden. Im Falle einer gegebenen Geschwindigkeit kann diese mit Hilfe der Länge des Bahnabschnittes, die der Bahnpunkt repräsentiert, in eine entsprechende Zeit umgerechnet werden.

Häufig wird nicht die Verweildauer an einem Bahnpunkt sondern der Zeitpunkt  $\tau_j$ , an dem der Bahnpunkt erreicht wird, angegeben. Aus diesem lässt sich ebenfalls die Verweildauer bestimmen zu  $\hat{t} = (t_{j-1} + t_j) / 2$  wobei  $t_j = \tau_{j+1} - \tau_j$  gilt.  $t_j$  beschreibt demnach die benötigte Zeit, um das Werkzeug von dem Bahnpunkt  $\chi_j$  zu dem Bahnpunkt  $\chi_{j+1}$  zu bewegen.

## 2.2 Werkstücke

Die digitale Modellierung und Speicherung von 3D-Modellen für Werkstücke geschieht meistens mittels NURBS-Flächen oder mit Hilfe von Dreiecksnetzen. Ein Vorteil bei der Nutzung von NURBS-Flächen ist die einfache und exakte Modellierung von Freiformflächen mit Hilfe von Flächen- und Kurvendefinitionen. Dreiecksnetze hingegen bieten den Vorteil effizienter Algorithmen im Bereich der Computergrafik. Ein Dreiecksnetz besteht aus einer Menge von Knoten (Vertices)  $\mathbf{v}_i, i = 0, \dots, m$ , einer Menge von Kanten  $e_i, i = 0, \dots, m_e$ , sowie einer Menge von Facetten  $f_i, i = 0, \dots, m_f$ . Jeder Knoten ist Endpunkt von mindestens zwei Kanten und eine Kante verbindet je zwei Knoten. Eine Facette wird von drei Eckpunkten und den drei Kanten, die diese Punkte untereinander verbinden, gebildet. Jede Kante hat entweder ein oder zwei benachbarte Dreiecke. Kanten mit nur einem benachbarten Dreieck bilden den Rand des Netzes. In dieser Arbeit werden nur Netze betrachtet, die eine 2-dimensionale Mannigfaltigkeit bilden. Dies bedeutet unter anderem, dass das Netz lokal euklidisch ist, also jeder Punkt eine Umgebung besitzt, die homöomorph zu einer offenen Teilmenge des  $\mathbb{R}^2$  ist [106]. Besondere Einschränkungen an die Form des Werkstücks werden nicht gestellt. Das zu beschichtende Gebiet auf der Oberfläche muss jedoch zusammenhängend sein. Des Weiteren können Löcher in der Fläche zu unintuitiven Ergebnissen führen, da die Punkt-zu-Punkt Bahnen um diese herumführen. Daher können Löcher bei Bedarf durch Vorverarbeitungsschritte gefüllt werden. Auch sehr stark ausgefranste Ränder können zu weniger glatt wirkenden Bahnen führen, stellen jedoch kein technisches Hindernis dar, da die Algorithmen nur erfordern, dass die geodätisch kürzesten Verbindungen zwischen Punkten auf der Oberfläche gene-

riert werden können. Komplex geformte Oberflächen, die Hinterschnitte bilden, können unter Umständen nicht kollisionsfrei beschichtet werden. Ein Test auf Kollisionsfreiheit wird im Rahmen dieser Arbeit nicht geleistet, stellt aber einen wesentlichen Aspekt bei der Umsetzung der Bahn in der Praxis dar. Effiziente Algorithmen zum Konvertieren von NURBS-Flächen in Dreiecksnetze können zum Beispiel in [26] gefunden werden.

Eine Normale  $\mathbf{n}$  einer Fläche ist ein Vektor, der senkrecht auf der Oberfläche steht und auf die Länge 1 normiert ist. Auf Dreiecksnetzen wird häufig zwischen Facetten- und Knotennormalen unterschieden. Die Facettennormalen werden dabei pro Dreieck angegeben und sind bis auf ihre Orientierung eindeutig. Es ist zweckmäßig, dass die Facettennormalen einheitlich orientiert werden, so dass sie zum Beispiel nach *außen* zeigen. Bei den Knotennormalen ergibt sich das Problem, dass diese allein aus der Geometrie des Netzes nicht eindeutig bestimmbar sind. Für den Fall, dass das Dreiecksnetz aus NURBS-Flächen oder einer anderen Quelle erstellt worden ist, sollten daher die Normalen direkt aus diesem Modell genutzt werden. Wenn kein solches Modell vorhanden ist, müssen die Normalen direkt aus dem Netz berechnet werden. In dieser Arbeit werden die Knotennormalen aus dem gewichteten Mittelwert aller Normalen der zu diesem Knoten benachbarten Facetten bestimmt. Als Gewichte dienen die an dem Punkt adjazenten Innenwinkel der anliegenden Dreiecke. Zusätzlich werden die Normalen mit Hilfe der Normalen der Nachbarknoten geglättet. Dazu werden die Normalen der Nachbarn abstandsgewichtet auf die Knotennormale addiert und der so bestimmte Vektor anschließend wieder normiert. Dies führt zu weniger Ausreißern in den Normalen, die bei schlechter Vernetzung mit unregelmäßiger Struktur und ungleichmäßigen Dreiecken vorkommen.

Ein Vorteil bei der Nutzung eines Dreiecksnetzes ist die einfache Speicherung von weiteren Parametern der Oberfläche. So können die Parameter direkt an den Knoten des Netzes gespeichert werden. So werden für das Beschichten beim thermischen Spritzen Werte wie die Schichtdicke  $z_i$ , die Zielschichtdicke  $Z_i$  oder auch die Temperatur  $\Theta_{i,\tau}$  an jedem Knoten gesichert. Diese Art der Speicherung der Daten ist für die Algorithmen der Bahnplanung besonders günstig, da zum Beispiel die Information, wie viel Materialvolumen auf einer Facette des Netzes abgelagert wird, für die Planung oder Optimierung der Bahn weniger hilfreich ist als der Fehler in der Schichtdicke an den Knoten des Netzes.

Bei der Nutzung von Dreiecksnetzen ist die Dichte und Verteilung der Knoten von besonderem Interesse. Je dichter die Knoten liegen, desto exakter wird eine gegebene Freiformfläche von dem Dreiecksnetz approximiert und desto exakter können entsprechende Berechnungen durchgeführt werden. Genauso hat die Verteilung der Knoten und die Vernetzung Einfluss auf die Güte des Netzes. Die Wahl eines zu grob aufgelösten Netzes birgt die Gefahr großer Diskretisierungsfehler. Auf der anderen Seite erhöht sich der Berechnungsaufwand der Algorithmen mit der Anzahl der Knoten. Eine besondere Herausforderung ist daher die Wahl einer geeigneten Auflösungsstufe, die die Diskretisierungsfehler ebenso wie die Laufzeit gering hält.

Für einige Algorithmen wird der Flächeninhalt benötigt, der von einem Knoten repräsentiert wird. Die Bestimmung dieses Flächeninhalts kann mit Hilfe des Voronoi-Gebiets

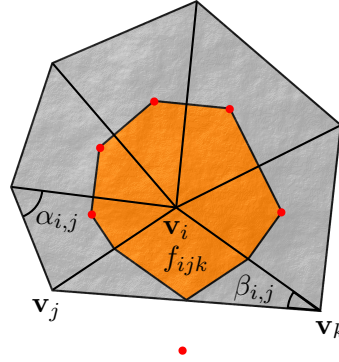


Abbildung 2.2: Darstellung der einem Knoten  $\mathbf{v}_i$  zugeordneten Fläche, die mit Hilfe des hybriden Voronoi-Verfahrens bestimmt wurde. In Orange ist die dem Knoten  $\mathbf{v}_i$  zugehörige Fläche dargestellt und in Rot die für die Berechnung wichtigen Mittelpunkte der Dreiecksumkreise.

um den Knoten erfolgen. Zu dem Voronoi-Gebiet eines Knotens zählen alle Punkte der Oberfläche, die bezüglich ihrer Distanz auf der Oberfläche näher an dem Knoten liegen als zu allen anderen. Eine Herleitung zur Bestimmung der Voronoi-Fläche auf Dreiecksnetzen ist in [122] und [81] zu finden. Die Fläche eines zu  $\mathbf{v}_i$  adjazenten Dreiecks  $f_{ijk}$ , die zum Voronoi-Gebiet von  $\mathbf{v}_i$  gehört, kann mit

$$A_{\text{Voronoi}, f_{ijk}} = \frac{1}{8} \left( \|\mathbf{v}_i - \mathbf{v}_j\|^2 \cot \angle f_{ijk}^k + \|\mathbf{v}_i - \mathbf{v}_k\|^2 \cot \angle f_{ijk}^j \right) \quad (2.1)$$

berechnet werden, wobei die Knoten  $\mathbf{v}_i$ ,  $\mathbf{v}_j$  und  $\mathbf{v}_k$  die Knoten des Dreiecks darstellen und  $\angle f_{ijk}^i$  den Winkel am Knoten mit Index  $i$  in diesem Dreieck. Daraus ergibt sich als Gesamtfläche für einen Knoten  $\mathbf{v}_i$

$$A_{\text{Voronoi}, \mathbf{v}_i} = \frac{1}{8} \sum_{l \in N_1(i)} (\cot \alpha_{i,l} + \cot \beta_{i,l}) \|\mathbf{v}_i - \mathbf{v}_l\|^2, \quad (2.2)$$

wobei  $N_1(i)$  dem 1-Ring um den  $i$ -te Knoten und  $\alpha_{i,l}$  und  $\beta_{i,l}$  den Winkeln gegenüber der Kante  $\overline{\mathbf{v}_i \mathbf{v}_l}$  entsprechen, siehe auch Abbildung 2.2.

Stumpfwinklige Dreiecke, das heißt Dreiecke, bei denen einer der drei Innenwinkel größer ist als  $90^\circ$ , werden gesondert betrachtet, so dass sich der zum  $i$ -ten Knoten gehörende Flächenanteil eines adjazenten Dreiecks wie folgt bestimmen lässt:

$$A_{\text{Mixed}, f_{ijk}} = \begin{cases} A_{\text{Voronoi}, f_{ijk}}, & \text{wenn } \angle f_{ijk}^i < 90 \wedge \angle f_{ijk}^j < 90 \wedge \angle f_{ijk}^k < 90, \\ A_{f_{ijk}}/4, & \text{wenn } \angle f_{ijk}^j \geq 90 \vee \angle f_{ijk}^k \geq 90, \\ A_{f_{ijk}}/2, & \text{wenn } \angle f_{ijk}^i \geq 90, \end{cases} \quad (2.3)$$

wobei  $f_{ijk}$  das Dreieck und  $A_{f_{ijk}}$  die Gesamtfläche des Dreiecks darstellt.

Somit ergibt sich für die gesamte hybride Voronoi-Fläche eines Knotens  $\mathbf{v}_i$

$$A_i = \sum_{f_{ijk}} A_{\text{Mixed}, f_{ijk}}, \quad (2.4)$$

wobei die Summe über die zu  $\mathbf{v}_i$  adjazenten Dreiecke läuft.

Geometrisch kann die dem Knoten zugeordnete Fläche bestimmt werden, indem für jedes Dreieck der Umkreismittelpunkt berechnet wird und dieser mit den Mittelpunkten der beiden Kanten, die von dem Knoten ausgehen und zu dem Dreieck adjazent sind, verbunden werden. In Abbildung 2.2 ist die Fläche  $A_{\text{mixed}}$  für den Knoten mit Index  $i$  in Orange dargestellt. Die Mittelpunkte der Umkreise der Dreiecke sind in Rot eingezeichnet. Für das untere Dreieck liegt der Umkreismittelpunkt außerhalb des Dreiecks. Das Dreieck ist somit stumpfwinkelig. Anstelle des Umkreismittelpunkts wird in diesem Fall der Mittelpunkt der gegenüber vom Knoten  $\mathbf{v}_i$  liegenden Dreiecksseite zur Konstruktion des dem Knoten zugeordneten Gebiets genutzt.

## 2.3 Industrieroboter

Die Verwendung von Robotern bei industriellen Prozessen bringt verschiedene Vorteile gegenüber der manuellen Fertigung mit sich [47]. Einige dieser Vorteile sind zum Beispiel, dass Roboter ohne Unterbrechung immer wiederkehrende Arbeitsschritte in gleichbleibender Qualität durchführen können. Außerdem sind Roboter dazu geeignet, in für Menschen gefährlichen Umgebungen zu arbeiten oder für Menschen gefährliche Arbeitsschritte durchzuführen. Gerade im Bereich des thermischen Spritzens können so die Gefahren für Menschen reduziert werden. Eine der größten Schwierigkeiten bei der Nutzung von Robotern ergibt sich durch die Notwendigkeit der Programmierung, die je nach Aufgabe ein sehr schwieriger und langwieriger Prozess ist. Des Weiteren eignen sich Roboter hauptsächlich für die Ausführung von exakt gleichbleibenden Arbeitsschritten. Die Flexibilität eines Menschen, auf unvorhergesehene Situationen zu reagieren, kann von einem Roboter bislang nicht erbracht werden.

### 2.3.1 Aufbau

Im Rahmen dieser Arbeit werden sechs-achsige Vertikal-Knickarmroboter betrachtet, wie sie für industrielle Anwendungszwecke häufig verwendet werden. Stellvertretend wird hier der ABB IRB 4600 verwendet, der in Abbildung 2.3 dargestellt ist. Der Roboter hat sechs Gelenke, die um jeweils eine Achse drehbar sind. Die möglichen Achsstellungen und die dabei erreichbaren Geschwindigkeiten sind in Tabelle 2.1 zusammengefasst.

Am Ende des Roboterarms befindet sich der Roboterflansch, an dem das Werkzeug angebracht wird, mit dem gearbeitet werden soll. In Abbildung 2.3a ist das 3D Modell einer Spritzpistole für den Lichtbogen-Spritzprozess an dem Roboter fixiert. Durch die Konstruktion der Spritzpistole ist die Ausrichtung der Spritzpistolendüse um  $90^\circ$  zum Flansch gedreht und die Position der Spitze der Sprühpistole ergibt sich durch eine Translation entlang aller Achsen vom Mittelpunkt des Flansches. Der Tool Center Point (TCP) des Werkzeugs liegt in einiger Entfernung von der Spitze der Spritzpistole. Dabei

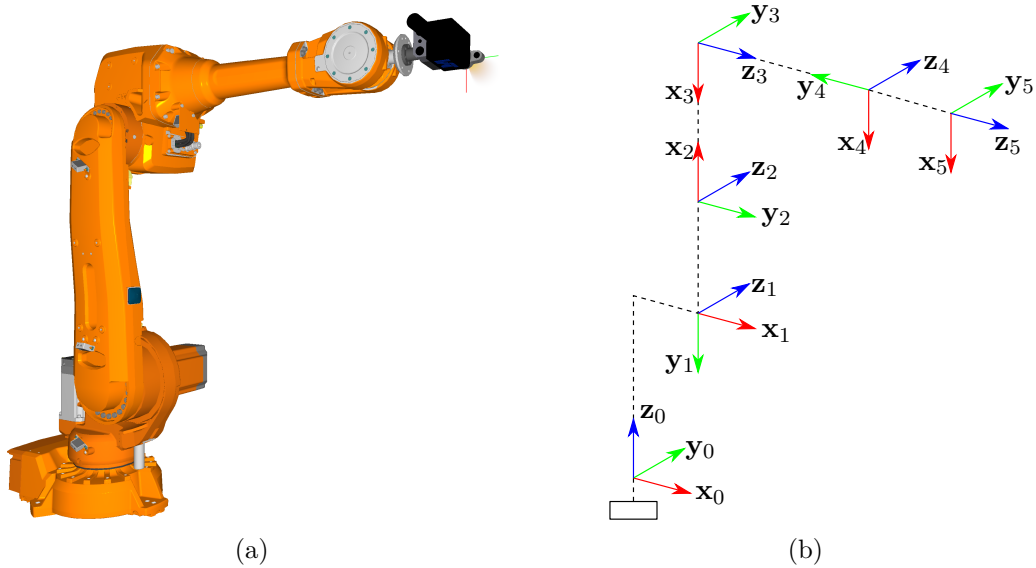


Abbildung 2.3: (a) CAD-Modelle des ABB IRB 4600 Roboters (Quelle <http://new.abb.com/products/robotics/industrial-robots/irb-4600/irb-4600-cad> (07.02.2017)) sowie einer Lichtbogen-Spritzpistole (Quelle: LWT, TU Dortmund); (b) Visualisierung der Denavit-Hartenberg Koordinatensysteme des ABB IRB 4600 Roboters.

Tabelle 2.1: Technische Daten des ABB IRB 4600 Roboters (Quelle: <http://new.abb.com/products/robotics/industrial-robots/irb-4600> (07.02.2017)).

Gelenk	Min. Winkel (°)	Max. Winkel (°)	Max. Geschwindigkeit (°/s)
0	-180	180	175
1	-90	150	175
2	-180	75	175
3	-400	400	250
4	-125	120	250
5	-400	400	360

wird die negative  $z$ -Richtung des Spritzpistolenkoordinatensystems als Spritzrichtung der Pistole genutzt.

### 2.3.2 Kinematik

Im Rahmen der Roboterkinematik, siehe [92], wird die Bewegung des Roboters beschrieben. Der Roboter kann jedes Gelenk bis zu einem gewissen Maximalwinkel um die jeweilige Rotationsachse drehen. Dabei sind sowohl die minimale und die maximale Achsstellung als auch die maximalen Winkelgeschwindigkeiten zu berücksichtigen. Eine

Tabelle 2.2: Denavit-Hartenberg Parameter des ABB IRB 4600 Roboters.

Index	$T_i^\alpha$ (rad)	$T_i^a$ (mm)	$T_i^d$ (mm)	$T_i^\Phi$ (rad)
1	$1,5\pi$	175	495	$0\pi$
2	$0\pi$	900	0	$-0,5\pi$
3	$0,5\pi$	-175	0	$-1,0\pi$
4	$1,5\pi$	0	960	$0\pi$
5	$0,5\pi$	0	0	$0\pi$
6	$0\pi$	0	135	$0\pi$

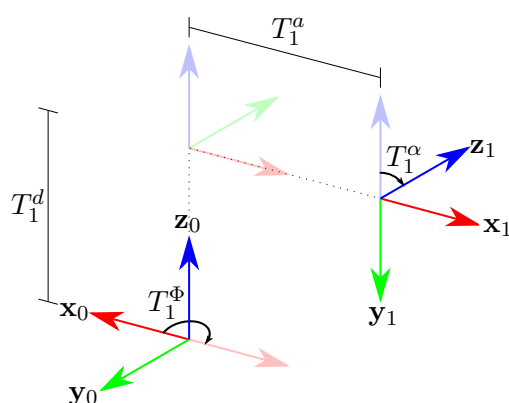


Abbildung 2.4: Beispiel für eine Denavit-Hartenberg Transformation. Das Koordinatensystem mit Index 0 wird in das mit Index 1 mittels der Parameter  $T_1^\alpha$ ,  $T_1^a$ ,  $T_1^d$  und  $T_1^\Phi$  überführt. In halb-Transparent sind Zwischenschritte kenntlich gemacht.

Betrachtung von Kräften und der damit verbundenen Roboterdynamik findet im Rahmen dieser Arbeit nicht statt.

Der Aufbau eines Industrieroboters kann mit Hilfe einer kinematischen Kette beschrieben werden. Dabei hat die Manipulation eines Gelenkwinkels Auswirkungen auf alle folgenden Achsen, Gelenke und auf die Endeffektorposition sowie -orientierung. Für die Beschreibung der kinematischen Kette können die Denavit-Hartenberg Transformationen genutzt werden, welche in der Robotik zum Standard geworden sind. Hierbei handelt es sich um homogene Transformationsmatrizen, die die Transformation eines Koordinatensystems von einem zum nächsten Element der kinematischen Kette beschreiben. Durch einfaches Multiplizieren der Matrizen kann die gesamte Transformation bestimmt werden. Um die Matrizen aufbauen zu können, müssen die Denavit-Hartenberg Parameter des Roboters bekannt sein oder bestimmt werden. Hierbei handelt es sich um je vier Parameter pro Gelenk. Die entsprechenden Parameter für den ABB IRB 4600 Roboter sind in Tabelle 2.2 aufgeführt und eine Visualisierung der lokalen Koordinatensysteme der Gelenke ist in Abbildung 2.3b dargestellt.

Die Transformation von einem in das nächste Koordinatensystem wird mittels der Denavit-Hartenberg Transformationen in vier Schritten durchgeführt, siehe Abbildung 2.4.



Der erste Schritt ist die Verschiebung entlang der  $\mathbf{z}_{i-1}$ -Achse um  $T_i^d$ , so dass der Koordinatenursprung auf dem Schnittpunkt der  $\mathbf{z}_{i-1}$ - und der  $\mathbf{x}_i$ -Achse liegt. Anschließend erfolgt eine Rotation um die  $\mathbf{z}_{i-1}$ -Achse um  $T_i^\Phi$ , wodurch die  $\mathbf{x}_{i-1}$ -Achse in die  $\mathbf{x}_i$ -Achse überführt wird. Die Verschiebung um  $T_i^a$  entlang der  $\mathbf{x}_i$ -Achse sorgt für die endgültige Positionierung des Systems. Als Letztes folgt noch eine Rotation um  $T_i^\alpha$  um die  $\mathbf{x}_i$ -Achse, um die  $\mathbf{z}_{i-1}$ - in die  $\mathbf{z}_i$ -Achse zu überführen. Damit ist die Transformation abgeschlossen.

Die Transformationsmatrix, die einem Gelenk zugeordnet ist, kann mit

$$\mathbf{T}_i(\Phi) = \begin{pmatrix} \cos(\hat{T}_i^\Phi) & -\sin(\hat{T}_i^\Phi) \cdot \cos(T_i^\alpha) & \sin(\hat{T}_i^\Phi) \cdot \sin(T_i^\alpha) & T_i^a \cdot \cos(\hat{T}_i^\Phi) \\ \sin(\hat{T}_i^\Phi) & \cos(\hat{T}_i^\Phi) \cdot \cos(T_i^\alpha) & -\cos(\hat{T}_i^\Phi) \cdot \sin(T_i^\alpha) & T_i^a \cdot \sin(\hat{T}_i^\Phi) \\ 0 & \sin(T_i^\alpha) & \cos(T_i^\alpha) & T_i^d \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

beschrieben werden. Hierbei ist  $\hat{T}_i^\Phi$  die Summe aus  $T_i^\Phi$  und dem aktuellen Rotationswinkel des entsprechenden Gelenkes.

Bei der Vorwärtskinematik wird die Werkzeugpose  $\boldsymbol{\rho}_j$ , also die Position und Orientierung des TCPs, bei gegebenen Achsstellungen gesucht. Dieses Problem ist dank der Denavit-Hartenberg Transformation sehr leicht zu lösen, indem die Transformationen der kinematischen Kette multipliziert werden. Die Position  $\mathbf{x}(\Phi)$  und die Orientierung  $\mathbf{R}(\Phi)$  können dann in Abhängigkeit von den Roboterachsstellungen  $\Phi$  durch die Gleichung

$$\begin{pmatrix} \mathbf{R}(\Phi) & \mathbf{x}(\Phi) \\ \mathbf{0} & 1 \end{pmatrix} = \mathbf{T}^r \cdot \prod_{i=1}^6 \mathbf{T}_i(\Phi_i) \cdot \mathbf{T}^t \quad (2.6)$$

berechnet werden.  $\mathbf{T}^r$  beschreibt die Initialtransformation des Roboters in Form einer homogenen Transformationsmatrix und gibt demnach die Position und Orientierung der Roboterbasis in Weltkoordinaten an.  $\mathbf{T}_i(\Phi_{i-1})$  sind die Denavit-Hartenberg Transformationsmatrizen aus Gleichung (2.5).  $\mathbf{T}^t$  beschreibt die Transformation vom Roboterflansch zum TCP ebenfalls in Form einer homogenen Transformationsmatrix.

Die inverse Kinematik befasst sich mit der Bestimmung der Gelenkwinkel  $\Phi_i$ ,  $i = 1, \dots, 6$ , einer kinematischen Kette ausgehend von einer gegebenen Werkzeugpose  $\boldsymbol{\rho}_j$ . Dies bedeutet, dass aus der Position und der Orientierung des TCPs auf die Gelenkwinkel geschlossen werden muss. Dieses Problem ist deutlich komplizierter als die Vorwärtskinematik. Eine Möglichkeit zur Lösung des allgemeinen Problems ist, die inverse Kinematik mit Hilfe der Jakobi Matrix zu lösen (siehe [120]). Eine Erleichterung bei den meisten Industrierobotern, insbesondere des hier betrachteten ABB IRB 4600, ist der spezielle Aufbau der letzten drei Achsen, deren Rotationsachsen sich alle in einem Punkt schneiden. In diesem Fall ist es möglich, die Position und die Orientierung mittels *Kinematic Decoupling* einzeln zu bestimmen (siehe [149]). Das Grundprinzip besteht darin, dass die ersten drei Achsen die Position und die letzten drei Achsen die Orientierung bestimmen. Mittels geometrischer Überlegungen können die gesuchten Achsstellungen dann analytisch bestimmt werden.



### 2.3.3 Programmierung

Die Programmierung eines Roboters ist bei komplexen Prozessen sehr schwierig und daher sowohl mit einem großen Zeit- wie auch Personalbedarf verbunden. In der Praxis werden häufig zwei unterschiedliche Ansätze verfolgt, die jeweils verschiedene Vor- und Nachteile mit sich bringen. Eine Übersicht über diese Möglichkeiten ist unter anderem in [131] zusammengestellt. Einige Aspekte aus [131] werden im Folgenden vorgestellt.

Die erste Möglichkeit – und die am weitesten verbreitete – ist das Online Programmieren. Ein Experte wird benötigt, der direkt am Roboter alle Bewegungen und alle Arbeitsschritte manuell ausführt. Während des Positionierens des Roboters werden die Bewegungen und Positionen vom Robot-Controller gespeichert und stehen danach zur wiederholten Ausführung bereit. Dieses sogenannte *Teachen* des Roboters ist sehr aufwändig und es kann viel Zeit vergehen, bis der Prozess zur vollen Zufriedenheit einprogrammiert wurde. Die Arbeit muss direkt am Roboter geschehen, so dass der Roboter für diese Zeit nicht zur Produktion zur Verfügung steht. Außerdem ist die Arbeit am Roboter nicht ungefährlich. Die Qualität des entstehenden Roboterprogramms ist abhängig von den Fähigkeiten des Experten, der die Programmierung vornimmt. Wenn sich das Werkstück oder die Prozessparameter ändern, muss das Teachen erneut durchgeführt werden, so dass dieses Verfahren sehr unflexibel ist.

Die zweite Möglichkeit zur Programmierung von Robotern ist das Offline Programmieren. Bei dieser Variante wird ein Roboterprogramm nicht direkt am Roboter, sondern mit Hilfe einer Software und 3D-Modellen vom Roboter und dem Werkstück erstellt. Je nach Möglichkeiten der Software können Bahnen für den Roboter auch teilautomatisiert generiert werden. Ein großer Vorteil dieses Programmierverfahrens ist, dass der Roboter für das Programmieren nicht benötigt wird und die Programmierung je nach Prozess auch schon frühzeitig mit Hilfe von Prototypen des Werkstücks begonnen werden kann. Kleinere Änderungen am Verfahren oder den Werkstücken können durch Anpassungen an der Bahn einfacher vorgenommen werden als beim Online Programmieren. Falls in der Software Simulationsmodule integriert sind, kann der Prozess bereits vor der tatsächlichen Ausführung simuliert und die Bahnen evaluiert werden. Hierdurch kann der Aufwand für die manuelle Korrektur im echten Prozess reduziert werden. Da der Prozess jedoch zunächst nur Offline geplant wird, ist ebenfalls ein Kalibriervorgang notwendig, um Abweichungen der Simulationsumgebung von der Realität zu kompensieren. Je nach Prozess und der Komplexität des Problems kann ebenso ein Programmierer benötigt werden, der die Software an die gegebenen Anforderungen anpasst.





# 3

## Simulation

Ein zentrales Element bei der hier entwickelten Bahnplanung ist die Simulation des Prozesses, für den die Bahn geplant werden soll. Durch die Nutzung von Simulationen können sowohl die Prozessergebnisse als auch der aktuelle Zustand des Werkstücks zu jeder Zeit während des Prozesses bestimmt werden. Für die Planung einer Bahn bedeutet dies, dass auf Grundlage dieser Informationen die Bahn geändert und optimiert werden kann, ohne dass teure Proben mit dem tatsächlichen Prozess erzeugt werden müssen. Voraussetzung hierfür ist jedoch, dass die Simulation realitätsnahe Ergebnisse liefert, um tatsächliche Experimente zu ersetzen.

In diesem Kapitel werden die entwickelten Simulationskomponenten vorgestellt, die im Bereich der Bahnplanung verwendet werden. Nach einer kurzen Einleitung in Abschnitt 3.1 werden die Kernkomponenten, bestehend aus der Auftragssimulation und der thermischen Simulation, vorgestellt. Die Auftragssimulation wird in Abschnitt 3.2 behandelt und hat zur Aufgabe, die Schichtbildung vorherzusagen, während die Simulation des thermischen Systems, welche in Abschnitt 3.3 beschrieben wird, die Temperaturen auf dem Werkstück während des Prozesses simuliert. In Abschnitt 3.4 wird eine Übersicht über die gesamte Simulationsumgebung gegeben und in Abschnitt 3.5 werden die Simulationskomponenten evaluiert. Abschnitt 3.6 schließt das Kapitel mit einem Fazit ab.

### 3.1 Einleitung

Die Simulation, die im Rahmen dieser Arbeit entwickelt wurde, soll den Prozess des thermischen Spritzens abbilden. Die wesentlichen Komponenten, die hier betrachtet werden, sind die Auftragssimulation und die thermische Simulation. Die Auftragssimulation modelliert die Schichtbildung beim thermischen Spritzen. Hierbei soll eine Vorhersage der Schichtdicke an jeder Stelle des Werkstücks getroffen werden. Die Simulation berücksichtigt dabei Faktoren wie den Massefluss im Spritzkegel und die Abhängigkeit

vom Einschlagwinkel auf der Oberfläche. Die Grundlage des Auftragsmodells bildet ein makroskopischer Ansatz, so dass einzelne Partikel nicht simuliert werden. Da die Oberfläche des Werkstücks als Dreiecksnetz modelliert wird, wird die Schichtdicke an den Knoten dieses Netzes berechnet und gespeichert. Die Dicke der Schicht wird in Bezug auf die Knotennormalen bestimmt. Die thermische Simulation soll eine Vorhersage der Temperaturen auf der Oberfläche des Werkstücks treffen. Dabei werden Faktoren wie der Wärmeeintrag, die Diffusion und die Abkühlung mittels verschiedener physikalischer Prozesse wie der Konvektion, der Diffusion und der Strahlung beschrieben. Auch die Temperatur wird an den Knoten des Netzes berechnet und gespeichert.

Das Hauptaugenmerk der Simulationen liegt auf der Nutzbarkeit im Rahmen der Bahnplanung. Dies bedeutet, dass neben der Genauigkeit der Simulationsergebnisse auch die Geschwindigkeit der Simulationen im Vordergrund steht. Die Ergebnisse der Simulationen werden dazu genutzt, die Güte der Bahnen vorherzusagen, verschiedene Bahnen zu vergleichen oder auf Grundlage der Simulationsdaten Stellen der Bahn zu identifizieren, deren Modifikation gewisse Eigenschaften verbessern. Ein wichtiger Aspekt der Auftragssimulation ist daher die Vorhersage der Schichtdicke. Diese wird direkt für die Qualitätsbestimmung der Bahn im Bereich der Initialbahnplanung genutzt, siehe Kapitel 4. Da hier allerdings sehr viele Bahnen erzeugt, simuliert und ausgewertet werden müssen, muss die gesamte Simulation sehr performant sein. Für die Bahnoptimierung aus Kapitel 5 wird zusätzlich eine bereits existierende, externe Simulation zur Bestimmung der Beschichtung genutzt, da diese in der Lage ist, besonders gute Vorhersagen zu liefern. Daher muss eine entsprechende Simulation nicht neu entwickelt werden. Die in diesem Kapitel vorgestellte Simulation wird im Bereich der Bahnoptimierung jedoch trotzdem verwendet. Da keine Gradienten von der externen Simulation zur Verfügung stehen, werden diese mit Hilfe der an der externen Simulation kalibrierten eigenen Simulation bestimmt. Dies ermöglicht den Einsatz von Gradientenabstiegsverfahren wie dem nichtlinearen konjugierten Gradientenverfahren. Das Simulationsmodell ist daher konzeptionell darauf ausgelegt, effizient ableitbar zu sein. Bei dem realen Prozess stellen im Bereich der Wärmeentwicklung im Werkstück vor allem thermische Varianzen ein Problem dar, so dass die relative Temperaturverteilung im Werkstück genauer als die absoluten Temperaturwerte betrachtet werden müssen. Des Weiteren ist die Modellierung des thermischen Systems physikalisch und mathematisch relativ komplex und daher bei exakter Simulation sehr zeitaufwändig. Aus diesen beiden Gründen werden daher zu Gunsten des Rechenzeitbedarfs an geeigneten Stellen Ersatzmodelle verwendet, die eine leichte Qualitätseinbuße gegenüber exakteren Berechnungen mit sich bringen, jedoch die relativen Temperaturunterschiede qualitativ korrekt abbilden.

## 3.2 Auftragssimulation

Die Auftragssimulation ist die erste große Komponente der Simulationsumgebung. Diese Komponente soll eine Vorhersage des Schichtauftrags an jedem Punkt des Werkstücks leisten.

### 3.2.1 Einordnung und Abgrenzung zu existierenden Modellen

Simulationen für Auftragsprozesse können in mikroskopische und makroskopische Modelle unterteilt werden. Bei Ersteren liegt der Fokus auf der präzisen Simulation auf Partikelebene, bei Letzteren wird von den Partikeln abstrahiert und ein Modell des Masseflusses betrachtet, um den Schichtauftrag vorherzusagen.

Die Simulation von Einzelpartikeln soll genauere Einblicke in das Verhalten der Partikel beim Auftreffen auf das Werkstück und die daraus resultierende Schichtbildung geben. Dabei stehen Fragestellungen rund um die Splat-Bildung, welche die Verformung der Partikel beim Aufschlag auf die Oberfläche beschreibt, sowie den Abkühl- und Erstarrvorgang im Vordergrund. Übersichten über den Stand der Forschung können unter anderem in [174, 33, 148, 60] gefunden werden. Die Erkenntnisse, die aus den Einzelsplatberechnungen gezogen werden, können in einem weiteren Schritt dazu verwendet werden, die Schichtbildung mit vielen Splats zu simulieren. Die Publikation [171] befasst sich genau mit dieser Fragestellung und simuliert die Schichtbildung, indem viele Einzelsplats betrachtet und mit Hilfe eines Voxelmodells übereinander geschichtet werden. Dabei werden je nach Untergrundsituation einzelne Partikel mittels bekannter Simulationsmodelle berechnet oder aber aus einer Datenbank geladen, wenn ähnliche Situationen bereits vorberechnet wurden. Dieser Ansatz erlaubt es erstmals, sehr viele, bis zu mehreren zehntausend Partikel, in akzeptabler Zeit zu berechnen.

Exakte Simulationen von Einzelpartikeln können auf Grund ihrer Komplexität nicht für die Schichtbildungssimulation großer Werkstücke herangezogen werden. Die Erkenntnisse, die aus dieser Art der Simulation gewonnen werden, kann jedoch als Grundlage genutzt werden, um makroskopische Modelle zu parametrisieren.

Bei makroskopischen Modellen wird von den vorhandenen Partikeln im Spritzstrahl abstrahiert und es wird eine Aussage über den Massefluss herangezogen. Die Beschreibung des Masseflusses wird häufig entweder mittels eines analytischen Modells oder über Look-Up Tabellen realisiert. Ein detailliertes analytisches Modell wurde in [54] vorgestellt und neuere Modelle, z. B. in [22, 142], basieren auf diesem. Alternativ ist ein anderer Ansatz entwickelt worden, der sogenannte Footprints [173, 170] verwendet. Bei einem Footprint handelt es sich um eine Art Look-Up Tabelle, die den Massefluss in dem Spritzstrahl modelliert. Die Footprints werden erzeugt, indem auf eine Flachprobe orthogonal aus konstantem Abstand für eine gewisse Zeit gespritzt wird. Der entstandene Auftrag wird vermessen und als Look-up Tabelle für die Bestimmung der Partikelverteilung im Spritzstrahl verwendet. Eine verbesserte Variante der Footprinterzeugung nutzt zusätzlich die Höhenprofile von Linearbahnen, um genauere Footprints zu erzeugen, siehe [170]. Der große Vorteil dieses Verfahrens ist die Einfachheit der Modellierung asymmetrischer Partikelverteilungen, wie sie in der Realität häufig vorzufinden sind. Auch andere Effekte, wie die Haftung der Partikel auf dem Werkstück unter verschiedenen Einschlagwinkeln, werden in solchen Modellen berücksichtigt. Insgesamt können makroskopische Modelle für die effiziente Vorhersage der Schichtverteilung auf großen Werkstücken verwendet werden.

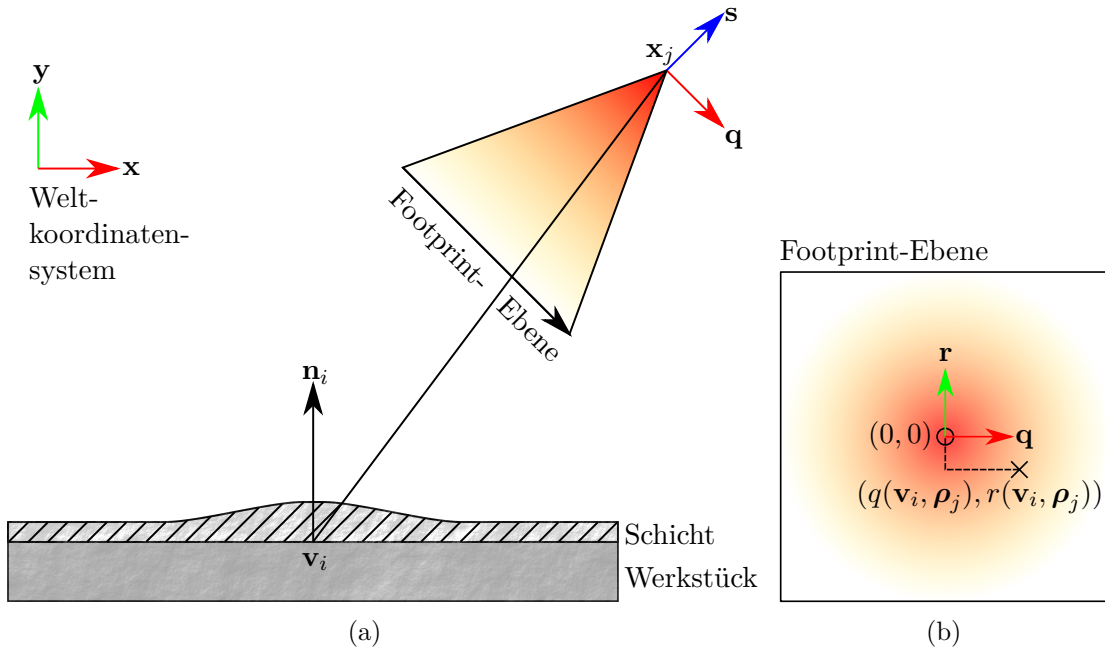


Abbildung 3.1: (a) Zweidimensionale, schematische Darstellung des Auftragsmodells. Der Spritzkegel hat seinen Ursprung im Bahnpunkt  $\mathbf{x}_j$  und zeigt in Richtung  $-\mathbf{s}$ . Der Oberflächenpunkt  $\mathbf{v}_i$  des Werkstücks (dunkelgrau) mit der Normalen  $\mathbf{n}_i$  wird mit Material (hellgrau) beschichtet. (b) Schematische Ansicht der Footprint-Ebene. In der Mitte der Ebene liegt der Punkt  $(0, 0)$ . Der Schnittpunkt der Verbindungsline von  $\mathbf{x}_j$  nach  $\mathbf{v}_i$  mit dieser Ebene liegt bei  $(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j))$  nach Gleichung (3.8). Durch die Schattierung ist der Massefluss angedeutet.

Im Folgenden wird ein Auftragsmodell beschrieben, das für die Initialbahnplanung sowie für die Gradientenbestimmung in der Bahnoptimierung verwendet wird. Hierbei handelt es sich um ein makroskopisches Modell, das an das Modell von [54] angelehnt ist.

### 3.2.2 Auftragsmodell

Das Auftragsmodell bestimmt den Schichtauftrag an einem Oberflächenpunkt, der bei der Verwendung einer bestimmten Bahn erzeugt wird. Dazu wird die Menge an Material bestimmt, die an jedem Oberflächenpunkt abgelagert wird, während sich die Spritzpistole entlang der Bahn bewegt. Da die Bahn als eine Menge dicht liegender Spritzpistolenposen repräsentiert wird, wird für jedes Paar aus Pose und Knoten des Oberflächennetzes der Auftrag bestimmt, der bei dieser Konfiguration abgelagert wird. Das Auftragsmodell wird für einen einzelnen Bahnpunkt schematisch in Abbildung 3.1a dargestellt. Die für

jeden Bahnpunkt entstehenden Materialbeiträge werden dann zum gesamten Auftrag aufsummiert. Der Materialauftrag am Oberflächenpunkt mit Index  $i$  ergibt sich daher zu

$$\tilde{z}_i = \sum_{j=0}^n z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j), \quad (3.1)$$

wobei  $\tilde{z}_i$  die Beschichtungsdicke am Oberflächenpunkt  $\mathbf{v}_i$  ausdrückt. Der normalisierte Normalenvektor an diesem Punkt wird mit  $\mathbf{n}_i$  bezeichnet und  $\boldsymbol{\chi}_j = (\boldsymbol{\rho}_j, \hat{t}_j)^\top$  ist der  $j$ -te Bahnpunkt.  $\hat{t}_j$  beschreibt die Zeit, die die Spritzpistole an einem Bahnpunkt verbringt. Diese lässt sich aus den Zeitpunkten berechnen, zu denen die Spritzpistole eine Bahnposition erreicht, siehe Abschnitt 2.1.

Das Beschichtungsmodell berücksichtigt verschiedene Komponenten, nämlich die Abhängigkeit des Auftrags von dem Einschlagwinkel der Partikel, die Verteilung der Partikel im Spritzstrahl und die Geschwindigkeit der Spritzpistole, welche durch die Spritzdauer am Bahnpunkt ausgedrückt wird. Dadurch kann die Summe in Gleichung (3.1) weiter verfeinert werden, um diese Komponenten zu beschreiben, und ergibt somit

$$z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_M(\mathbf{v}_i, \boldsymbol{\rho}_j) \cdot \frac{t_{j-1} + t_j}{2}. \quad (3.2)$$

Der Term  $a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j)$  beschreibt die Abhängigkeit des Materialauftrags vom Einschlagwinkel der Partikel auf das Werkstück und der Term  $z_M(\mathbf{v}_i, \boldsymbol{\rho}_j)$  beschreibt die Materialverteilung im Spritzstrahl.

Die Abhängigkeit des Auftrages vom Einschlagwinkel der Partikel auf das Werkstück wird in diesem Modell über den Cosinus des Winkels beschrieben. Je größer der Winkel zwischen der Einschlagrichtung und der Normalen der Oberfläche ist, desto größer ist die Fläche, die von einem gegebenen Winkelausschnitt beschichtet wird. Damit kann der Term mit

$$a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) = \frac{(\mathbf{x}_j(\boldsymbol{\rho}_j) - \mathbf{v}_i)^\top \mathbf{n}_i}{\|\mathbf{x}_j(\boldsymbol{\rho}_j) - \mathbf{v}_i\|} \quad (3.3)$$

beschrieben werden, wobei  $\mathbf{x}_j(\boldsymbol{\rho}_j)$  die Position der Spritzpistole bezeichnet. Je nach Repräsentationsform der Bahn liegt die Position entweder direkt vor oder kann durch Gleichung (2.6) aus den gegebenen Achsstellungen des Roboters durch Anwendung der Vorwärtskinematik bestimmt werden.

Über die Materialverteilung im Spritzstrahl wird das an einem Punkt auftreffende Material bestimmt. Um dieses zu ermitteln, wird  $z_M(\mathbf{v}_i, \boldsymbol{\rho}_j)$  entsprechend definiert zu

$$z_M(\mathbf{v}_i, \boldsymbol{\rho}_j) = z_M(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)). \quad (3.4)$$

Die konkrete Menge an Material, welche an einem Oberflächenpunkt abgelagert wird, ist daher abhängig von der Distanz

$$d(\mathbf{v}_i, \boldsymbol{\rho}_j) = \|\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)\| \quad (3.5)$$

der Spritzpistole von dem Oberflächenpunkt und der relativen Orientierung der zentralen Spritzrichtung am Punkt  $\mathbf{x}_j(\boldsymbol{\rho}_j)$  und der Verbindungslinie von  $\mathbf{x}_j(\boldsymbol{\rho}_j)$  nach  $\mathbf{v}_i$ .  $q(\mathbf{v}_i, \boldsymbol{\rho}_j)$  und  $r(\mathbf{v}_i, \boldsymbol{\rho}_j)$  sind die Distanzen vom Ursprung zu dem Punkt  $(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j))$  in der Footprint-Ebene, siehe Abbildung 3.1b. Die Footprint-Ebene liegt einen Millimeter vor der Spritzpistole und spannt den Parameterraum der konkreten Materialverteilungsfunktion auf.

Um die Koordinaten  $(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j))$  bestimmen zu können, muss die relative Lage der zu beschichtenden Oberflächenpunkte bezüglich des Spritzpistolenkoordinatensystems bestimmt werden. Für die Verbindung von  $\mathbf{x}_j(\boldsymbol{\rho}_j)$  nach  $\mathbf{v}_i$  gilt dann

$$\overline{(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))} = \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)). \quad (3.6)$$

Hierbei wird die Transformation  $\mathbf{R}(\boldsymbol{\rho}_j)$  der Koordinatensysteme abhängig von der Repräsentationsform der Bahn bestimmt. Wenn die Orientierung des Werkzeugs über drei Winkel gegeben ist, dann kann die Transformation mittels

$$\mathbf{R}(\boldsymbol{\rho}_j) = \mathbf{R}_z(\phi_{j,2}) \cdot \mathbf{R}_x(\phi_{j,1}) \cdot \mathbf{R}_z(\phi_{j,3}) \quad (3.7)$$

bestimmt werden. Dabei bezeichnet  $\mathbf{R}_w(\phi)$  eine Rotation um die Weltkoordinatenachse  $\mathbf{w}$  um den Winkel  $\phi$ . Bei Vorliegen der Bahnrepräsentation über Roboterachsstellungen kann die Transformation  $\mathbf{R}(\boldsymbol{\rho}_j)$  mit Hilfe der Gleichung (2.6) berechnet werden.

Die Werte  $q(\mathbf{v}_i, \boldsymbol{\rho}_j)$  und  $r(\mathbf{v}_i, \boldsymbol{\rho}_j)$  sind die Koordinaten des Schnitts von  $\overline{(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))}$  mit der ( $s = -1$ )-Ebene (Footprint-Ebene, siehe Abbildung 3.1b) und werden durch

$$q(\mathbf{v}_i, \boldsymbol{\rho}_j) = -\frac{\overline{(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))}_q}{\overline{(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))}_s} \quad (3.8)$$

und

$$r(\mathbf{v}_i, \boldsymbol{\rho}_j) = -\frac{\overline{(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))}_r}{\overline{(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))}_s} \quad (3.9)$$

berechnet. Durch die Indizes  $q$ ,  $r$  und  $s$  wird jeweils die Nutzung der  $q$ -,  $r$ - bzw.  $s$ -Koordinate eines Vektors ausgedrückt. Einsetzen der Transformation liefert

$$q(\mathbf{v}_i, \boldsymbol{\rho}_j) = -\frac{\left(\mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_q}{\left(\mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s} \quad (3.10)$$

für  $q(\mathbf{v}_i, \boldsymbol{\rho}_j)$  und äquivalent für  $r(\mathbf{v}_i, \boldsymbol{\rho}_j)$ .

Das konkrete Modell der Materialverteilungsfunktion wird für diese Arbeit über eine Menge von überlagerten Gaußfunktionen beschrieben, wobei eine Gaußfunktion mittels

$$z_{M,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) = \frac{c_k}{d(\mathbf{v}_i, \boldsymbol{\rho}_j)^2} \cdot \exp \left( - \left( \begin{array}{c} q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) \\ r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) \end{array} \right)^\top \boldsymbol{\Sigma}_k^{-1} \left( \begin{array}{c} q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) \\ r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) \end{array} \right) \right) \quad (3.11)$$



beschrieben werden kann. Durch den Index  $d$  wird eine Verschiebung der Werte ausgedrückt, damit auch eine Verschiebung des Masseflusses bezüglich der Spritzrichtung beschrieben werden kann. Dies wird durch

$$q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) = q(\mathbf{v}_i, \boldsymbol{\rho}_j) - \mu_{q,k}, \quad (3.12)$$

$$r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) = r(\mathbf{v}_i, \boldsymbol{\rho}_j) - \mu_{r,k} \quad (3.13)$$

realisiert. Die Matrizen sind über

$$\boldsymbol{\Sigma}_k^{-1} = \begin{pmatrix} \frac{1}{\sigma_{1,k}^2} & \frac{1}{\sigma_{3,k}} \\ \frac{1}{\sigma_{3,k}} & \frac{1}{\sigma_{2,k}^2} \end{pmatrix} \quad (3.14)$$

gegeben, wobei  $\sigma_{1,k}, \sigma_{2,k} \in (0, \infty)$  und  $\sigma_{3,k} \in (0, \infty]$ . Da die Matrix positiv definit sein soll, müssen die Eigenwerte der Matrix positiv sein. Aus dieser Eigenschaft folgt die Bedingung  $\sigma_{3,k}^2 \geq \sigma_{1,k}^2 \sigma_{2,k}^2$ . Damit ergibt sich für  $z_{M,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)$  zusammengefasst:

$$z_{M,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) = \frac{c_k}{d(\mathbf{v}_i, \boldsymbol{\rho}_j)^2} \cdot \exp \left( - \left( \frac{q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)^2}{\sigma_{1,k}^2} + \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{3,k}} + \frac{r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)^2}{\sigma_{2,k}^2} \right) \right). \quad (3.15)$$

Das gesamte Modell ist dann:

$$\tilde{z}_i = \sum_{j=0}^n z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j), \quad (3.16)$$

$$z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot \frac{t_{j-1} + t_j}{2} \cdot z_M(\mathbf{v}_i, \boldsymbol{\rho}_j) \quad (3.17)$$

$$= a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot \frac{t_{j-1} + t_j}{2} \cdot \sum_{k=0}^K z_{M,k}(\mathbf{v}_i, \boldsymbol{\rho}_j). \quad (3.18)$$

### 3.2.3 Kalibrierung

Um die Parameter  $c_k$ ,  $\sigma_{1,k}$ ,  $\sigma_{2,k}$ ,  $\sigma_{3,k}$ ,  $\mu_{q,k}$  und  $\mu_{r,k}$  des Modells für einen konkreten Prozess zu bestimmen, soll ein einfaches Gradientenabstiegsverfahren genutzt werden. Das Verfahren nutzt dazu die Schichtdicken eines Footprints, der mit Hilfe einer Flachprobe erzeugt wird, die mit einer unbewegten Spritzpistole für eine festgelegte Zeit beschichtet wird. Diese Daten können entweder messtechnisch nach einem realen Experiment erfasst oder alternativ mit Hilfe eines vorhandenen Footprints aus der externen Simulation *SimSpray*<sup>1</sup> erzeugt werden.

Die Simulation wird auf Basis dieser Daten so konfiguriert, dass die Abweichungen vom Prozess bzw. Footprint minimiert werden. Die Parameter des Modells werden

<sup>1</sup>Zur Verfügung gestellt von Thomas Wiederkehr.

initial geschätzt und die Simulation wird basierend auf diesen Daten ausgeführt. Die entstehenden Schichtdicken auf dem Werkstück werden mit dem realen Experiment oder Footprint verglichen und die quadratischen Abweichungen der Dicken werden als Fehlerfunktion verwendet. Die Funktion hat demnach die Form

$$E = \sum_{i=0}^m \left( z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) - Z(\mathbf{v}_i) \right)^2, \quad (3.19)$$

wobei  $Z(\mathbf{v}_i)$  für die durch den Footprint gegebene Schichtdicke an dem Knoten  $\mathbf{v}_i$  steht. Für das Gradientenabstiegsverfahren wird diese Fehlerfunktion nach den Modellparametern abgeleitet. Da der Gradient der Funktion die Richtungen des steilsten Anstiegs im Parameterraum darstellt und die Funktion minimiert werden soll, kann durch Subtraktion des (skalierten) Gradienten eine Verringerung des Fehlers erreicht werden, wobei der Betrag von  $\sigma_{3,k}$  bei Verletzung der Nebenbedingung auf  $\sigma_{1,k}\sigma_{2,k}$  gesetzt wird. Durch Iteration des Verfahrens kann der Fehler solange verkleinert werden, bis ein (lokales) Optimum erreicht wird. Die partiellen Ableitungen der Fehlerfunktion ergeben sich zu

$$\nabla_p E = 2 \cdot \left( z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) - Z(\mathbf{v}_i) \right) \cdot \nabla_p z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j), \quad (3.20)$$

wobei  $\nabla_p$  die partielle Ableitung nach einem der Parameter  $c_k, \sigma_{1,k}, \sigma_{2,k}, \sigma_{3,k}, \mu_{q,k}$  und  $\mu_{r,k}$  darstellt.  $\nabla_p z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j)$  ergibt somit konkret für jede der Komponenten

$$\nabla_{c_k} z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = c \cdot \frac{1}{c_k}, \quad (3.21)$$

$$\nabla_{\sigma_{1,k}} z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = c \cdot \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)^2}{\sigma_{1,k}^3}, \quad (3.22)$$

$$\nabla_{\sigma_{2,k}} z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = c \cdot \frac{2r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)^2}{\sigma_{2,k}^3}, \quad (3.23)$$

$$\nabla_{\sigma_{3,k}} z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = c \cdot \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{3,k}^2}, \quad (3.24)$$

$$\nabla_{\mu_{q,k}} z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = c \cdot \left( \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{1,k}^2} + \frac{2r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{3,k}} \right), \quad (3.25)$$

$$\nabla_{\mu_{r,k}} z(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j, \hat{t}_j) = c \cdot \left( \frac{2r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{2,k}^2} + \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{3,k}} \right), \quad (3.26)$$

mit

$$c = a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot \frac{t_{j-1} + t_j}{2} \cdot z_{M,k}(\mathbf{v}_i, \boldsymbol{\rho}_j). \quad (3.27)$$

Wenn sich nach einigen Iterationen der Funktionswert nicht mehr ändert, ist ein lokales Optimum erreicht. Der verbleibende Fehler ist für jeden konkreten Footprint unterschiedlich und wird für drei verschiedene Footprints in Abschnitt 3.5.1 angegeben. Komplexere Verfahren wie das konjugierte Gradientenverfahren können ein (lokales) Optimum in weniger Iterationen finden. Da die Modellparameter für einen Prozess bzw. Footprint nur ein einziges Mal bestimmt werden müssen, ist die Implementierung komplexerer Verfahren jedoch nicht notwendig.

### 3.3 Thermische Simulation

Die thermische Simulation ist die zweite große Komponente der Simulationsumgebung. Die Aufgabe dieser Komponente besteht in der Vorhersage der Temperatur an jeder Stelle des Werkstücks während des Beschichtungsvorgangs.

#### 3.3.1 Einordnung und Abgrenzung zu existierenden Modellen

Die Simulation des thermischen Verhaltens beim thermischen Spritzen hat weniger Aufmerksamkeit in der Literatur erhalten als die Beschichtungssimulation. Ein ausführliches Modell wird von Jones et. al [95] vorgestellt. In diesem Modell spielen Faktoren wie der Wärmeeintrag, die Diffusion sowie die Abkühlung eine Rolle. Das Modell wird genutzt, um eine analytische Optimierung der Parameter für Billardbahnen beim thermischen Spritzen durchzuführen. In [137] wird ein komplexes Wärmemodell für das mit dem thermischen Spritzen verwandten Sprühkompaktieren vorgestellt und experimentell bestätigt.

Das in dieser Arbeit vorgestellte thermische Modell basiert auf den oben genannten Modellen von Jones et. al [95] und Rayment et. al [137], wird an einigen Stellen jedoch erweitert. So werden verschiedene physikalische Prozesse im Bereich Wärmeeintrag, Abkühlung und Wärmediffusion berücksichtigt. Während das primäre Ziel der Simulation die exakte Bestimmung der Temperaturwerte ist, muss auf der anderen Seite eine geringe Laufzeit der Simulation erzielt werden, um eine praktische Anwendbarkeit in den Optimierungen zu gewährleisten. Auf Grund der Komplexität der physikalischen Prozesse ist es daher nicht immer sinnvoll, diese exakt abzubilden. An manchen Stellen ist es eher wünschenswert, die Prozesse durch Näherungsmodelle zu beschreiben, um damit die Laufzeitanforderungen zu erfüllen. Im vorliegenden Fall wird der Diffusionsprozess auf zwei Dimensionen beschränkt. Die Diffusion kann demnach entlang der Kanten des gegebenen Oberflächendreiecksnetzes erfolgen. Eine Diffusion ins Innere des Werkstücks kann mit Hilfe von Finiten Elementen Methoden und der Nutzung von Tetraedernetzen erfolgen [20], was die Komplexität der Simulation deutlich erhöht. An dieser Stelle ist der Diffusionsprozess ins Innere des Werkstücks daher durch eine Wärmeausgleichsrechnung ersetzt worden.

Abbildung 3.2 zeigt alle beteiligten Prozesse. Dazu gehören der Wärmeeintrag, die Diffusion im Werkstück und die Abkühlung, welche durch die physikalischen Prozesse der Konvektion, Diffusion und Strahlung entstehen.

#### 3.3.2 Thermisches Modell

Der erste Prozess, der im Rahmen des Modells betrachtet wird, ist der Wärmeeintrag durch das Auftreffen der heißen Partikel auf das Werkstück. Die Partikel bringen eine große Menge Wärmeenergie mit sich, die zu der bereits vorhandenen Wärme an dem

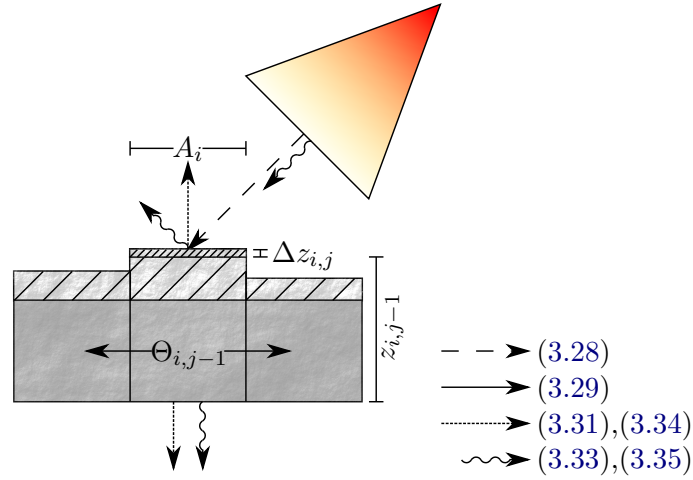


Abbildung 3.2: Zweidimensionale, schematische Darstellung des thermischen Modells. In Dunkelgrau ist das Werkstück und in Hellgrau der Auftrag dargestellt.  $z_{i,j-1}$  gibt die Gesamtdicke des Werkstücks inklusive der aktuellen Schicht und  $\Delta z_{i,j}$  die Dicke der neu hinzukommenden Schicht an.  $\Theta_{i,j-1}$  ist die Wärme an dem Knoten.  $i$  ist der Index des Knotens und  $j-1$  gibt den Zeitpunkt vor dem Beschichten mit dem Bahnpunkt  $j$  an. Die Pfeile deuten den Wärmetransfer durch die Formeln (3.28), (3.29), (3.31), (3.33), (3.34) und (3.35) an.

Oberflächenpunkt hinzuaddiert wird [18]. Damit ergibt sich die Temperatur, welche in °C angegeben wird, zu

$$\Theta_{i,j} = \frac{\tilde{\Theta} \Delta z_{i,j} + \Theta_{i,j-1} z_{i,j-1}}{\Delta z_{i,j} + z_{i,j-1}}, \quad (3.28)$$

wobei durch  $\tilde{\Theta}$  die Temperatur der Partikel kurz vor dem Auftreffen beschrieben wird.  $\Delta z_{i,j}$  ist die Dicke der neu hinzukommenden Beschichtung an dem Oberflächenpunkt mit Index  $i$ , welche durch das Material aufgetragen wird, das am Bahnpunkt mit Index  $j$  ausgegeben wird.  $\Theta_{i,j-1}$  beschreibt die Temperatur der Oberfläche vor dem Eintreffen der neuen Partikel und  $z_{i,j-1}$  ist die kombinierte Dicke der Schicht und des Werkstücks.

Die Wärmediffusion sorgt für eine Umverteilung der Wärme von wärmeren zu kälteren Bereichen. Mathematisch lässt sich dieser Vorgang durch eine Differentialgleichung beschreiben [95]. Die Diskretisierung dieser Gleichung mit Hilfe des Laplace-Beltrami Operators [140] führt zu

$$\Theta_{i,j} = \Theta_{i,j-1} + \frac{1}{2A_i} \frac{M_k \hat{t}_j}{M_c M_\rho} \sum_{l \in N_{i,1}} b_{i,l,j-1} (\cot(\alpha_{i,l}) + \cot(\beta_{i,l})) (\Theta_{l,j-1} - \Theta_{i,j-1}). \quad (3.29)$$

In dieser Gleichung wird durch  $A_i$  die Größe der Fläche beschrieben, die dem Knoten  $\mathbf{v}_i$  zugewiesen ist. Diese Fläche wird durch das hybride Voronoi-Verfahren bestimmt, siehe [122, 81] und Abbildung 2.2.  $M_k$  ist die Wärmeleitfähigkeit des Materials in W/(mK),

$M_c$  die Wärmekapazität in J/(kg K) und  $M_\rho$  die Dichte des Materials in kg/m<sup>3</sup>. Die Winkel  $\alpha_{i,l}$  und  $\beta_{i,l}$  liegen jeweils gegenüber der Kante, die gerade betrachtet wird, siehe Abbildung 2.2. Der Faktor  $b_{i,l,j}$  dient dazu, die unterschiedlichen Schichtdicken benachbarter Punkte zu berücksichtigen, um damit der Energieerhaltung zu genügen:

$$b_{i,l,j} = \begin{cases} \frac{z_{l,j}}{z_{i,j}}, & \text{wenn } z_{i,j} > z_{l,j}, \\ 1, & \text{sonst.} \end{cases} \quad (3.30)$$

Konvektion findet an der Oberfläche des Werkstücks statt. Die Wärmeenergie geht dabei nach dem Gesetz des Wärmetransfers an die Luft über dem Werkstück über. Direkt unter der Flamme der Spritzpistole ist die Luft jedoch sehr heiß, so dass hier umgekehrt Wärmeenergie an das Werkstück übertragen wird. Die Energieübertragung kann mit folgender Gleichung [19, 154] bestimmt werden:

$$\Theta_{i,j} = \Theta_{i,j-1} - \frac{M_{\alpha,i,j}(\Theta_{i,j-1} - \hat{\Theta}_{i,j-1})\hat{t}_j}{M_c M_\rho z_{i,j-1}}. \quad (3.31)$$

Hier bezeichnet  $M_{\alpha,i,j}$  den Wärmeübergangskoeffizienten in W/(m<sup>2</sup> K) und  $\hat{\Theta}_{i,j-1}$  die Temperatur direkt über dem Oberflächenpunkt. Die Luft über der Oberfläche ist an den Stellen, die die Spritzpistole nicht beeinflusst, relativ kühl (Raumtemperatur) und direkt unter der Flamme sehr heiß. Die Temperaturverteilung der Flamme wird mit einer Gaußfunktion beschrieben, die ihren heißesten Punkt direkt in der Mitte des Spritzkegels hat. Der Wärmeübergangskoeffizient ist ebenfalls variabel, da dieser unter der Flamme auf Grund des dort herrschenden Drucks deutlich größer ist. Daher wird der Wärmeübergangskoeffizient ebenfalls mittels einer Gaußfunktion beschrieben, wie es in [72] vorgeschlagen wird.

Ein weiterer Effekt, der die Temperatur des Werkstücks beeinflusst, ist die Wärmestrahlung. Diese kann über das Boltzmann Gesetz

$$P = \sigma_B A \Theta^4 \quad (3.32)$$

beschrieben werden [121], wobei  $P$  die Leistung in W,  $\sigma_B$  die Stefan-Boltzmann Konstante,  $A$  die Größe der Fläche und  $\Theta$  die Temperatur ist. Nach Diskretisierung der Formel und Anpassung an die Temperaturänderung anstelle der Energiebetrachtung ergibt sich

$$\Theta_{i,j} = \Theta_{i,j-1} - \frac{M_\epsilon \sigma_B (\Theta_{i,j-1}^4 - \hat{\Theta}_{i,j-1}^4) \hat{t}_j}{M_c M_\rho z_{i,j-1}}. \quad (3.33)$$

Durch den Wert von  $M_\epsilon$  wird das Strahlungsverhalten des Materials angegeben, das für einen idealen schwarzen Strahler mit  $M_\epsilon = 1$  angegeben wird.

Die Wärme, die an der Oberfläche des Werkstücks eingetragen wird, diffundiert nicht nur zu den Seiten, sondern auch in das Werkstück hinein. Dieser Diffusionsprozess kann

über einen Finite Elemente Ansatz modelliert werden. Dieser kann mittels eines feinaufgelösten Voxelmodells oder eines Tetraedernetzes modelliert werden. Beide Verfahren benötigen jedoch eine große Menge Rechenkapazität. Auf Grund der Notwendigkeit vieler Simulationsläufe während der Optimierungen wird an dieser Stelle auf eine exakte Berechnung verzichtet. Stattdessen wird eine einfachere Wärmeausgleichsrechnung durchgeführt, die jedoch die Energieerhaltung beachtet.

Nach der Diffusion in das Werkstück wird die Wärmeenergie an der Unterseite des Werkstücks ebenfalls durch den Abkühlungs- sowie Strahlungsprozess abgegeben. Die entsprechenden Formeln ergeben sich zu

$$\Theta_{i,j} = \Theta_{i,j-1} - \frac{M_\alpha(\Theta_{i,j-1} - \hat{\Theta})\hat{t}_j}{M_c M_\rho z_{i,t-1}} \quad (3.34)$$

und

$$\Theta_{i,j} = \Theta_{i,j-1} - \frac{M_\epsilon \sigma_B (\Theta_{i,j-1}^4 - \hat{\Theta}^4)\hat{t}_j}{M_c M_\rho z_{i,j-1}} \quad (3.35)$$

mit konstanten Werten für  $M_\alpha$  (materialabhängig) und  $\hat{\Theta}$  (z. B. Raumtemperatur).

### 3.3.3 Gesamtmodell der thermischen Simulation

Die Temperaturen  $\Theta_{i,j}$  an jedem Oberflächenknoten werden durch Anwendung der oben beschriebenen Formeln bestimmt. Da es sich bei der Diffusionsgleichung (3.29) um eine Diskretisierung einer Differentialgleichung handelt, muss darauf geachtet werden, dass der Zeitschritt nicht zu groß gewählt wird, um numerische Stabilität zu gewährleisten. Der native Zeitschritt der Simulation ist durch den zeitlichen Abstand zweier benachbarter Bahnpunkte gegeben. Dieser Zeitschritt ist jedoch häufig bereits zu groß. Die Bestimmung eines geeigneten Zeitschritts geschieht durch die Limitierung des maximalen Wärmetransfers zwischen den Knoten. Wenn der Wärmetransfer über eine Kante des Netzes einen gegebenen Schwellwert überschreiten würde, wird die Anzahl der Schritte bestimmt, die nötig ist, damit pro Schritt nicht mehr Wärme als vom Schwellwert vorgegeben übertragen wird. Die Diffusion wird also pro eigentlichem Zeitschritt möglicherweise wesentlich häufiger, jedoch kleinschrittiger als der Auftrag und die anderen thermischen Komponenten berechnet. Als Schwellwert wird in dieser Arbeit ein Tausendstel der Temperatur der eintreffenden Partikel genutzt.

### 3.3.4 Kalibrierung

Wie in den vorhergehenden Abschnitten beschrieben, nutzt die thermische Simulation ein Modell, das verschiedene physikalische Prozesse kombiniert, um die thermische Situation auf der Werkstückoberfläche vorherzusagen. Zur Reduktion der Rechenzeit wird dabei jedoch zum Teil auf Näherungsmodelle zurückgegriffen. Daher soll das hier

genutzte Modell anhand einer exakteren externen Simulation, die auf einem Finite Elemente Ansatz basiert, kalibriert werden. Dadurch sollen die Abweichungen in den berechneten Temperaturen durch das hier entwickelte Modell und durch die exaktere externe Simulation minimiert werden.

Die Kalibrierung der thermischen Simulation wird mittels eines iterativen, sequenziellen Finite Differenzen Verfahrens realisiert. Ein Verfahren, das auf echten Gradienten basiert, ist nicht ohne Weiteres anwendbar, da sich die benötigten Gradienten nicht einfach aus den Rekursionsformeln des thermischen Modells herleiten lassen. Das Verfahren geht von einer Schätzung der Parameter aus und verändert diese so, dass die Abweichungen der Simulationsergebnisse zu denen der externen Simulation minimiert werden. Dazu wird ausgehend von den aktuellen Parametern der Reihe nach ein Parameter ausgewählt. Durch leichte Modifikation des Wertes in beide Richtungen und anschließende Simulation wird bestimmt, ob eine Verkleinerung oder Vergrößerung des Wertes eine bessere Übereinstimmung der Ergebnisse mit denen der externen Simulation ergibt. In der entsprechenden Richtung wird dann ein Line-Search Verfahren angewendet, um den Wert dieses Parameters weiter zu verbessern. Anschließend wird der nächste Parameter gewählt und das Verfahren neu gestartet. Da die Parameter nicht voneinander unabhängig sind, wird das Verfahren iteriert, bis sich keine wesentliche Verbesserung mehr erzielen lässt. Als in Frage kommende Parameter werden  $M_c$ ,  $M_k$ ,  $M_\epsilon$ ,  $M_\alpha$ , die Temperatur der Flamme sowie die Standardabweichung des gaußförmigen Spritzkegels genutzt. Obwohl es sich bei den meisten Parametern um Materialkonstanten handelt, werden diese durch die Optimierung möglicherweise nicht auf die tatsächlichen Werte abgebildet. Dies liegt an der Nutzung der vereinfachten Modelle. Das Erreichen der genauen Materialparameter ist allerdings auch nicht das Ziel der Optimierung. Vielmehr sollen die Parameter so eingestellt werden, dass das simulierte thermische Verhalten an der Werkstückoberfläche möglichst exakt an das reale Verhalten herankommt. Allerdings ist die Stärke der Abweichung der Parameter von den durch die externe Simulation bekannten Parameter auch ein Hinweis auf die Güte der Ersatzmodelle, die in dieser Arbeit verwendet werden. Daher werden diese Abweichungen in der Evaluation ebenfalls diskutiert.

In der Zielfunktion werden die Abweichungen der Temperaturen auf der Werkstückoberfläche verwendet. Da die thermische Situation auf dem Werkstück nicht nur nach Beendigung der Simulation, sondern vielmehr während des gesamten Prozesses von Bedeutung ist, kann nicht einfach das Endergebnis eines Simulationslaufs genutzt werden, wie es bei der Kalibrierung des Auftragsmodells gemacht wird. Vielmehr muss die thermische Situation auf der Oberfläche während des gesamten Prozesses betrachtet werden. Daher werden die Temperaturen auf der Oberfläche für alle Simulationsschritte gespeichert, so dass die entsprechenden Daten zur Auswertung der Zielfunktion zur Verfügung stehen.

Die zu minimierende Zielfunktion verwendet die Temperaturdifferenzen an allen Werkstückknoten zu allen Simulationszeitpunkten und kann wie folgt beschrieben werden:

$$E = \sum_{j=0}^n \sum_{i=0}^m (\Theta_{i,j} - \Theta_{i,j,e})^2. \quad (3.36)$$

Der quadratische Fehler gibt an, wie groß die simulativen Temperaturunterschiede auf der Werkstückoberfläche sind, wobei  $\Theta_{i,j}$  die Temperatur darstellt, die mit dem hier vorgestellten Simulationsmodell bestimmt wird, und  $\Theta_{i,j,e}$  die Temperatur beschreibt, die auf Grund der externen Simulation berechnet wird. Die Differenzen sollen entsprechend minimiert werden.

Da die externe Simulation nur die unbeladene Flamme simulieren kann und den Wärmeeintrag der Partikel nicht berücksichtigt, wird die entsprechende Komponente auch in dieser Simulation deaktiviert.

Alternativ zur Nutzung einer externen Simulation können auch reale Messdaten genutzt werden. Diese sind jedoch sehr schwierig zu ermitteln, da die Oberflächentemperatur nicht einfach gemessen werden kann. Die Flamme überstrahlt die Oberfläche bei Nutzung einer Thermo-Kamera und Messfühler können nicht einfach an die Werkstückoberfläche herangebracht werden. Dennoch kann bei Vorhandensein solcher Daten dasselbe Prinzip zur Kalibrierung genutzt werden.

### 3.4 Gesamtmodell

Bei der Simulation einer Bahn wird die Beschichtung, die von jedem einzelnen Bahnpunkt erzeugt wird, an jedem Knoten des Werkstücks ermittelt. Mit Hilfe der dabei bestimmten Schichtdicken, die auf Grund eines Bahnpunkts entstehen, kann die thermische Simulation angewendet werden. Diese bestimmt dann die Temperatur an jedem Oberflächenpunkt des Werkstücks zu dem entsprechendem Zeitpunkt. Aus den Temperaturen kann die thermische Varianz zu dem gegebenen Zeitpunkt bestimmt werden.

Wenn ein Robotermodell zur Verfügung steht, werden zusätzlich die Roboterposen, die zum Erreichen der Bahnpositionen und -ausrichtungen nötig sind, bestimmt. Für jeden Bahnpunkt werden über die inverse Kinematik die entsprechenden Roboterachsstellungen berechnet. Aus dem Verlauf dieser Daten können auch die Achsgeschwindigkeiten und die Achsbeschleunigung des Roboters bestimmt werden.

Eine Verringerung des Berechnungsaufwands der Simulation kann dadurch erreicht werden, dass die Simulation nicht für alle Paare von Bahn- und Oberflächenpunkten berechnet wird. Wenn das Skalarprodukt zwischen der Einschlagrichtung und der Oberflächennormalen (Gleichung (3.3)) negativ ist, wird kein Material abgelagert. Außerdem wird der maximale Radius des Spritzkegels bestimmt, bei dem noch Material aufgetragen oder Wärme eingetragen wird. Wenn der Oberflächenpunkt außerhalb dieses Bereichs liegt, kann auch hier die Berechnung abgebrochen werden.



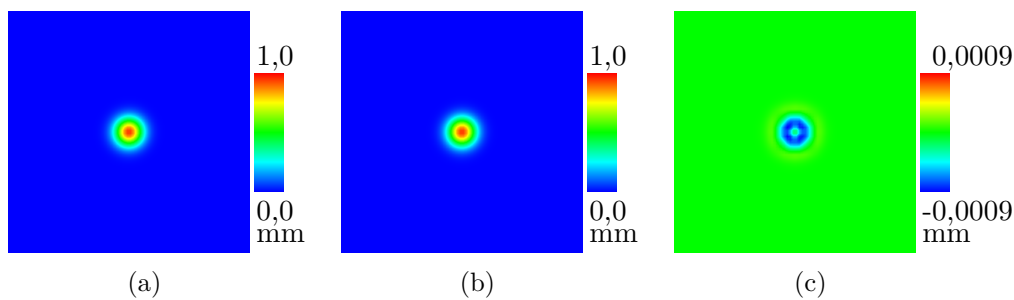


Abbildung 3.3: Kalibrierung eines Gauß-Footprints. (a) Ein gaußförmiger Footprint (max: 0,937551 mm). (b) Footprint des kalibrierten Modells (max: 0,937790 mm). (c) Die Differenz des originalen und des Modellfootprints. (min:  $-0,000916$  mm, max:  $0,000149$  mm)

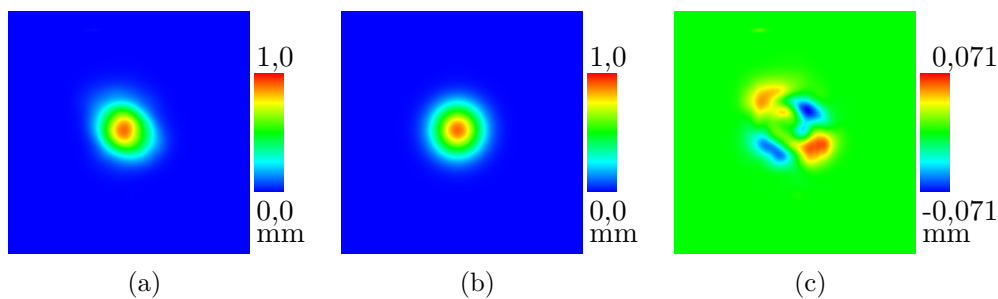


Abbildung 3.4: Kalibrierung eines Libo-Footprints. (a) Der reale Footprint erzeugt durch die externe Simulation (max:  $0,899721$  mm). (b) Footprint des kalibrierten Modells mit einer Gaußfunktion (max:  $0,899435$  mm). (c) Die Differenz des originalen und des Modellfootprints (min:  $-0,070661$  mm, max:  $0,061332$  mm).

## 3.5 Evaluation

In diesem Abschnitt werden die verwendeten Simulationen evaluiert. Dazu werden als Erstes die Ergebnisse der Kalibrierung für die Auftragssimulation untersucht. Anschließend wird der simulierte Auftrag auf einem komplexen Werkstück mit den Resultaten der externen Simulation, die auf real vermessenen Footprints beruht, verglichen. Zum Schluss werden die Ergebnisse der Kalibrierung des thermischen Modells evaluiert.

### 3.5.1 Auftragssimulation

Auf Basis des in Abschnitt 3.2.3 vorgestellten Verfahrens zur Kalibrierung werden die Modellparameter der Auftragssimulation bestimmt. Hierzu werden zwei unterschiedliche Footprints untersucht. Bei dem ersten Footprint handelt es sich um einen synthetischen, gaußförmigen Footprint, der zweite ist ein real vermessener Libo-Footprint (Lichtbogen Spritzprozess). Die Daten der Schichtdicken der Footprints, die für die Kalibrierung benötigt werden, stammen aus der externen Simulation *SimSpray*.

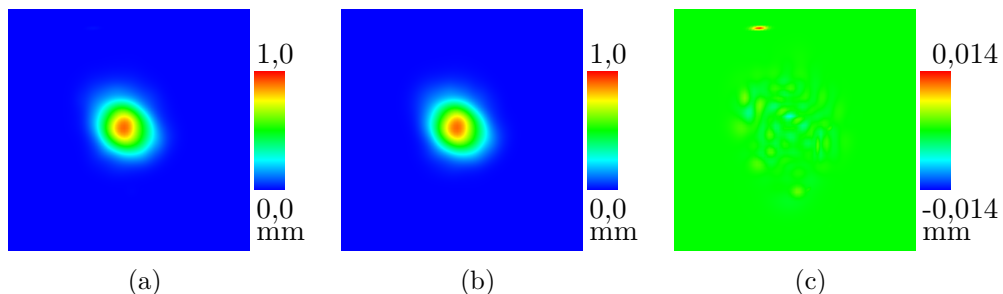


Abbildung 3.5: Kalibrierung eines Libo-Footprints. (a) Der reale Footprint erzeugt durch die externe Simulation (max: 0,899721 mm). (b) Footprint des kalibrierten Modells mit 20 Gaußfunktionen (max: 0,899358 mm). (c) Die Differenz des originalen und des Modellfootprints (min:  $-0,006427$  mm, max: 0,014106 mm).

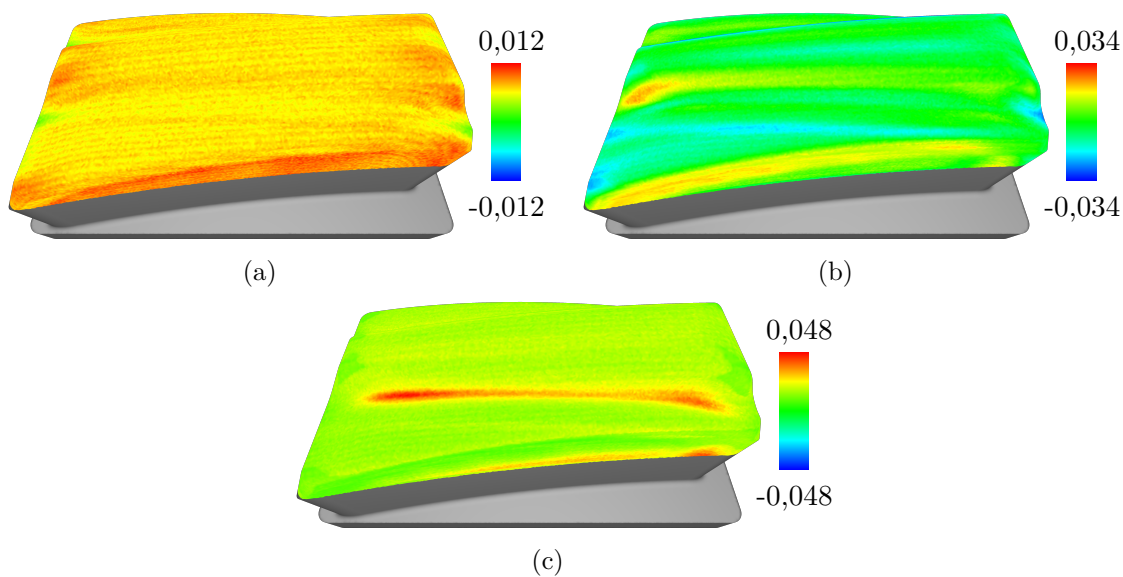


Abbildung 3.6: Relativer Auftragsfehler einer Bahn auf einem komplexen Werkstück für (a) den Gauß-Footprint (min: 0,001239, max: 0,011514), (b) den Libo-Footprint mit einer Gaußfunktion (min:  $-0,034264$ , max: 0,031597) und (c) den Libo-Footprint mit 20 Gaußfunktionen (min: 0,002235, max: 0,047583).

Die Kalibrierung der Modellparameter soll dafür sorgen, dass der durch den Footprint gegebene Massefluss durch das Modell möglichst gut approximiert wird. Ausschlaggebend hierfür ist neben den Parametern auch die Anzahl der verwendeten Gaußfunktionen. Je mehr Funktionen verwendet werden, desto besser können beliebige Footprints abgebildet werden. Auf der anderen Seite ist es erstrebenswert, möglichst wenige Funktionen zu nutzen, um den Footprint durch das Modell zu generalisieren. Die Anzahl der benötigten Funktionen hängt stark vom Anwendungszweck ab. Bei der Initialbahnplanung wird das Modell zur schnellen, approximativen Auftragsbestimmung genutzt, wobei leichte Fehler akzeptabel sind. Bei der Bahnoptimierung wird das Modell für die Gradientenschätzung genutzt, während der Auftrag durch eine externe Simulation bestimmt wird. Auch hier muss das Modell nicht vollständig exakt sein. Die Auswirkungen von approximativen Modellparametern wird an den entsprechenden Stellen daher ausführlich evaluiert.

Im Folgenden wird die Kalibrierung für den synthetischen Gauß-Footprint mit einer einzelnen Gaußfunktion und für den realen Libo-Footprint sowohl mit nur einer als auch mit 20 überlagerten Gaußfunktionen untersucht.

Für den gaußförmigen Footprint werden die Parameter des Modells durch die Kalibrierung so bestimmt, dass der Footprint beinahe fehlerfrei abgebildet wird. In Abbildung 3.3 sind die Footprints jeweils mit Hilfe einer Farbskala gegenübergestellt. Der Footprint der externen Simulation erzeugt eine maximale Auftragsdicke von 0,937551 mm, während der Modellfootprint eine Maximaldicke von 0,937790 mm erzeugt. Die maximale Differenz ist gerade einmal 0,916  $\mu\text{m}$ . Die ermittelten Parameter sind in Tabelle 3.1 zusammengestellt.

Die Kalibrierung des real vermessenen Libo-Footprints wird zunächst für ein Modell mit einer einzelnen Gaußfunktion durchgeführt. Die maximale Schichtdicke für den realen Footprint ergibt sich zu 0,899721 mm während das kalibrierte Modell einen Maximalwert von 0,899435 mm erreicht. Die maximale Abweichung zwischen Modell und Footprint ergibt sich zu 70,661  $\mu\text{m}$ . Die Footprints und die Abweichung der Footprints sind in Abbildung 3.4 und die Parameter in Tabelle 3.1 zusammengestellt.

Bei der letzten Kalibrierung wird das Modell mit 20 Gaußfunktionen genutzt. Die maximale Schichtdicke ist hier 0,899358 mm und die maximale Abweichung ist 14,106  $\mu\text{m}$ . Abbildung 3.5 stellt die Footprints und die Abweichung grafisch gegenüber und in Tabelle 3.1 sind die ermittelten Parameterwerte zu finden.

Die Abweichungen der Schichtdicken zwischen der externen Simulation und den jeweils kalibrierten Simulationsmodellen für eine Bahn auf einem komplexen Werkstück sind in Abbildung 3.6 für beide analysierten Footprints dargestellt. Die maximalen prozentualen Abweichungen in allen Testfällen liegen im Bereich von  $\pm 4,8\%$ , was für den geplanten Einsatzzweck der Simulation einen guten Wert darstellt. Für die Testfälle mit dem synthetischen Gauß-Footprint und dem Libo-Footprint bei Nutzung von 20 Gaußfunktionen kann eine systematische Überbeschichtung durch das Modell festgestellt werden. Dieser Fehler liegt nicht an der Kalibrierung des Modells, was an den sehr kleinen Fehlern bei der Differenz der Footprints zu sehen ist, sondern an dem Beschichtungsmodell selbst, das für die hier benötigten Anwendungsfälle vor allem im Bereich der Bahnoptimierung

Tabelle 3.1: Parameter der Modelle nach der Kalibrierung. Gauß: Modell mit einer Gaußfunktion für einen synthetischen Footprint. Libo: Modell mit einer Gaußfunktion für einen Libo-Footprint. Libo20: Modell mit 20 Gaußfunktion für einen Libo-Footprint.

Footprint	$c_k$	$\sigma_{1,k}$	$\sigma_{2,k}$	$\sigma_{3,k}$	$\mu_{q,k}$	$\mu_{r,k}$
Gauß	11622,24	0,041594	0,041594	$10^{10}$	0,000000	0,000000
Libo	10908,26	0,072389	0,073717	$10^{10}$	0,013266	0,007815
Libo20-1	-688,73	0,055891	0,052171	$10^{10}$	0,020644	0,123502
Libo20-2	-3082,23	0,047065	0,046589	$10^{10}$	0,030011	0,064813
Libo20-3	145,93	0,043441	0,023804	$10^{10}$	0,076630	0,072140
Libo20-4	-1888,90	0,029759	0,044034	$10^{10}$	0,081715	-0,017741
Libo20-5	6035,52	0,037685	0,040848	$10^{10}$	0,073520	-0,020117
Libo20-6	-3064,83	0,038198	0,036025	$10^{10}$	0,066473	-0,023480
Libo20-7	899,96	0,051145	0,060474	$10^{10}$	0,093513	-0,083185
Libo20-8	-609,63	0,035715	0,021504	$10^{10}$	0,034520	-0,092383
Libo20-9	5091,43	0,046234	0,060213	$10^{10}$	-0,014011	-0,065350
Libo20-10	-5902,29	0,037018	0,049304	$10^{10}$	-0,011711	-0,087382
Libo20-11	1929,85	0,030385	0,041992	$10^{10}$	-0,012098	-0,102273
Libo20-12	-821,19	0,031222	0,031757	$10^{10}$	-0,044132	-0,062371
Libo20-13	952,12	0,028864	0,065852	$10^{10}$	-0,096416	-0,011811
Libo20-14	-3654,99	0,034002	0,054732	$10^{10}$	-0,089148	0,000522
Libo20-15	4526,88	0,042011	0,052140	$10^{10}$	-0,077218	0,007520
Libo20-16	-582,07	0,022591	0,035548	$10^{10}$	-0,062319	0,034224
Libo20-17	1795,32	0,056315	0,040502	$10^{10}$	-0,027535	0,053833
Libo20-18	-713,83	0,050798	0,026516	$10^{10}$	-0,022188	0,060054
Libo20-19	-700,27	0,037121	0,044224	$10^{10}$	-0,007722	0,084814
Libo20-20	10143,13	0,057193	0,089093	$10^{10}$	0,019382	0,006956

einfacher gehalten ist und nicht alle Komponenten enthält, die in der externen Simulation berücksichtigt werden.

### 3.5.2 Thermische Simulation

Durch die Kalibrierung der thermischen Simulation soll das Modell einen Temperaturverlauf an allen Stellen der Oberfläche liefern, der dem Verlauf der externen Simulation möglichst nahekommt. Dabei werden die Werte der in Abschnitt 3.3.4 vorgestellten Parameter so bestimmt, dass die Abweichung der Verläufe minimal wird.

Für die Kalibrierung wird als Werkstück eine einfache Flachprobe mit einer planaren Oberfläche gewählt. Die Werkstückdicke beträgt 15 mm. Die Bahn wird so ausgewählt, dass die Spritzpistole für 10s auf die Mitte der Werkstückoberfläche ausgerichtet ist. Anschließend wird die Simulation ohne Einfluss der Spritzpistole für weitere 10s laufen gelassen, damit sich das Werkstück abkühlen kann.

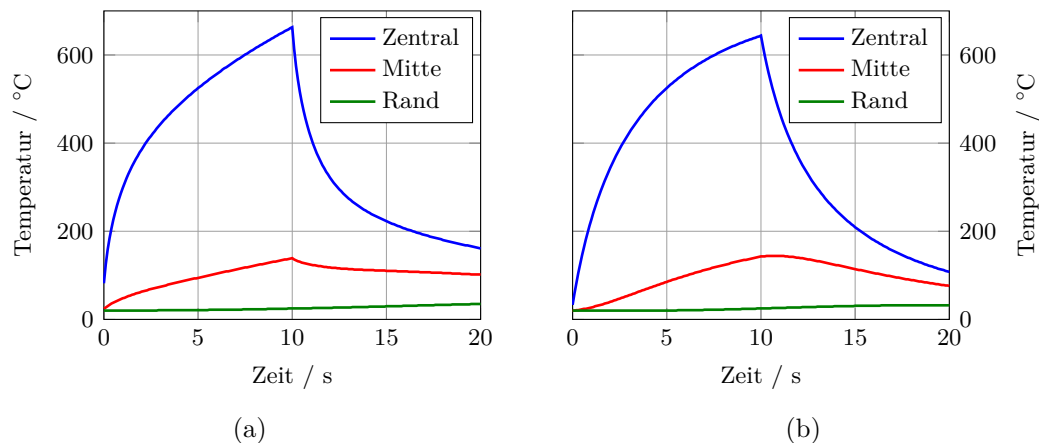


Abbildung 3.7: Temperaturentwicklung auf einer Flachprobe simuliert (a) mit einer externen Simulation und (b) mit der hier entwickelten Simulation. Gezeigt sind die Temperaturverläufe an drei Stellen der Oberfläche. Die Spritzpistole wurde nach 10 Sekunden abgeschaltet.

Tabelle 3.2: Parameter der thermischen Simulation ( $\diamond$ : Nicht variabel. \*: Konstanter Term mit Wert 422,0 plus linearer, quadratischer und kubischer Term. \*: Konstanter Term mit Wert 58,4 plus linearer, quadratischer und kubischer Term.  $\dagger$ : Zwei Werte mit 0,89 und 0,91.  $\ddagger$ : andere Definition bezüglich Footprint-Ebene.)

Parameter	Wert des Modells	Wert der ext. Sim.	Einheit
Temperatur Flamme	2868,3462	3000,0000	$^{\circ}\text{C}$
Werkstückdicke $\diamond$	15,0000	15,0000	mm
Lufttemperatur $\diamond$	20,0000	20,0000	$^{\circ}\text{C}$
$M_{\rho}$	6,3100	7,8500	$\text{g}/\text{cm}^3$
$M_c$	416,3352	422,0000*	$\text{J}/(\text{kg K})$
$M_k$	51,5800	58,4000*	$\text{W}/(\text{m K})$
$M_{\epsilon}$	1,0000	0,9000 $\dagger$	
$M_{\alpha}$	0,0014	0,0020	$\text{W}/(\text{m}^2 \text{K})$
std.var.	9,3281	15,0000 $\ddagger$	

In Tabelle 3.2 sind die optimierten Parameterwerte nach der Kalibrierung zusammengestellt. Die ermittelten Werte stimmen sehr gut mit den Parameterwerten überein, die in der externen Simulation genutzt werden, obwohl die Parameter zum Teil nur schwer miteinander verglichen werden können, da die externe Simulation neben den weiterreichenden Diffusionsmodellen auch andere Effekte, wie lineare, quadratische und kubische Terme für die Materialkonstanten verwendet. Die jeweiligen konstanten Anteile sind in der Tabelle angegeben und stimmen von ihren Werten her sehr gut mit den optimierten Parametern überein. Hieraus kann gefolgert werden, dass das hier genutzte Simulationsmodell die wichtigsten physikalischen Effekte trotz der Vereinfachungen gut abbildet.

In Abbildung 3.7a sind exemplarisch drei Temperaturverläufe für die exaktere externe Simulation aufgetragen. Als Messpunkte werden drei Knoten des Werkstücknetzes genutzt, wobei der erste fast direkt zentral unter dem Mittelpunkt der Flamme liegt, der zweite 5 cm neben dem ersten und der dritte weitere 5 cm versetzt. Das Werkstück hat eine Anfangstemperatur von 20 °C. Sobald die Spritzpistole eingeschaltet wird, wärmt sich das Werkstück direkt unter der Flamme stark auf. Dabei ist der Temperaturanstieg direkt unter der Flamme am schnellsten. Nach dem Abschalten der Spritzpistole fällt die Temperatur ähnlich einer inversen Exponentialfunktion ab. Die weiter außen liegenden Knoten werden je nach Abstand zum Mittelpunkt des Spritzkegels weit weniger von der Flamme beeinflusst. Daher verläuft der Temperaturanstieg deutlich schwächer und wird weiter außen vornehmlich von der Diffusion getragen. Auch nach Abschalten der Spritzpistole erwärmen sich die äußeren Punkte noch weiter, da die Wärmeenergie aus der Mitte des Werkstücks stärker nach außen diffundiert, als sich die äußeren Punkte abkühlen.

In Abbildung 3.7b sind die Verläufe der Temperaturen dargestellt, die durch die Simulation mit den Parametern aus Tabelle 3.2 für die gleichen Knoten berechnet werden wie für die externe Simulation. Das qualitative Verhalten der Kurven aus der externen Simulation lässt sich hier sehr gut wiederfinden. Zunächst steigen die Temperaturen der Oberflächenpunkte direkt unter der Flamme stark unter dem Einfluss der Spritzpistole an. Nach Abschalten der Spritzpistole fallen die Kurven qualitativ wie eine inverse Exponentialfunktion ab. Auch die Kurven der weiter außen liegenden Knoten zeigen ein sehr ähnliches Verhalten, das stark von der Diffusion beeinflusst wird. Auf Grund der einfacheren Modelle stimmen die Kurven jedoch nicht vollkommen exakt überein. Eine Fehlerquelle ergibt sich dadurch, dass nicht wie bei der externen Simulation ein Tetraedermodell und eine Finite Elemente Methode genutzt wird, um die Diffusion ins Innere des Bauteils zu berechnen. Stattdessen wird die Diffusion über einen Wärmeausgleich approximiert.

Obwohl die Nutzung des einfacheren Modells zu geringen Abweichungen in den Temperaturbestimmungen führt, sollen diese Abweichungen in Kauf genommen werden, da die einfacheren Modelle zu deutlich geringeren Rechenzeiten führen, so dass viele Simulationen in der Evolutionären Optimierung, welche in der Initialbahnplanung genutzt wird, durchgeführt werden können. Die Rechenzeit für die externe Simulation für das gegebene

Szenario beträgt ca. 101 s trotz Nutzung der GPU, während die gleiche Simulation mit dem hier genutzten Modell ca. 0,015 s benötigt.

Trotz der kleineren Ungenauigkeiten in dem Modell, die auf Grund der Näherungsrechnungen auch zu erwarten sind, stimmen die Temperaturverläufe qualitativ sehr gut überein. Auch wenn exakte Vorhersagen der Temperatur günstig sind, sind genauere Temperaturwerte für den Anwendungsfall der Bahnplanung nicht notwendig. Die Bahnplanung berücksichtigt bei der Optimierung die Temperaturvarianzen und versucht diese im Verlauf der Optimierung zu verringern. Für dieses Vorhaben wird der absolute Betrag der Temperatur nicht benötigt, sondern nur eine verlässliche Möglichkeit, den Temperaturverlauf, der von verschiedenen Bahnen erzeugt wird, zu vergleichen. Dieses Ziel wird mit dem angegebenen Modell erreicht. Des Weiteren ist eine geringe Rechenzeit für die Menge an notwendigen Simulationen für die Initialbahnplanung von großer Bedeutung, so dass ein weiteres wichtiges Ziel mit diesem Simulationsmodell erreicht wird.

### 3.6 Fazit

In diesem Kapitel ist ein Simulationsmodell vorgestellt worden, das in der Lage ist, sowohl den Auftrag als auch die Temperaturen, die während des Prozesses auf dem Werkstück entstehen, in für die Initialbahnplanung und Bahnoptimierung geeigneter Qualität zu simulieren. Das Auftragsmodell orientiert sich an bekannten Modellen der Literatur und erweitert diese an bestimmten Stellen. Das Modell ist des Weiteren darauf ausgelegt, Gradienten für die Bahnoptimierung zur Verfügung zu stellen. Das Modell wird für einen konkreten Prozess mittels Footprint-Daten kalibriert und kann demnach für verschiedene Prozesse eingesetzt werden.

Das thermische Modell berücksichtigt die physikalischen Prozesse, die für das Aufwärmen, Abkühlen und für die Wärmediffusion verantwortlich sind. Das Modell verwendet an wenigen Stellen anstelle exakter Rechnungen einfachere Ersatzmodelle, um eine schnelle Berechnung zu gewährleisten. Mittels eines Vergleichs mit einer externen Simulation ist die qualitative Korrektheit des Modells jedoch bestätigt worden.

Zusammenfassend kann festgehalten werden, dass die Simulationsmodelle ausreichend exakte Ergebnisse in einer für die Optimierung geeigneten Zeit liefern.





# Initialbahnplanung

In diesem Kapitel wird die Initialbahnplanung vorgestellt, deren Aufgabe die Generierung einer flächenüberdeckenden Bahn für ein gegebenes Werkstück ist, die die geforderten Zielkriterien bestmöglich einhält. In der Literatur zur Offline Bahnplanung wird normalerweise nicht zwischen Initialbahnplanung und Bahnoptimierung unterschieden. Dies liegt an den häufig verwendeten, recht einfachen Bahnstrukturen, deren Parameter mittels einer Optimierung in einem Schritt angepasst werden. Ein prominentes Beispiel für eine derartige Bahnstruktur sind Mäanderbahnen. In dieser Arbeit wird abweichend von den herkömmlichen Strategien ein neuartiges Konzept zur Trennung der Bahnplanung in eine Initialbahnplanung und eine Bahnoptimierung vorgestellt. Die Initialbahnplanung ist für das Generieren einer initialen Bahn verantwortlich, setzt jedoch, im Gegensatz zu üblichen Verfahren, deutlich komplexere Bahnstrukturen ein. Hierdurch ist es möglich, die Bahn flexibel an die Gegebenheiten komplexer Werkstückgeometrien anzupassen, was zum Beispiel mit Mäanderbahnen nur in sehr begrenztem Umfang möglich ist. Vor allem im Bereich der Minimierung der Roboterachsbeschleunigungen ist eine solche Flexibilität von großem Vorteil. Nach der Fertigstellung der initialen Bahn, die die globale Struktur der Bahn festlegt, wird diese anschließend von der Bahnoptimierung lokal optimiert, um Zielkriterien bestmöglich zu erreichen, siehe Kapitel 5.

In Abschnitt 4.1 wird eine kurze Einführung in die Initialbahnplanung gegeben. Anschließend wird in Abschnitt 4.2 der Stand der Forschung im Bereich der Bahnplanung und insbesondere im Bereich der Bahnplanung unter Berücksichtigung von Wärme vorgestellt. Es folgt eine Übersicht über den Aufbau der Initialbahnplanung in Abschnitt 4.3. Abschnitt 4.4 stellt das Verfahren als Optimierungsproblem vor und Abschnitt 4.5 beschreibt drei Möglichkeiten der Bahngenerierung. In Abschnitt 4.6 wird die Zielfunktion mathematisch beschrieben. Die Optimierung geschieht mittels eines Evolutionären Algorithmus, dessen grundlegender Aufbau in Abschnitt 4.7 dargestellt wird. In Abschnitt 4.8 wird der Zusammenhang der Problemstruktur mit bekannten Problemstellungen genauer untersucht. Die konkreten Realisierungen der Algorithmen für die drei Bahnstrukturen

werden in den Abschnitten 4.9-4.11 beschrieben. Schließlich werden die vorgestellten Verfahren in Abschnitt 4.12 evaluiert.

## 4.1 Einleitung

Das Ziel der Initialbahnplanung ist die Generierung flächenüberdeckender Bahnen auf komplexen Werkstücken für das thermische Spritzen. Eine so erstellte Bahn soll sich dadurch auszeichnen, dass durch ihre Ausführung die vorgegebenen Schichtdicken auf dem Werkstück aufgetragen werden und die thermischen Varianzen, die während des Prozesses auf der Oberfläche entstehen, gering gehalten werden. Des Weiteren soll die Bahn gute kinematische Eigenschaften aufweisen, so dass ein Roboter, der für den echten Prozess verwendet wird, die Bahn ohne Probleme ausführen kann. Als letztes Kriterium soll der Overspray, der entsteht, wenn neben die zu beschichtende Oberfläche gespritzt wird, gering gehalten werden. Alle genannten Zielkriterien können über die zur Verfügung stehenden Simulationen (siehe Abbildung 1.4) ausgewertet werden.

Bisherige Ansätze zur Bahnerzeugung setzen häufig einfache Bahnstrukturen ein (siehe Abschnitt 4.2), wobei ein prominentes Beispiel einfache Mäanderbahnen sind. Diese Bahnstrukturen zeichnen sich dadurch aus, dass sie mit wenigen Parametern beschrieben werden können, wodurch es möglich wird, die optimale Bahn bezüglich dieser Bahnstruktur durch die Optimierung der Parameter zu finden. Diese Art der Bahngenerierung ist für manche Prozesse (Lackieren, Fräsen) und einfache Werkstückgeometrien (ebene Oberflächen) sehr gut geeignet. Wenn allerdings die Werkstückgeometrie komplex ist, dann reichen einfache Bahnstrukturen nicht mehr aus. Zum Beispiel kann sich die maximal mögliche Oberflächengeschwindigkeit beim Überfahren einer konvex gekrümmten Oberfläche auf Grund der kinematischen Einschränkungen des Roboters deutlich reduzieren, so dass beim thermischen Spritzen zu viel Material an diesen Stellen aufgetragen wird. Ein weiterer Nachteil von einfachen Mäanderbahnen ist die hohe thermische Varianz, die in dem Werkstück entsteht, siehe Abschnitt 6.5.3. Daher sind in diesen Fällen alternative Ansätze notwendig.

Die Initialbahnplanung, die in diesem Kapitel vorgestellt wird, soll die genannten Defizite herkömmlicher Bahnstrukturen überwinden. Dazu wird die Initialbahnplanung als erster von zwei Schritten verstanden, die die vollständige Bahnplanung darstellen. Die Initialbahnplanung soll auf der Basis von komplexen Bahnstrukturen arbeiten, die weitaus flexibler sind als herkömmliche Mäanderbahnen. Dadurch sind wesentlich bessere Lösungen möglich, jedoch ist der Parameterraum so groß, dass dieser auch mit globalen Optimierungsmethoden nicht vollständig in akzeptabler Zeit untersucht werden kann. Daher soll sich die Initialbahnplanung darauf konzentrieren, die globale Struktur der Bahn zu optimieren. Die Bahnoptimierung (siehe Kapitel 5) kann anschließend die Bahn lokal verändern, um die Zielkriterien bestmöglich zu erfüllen. Ein wesentliches Zielkriterium, das von der globalen Struktur der Bahn abhängt, ist die thermische Varianz, die durch die lokale Änderung der Bahn kaum beeinflusst werden kann, siehe Abschnitt 4.9.14. Daher soll die Initialbahnplanung dieses Zielkriterium besonders beachten.

## 4.2 Stand der Forschung

Im Folgenden wird eine Übersicht über verwandte Arbeiten im Bereich der Bahnplanung beim thermischen Spritzen gegeben. Dazu werden zunächst allgemeine Verfahren vorgestellt, bevor anschließend speziell auf die Bahnplanung unter Berücksichtigung der Wärme eingegangen wird.

### 4.2.1 Bahnplanung beim thermischen Spritzen

Im Gegensatz zur Bahnplanung beim thermischen Spritzen ist ein verwandter Prozess, das Lackieren, deutlich mehr in der Literatur untersucht worden. Eine Übersicht über Verfahren zur Bahnplanung beim Lackieren ist in [35] zu finden. Das Lackieren hat viele Gemeinsamkeiten mit dem thermischen Spritzen, so dass die dort entwickelten Verfahren auch für das thermische Spritzen interessant sein können. So befassen sich einige Arbeiten mit dem Aufstellen von Auftragsmodellen, die den Massefluss der Partikel beschreiben und damit die Grundlage einer Bahnplanung darstellen können [39]. Für die eigentliche Bahnplanung werden häufig feste Bahnstrukturen genutzt, wie zum Beispiel Mäanderbahnen, deren Parameter über verschiedene Methoden festgelegt werden. Atkar et al. [8] nutzen zum Beispiel geodätische Abstandsmaße zur Bestimmung des Abstandes zwischen Überläufen und Wang et al. [167] bestimmen den idealen Abstand der Scans um eine möglichst homogene Schichtdicke zu erreichen. Auch andere fortgeschrittene Fragestellungen werden untersucht, wie die Segmentierung von großen Werkstücken [7] oder die Kooperation von Robotern mittels Ameisenalgorithmen [65]. Eine vollständige Software namens Flex Paint wird in [163] beschrieben. Es gibt aber auch einige wichtige Unterschiede zwischen dem Lackieren und dem thermischen Spritzen, die die Verfahren des Lackierens für das thermische Spritzen ungeeignet machen. Zum Beispiel ist das Einhalten von genauen Schichtdicken beim Lackieren weniger wichtig, da häufig nur eine relativ gleichmäßige Deckung der Oberfläche nötig ist. Bei thermisch gespritzten Schichten muss die Kontur des Werkstücks, je nach späterem Einsatzbereich, sehr genau mit den Vorgaben übereinstimmen. Als Beispiel sei hier eine Verschleißschicht für Werkzeuge im Bereich des Tiefziehens genannt, die hohe Anforderungen an die Endkontur stellt. Ein anderes Problem beim thermischen Spritzen ist die thermische Belastung des Werkstücks, welche beim Lackieren keine Rolle spielt. Häufig verwendete Mäanderbahnen oder ähnliche Bahnstrukturen sind daher für das thermische Spritzen auf komplexen Werkstückgeometrien ungeeignet, wie die Analysen in Abschnitt 6.5.3 zeigen.

Auch im Bereich des thermischen Spritzens werden häufig feste Bahnstrukturen genutzt. Beispiele hierfür sind Mäanderbahnen [59], parallele und quasi-parallele Bahnen oder kontur- bzw. quasi-konturparallele Bahnen, siehe [102, 103, 100]. Diese werden vor allem für geometrisch einfache Werkstücke genutzt, wie zum Beispiel planare Flächen oder einfachere extrudierte Kurven. Es ist bekannt, dass parallele Bahnen auf diesen Flächen sehr gute Beschichtungsergebnisse liefern, da bei geeignetem Bahnabstand eine sehr

homogene Schichtdicke erzeugt werden kann. Ein Problem auf komplexeren Flächen ist, dass der Bahnabstand adaptiv verändert werden muss, um diese homogene Beschichtung zu erreichen, siehe hierzu zum Beispiel [102, 103, 100]. Ein dabei wenig beachtetes Problem ist jedoch der Wärmeeintrag in das Werkstück und die Schicht, der an komplexen Stellen des Werkstücks, an denen geringe Bahnabstände oder auf Grund der Eigenschaften des Roboters nur geringe Geschwindigkeiten erreicht werden können, sehr groß werden kann. Ein großer Wärmeeintrag kann zu großen thermischen Varianzen führen, was wiederum zu Spannungen in dem Werkstück und der Schicht führen kann. Diese Spannungen vermindern die Schichtqualität und können im Extremfall zu Abplatzungen und damit zur Zerstörung der Schicht führen. Das Problem verstärkt sich mit zunehmend komplexen Werkstücken, die starke Krümmungen aufweisen. Ein alternativer Ansatz, der die klassischen Bahnstrukturen zu Gunsten geringerer thermischer Varianz auflöst, wird in [84] vom Autor dieser Arbeit beschrieben. Hier werden Rand-zu-Rand-Bahnen erzeugt, die explizit auf geringe thermische Varianz hin optimiert werden. Dieser Ansatz, der in der genannten Arbeit nur auf planaren, konvexen Werkstücken funktioniert, dient als Grundlage des hier entwickelten Algorithmus. Die Bahnplanung mit Rand-zu-Rand-Bahnen, aber auch mit anderen Bahnstrukturen, beinhaltet eine Komponente, die an das Travelling Salesman Problem (Problem des Handlungsreisenden, TSP) erinnert. Hier soll eine Rundreise gefunden werden, die alle gegebenen Städte mit geringst möglicher Reiselänge besucht [4]. Das TSP-Problem ist sehr gut untersucht und es gibt viele exakte und heuristische Lösungsansätze, wie zum Beispiel Genetische Algorithmen, siehe [104]. Die Verbindung zur Bahnplanung beim thermischen Spritzen ist die Überdeckungskomponente. Jeder Punkt auf der Oberfläche muss besucht werden, jedoch ist der Unterschied, dass der Spritzkegel ausgedehnt ist, und ein ausgedehnter Teil der Oberfläche gleichzeitig beschichtet wird. Auch muss jeder Punkt eine genaue Zielschichtdicke erreichen, welche eventuell mehrere Überläufe, also Besuche, erfordert. Dies ist ähnlich dem Close Enough Travelling Salesman Problem oder auch dem Covering Salesman Problem [41, 73, 75]. Eine andere Komponente beim thermischen Spritzen ist die Wärmevarianz, die durch ungleichmäßigen Wärmeeintrag entsteht. Dies deutet eine Abhängigkeit des Beschichtungsvorgangs von der Zeit an. Daraus kann auf einen Zusammenhang mit der Zeitabhängigkeit der Bearbeitung von Jobs bei Scheduling-Problemen geschlossen werden.

Ähnlich wie das Bahnplanungsproblem beim thermischen Spritzen mit dem Close Enough Travelling Salesman Problems verwandt ist, kann dieses auch mit dem Coverage Path Planning in Verbindung gesetzt werden. Beim Coverage Path Planning wird eine Bahn für einen Roboter gesucht, die ein gegebenes Gebiet vollständig überdeckt [67]. Beispiele für Anwendungsfälle sind Staubsauger- oder Rasenmäher-Roboter [43]. Auch wenn die Überdeckungskomponente in diesem Problem gut untersucht wird, fehlt genau wie bei dem Close Enough TSP eine vergleichbare Komponente zum Wärmeeintrag und der thermischen Varianz.

Andere Arbeiten im Bereich des thermischen Spritzens betrachten unterschiedliche speziellere Problemstellungen. Cai et al., [27] untersuchen die Einflüsse verschiedener Parameter, wie Spritzabstand und Bahnabstand beim Kaltspritzen, um die Erkenntnisse

zur Bahnoptimierung zu nutzen. Dazu wird der Footprint des Prozesses analysiert, um den Spritzabstand und den Bahnabstand der genutzten Mäanderbahnen so einzustellen, dass die Schichtdicke optimiert wird. Andere Bahnstrukturen werden nicht untersucht.

Ein weiterer Aspekt, der betrachtet worden ist, ist die Beschichtung von rotations-symmetrischen Werkstücken, siehe zum Beispiel [78]. Diese können beschichtet werden, indem eine externe Achse benutzt wird, um das Werkstück während des Prozesses zu drehen. Damit kann eine Erreichbarkeit von allen Seiten gewährleistet werden. Das Verfahren eignet sich allerdings nur für rotations-symmetrische oder nahezu rotations-symmetrische Werkstücke. Dennoch ist das Vorhandensein einer externen Achse für Beschichtungsprozesse vorteilhaft, um das Problem der Erreichbarkeit mit Robotern zu reduzieren.

Eine alternative Strategie, die auch für die Verminderung von Erreichbarkeitsproblemen eingesetzt werden kann, ist die Segmentierung von großen Werkstücken. Für einen verwandten Prozess, dem Sprühkompaktieren, ist ein solcher Ansatz in [36] vorgestellt worden. Dieser nutzt Mäanderbahnen auf jedem Segment des Werkstücks und nutzt eine Optimierung, um die einzelnen Segmente optimal zusammenzufügen.

Die meisten Arbeiten für die Bahnplanung beim thermischen Spritzen berücksichtigen den Einsatz von Robotern. Ein allgemeiner Einblick in die Nutzung von Industrierobotern beim thermischen Spritzen sowie Methoden zu deren Programmierung sind in [47] zusammengetragen. Die gleichen Autoren haben später in [48] eine Arbeit zur Kinematikanalyse des Roboters vorgestellt. Diese kann dazu genutzt werden, kinematische Probleme einer geplanten Bahn zu erkennen und die Stellen der Bahn, die diese Probleme verursachen, zu überarbeiten. In dieser Arbeit werden ähnliche Kriterien berücksichtigt, indem zunächst eine Roboterplatzierung und später eine Minimierung der Roboterachsbeschleunigungen durchgeführt wird.

Bahnplanungsmethoden werden auch in vielen anderen Bereichen genutzt, die nur wenig mit dem thermischen Spritzen gemeinsam haben. Spezielle Eigenschaften des thermischen Spritzens werden in diesen Methoden entsprechend nicht berücksichtigt. Dennoch können die Ansätze interessante Ideen für die Übertragung auf das thermische Spritzen liefern. Im Bereich des Fräsens, siehe zum Beispiel [105] für eine Übersicht, ist die primäre Zielsetzung die Minimierung der Rauhtiefe bei gleichzeitiger Minimierung der Bahnlänge [63, 107]. Häufig werden auch hier Mäanderbahnen genutzt, deren Bahnabstand für die akzeptable Rauhtiefe optimiert wird. Leerfahrten an bereits bearbeiteten Stellen werden reduziert, um so die Bahnlänge zu verringern. Die Optimierung von Mäanderbahnabständen wird in ähnlicher Form bei einigen Verfahren für das thermische Spritzen genutzt. Auch die Reduktion von Leerfahrten ist ähnlich dem Ziel der Minimierung von Overspray, der entsteht, wenn neben das Werkstück gespritzt wird, zum Beispiel um die Spritzpistole ohne Ablagerungen auf dem Werkstück an eine andere Stellen der Oberfläche zu bewegen. Eine Leerfahrt über das Werkstück ist in vielen Fällen nicht möglich, da die Spritzpistole meist nicht für kurze Zeit deaktiviert werden kann.

### 4.2.2 Bahnplanung unter Berücksichtigung der Wärme

Für die Bahnplanung beim thermischen Spritzen spielt das Wärmeverhalten auf der Oberfläche eine spezielle Rolle. Durch die heißen Partikel und die Flamme der Spritzpistole wird sehr viel Wärme lokal in das Werkstück eingetragen, während das Werkstück durch verschiedene Prozesse wie Konvektion, Diffusion und Strahlung an anderen Stellen abkühlt. Durch die zum Teil erhebliche Wärmevarianz entstehen Spannungen in dem Werkstück, der bereits aufgetragenen Schicht und an der Grenze zwischen Schicht und Werkstück. Ungleichmäßige oder starke Spannungen führen zu einer Reduktion der Qualität der Schicht und im Extremfall zum Abplatzen der Schicht. Dieses Problem wird um so größer, je komplexer das Werkstück geformt ist. An Radien ist zum Beispiel eine Umorientierung der Spritzpistole notwendig, die zum Teil erhebliche Einbußen der Oberflächengeschwindigkeit und damit einen sehr großen Wärmeeintrag verursacht. Wenn durch die komplexe Form keine Richtung gefunden werden kann, die solche Probleme komplett verhindern kann (zum Beispiel durch Drehung der Mäanderrichtung), müssen alternative Bahnstrukturen genutzt werden.

Candel und Gadow nutzen Mäanderbahnen zur Bahnplanung auch auf komplexeren Flächen, wie den in [29] und [66] untersuchten Turbinenschaufeln. Des Weiteren verwenden sie eine numerische Simulation des Wärmeverhaltens auf der Oberfläche. Diese Simulation soll die Optimierung der Bahn ermöglichen. Ein konkretes Verfahren zur Optimierung wird allerdings nicht genannt.

Eine tiefgehende Untersuchung zur Reduktion von thermischen Varianzen auf einem Werkstück ist von Jones et al. in [95, 54] durchgeführt worden. Es werden so genannte Billardbahnen vorgestellt, die bezüglich des Winkels, des Startpunkts und der Geschwindigkeit so optimiert werden, dass die thermischen Varianzen reduziert werden. Das Verfahren ist auf ebene rechteckige Werkstücke beschränkt, kann dafür aber analytisch gelöst werden. Die bereits vorgestellten Rand-zu-Rand Bahnen aus [84] können als Erweiterung dieser Bahnen gesehen werden.

Grant et al. [74] untersuchen ebenfalls den Temperatureinfluss von Bahnen, jedoch speziell im Bereich des Sprühkompaktierens. Auch hier werden die Billardbahnen als Verbesserung zu den häufiger verwendeten Mäanderbahnen genannt. Im gleichen Bereich des Sprühkompaktierens haben auch Duncan et al. [54] Messungen zur Bestimmung von Temperaturprofilen erarbeitet. Diese Arbeiten können anschließend genutzt werden, um Modelle des Wärmeflusses aufzustellen, wie es in [137] gemacht wird. Das Modell wird anhand der Messdaten ermittelt und anschließend evaluiert.

Andere Bahnstrukturen werden in [64] auf den Einfluss auf Spannungen in dem Werkstück untersucht. Das Ergebnis der Untersuchung zeigt sowohl simulativ als auch durch Experimente bestätigt, dass verschiedene Bahnstrukturen zu unterschiedlichen Spannungen im Werkstück führen. Da die Spannungen hauptsächlich über thermische Varianzen erzeugt werden, sollte das Werkstück zu keiner Zeit große Temperaturunterschiede aufweisen. Dies ist daher eines der primären Ziele der hier entwickelten Initialbahnplanung.

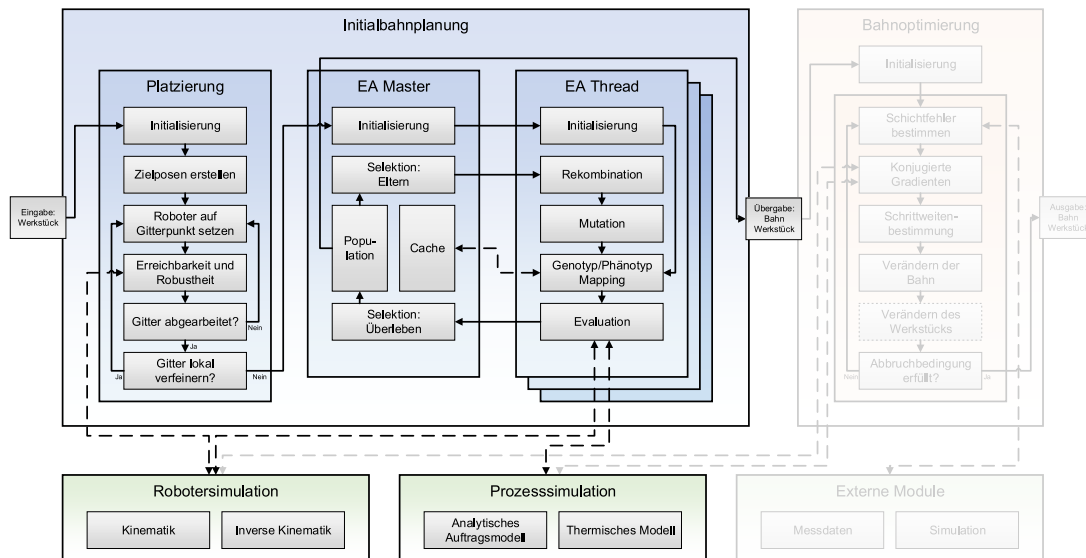


Abbildung 4.1: Konzeptioneller Aufbau der Initialbahnplanung. Die Komponenten des Systems sind durch graue Rechtecke dargestellt. Optionale Komponenten haben einen gepunkteten Rand. Die Kontrollflüsse sind durch durchgezogene Pfeile gekennzeichnet, die Datenkommunikation über gestrichelte Pfeile.

Eine Analyse des Wärmeverhaltens beim Plasmaspritzen ist auch von Zhang durchgeführt worden [176]. Unter anderem sind die Effekte verschiedener Bahneigenschaften wie die Spritzdistanz, Geschwindigkeit und Bahnabstand untersucht worden. Als Bewertungskriterien dienen die periodische Durchschnittstemperatur und die periodische Standardabweichung. Letztere ist vergleichbar mit der thermischen Varianz, deren Reduktion in dieser Arbeit zentral ist.

Chen et al. [37] haben einen auf Finite Elemente basierenden Ansatz zur Beschichtungs- und Temperatursimulation entwickelt. Das Modell wird mit einem traditionellen Schichtmodell zur Wärmesimulation verglichen. Dabei stellt sich heraus, dass die Finite Elemente Simulation wesentlich exakter ist, da sie die Bewegung der Spritzpistole sowie die (zufällige) Partikelverteilung im Spritzstrahl modelliert. Diese beiden Komponenten werden auch in der in dieser Arbeit genutzten Simulation verwendet.

## 4.3 Übersicht

In Abbildung 4.1 ist das Konzept der Initialbahnplanung dargestellt. Bei der Initialbahnplanung handelt es sich um eine Optimierung, die basierend auf einer Bahnstruktur eine möglichst gute initiale Bahn für ein gegebenes Werkstück erstellt. Die Optimierung wird mittels eines Evolutionären Algorithmus durchgeführt, der die Parameter der gegebenen Bahnstruktur optimiert. In Abschnitt 4.5 werden insgesamt drei verschiedene



Bahnstrukturen vorgestellt. Zwei dieser Bahnstrukturen, die Rand-zu-Rand Bahnen und die Punkt-zu-Punkt Bahnen, stellen komplexe Möglichkeiten zur Überdeckung der Oberfläche mit Bahnsegmenten dar. Bei der dritten Struktur handelt es sich um eine Erweiterung herkömmlicher, konturparalleler Mäanderbahnen auf komplexe Oberflächen. Diese Struktur soll einen Vergleich der anderen beiden Methoden mit den klassischen Bahnerzeugungsstrategien ermöglichen.

Die Eingabe der Initialbahnplanung besteht aus einem Werkstück, welches in Form eines Dreiecksnetzes vorliegen muss, da die Bahngenerierung (Genotyp/Phänotyp Mapping) die Knoten und die Netztopologie als Basis zur Konstruktion der Bahnen verwendet. Dies wird in den entsprechenden Abschnitten zur Bahngenerierung (Abschnitt 4.5) genauer beschrieben. Außerdem wird das Simulationsmodell aus Kapitel 3 verwendet. Sowohl das Auftrags- als auch das thermische Modell werden genutzt, um die Güte der Bahn (Fitness) zu bewerten. Die Robotersimulation wird ebenfalls verwendet. Über diese können die Achsbeschleunigungen des Roboters bestimmt werden, die von der Bahn verursacht werden. Diese sind ebenfalls Kriterien für die Güte der Bahn.

Die Initialbahnplanung ist in ein Mastermodul und mehrere Threads aufgeteilt, so dass sich die Arbeit mit Hilfe der Threads parallel auf die zur Verfügung stehenden Prozessorkerne verteilen lässt. Die Hauptaufgabe des Masters ist, die Population zu verwalten, mit den Threads zu kommunizieren sowie, wenn nötig, diese zu synchronisieren. Synchronisation ist immer dann notwendig, wenn die Population manipuliert werden soll. Hierunter fällt unter anderem die Aufnahme neuer, besserer Individuen in die Population durch die Selektion. Des Weiteren hält der Master einen Cache mit bereits berechneten Bahnsegmenten (Phänotyp) vor. Dieser wird von den Threads gefüllt und kann von ihnen gelesen werden, um das mehrfache Berechnen gleicher Bahnsegmente zu vermeiden.

Die Threads arbeiten unabhängig voneinander und führen nacheinander die elementaren Schritte des Evolutionären Algorithmus aus. Hierzu zählen die Rekombination, die Mutation, das Genotyp/Phänotyp Mapping sowie die Evaluation der erzeugten Individuen. Die letzten beiden Schritte sind die jeweils rechenintensivsten, so dass die Parallelisierung hier den größten Vorteil bringt.

Am Ende des Algorithmus steht eine Population von Individuen zur Verfügung, die jeweils die besten Bahnen repräsentieren, die bestimmt werden konnten. Die Bahn, die durch eines dieser Individuen repräsentiert wird, wird als Ergebnis an den nachfolgenden Schritt, die Bahnoptimierung, weitergegeben. Das Werkstück wird unverändert ebenfalls an die Bahnoptimierung durchgereicht.

## 4.4 Formulierung als Optimierungsproblem

Wie in Abschnitt 4.3 bereits erläutert, wird die Initialbahnplanung als Optimierungsproblem aufgefasst. Hierzu ist es notwendig, das Problem entsprechend zu formulieren. Die Kernaufgaben bei der Formulierung sind erstens die Beschreibung einer Bahnstruktur,



zweitens die mathematische Angabe einer Zielfunktion und drittens die Wahl einer geeigneten Optimierungsmethode. Diese drei Komponenten beschreiben zusammengenommen das Optimierungsproblem. In diesem Abschnitt werden diese Punkte informell vorgestellt. Eine formelle Beschreibung wird anschließend in den folgenden Abschnitten gegeben.

Wie in der Einleitung bereits erläutert sollen zwei komplexere Bahnstrukturen die Grundlage der Optimierung darstellen und durch eine einfachere Mäanderstruktur zum Vergleich ergänzt werden. Daher werden als Bahnstrukturen folgende Möglichkeiten betrachtet:

- **Quasi-konturparallele Bahnen:** Diese Bahnen sind einfache Erweiterungen konturparalleler Bahnen auf komplexe Oberflächen. Eine beliebige Kante der Oberfläche wird als Quelle ausgewählt und jeder Scan der Bahn, d. h. ein einzelner Überlauf von einer Seite des Werkstücks zur anderen, wird in geodätisch äquidistantem Abstand hierzu gewählt. Auf nicht allzu gekrümmten Oberflächen wirken diese Bahnen quasi parallel. Diese Bahnstruktur hat relativ wenige Freiheitsgrade und soll als Vergleich zu den anderen, komplexeren Strukturen dienen.
- **Rand-zu-Rand Bahnen:** Die Segmente dieser Bahnen werden so erzeugt, dass sie an zwei beliebigen, jedoch unterschiedlichen, Randpunkten auf der Kontur des Werkstücks starten bzw. enden. Zwischen diesen Punkten verläuft das Bahnsegment über die zu beschichtende Oberfläche. Aufeinanderfolgende Bahnsegmente werden über außerhalb des Werkstücks laufende Bahnen verbunden. Es kann bewiesen werden, dass diese Art von Bahnen die komplette Fläche überdecken kann (siehe Abschnitt 4.5.2).
- **Punkt-zu-Punkt Bahnen:** Diese Bahnen werden über Verbindungen zwischen Knoten definiert. Von einem Startknoten wird die geodätisch kürzeste Verbindung zu einem Zielknoten als Bahnsegment verwendet. Eine Folge dieser Segmente definiert die Bahn. Es ist sofort ersichtlich, dass das ganze Werkstück überdeckt werden kann, jedoch können erhebliche Knicke in der Bahn erzeugt werden, was zu praktischen Problemen führt und daher durch die Optimierung vermieden werden sollte.

Die Zielfunktion spezifiziert die Zielkriterien, die durch die Optimierung berücksichtigt werden sollen. Im Falle der Initialbahnplanung beim thermischen Spritzen haben sich folgende Zielkriterien als sinnvoll herausgestellt:

- Die durchschnittliche Abweichung der Schicht von der Zielschichtdicke an jedem Knoten des Netzes.
- Die maximale Abweichung der Schicht von der Zielschichtdicke.
- Die durchschnittliche thermische Varianz auf dem Werkstück während des Spritzvorgangs.
- Die maximale thermische Varianz.

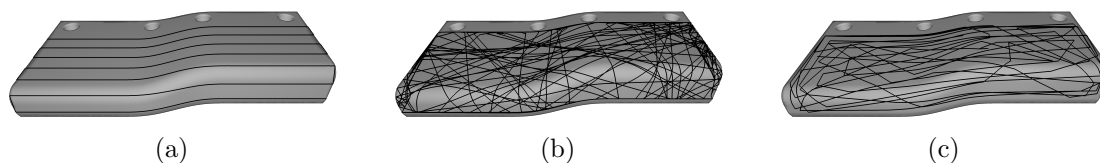


Abbildung 4.2: Vergleich verschiedener Bahnstrukturen. (a) Quasi-konturparallele Bahn bezüglich der vorderen, unteren Kante. (b) Beispiel einer Rand-zu-Rand Bahn. (c) Beispiel einer Punkt-zu-Punkt Bahn.

- Die durchschnittliche Roboterachsbeschleunigung.
- Die maximale Roboterachsbeschleunigung.
- Der Overspray.

Als Optimierungsverfahren kommen eine Menge verschiedener Methoden in Frage. Für das vorliegende Problem ist es sinnvoll, eine Metaheuristik zu verwenden. Einerseits ist der Lösungsraum sehr groß und unstrukturiert, andererseits stehen keine Gradienten zur Verfügung, um ein gradientenbasiertes Verfahren zu nutzen. Daher ist ein Evolutionärer Algorithmus verwendet worden. Dieser zeichnet sich dadurch aus, dass er den großen Lösungsraum großflächig untersuchen kann und dennoch in (lokale) Optima strebt.

## 4.5 Bahnerstellung

Durch die Bahnstruktur wird die globale Form der Bahn vorgegeben. Eine häufig verwendete Bahnstruktur ist die Mäanderbahn. Hier werden parallele bzw. quasi-parallele Bahnen in Form eines Mäanders verbunden. Im Folgenden werden drei verschiedene Bahnstrukturen vorgestellt, die in dieser Arbeit verwendet werden. Ein Vergleich der Bahnstrukturen ist beispielhaft in Abbildung 4.2 dargestellt.

Verschiedene Bahnstrukturen haben jeweils unterschiedliche Eigenschaften, so dass sie für den zu betrachtenden Prozess unterschiedlich gut geeignet sind. Im Bereich des thermischen Spritzens sind vor allem die Einhaltung der geforderten Schichtdicke an jeder Stelle der Oberfläche sowie die Minimierung der thermischen Varianzen während des Prozesses von Bedeutung. Diese Eigenschaften können durch die Bahnstruktur stark beeinflusst werden. Zum Beispiel liegen aufeinanderfolgende Scans einer Mäanderbahn sehr dicht beieinander, so dass die thermischen Varianzen bei der Optimierung der Parameter der Bahnstruktur kaum beeinflusst werden können. Rand-zu-Rand und Punkt-zu-Punkt Bahnen können jedoch dahingehend optimiert werden, dass die Spritzpistole jeweils möglichst weit von sehr heißen Stellen der Oberfläche wegbewegt wird, was die thermischen Varianzen verringert. Weitere Zielkriterien sind die kinematische Qualität der Bahn, die durch die entstehenden Achsbeschleunigungen des Roboters beim Abfahren der Bahn oder durch geometrische Überlegungen der Bahn bestimmt werden kann, sowie der Overspray.

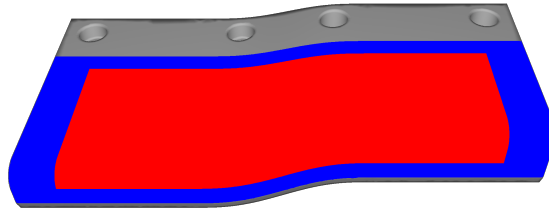


Abbildung 4.3: Beispiel für ein Region of Interest (ROI, rot) und ein erweitertes Region of Interest (EROI, blau). Auf dem ROI wird die Zielfunktion ausgewertet, während die Bahn auf dem EROI konstruiert wird.

In den folgenden Abschnitten wird die Konstruktion der in dieser Arbeit verwendeten Bahnstrukturen genauer erläutert.

#### 4.5.1 Region of Interest

Die Konstruktion der Bahnen basiert auf zwei sogenannten Region of Interests (ROIs), also Regionen des Werkstücks, die den funktionalen Bereich darstellen. Abbildung 4.3 stellt die beiden Typen von ROIs exemplarisch dar. Das erste ROI beschreibt den Teil des Werkstücks, der durch das thermische Spritzen beschichtet werden soll. Dies ist der funktionale Teil der Oberfläche und kann durchaus die gesamte Fläche sein. In der Optimierung wird die Zielfunktion auf der durch das ROI vorgegebenen Fläche ausgewertet.

Das erweiterte Region of Interest (EROI) ist etwas größer als das ROI und umfasst dieses komplett. Das EROI kann verwendet werden, um die Bahn zu konstruieren. Dies hat bei einigen Verfahren den Vorteil, dass die Bahn ihre Wendestellen nicht direkt auf der funktionalen Oberfläche hat. An diesen Stellen ist die Oberflächengeschwindigkeit auf Grund der Eigenschaften des Roboters besonders niedrig, so dass dort viel Material und viel Wärmeenergie aufgetragen würde. Wenn das ROI bereits das ganze Werkstück umfasst, kann das EROI dennoch durch virtuelle Erweiterung des Werkstücks entsprechend größer gewählt werden.

Die beiden Region of Interests müssen jeweils ein zusammenhängendes Gebiet der Werkstückoberfläche beschreiben. Löcher in den Regionen sind zulässig, solange immer eine geodätisch kürzeste Verbindung zwischen allen Oberflächenpunkten bestimmt werden kann. Da die Löcher jedoch zu unintuitiven Ergebnissen führen können, ist es manchmal empfehlenswert, diese in einem Vorverarbeitungsschritt zu füllen.

#### 4.5.2 Rand-zu-Rand Bahnen

Eine der am häufigsten eingesetzten Bahnstrukturen sind die Mäanderbahnen. Da diese aber auf Grund ihrer nahe beieinander liegenden Scans sehr schlechte thermische

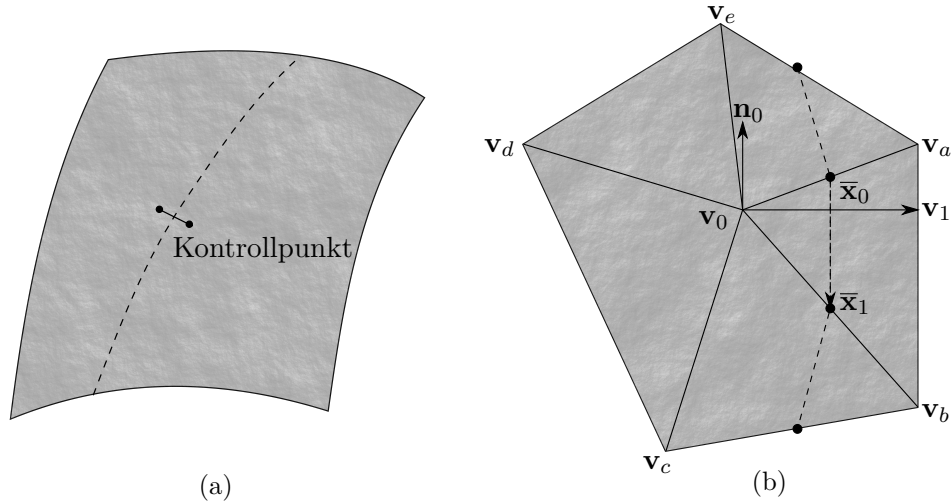


Abbildung 4.4: (a) Fläche mit einem Oberflächenbahnsegment (gestrichelte Linie), wobei jeder Punkt auf dem Oberflächenbahnsegment den gleichen geodätischen Abstand zu beiden Kontrollpunkten hat. (b) Konstruktion von Rand-zu-Rand Bahnen. Der Seedknoten  $v_0$  und ein zweiter Punkt  $v_1$  erzeugen das Oberflächenbahnsegment (gestrichelte Linie). Die Richtung der Bahn wird über die Beziehung der Vektoren  $\mathbf{n}_0$ ,  $\bar{x}_0 - v_0$  und  $\bar{x}_1 - \bar{x}_0$  bestimmt.

Varianzen erzeugen, sind alternative Bahnstrukturen vorgeschlagen worden. In [95] und [54] sind zum Beispiel Billardbahnen benutzt worden, um die thermischen Varianzen zu verringern. Da diese Art von Bahnen auf rechteckige und planare Flächen beschränkt sind, sind in einer Vorarbeit zu dieser Arbeit Rand-zu-Rand Bahnen in [84] vorgestellt worden. Diese können auf konvexen, allerdings ebenfalls nur planaren Oberflächen angewendet werden. Angelehnt an diese Bahnstruktur werden die Rand-zu-Rand Bahnen in dieser Arbeit modifiziert, damit sie auf Werkstücken mit gekrümmten Oberflächen eingesetzt werden können.

### Konstruktion der Bahnsegmente

Der Übertrag von Rand-zu-Rand Bahnen von planaren Flächen, siehe [84], auf komplexere Freiformflächen ist nicht intuitiv möglich. Auf planaren Flächen können die Bahnsegmente über zwei Punkte am Rand definiert werden, so dass sich ein Bahnsegment als direkte Verbindung dieser beiden Punkte ergibt. Während die Bahnsegmente auf konvexen Flächen immer über die Fläche verlaufen, können die Bahnsegmente auf nicht konvexen Flächen auch außerhalb der Fläche verlaufen. Die Überdeckung der gesamten Fläche mittels einer Menge dieser Bahnsegmente ist jedoch immer möglich. Eine einfache Erweiterung der Rand-zu-Rand Bahnen auf komplexere Freiformflächen könnte dadurch realisiert werden, dass die direkten Verbindungen von Punkten auf dem Rand durch geodätisch kürzeste Verbindungen (siehe [125]) über die Fläche ersetzt werden. In diesen Fall ist die Überdeckung der gesamten Fläche jedoch nicht mehr gewährleistet. Ein

Beispiel für eine Fläche, die auf diese Weise nicht überdeckt werden kann, ist eine Ebene mit einem einzelnen, hohen Berg. Daher ist eine alternative Konstruktionsregel notwendig.

Die alternative Konstruktionsregel basiert nicht auf der Angabe zweier Punkte am Rand sondern verwendet zwei innere Punkte auf der Oberfläche. In Abbildung 4.4a ist die grundsätzliche Idee dargestellt. Durch die Auswahl zweier Kontrollpunkte auf einer Fläche kann ein Oberflächenbahnsegment erzeugt werden, so dass alle Punkte auf dem Oberflächenbahnsegment den gleichen Abstand zu beiden Kontrollpunkten besitzen. Die Ausrichtung des Oberflächenbahnsegments wird durch die Wahl der Kontrollpunkte bestimmt. Auf einem Dreiecksnetz kann einer der Kontrollpunkte durch einen Knoten des Netzes repräsentiert werden. Dadurch wird das Oberflächenbahnsegment in der direkten Nachbarschaft des entsprechenden Knotens erzeugt, so dass dieser überdeckt wird. In dem später in diesem Abschnitt folgenden Beweis zur Überdeckung wird gezeigt, dass für jeden inneren Knoten ein in der Nachbarschaft liegendes Bahnsegment erzeugt werden kann, das an zwei Punkten des Randes endet. Damit ist die Überdeckung der Oberfläche gewährleistet.

Wie bereits erwähnt, basiert die Konstruktion eines Bahnsegments auf der Auswahl zweier Kontrollpunkte auf der Oberfläche, wobei einer der Kontrollpunkte der Knoten des Dreiecksnetzes ist, der von dem Bahnsegment überdeckt werden soll. Dieser Knoten wird im folgenden mit  $\mathbf{v}_0$  bezeichnet. Als zweiter Kontrollpunkt wird ein beliebiger Punkt auf dem 1-Ring des Knotens  $\mathbf{v}_0$  genutzt. Dieser wird mit  $\mathbf{v}_1$  bezeichnet. In Abbildung 4.4b sind ein Teil eines Oberflächennetzes sowie die entsprechenden Punkte dargestellt. In der Abbildung ist ebenfalls der Teil des Oberflächenbahnsegments gestrichelt eingezeichnet, der über dieses Oberflächenstück verläuft. In bestimmten Fällen kann es vorkommen, dass ein auf diese Weise konstruiertes Oberflächenbahnsegment einen Kreis bildet. Im Zusammenhang mit Rand-zu-Rand Bahnen sind diese Oberflächenbahnsegmente nicht sinnvoll und werden verworfen. Des Weiteren kann es vorkommen, dass ein Oberflächenbahnsegment in mehrere Teile zerfällt, wenn die Oberfläche z. B. Löcher besitzt. In diesem Fall können die einzelnen Teile durch eine direkte Verbindung zu einem Bahnsegment verschmolzen werden. Bei Oberflächen mit verhältnismäßig langer Kontur starten und enden die Bahnsegmente in der Regel jedoch am Rand der Oberfläche. Im folgenden Beweis zur Überdeckung wird auch gezeigt, dass es in der Nachbarschaft jedes inneren Knotens ein Bahnsegment gibt, das am Rand endet.

Zur konkreten Konstruktion des Bahnsegments werden alle Schnittpunkte der Netzkanten mit dem Oberflächenbahnsegment gesucht. In Abbildung 4.4b entsprechen diese Punkte den schwarzen Kreisen. Diese lassen sich dadurch ermitteln, dass die geodätischen Abstände der Knoten des Dreiecksnetzes zu den beiden Punkten  $\mathbf{v}_0$  und  $\mathbf{v}_1$  bestimmt werden. Wenn einer von zwei benachbarten Knoten näher an  $\mathbf{v}_0$  liegt und der andere näher an  $\mathbf{v}_1$ , so existiert ein Schnittpunkt des Oberflächenbahnsegments mit der Netzkante, die die beiden betrachteten Knoten verbindet. Die Lage des Schnittpunkts wird bestimmt und der entstandene Punkt (in der Abbildung z. B.  $\bar{\mathbf{x}}_0$ ) gespeichert.

Die Menge der Punkte, die die Schnitte zwischen dem Oberflächenbahnsegment und den Netzkanten beschreiben, liegen bisher unsortiert vor. Im nächsten Schritt sollen die Punkte nun so sortiert werden, dass die Verbindung von je zwei aufeinander folgenden Punkten innerhalb eines Dreiecks verlaufen und somit insgesamt das Bahnsegment ergeben. Um die entsprechende Sortierung zu erzeugen kann die Struktur des Dreiecksnetzes ausgenutzt werden. Wie in Abbildung 4.4b an den Punkten  $\bar{x}_0$  und  $\bar{x}_1$  zu erkennen ist, liegen zwei benachbarte Punkte auf Netzkanten, die ein gemeinsames adjazentes Dreieck besitzen. Dank dieser Eigenschaft lassen sich die Punkte durch Verfolgen der Netzstruktur so in eine Reihenfolge bringen, dass sie durch einfaches Verbinden aufeinanderfolgender Punkte das Bahnsegment erzeugen. Sollten auf diese Weise nicht alle Punkte verbunden werden, zerfällt das Bahnsegment in mehrere Teile. In diesem Fall werden die Teile zunächst einzeln betrachtet und schließlich mit einer möglichst kurzen direkten Verbindung verschmolzen.

Für die Bahnplanung ist es wichtig, dass jedes Bahnsegment eine eindeutige Richtung aufweist, das heißt, dass bei der wiederholten Generierung eines Bahnsegments nicht zufällig der Start- und der Endpunkt des Segments vertauscht werden. Um die Eindeutigkeit der Richtung sicherzustellen, sollen die Eigenschaften des Dreiecksnetzes ausgenutzt werden. Zunächst wird der Punkt des Bahnsegments bestimmt, der den geringsten Abstand zum erzeugenden Punkt  $\mathbf{v}_0$  hat. In Abbildung 4.4b ist dies der Punkt mit der Bezeichnung  $\bar{x}_0$ . Durch den vorangegangenen Schritt der Sortierung der Bahnpunkte wurden die Punkte des Bahnsegments bereits in einer bestimmten Reihenfolge angeordnet. Der durch diese Reihenfolge auf  $\bar{x}_0$  folgende Bahnsegmentpunkt wird nun betrachtet und im folgenden mit  $\bar{x}_1$  bezeichnet, siehe auch Abbildung 4.4b. Nun wird der Wert des Ausdrucks

$$(\mathbf{n}_0 \times (\mathbf{v}_0 - \bar{x}_0))^T (\bar{x}_1 - \bar{x}_0) \quad (4.1)$$

ausgewertet, wobei  $\mathbf{n}_0$  die Normale an dem Knoten  $\mathbf{v}_0$  bezeichnet. Wenn dieser Ausdruck negativ ist, wird die Reihenfolge der Punkte, die das Bahnsegment bilden, invertiert. Damit ist sichergestellt, dass die Richtung des Segments eindeutig ist.

Das Bahnsegment kann als Eingriffsbahn, siehe Abschnitt 2.1, betrachtet werden. Der nächste Schritt ist daher die Konvertierung des Bahnsegments in eine Werkzeugbahn. Dieser Schritt ist für alle Bahnstrukturen gleich und wird in Abschnitt 4.5.6 beschrieben.

### Konstruktion der Bahn

Eine vollständige Bahn wird aus einer Menge aufeinanderfolgender Bahnsegmente zusammengesetzt, die nacheinander abgefahren werden sollen. Sobald ein Bahnsegment ausgeführt wurde, also der Endpunkt des Bahnsegments erreicht wird, muss die Sprühpistole an den Startpunkt des nächsten Bahnsegments bewegt werden, wobei sowohl die Start- wie auch die Endpunkte auf dem Rand liegen. Die direkte Verbindung vom Endpunkt eines Bahnsegments zum Startpunkt des folgenden Bahnsegments könnte jedoch direkt über die funktionale Oberfläche verlaufen, so dass viel Auftrag entsteht, der nicht gewünscht ist. Daher muss ein alternativer Weg gefunden werden, der keinen Auftrag auf

der funktionalen Oberfläche erzeugt, also außerhalb dieser Fläche verläuft. Um dies zu erreichen wird das EROI verwendet. Das EROI umfasst das ROI, ist jedoch größer. Bei günstiger Wahl des EROIs ist dieses so groß, dass eine Bahn, die auf dem äußeren Rand des EROIs verläuft, keinen Auftrag auf der funktionalen Fläche erzeugt. Daher wird der kürzeste Weg gesucht, der entlang des äußeren Randes des EROIs verläuft und den Endpunkt eines Bahnsegments mit dem Startpunkt des nächsten Segments verbindet. Das Abfahren dieser Bahn erzeugt keinen Auftrag auf dem ROI, jedoch einen großen Overspray, da nur eine gewisse maximale Geschwindigkeit von dem Roboter erreicht wird, so dass einige Zeit benötigt wird, um den entsprechenden Weg zurückzulegen. Die gesamte Bahn ergibt sich demnach aus allen Bahnsegmenten sowie den entsprechenden Wegen entlang des äußeren Randes des EROIs, die die aufeinanderfolgenden Bahnsegmente verbinden.

### Beweis zur Überdeckung

Da für die Rand-zu-Rand Bahnen nicht direkt offensichtlich ist, dass diese die Oberfläche überdecken können, soll dies bewiesen werden. Genauer soll gezeigt werden, dass es für jeden inneren Knoten oder einen direkten Nachbarn ein entsprechendes Bahnsegment gibt, das am Rand endet. Wenn das Netz entsprechend dicht liegende Knoten besitzt, kann jeder Knoten durch die Bahn beschichtet, also überdeckt, werden.

**Satz 1.** *Zu jedem inneren Knoten  $p$  eines 2-mannigfaltigen Dreiecksnetzes oder zu einem seiner Nachbarknoten  $p'$  gibt es ein Bahnsegment, das einen gegebenen Punkt  $x$  auf dem Rand des Netzes erreicht.*

Zunächst wird der Umkreis um einen Punkt definiert.

**Definition 1.** *Der Umkreis um einen Punkt  $p$  mit Radius  $r$  besteht aus allen Punkten der Oberfläche, die einen geodätischen Abstand  $r$  von  $p$  haben.*

Hieraus folgt, dass der Umkreis eines Punkts auf einem Netzknoten oder einer Netzkante aus einer oder mehreren Zusammenhangskomponenten besteht, die jeweils entweder eine geschlossene Kurve bilden oder am Rand enden. Diese Zusammenhangskomponenten werden im Folgenden auch Umkreiskomponenten genannt.

Zunächst werden zwei Lemmata gezeigt:

**Lemma 1.** *Der 1-Ring innerer Knoten ist eine stetige, zusammenhängende, geschlossene Kurve.*

**Beweis:** Klar. ■

**Lemma 2.** *Eine Umkreiskomponente ist stetig.*



**Beweis:** Der Umkreis um einen Punkt  $p$  kann als implizite Funktion  $U(p, p') = r$  angesehen werden, wobei  $p'$  ein Punkt auf der Oberfläche des Dreiecksnetzes und  $r$  der Radius des Umkreises um  $p$  ist. Die Funktion  $U$  nutzt ein Distanzfeld  $D_S$  mit  $U(p, p') = D_p(p')$ , das auf dem Dreiecksnetz definiert werden kann als [101]:

$$D_S(m) = d(m, S) = \min\{d(m, m') \mid m' \in S\}. \quad (4.2)$$

$S$  ist dabei die Quelle des Distanzfeldes, also ein Punkt oder eine Kurve auf der Oberfläche des Dreiecksnetzes,  $d(., .)$  die geodätische Distanz auf dieser Fläche und  $m$  bzw.  $m'$  sind Punkte auf der Oberfläche. Die Stetigkeit der Umkreiskomponenten ergibt sich dann aus dem Hauptsatz über implizite Funktionen [57]. ■

**Satz 2.** *Entweder schneidet eine Umkreiskomponente von  $x$  mit Radius  $r$  den 1-Ring um  $p$ , oder eine Umkreiskomponente von  $x$  mit Radius  $r'$  schneidet den 1-Ring eines Nachbarknotens  $p'$  von  $p$ . Hierbei ist  $r$  der Abstand von  $x$  zu  $p$  und  $r'$  der Abstand von  $x$  zu  $p'$ .*

**Beweis:** Ist mindestens ein Punkt auf dem 1-Ring von  $p$  weiter entfernt von  $x$  als  $p$  und mindestens einer näher, dann gibt es auf Grund der Tatsache, dass  $p$  selber Teil der Umkreiskomponente sein muss, der Stetigkeiten vom 1-Ring und der Umkreiskomponenten sowie dem Zwischenwertsatz [56] einen Schnitt zwischen der Umkreiskomponente und dem 1-Ring um  $p$ . Andernfalls könnte die Umkreiskomponente vollständig innerhalb des 1-Ringes verlaufen. Dann gibt es jedoch einen Nachbarn  $p'$  von  $p$ , für den mindestens ein Punkt des 1-Ringes (nämlich  $p$ ) weiter entfernt ist als  $p'$ , und die entsprechende Umkreiskomponente schneidet diesen 1-Ring. Der jeweilige Schnittpunkt kann als Punkt  $q$  für die Berechnung des gesuchten Bahnsegments genutzt werden. ■

**Korollar 1.** *Der betrachtete Randpunkt  $x$  liegt auf dem Bahnsegment von  $p$  bzw.  $p'$  und dem ermittelten Schnittpunkt  $q$ .*

**Beweis:** Das Bahnsegment wird dadurch konstruiert, dass alle Punkte auf Netzkanten mit gleichem Abstand zu den Punkten  $p$  ( $p'$ ) und  $q$  zu einem Bahnsegment verbunden werden. Daher muss auch der Randpunkt  $x$  auf diesem Segment liegen. ■

### 4.5.3 Punkt-zu-Punkt Bahnen

Eine weitere Möglichkeit für eine Bahnstruktur ergibt sich intuitiv aus der Repräsentation der Oberfläche als Dreiecksnetz. Sofern die Oberfläche gleichmäßig von den Knoten des Netzes abgetastet ist, können die Knoten direkt als Start- und Zielpunkte von Bahnsegmenten dienen.

#### Konstruktion der Bahnsegmente

Bei dieser Bahnstruktur geschieht die Konstruktion eines Bahnsegments durch die Angabe zweier Knoten auf dem Netz. Die kürzeste geodätische Verbindung [125] dieser



beiden Knoten beschreibt dann das Bahnsegment. Wie bei den Rand-zu-Rand Bahnen wird ein Bahnsegment über eine Sequenz von Punkten auf den Netzkanten beschrieben, die auf dem kürzesten Weg liegen. Diese werden durch eine Algorithmus zur Bestimmung der geodätisch kürzesten Verbindung bestimmt, siehe [125]. Aufeinanderfolgende Punkte haben ein gemeinsames adjazentes Dreieck und können direkt verbunden werden, so dass die Bahnen in die Dreiecke interpoliert werden. Die Voraussetzung für die Konstruktion ist ein zusammenhängendes Gebiet, so dass die kürzeste geodätische Verbindung über das Gebiet bestimmt werden kann.

Da sich die so erstellten Bahnsegmente als Eingriffsbahnen interpretieren lassen, müssen sie noch in Werkzeugbahnen überführt werden. Ein entsprechendes Verfahren hierzu wird in Abschnitt 4.5.6 beschrieben.

### Konstruktion der Bahn

Die Konstruktion einer Bahn aus den Bahnsegmenten ergibt sich in natürlicher Weise durch die Hintereinanderreihung der Bahnsegmente. Wenn als Startpunkt des nächsten Bahnsegments der Zielpunkt des vorherigen gewählt wird, sind keine Zwischenbahnen notwendig. Problematisch könnten jedoch die durch diese Konstruktion entstehenden Knicke in der Bahn sein. Wenn das an einem Punkt ankommende Bahnsegment mit dem an diesem Punkt startenden Bahnsegment einen spitzen Winkel bildet, muss der Roboter einen Brems- und Beschleunigungsvorgang durchführen, der zu einem hohen Auftrag und einem großen Wärmeeintrag führen kann. Daher sind solche Situationen in der Optimierung der Bahn zu berücksichtigen.

Auch wenn es bei der Konstruktion der Bahnen nicht unbedingt notwendig ist, bietet es sich an, zur Bahnkonstruktion auf das EROI zurückzugreifen. Dadurch werden die Ränder des ROIs besser berücksichtigt.

#### 4.5.4 Quasi-konturparallele Bahnen

Konturparallele Bahnen werden auf planaren Flächen häufig verwendet. Eine Erweiterung dieser Bahnstruktur auf freigeformte Flächen kann mit Hilfe von quasi-konturparallelen Bahnen erreicht werden. Auch wenn die Abstände der Scans der Bahn (geodätisch) gleich sind, können die Bahnen subjektiv recht wenig parallel wirken. Da diese Bahnstrukturen die Krümmungen der Oberfläche nicht berücksichtigen, ergeben sich an vielen Stellen sehr starke Umorientierungen der Sprühpistole sowie sehr hohe Roboterachsbeschleunigungen, so dass die Bahn für die Ausführung mit einem echten Roboter an diesen Stellen problematisch ist.

### Konstruktion der Bahnsegmente

Die Konstruktion der Bahnsegmente beginnt mit der Angabe einer Quellmenge. Diese kann theoretisch beliebig sein, sollte im Falle quasi-konturparalleler Bahnen jedoch aus einer Linie entlang einer Kontur bestehen.

Als Nächstes werden die Iso-Linien generiert. Dazu werden alle Punkte der Oberfläche gesucht, die einen vorgegebenen Abstand von der Quellmenge besitzen. Der Abstand kann auf freigeformten Oberflächen mittels des minimalen geodätischen Abstands [125] zur Quellmenge bestimmt werden. Auf einem Dreiecksnetz bietet es sich an, die Punkte auf den Kanten zu suchen, die den gesuchten Abstand besitzen, und zwischen benachbarten Punkten zu interpolieren. Wenn die Quellmenge nicht am Rand der Oberfläche liegt, muss ebenfalls beachtet werden, dass Punkte mit dem gesuchten Abstand auf beiden Seiten der Quellmenge liegen können und es pro Abstandswert entsprechend zwei Bahnsegmente geben kann. Dieses Verfahren wird für jeden Iso-Wert durchgeführt, so dass die Punkte auf den Kanten für die äquidistanten Bahnsegmente ermittelt werden.

Die Punkte auf den Kanten, die zu einem Iso-Wert gehören, werden anschließend in eine Reihenfolge gebracht, so dass zwei aufeinander folgende Punkte auf den Kanten eines gemeinsamen Dreiecks liegen. Durch Interpolation zwischen den aufeinanderfolgenden Punkten wird dann das gewünschte Bahnsegment gebildet. Sollte ein Bahnsegment in mehrere Teile zerfallen, werden die einzelnen Teile separat behandelt und dann mit kurzen Verbindungen zusammengefügt. Bei geeigneter Wahl der Quellmenge verläuft jedes Bahnsegment von einem Punkt am Rand zu einem anderen Punkt am Rand des EROIs. Ansonsten können auch Kreise entstehen.

Die Bahnsegmente haben auf Grund ihrer Abstände eine natürliche Sortierung, aufsteigend nach Abstand zur Quellmenge. Auch hier ist wieder zu beachten, dass es zwei Bahnsegmente mit gleichem Abstand zur Quellmenge auf der jeweils anderen Seite der Quellmenge geben kann. In dem Fall wird zuerst absteigend auf der einen Seite, dann die Quelle selbst und schließlich aufsteigend auf der anderen Seite der Quelle sortiert.

Die Bahnsegmente stellen die Eingriffsbahn dar und müssen entsprechend in Werkzeugbahnen überführt werden, wie es in Abschnitt 4.5.6 beschrieben wird.

### Konstruktion der Bahn

Nach dem Erstellen der Bahnsegmente und dem Sortieren der Segmente können diese nun zu einer vollständigen Bahn zusammengefügt werden. Dazu wird entsprechend der Sortierung der Bahnsegmente das vorhergehende mit dem nachfolgenden Segment verbunden. Die Verbindung geschieht dabei jeweils so, dass das Ende der nachfolgenden Bahn verbunden wird, das näher an dem offenen Ende der vorhergehenden Bahn liegt. Dadurch entstehen automatisch die bekannten Mäanderstrukturen.

Tabelle 4.1: Qualitativer Vergleich der Bahnstrukturen

Kriterium	Quasi-konturparallel	Rand-zu-Rand	Punkt-zu-Punkt
Struktur	einfach	komplex	komplex
Parametrisierung	einfach	komplex	einfach
Eignung für			
komplexe Werkstücke	schlecht	gut	gut
Optimierungsparameter	wenige	viele	viele
Materialauftrag	gut	mittel	mittel
Thermische Varianzen	gut	gut	gut
Roboterachs-			
beschleunigungen	sehr schlecht	gut	gut
Overspray	schlecht	sehr schlecht	gut

#### 4.5.5 Vergleich der Bahnstrukturen

In Tabelle 4.1 werden einige Vor- und Nachteile der Verfahren nochmals gegenübergestellt. Jede Bahnstruktur hat dabei seine eigenen Stärken und Schwächen und keines der Verfahren ist in allen Punkten optimal. Während die quasi-konturparallelen Bahnen einfache Strukturen besitzen und auf Grund der geringen Anzahl an Parametern einfach zu optimieren sind, eignen sie sich wegen der sehr schlechten Roboterachsbeschleunigungen an Oberflächenkrümmungen nicht für beliebige Freiformflächen. Rand-zu-Rand Bahnen und Punkt-zu-Punkt Bahnen sind extra für komplexe Freiformflächen angepasst worden, sind dafür jedoch schwieriger zu optimieren.

#### 4.5.6 Erstellen der Werkzeugbahn

Alle oben genannten Verfahren erzeugen eine Eingriffsbahn. Für die Simulation und die Optimierung werden jedoch Werkzeugbahnen benötigt. Hier wird ein Verfahren beschrieben, das eine Werkzeugbahn an der Spitze der Spritzpistole erzeugt, siehe auch Abbildung 2.1.

Zunächst werden die Bahnen von der Oberfläche in den Raum über der Oberfläche verschoben. Da alle Punkte, die die Bahnsegmente beschreiben, auf den Kanten des Dreiecksnetzes liegen, kann für all diese Punkte durch einfache Interpolation der Normalen der Endpunkte dieser Kanten eine Normale für die Punkte auf den Kanten erzeugt werden. Entlang dieser Normalen werden die Bahnpunkte um den idealen Spritzabstand, der vom Prozess vorgegeben wird, verschoben.

Die Punkte der Bahn definieren jedoch noch keine vollständige Werkzeugbahn, da die Orientierung der Spritzpistole angegeben werden muss. Da die Spritzrichtung definitionsgemäß entlang der negativen  $\mathbf{z}$ -Achse verläuft, wird diese fest auf die jeweiligen negativen Normalenrichtungen gesetzt, entlang derer die Bahnpunkte verschoben wur-

den. Entsprechend müssen nur noch die  $\mathbf{x}$ - und  $\mathbf{y}$ -Achsen des Spritzkoordinatensystems festgelegt werden.

Kinematisch ist es sinnvoll, wenn sich die Spritzkoordinatensysteme aufeinanderfolgender Bahnpunkte so wenig wie möglich ändern, damit der Roboter möglichst wenig umorientieren muss. Daher wird folgendes Verfahren angewendet, um die Spritzkoordinatensysteme möglichst ähnlich zu gestalten.

Alle Normalen des EROIs werden betrachtet, um die Weltkoordinatenachse zu bestimmen, in die die wenigsten Normalenanteile der Oberflächenknoten fallen. Dies ist demnach die Achse  $\mathbf{w}$ , in deren Richtung am seltensten gespritzt wird. Im Folgenden soll das Spritzkoordinatensystem so ausgerichtet werden, dass die lokale  $\mathbf{x}$ -Achse so gut wie möglich an der  $\mathbf{w}$ -Achse ausgerichtet wird, ohne die Spritzrichtung zu verändern. Dazu wird zunächst das Spritzkoordinatensystem so ausgerichtet, dass die Spritzrichtung korrekt ist, die anderen Achsen jedoch beliebig sind. Die  $\mathbf{x}$ - und  $\mathbf{y}$ -Achsen des Spritzkoordinatensystems spannen eine Ebene auf, in die die  $\mathbf{w}$ -Achse projiziert wird. Nun kann der Rotationswinkel bestimmt werden, um den um die Spritzrichtung rotiert werden muss, damit die  $\mathbf{x}$ -Achse mit der projizierten  $\mathbf{w}$ -Achse übereinstimmt. Nach Ausführung der Rotation ist das Spritzkoordinatensystem wie gewünscht ausgerichtet. Das Verfahren funktioniert für alle Bahnpunkte, an denen nicht genau entlang der  $\mathbf{w}$ -Achse gespritzt wird.

#### 4.5.7 Zu- und Abföhrbahnen

Für den Einsatz von Bahnen in der Praxis werden in der Regel Zu- und Abföhrbahnen benötigt. Diese Bahnen erlauben es dem Roboter, von einer gewissen Startposition den Startpunkt der eigentlichen Bahn zu erreichen und diese Bahn dann abzufahren. Bei den Rand-zu-Rand Bahnen und den quasi-konturparallelen Bahnen beginnt und endet die Bahn am Rand des EROIs und es werden keine speziellen Zu- oder Abföhrbahnen benötigt. Die Punkt-zu-Punkt Bahnen können jedoch mitten auf dem funktionalen Teil der Oberfläche beginnen. Da die Spritzpistole normalerweise nicht einfach an- und abgeschaltet werden kann, muss eine Zu- und Abföhrbahn erzeugt werden, so dass der Roboter von außerhalb an das Werkstück heranzufahren kann.

Bei den Punkt-zu-Punkt Bahnen können daher zwei Knoten am Rand des EROIs bestimmt werden, die als Start- und Endpunkt der gesamten Bahn dienen. Dadurch wird verhindert, dass die Bahn mitten auf dem Werkstück startet bzw. endet. Größere Zu- und Abföhrbahnen aus einer entsprechenden Ruhestellung des Roboters sind dann einfach an diese Punkte anzuschließen, ohne dass ein Auftrag an den funktionalen Stellen der Werkstückoberfläche entsteht.

### 4.5.8 Roboterplatzierung

Nachdem eine Werkzeugbahn aus der Eingriffsbahn erzeugt worden ist, kann in einem nachfolgenden Schritt eine Roboterbahn erzeugt werden. Dazu müssen die entsprechenden Posen der Werkzeugbahn in Roboterachswinkel umgerechnet werden. Bevor dies mit Hilfe der inversen Kinematik geschehen kann, muss das Werkstück relativ zum Roboter platziert werden. Hier wird die relative Positionierung der Roboterbasis bezüglich des Werkstücks vorgenommen. Die Positionierung beinhaltet dabei sowohl die Position als auch die Orientierung der Roboterbasis. Das wesentliche Kriterium für eine erfolgreiche Platzierung ist die Erreichbarkeit aller Bahnpunkte durch den Roboter.

Das Problem der Roboterplatzierung wird in der Literatur vielfach untersucht. Eine Auswahl an Arbeiten zu diesem Thema ist zum Beispiel [5, 25, 58, 62, 89, 96, 109, 126, 53, 160]. In den Arbeiten werden verschiedene Lösungswege basierend auf unterschiedlichen Zielfunktionen vorgeschlagen. Das wichtigste Ziel ist die Erreichbarkeit aller geforderten Bahnpunkte und muss zwingend eingehalten werden. Über die reine Erreichbarkeit hinaus gibt es jedoch auch verschiedene Kriterien, die optimiert werden können. Dazu zählen in erster Linie die so genannte *Cycle time* [6, 58, 62, 96], der *Energiebedarf* [6, 109] oder auch die *Manipulierbarkeit* [5, 126, 53].

Grundsätzlich wäre es möglich, für jede Bahn eine Platzierung durchzuführen, um eine ideale Position für den Roboter zu finden. Jedoch werden im Verlauf der Initialbahnplanung viele tausend Bahnen erzeugt, so dass der Aufwand für jede Bahn immens wäre. Daher wird eine bahnunabhängige Platzierung im Voraus durchgeführt, die für alle erzeugten Bahnen zumindest die Erreichbarkeit sicherstellen soll, siehe auch Abbildung 4.1. Zusätzlich wird die Robustheit der Platzierung maximiert, die zum Ziel hat, dass kleine Änderungen an einer Bahn keine Erreichbarkeitsprobleme verursachen.

Prinzipiell ist allen in dieser Arbeit vorgestellten Bahnerstellungsverfahren gemeinsam, dass sie Eingriffsbahnen erzeugen, die auf dem EROI definiert sind. Die Eingriffsbahnen aller Verfahren werden mit der gleichen Methode in den Raum projiziert, so dass sich die Werkzeugbahnen ergeben. Daher muss sichergestellt werden, dass der Roboter so platziert wird, dass alle Punkte, die auf diese Weise erzeugt werden können, auch von dem Roboter erreicht werden können. Das Verfahren geht daher so vor, dass die Knoten, die zu dem EROI gehören, als Punkte einer generalisierten Eingriffsbahn dienen. Da die Knoten bei Werkstücknetzen mit einigermaßen dichter und regelmäßiger Abtastung sehr gut auf dem EROI verteilt sind, kann das EROI über diese Punkte gut abgedeckt werden. Jeder Knoten ist demnach ein Punkt einer generalisierten Eingriffsbahn. Die Konnektivität der Knoten spielt keine Rolle, da es grundsätzlich nur um die Erreichbarkeit der einzelnen Punkte geht. Daher werden die Knoten gemäß ihrer Indizierung zu einer Eingriffsbahn verbunden. Diese Bahn wird nun mit dem Verfahren aus Abschnitt 4.5.6 in eine Werkzeugbahn überführt. Für alle Punkte dieser Werkzeugbahn kann für jede Position des Roboters überprüft werden, ob der Bahnpunkt erreichbar ist oder nicht. Zusätzlich wird eine Metrik angewendet, die eine Aussage über die Güte der Platzierung treffen kann. Dazu wird für jede Roboterpose, die sich für einen Bahnpunkt ergibt,

ermittelt, wie nah sich die Achsstellungen an ihren Extrema befinden. Wenn die Achsen sehr dicht bei ihren extremen Stellungen stehen, ist das Risiko relativ groß, dass eine geringfügige Verschiebung eines Bahnpunkts zum Überschreiten der maximal möglichen Achsstellung führt. Da im Bereich der Bahnoptimierung (siehe Kapitel 5) viele kleine Änderungen an der Bahn durchgeführt werden, sollten extreme Achsstellungen möglichst vermieden werden. Außerdem muss ein gewisser Spielraum auch für die Punkte eingeplant werden, die zwischen den Knoten auf den Netzkanten oder in den Dreiecken liegen.

Die Suche nach einer optimalen Platzierung für den Roboter wird auf Basis eines hierarchischen Grid Search Verfahrens realisiert, das an effizienzsteigernde Methoden des Ray Tracings angelehnt ist, siehe [70]. Zunächst wird dazu ein Gitter in einem sechs dimensionalen Raum aufgespannt. Die ersten drei Dimensionen entsprechen den Raumrichtungen, in denen der Roboter verschoben werden kann. Die anderen drei Dimensionen entsprechen den Rotationen. Der Roboter kann dementsprechend uneingeschränkt positioniert werden. Dennoch ist es möglich, mit Hilfe von oberen und unteren Schranken in jeder Dimension auch bestimmte Randbedingungen zu modellieren. So könnte es zum Beispiel notwendig sein, dass der Roboter auf einen vorhandenen Boden platziert wird, der relativ zur Werkstückhalterung nicht verändert werden kann.

Der erste Schritt der Optimierung erzeugt ein Gitter mit einstellbarer Gitterauflösung im betrachteten Bereich zwischen den vorgegebenen Schranken. Der Roboter wird für jeden der Gitterpunkte entsprechend den Parametern testweise platziert und die Erreichbarkeit jedes Bahnpunkts wird mit Hilfe der inversen Kinematik getestet. Wenn es keine Einschränkungen bezüglich der Erreichbarkeit gibt, wird die Robustheit durch die oben beschriebene Metrik bestimmt. Die Platzierung mit der global besten Robustheit wird gespeichert.

Wenn alle Gitterpunkte getestet wurden, wird das Verfahren wahlweise beendet oder das Gitter verfeinert. Dazu wird ein neues Gitter erzeugt, das um den Gitterpunkt mit dem besten Zielfunktionswert gelegt wird. Die obere und untere Schranke jeder Achse wird um die Hälfte reduziert und die Auflösung des Gitters wird um einen Faktor (zum Beispiel zwei) je Dimension erhöht. Das Verfahren wird iteriert, bis der Benutzer mit dem Ergebnis zufrieden ist.

Im Gegensatz zu einer Platzierungsoptimierung für eine einzelne Bahn ist das genutzte Verfahren auf beliebige Bahnen über dem EROI optimiert. Dementsprechend kann es durchaus bessere Lösungen für einzelne Bahnen geben. Jedoch ist es auf Grund der relativ langen Rechenzeiten nicht möglich, eine Platzierung für tausende von Bahnen durchzuführen. Daher ist es günstig, eine Lösung für die Platzierung zu bestimmen, die für alle möglichen Bahnen eine gute Lösung bietet.

Alternativ zum iterativen Grid Search Verfahren könnten auch komplexere Methoden wie zum Beispiel gradientenbasierte Verfahren eingesetzt werden. Vorteile des Grid Search Verfahrens sind die relativ schnelle Generierung guter Lösungen und die Berücksichtigung des gesamten Zielfunktionsraums. Zudem lohnt es nicht, eine optimale Position des Roboters zu finden, solange für die Platzierung nur eine generalisierte Bahn basierend auf dem EROI genutzt wird. Für die finale Bahn könnte schließlich noch eine genaue

Platzierung durchgeführt werden. Hier könnten zudem weitere Zielkriterien nützlich sein, wie der Energiebedarf oder die Manipulierbarkeit.

Das Verfahren zur Platzierung wurde hier ausschließlich als externer Schritt betrachtet. Dies betrifft sowohl die genutzte Vorberechnung als auch eine Platzierung pro Bahn. Es wäre jedoch möglich, dass die Platzierung des Roboters als Variable in die Bahnoptimierung (Kapitel 5) aufgenommen wird. Dadurch würde sich der Parameterraum jedoch stark vergrößern, so dass die Optimierung deutlich komplexer werden würde.

## 4.6 Zielfunktion

Die Zielfunktion beschreibt die Komponenten, die von der Optimierung berücksichtigt werden sollen. Mögliche Komponenten sind in Abschnitt 4.4 bereits informell zusammengestellt worden. Damit die Zielfunktion ausgewertet werden kann, muss als Eingabe sowohl eine Bahn  $\boldsymbol{\chi} = (\chi_0, \dots, \chi_n)^\top$  als auch das Werkstück in Form eines Dreiecksnetzes mit den Knoten  $\mathbf{v}_i$ ,  $i = 0, \dots, m$ , zur Verfügung stehen. Die benötigten Daten zur Auswertung der Zielfunktion, wie zum Beispiel der Schichtauftrag  $z_i$  oder die Temperaturen  $\Theta_{i,j}$ , werden über die Simulationen (Kapitel 3) bestimmt. Im Folgenden werden die einzelnen Komponenten der Zielfunktion vorgestellt. Diese Komponenten können dann in einem Vektor zusammengefasst und in der mehrkriteriellen Optimierung als Zielfunktionswerte genutzt werden.

Die erste Zielfunktionskomponente beschreibt den Fehler in der Beschichtung. Dies bedeutet, dass der Fehler größer ist, je weiter die Schichtdicke von dem geforderten Soll abweicht. Die Auswertung der Schichtdicke geschieht in der Simulation an den Knoten des Netzes. Entsprechend wird als Eingabe eine Zielschichtdicke  $Z_i$  für jeden Knoten  $\mathbf{v}_i$  erwartet. Diese ist in vielen Fällen konstant für alle Knoten, kann jedoch im Allgemeinen beliebige Werte annehmen.

Der Fehler in der Beschichtung kann über

$$E_l(\boldsymbol{\chi}) = \frac{1}{A} \sum_{i=0}^m A_i \left( \frac{z_i(\boldsymbol{\chi}) - Z_i}{Z_i} \right)^2 \quad (4.3)$$

angegeben werden. Hierin ist  $A$  die gesamte Fläche des ROIs, auf dem die Beschichtung ausgewertet wird.  $A_i$  beschreibt die Fläche, die von einem Knoten repräsentiert wird, siehe Abbildung 2.2. Dadurch wird eine Flächengewichtung des Fehlers erreicht. Der Fehler selber wird über die quadratische Differenz zwischen Soll- und Ist-Schichtdicke dividiert durch die quadratische Zielschichtdicke bestimmt und wird über alle Knoten gewichtet aufsummiert.

Prinzipiell ist sowohl eine Unter- wie auch eine Überbeschichtung problematisch, jedoch kann eine Überbeschichtung in einem Nachbearbeitungsschritt, wie zum Beispiel Schleifen, korrigiert werden. Eine Korrektur einer Unterbeschichtung kann im Allgemeinen nicht

oder nur mit großem Aufwand realisiert werden. Daher kann es sinnvoll sein, eine Unterbeschichtung durch eine Multiplikation mit einer Konstanten höher zu gewichten.

Die zweite Zielfunktionskomponente ist die thermische Varianz auf der Werkstückoberfläche. Diese kann im Gegensatz zu dem Fehler in der Beschichtung nicht am Ende der Simulation ausgewertet werden, sondern muss während der Simulation in geeigneten Zeitabständen gemessen werden. Als Zeitabstand bietet sich der in der Simulation verwendete Zeitschritt an, der durch die Bahnpunkte gegeben ist. Die thermische Varianz nach einem Simulationsschritt kann mittels

$$E_{t,j}(\boldsymbol{\chi}) = \frac{1}{(m+1)} \sum_{i=0}^m \left( \frac{\Theta_{i,j} - \bar{\Theta}_j}{\bar{\Theta}_j} \right)^2 \quad (4.4)$$

berechnet werden. Die Varianz berechnet sich als über der Anzahl der Knoten gemittelten quadratischen Differenzen der Temperaturen an den Knoten  $\mathbf{v}_i$  zum Mittelwert der Temperaturen zum Simulationszeitpunkt  $j$  geteilt durch den quadratischen Mittelwert der Temperaturen auf der Oberfläche. Der flächengewichtete Mittelwert der Temperaturen zu einem Zeitpunkt  $j$  ergibt sich zu

$$\bar{\Theta}_j = \frac{1}{A} \sum_{i=0}^m A_i \Theta_{i,j}. \quad (4.5)$$

Als Gesamtwert kann dann das zeitgewichtete Mittel über all diese Varianzen genutzt werden mit

$$E_t(\boldsymbol{\chi}) = \frac{1}{\tau_n} \sum_{j=0}^n E_{t,j}(\boldsymbol{\chi}) \hat{t}_j, \quad (4.6)$$

wobei  $\tau_n$  die Gesamtausführungszeit der Bahn und  $\hat{t}_j$  die Spritzdauer an einem Bahnpunkt ist.

Im Falle der Rand-zu-Rand Bahnen ergeben sich eine Menge von Zwischenbahnen, die am äußeren Rand des EROIs verlaufen. Diese Tragen nicht zur Beschichtung bei, erzeugen jedoch Overspray und sorgen für längere Ausführungszeiten. Daher ist es erstrebenswert, diese Zwischenbahnlängen zu reduzieren. Als einfacher Wert in Form einer weiteren Zielfunktionskomponenten kann also einfach die Summe der quadratischen Längen aller Zwischenbahnen verwendet werden, damit insbesondere lange Wege außerhalb der funktionalen Oberfläche, die viel Overspray erzeugen, vermieden werden. Dies kann mittels

$$E_z(\boldsymbol{\chi}) = \frac{1}{n} \sum_{j=1}^n l(\boldsymbol{\chi}_{j-1}, \boldsymbol{\chi}_j)^2 \quad (4.7)$$

geschehen. Hier stellt  $l(\boldsymbol{\chi}_{j-1}, \boldsymbol{\chi}_j)$  eine Funktion dar, die die Länge der Zwischenstrecke bestimmt, für den Fall, dass es sich bei  $\boldsymbol{\chi}_{j-1}$  und  $\boldsymbol{\chi}_j$  um Endpunkte zweier Bahnsegmente handelt.



Eine weitere wichtige Zielfunktionskomponente ist die Beschleunigung in den Roboterachsen. Die Beschleunigungen sind ein Indikator dafür, ob der Roboter in der Lage ist, die gegebene Bahn in der gewünschten Geschwindigkeit abzufahren. Entsprechend ist es erstrebenswert, wenn die Beschleunigungen gering gehalten werden. Eine Möglichkeit, die Roboterachsbeschleunigungen zu beschreiben ist mittels

$$E_m(\boldsymbol{x}) = \frac{1}{6(n-1)} \sum_{j=1}^{n-1} \sum_{k=1}^6 \left( \frac{\Phi_{j+1,k} - \Phi_{j,k}}{t_j t_{j-1}} - \frac{\Phi_{j,k} - \Phi_{j-1,k}}{t_{j-1}^2} \right)^2, \quad (4.8)$$

wobei  $\Phi_{j,k}$  die Achsstellung der  $k$ -ten Achse am Bahnpunkt mit Index  $j$  darstellt.  $E_m(\boldsymbol{x})$  summiert demnach über die Beschleunigungen aller Achsen über die gesamte Bahn. Eine alternative Möglichkeit, die Güte der Bahn zu bestimmen, ergibt sich über die geometrischen Bahneigenschaften. Diese Auswertung ist sinnvoll, wenn keine Robotersimulation zur Verfügung steht. Dann kann die Auswertung über die Formeln (5.6)-(5.8) geschehen.

Als letzte Zielfunktionskomponente kann der Overspray berücksichtigt werden. Overspray entsteht, wenn Material neben die funktionale Fläche gespritzt wird. Der relative Overspray lässt sich dadurch bestimmen, dass der Quotient zwischen dem auf der funktionalen Flächen abgelagerten und dem ausgegebenen Material bestimmt wird:

$$E_o(\boldsymbol{x}) = \left( 1 - \frac{\sum_{i=0}^m A_i z_i(\boldsymbol{x})}{V_s \tau_n} \right)^2. \quad (4.9)$$

Hier ist  $V_s$  das Volumen an Material, das pro Zeiteinheit ausgegeben wird.  $V_s \tau_n$  ist demnach das Materialvolumen, das während des gesamten Beschichtungsvorgangs verspritzt wird. Der Quotient aus dem auf der Oberfläche abgelagerten Materialvolumen, welches über die Summe der Schichtdicke multipliziert mit den entsprechenden Flächen berechnet wird, und dem gesamt ausgegebenen Volumen wird schließlich bestimmt. Dieser Wert wird von 1 subtrahiert und quadriert und liefert so den finalen Zielfunktionswert.

Die meisten der oben genannten Zielfunktionswerte sind Mittelwerte. Manchmal kann es jedoch von Vorteil sein, den maximalen Wert zu bestimmen und zu minimieren. Zum Beispiel können im Mittel hohe Achsbeschleunigungen vom Roboter noch akzeptabel sein, während eine einzelne sehr hohe Beschleunigung bei ansonsten geringen Beschleunigungen nicht mehr ausgeführt werden kann. Die maximalen Werte der entsprechenden Zielfunktionskomponenten sind

$$E_{ml}(\boldsymbol{x}) = \max_{i=0,\dots,m} \left( \frac{z_i(\boldsymbol{x}) - Z_i}{Z_i} \right)^2, \quad (4.10)$$

$$E_{mt}(\boldsymbol{x}) = \max_{j=0,\dots,n} E_{t,j}(\boldsymbol{x}), \quad (4.11)$$

$$E_{mz}(\boldsymbol{x}) = \max_{j=1,\dots,n} l(\boldsymbol{x}_{j-1}, \boldsymbol{x}_j), \quad (4.12)$$

$$E_{mm}(\boldsymbol{x}) = \max_{\substack{j=1,\dots,n-1 \\ k=1,\dots,6}} \left| \frac{\Phi_{j+1,k} - \Phi_{j,k}}{t_j t_{j-1}} - \frac{\Phi_{j,k} - \Phi_{j-1,k}}{t_{j-1}^2} \right|. \quad (4.13)$$

Die Zielfunktionswerte können schließlich in einem Vektor zusammengefasst werden, um in der mehrkriteriellen Optimierung genutzt zu werden:

$$\mathbf{E}(\boldsymbol{\mathcal{X}}) = (E_l(\boldsymbol{\mathcal{X}}), E_t(\boldsymbol{\mathcal{X}}), E_z(\boldsymbol{\mathcal{X}}), E_m(\boldsymbol{\mathcal{X}}), E_o(\boldsymbol{\mathcal{X}}), E_{ml}(\boldsymbol{\mathcal{X}}), E_{mt}(\boldsymbol{\mathcal{X}}), E_{mz}(\boldsymbol{\mathcal{X}}), E_{mm}(\boldsymbol{\mathcal{X}}))^T. \quad (4.14)$$

Nicht in allen Fällen und für alle Bahnstrukturen sind alle Zielfunktionskomponenten gleich anzuwenden. Zum Beispiel spielen Zwischenbahnen bei Punkt-zu-Punkt Bahnen keine Rolle. Daher können beliebig viele dieser Zielfunktionskomponenten ausgewählt und genutzt werden. Auch eine Gewichtung der Komponenten in der Optimierung ist denkbar.

## 4.7 Evolutionäre Optimierung

Bei Evolutionären Algorithmen handelt es sich um Metaheuristiken, mit denen Optimierungsprobleme näherungsweise gelöst werden können. Metaheuristiken eignen sich vor allem dann, wenn kein exaktes Verfahren zur Optimierung anwendbar ist oder wenn die Zielfunktion nur durch komplexe Simulation ausgewertet werden kann und keine Gradienten der Zielfunktion bestimmt werden können. Die Funktionsweise Evolutionärer Algorithmen ist an die natürliche Evolution angelehnt und beinhaltet die Konzepte der Population, Mutation, Rekombination sowie Selektion. Eine Übersicht über die Komponenten von Evolutionären Algorithmen sowie eine Zusammenfassung der frühen Entwicklung sind in [11] zu finden. In [12] wird ebenfalls die Entwicklung von Evolutionären Algorithmen betrachtet, jedoch werden vor allem die Unterschiede der sich parallel entwickelten Paradigmen der Evolutionären Strategien, der Evolutionären Programmierung und der Genetischen Algorithmen herausgestellt. Die Hauptunterschiede liegen in der Nutzung verschiedener Repräsentationen und der Nutzung unterschiedlicher Operatoren mit unterschiedlichen Wahrscheinlichkeiten. Auch wenn die meisten hier genutzten Operatoren an die der Genetischen Algorithmen angelehnt sind, wird im Folgenden einfach von Evolutionären Algorithmen gesprochen.

### 4.7.1 Komponenten eines Evolutionären Algorithmus

Ein Evolutionärer Algorithmus basiert auf einer Population von Individuen, wobei jedes Individuum eine Lösung für das Problem darstellt. Jedes Individuum besitzt ein Chromosom, das die entsprechende Lösung codiert. Das Chromosom wiederum besteht aus einzelnen Genen. Der genaue Aufbau des Chromosoms und der Gene ist abhängig von der Repräsentation des Problems und muss in jedem Einzelfall neu bestimmt werden. Häufig bestehen die Gene aus binären Werten oder ganzen oder reellen Zahlen. Beim Genotyp/Phänotyp Mapping wird das Chromosom eines Individuums in eine Lösung für das konkrete Problem überführt. Über eine Zielfunktion, die für komplexe Probleme häufig mit Hilfe einer Simulation ausgewertet wird, wird eine solche Lösung bewertet.

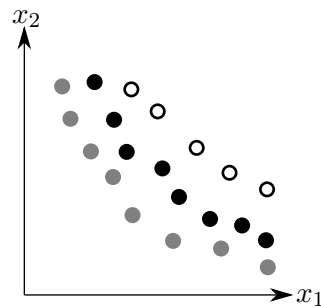


Abbildung 4.5: Beispiel für eine mehrkriterielle Optimierung, in der  $x_1$  und  $x_2$  minimiert werden sollen. Grau: Echte Pareto-Front; Schwarz, ausgefüllt: Aktuelle Pareto-Front; Schwarz, unausgefüllt: Dominierte Individuen.

Die Güte der Lösung wird als Fitness bezeichnet. Diese kann über einen einzelnen Wert oder, bei mehrkriteriellen Problemen, auch über einen Vektor ausgedrückt werden.

Um aus einer Population von Individuen neue Lösungen zu generieren, werden verschiedene Operatoren verwendet. Der erste Operator ist die Rekombination, die aus mindestens zwei bestehenden Individuen ein oder mehrere neue generiert. Hierbei werden große Teile der Elternchromosomen für das Kind übernommen und neu zusammengesetzt. Der zweite Operator ist die Mutation, die eher geringfügige Änderungen an einem Chromosom durchführt, um den Suchraum zu explorieren. Die Elternselektion ist für die Auswahl der Individuen zuständig, die für die Rekombination und Mutation verwendet werden, während die Überlebensselektion entscheidet, welche Individuen in die Population aufgenommen bzw. aus dieser entfernt werden. Die Operatoren und Komponenten des Evolutionären Algorithmus werden im Folgenden genauer vorgestellt.

#### 4.7.2 Zielfunktion, Fitness und mehrkriterielle Optimierung

Der Evolutionäre Algorithmus benutzt die sogenannte Fitness, um die Güte von Individuen zu bestimmen. Die Fitness kann im einfachsten Fall über den Wert der Zielfunktion ausgedrückt werden, wenn es sich um ein Maximierungsproblem handelt. Bei Minimierungsproblemen wird häufig der Kehrwert des Zielfunktionswertes genommen. Je größer demnach die Fitness ist, desto höher ist die Güte der durch das Individuum repräsentierten Lösung.

Bei mehrkriteriellen Problemen gibt es anstelle eines Zielfunktionswertes einen Zielfunktionsvektor, dessen Komponenten die einzelnen Werte der Zielfunktionskomponenten beinhalten. Das Ziel der Optimierung ist in solchen Fällen die Optimierung aller Komponenten gleichzeitig. Bei fast allen praktischen Problemen sind die Zielfunktionskomponenten nicht unabhängig voneinander und es kann eine der Komponenten häufig nicht verbessert werden, ohne eine andere zu verschlechtern. Eine Möglichkeit, mit dieser Situation umzugehen, ist mittels des Konzepts der Pareto-Optimalität [112]. Bei diesem Konzept wird nicht eine einzelne beste Lösung gesucht, sondern alle Lösungen, die

Pareto-Optimal sind. Eine Lösung ist genau dann Pareto-Optimal, wenn es nicht möglich ist, eine Komponente zu verbessern, ohne gleichzeitig eine andere zu verschlechtern. Alle Lösungen, die diese Eigenschaft erfüllen, bilden die sogenannte Pareto-Front. Ein weiterer Begriff in diesem Zusammenhang ist die Pareto-Dominanz. Eine Lösung dominiert eine andere, wenn die dominierende Lösung in allen Komponenten mindestens genauso gut und in mindestens einer Komponenten besser ist, als die dominierte Lösung. In Abbildung 4.5 ist beispielhaft eine Pareto-Front für ein zweikriterielles Optimierungsproblem dargestellt, wobei beide Kriterien minimiert werden sollen. Die tatsächliche, jedoch in den meisten Fällen unbekannte Front ist das Optimierungsziel. Die Population nähert sich im Verlaufe der Optimierung dieser Front an. Während dominierte Individuen schlechter sind als nicht dominierte Individuen, lassen sich zwei nicht wechselseitig dominierende Individuen nicht miteinander vergleichen.

Evolutionäre Algorithmen eignen sich direkt dazu, mehrkriterielle Probleme zu lösen. Die Population besteht in dem Fall aus den Individuen der aktuellen Pareto-Front und, je nach Populationsgröße, einigen dominierten Individuen. In der Selektion können verschiedene Techniken wie zum Beispiel Skalarisierung [165, 161] genutzt werden, um zu ermitteln, welche Individuen zur Rekombination genutzt werden und welche überleben. Die anderen Operatoren werden bei einfacher und mehrkriterieller Optimierung auf die gleiche Art und Weise verwendet.

### 4.7.3 Repräsentation

Die Wahl einer geeigneten Repräsentation des Problems mittels eines Chromosoms ist sehr wichtig für die Nutzung von Evolutionären Algorithmen und muss zu Beginn des Algorithmendesigns festgelegt werden. Dabei ist die Eignung einer Codierung von dem konkreten Problem abhängig. Häufig bieten sich binäre, ganzzahlige oder reellwertige Strings als Repräsentation an. Auch andere Möglichkeiten werden genutzt, wie zum Beispiel die Repräsentation als Baum, welche vor allem für die Darstellung von Programmcode verwendet wird. Bei Problemen mit indizierten Werten wird häufig auf die Repräsentation mit ganzen Zahlen zurückgegriffen, wie zum Beispiel beim Travelling Salesman Problem (TSP), bei dem die Zahlen die Indizes der Städte angeben, die besucht werden müssen [104].

In vielen Fällen ist die Länge des Chromosoms im Vorhinein bekannt und fest. Zum Beispiel werden bei der direkten Repräsentation des TSPs die Indizes der Städte als Gene genutzt, deren Anzahl die Länge des Chromosoms bestimmt. In anderen Fällen, wie bei der Bahnplanung in dieser Arbeit, ist die Länge des Chromosoms unbekannt und kann sich im Laufe des Verfahrens ändern.

Die Repräsentation hat des Weiteren Auswirkungen auf die nutzbaren Operatoren. Je nach Codierung können einige weit verbreitete Operatoren nicht verwendet werden.

#### 4.7.4 Initialisierung

Die Initialisierung hat zur Aufgabe, die erste Population von Individuen zu erstellen. Hier werden häufig zufällige Individuen erzeugt. Um eine möglichst gleichmäßige Überdeckung des Lösungsraums zu gewährleisten wird manchmal das Latin Hypercube Sampling [114] genutzt. Die initiale Population wird ausgewertet, um die Fitness der Individuen zu bestimmen. Danach werden die Operatoren des Evolutionären Algorithmus in einer Schleife angewendet.

#### 4.7.5 Rekombination

Der erste Operator zur Modifikation von Individuen bei Evolutionären Algorithmen ist die Rekombination, auch Crossover genannt. Hierbei handelt es sich um einen Operator, der aus zwei oder mehr Eltern ein oder mehrere Kinder erzeugt. Hierzu werden große Teile der Chromosomen der Eltern übernommen und neu zusammengesetzt. Die Idee ist, dass die Kombination bereits guter Individuen wieder eine gute oder noch bessere Lösung ergibt. Mit Hilfe dieses Operators soll Fortschritt erzielt werden. Bei der Repräsentation mittels reell- oder ganzzahligen Werten werden die drei Operatoren One-Point-Crossover, Two-Point Crossover und Uniform-Crossover häufig verwendet. Diese Rekombinationsoperatoren werden unter anderem in [112] und speziellere in [104, 97, 135, 156] beschrieben. Neben diesen sehr bekannten Operatoren können aber auch viele andere genutzt werden. Häufig bietet es sich an, besonders auf das gegebene Problem und dessen Repräsentation zugeschnittene Operatoren zu erstellen.

#### 4.7.6 Mutation

Die Mutation ist der zweite klassische Operator, der neben der Rekombination in Evolutionären Algorithmen genutzt wird. Im Gegensatz zur Rekombination handelt es sich bei der Mutation um einen unären Operator, der auf einem einzelnen Chromosom arbeitet. Das Ziel des Operators ist es, durch kleinere, zufällige Änderungen an dem Chromosom den Suchraum zu explorieren.

Es sind eine Menge Operatoren entwickelt worden, wobei viele dieser Operatoren auf die jeweiligen spezifischen Probleme zugeschnitten sind. Einige Operatoren sind jedoch problemunabhängig und können bei gleicher Repräsentation immer verwendet werden. Eine weitere Unterscheidung der Operatoren kann über den Einflussbereich der Mutation geleistet werden. Einige Operatoren operieren auf einzelnen Genen, während andere auf ganzen Blöcken von Genen arbeiten. Eine Übersicht über viele Operatoren, vor allem im Bereich des TSPs, sind in [104] zusammengestellt.

Wenn problemspezifisches Wissen über das zu lösende Problem vorhanden ist, können neue Operatoren entwickelt oder vorhandene Operatoren gezielt auf Stellen des Chromosoms angewendet werden, die für eine Veränderung besonders problematischer Gene

sorgen. Voraussetzung hierfür ist die genauere Kenntnis des Problems sowie dessen Repräsentation mittels des Chromosoms, was im Allgemeinen nicht gegeben sein muss.

#### 4.7.7 Elternselektion

Die Aufgabe der Elternselektion ist die Auswahl von zwei (oder mehr) Individuen der aktuellen Population, aus denen mit Hilfe der Rekombination und Mutation ein neues Individuum erzeugt wird. Die Wahl der Eltern ist von besonderer Bedeutung, da über die konkrete Auswahl die Höhe des sogenannten Selektionsdrucks bestimmt wird, der angibt, ob nur besonders fitte oder auch weniger fitte Individuen ausgewählt werden. Der Selektionsdruck ist damit für die Konvergenz des Verfahrens mitverantwortlich. Des Weiteren hat die Elternselektion im mehrkriteriellen Fall Einfluss auf die Verteilung der Individuen auf der Pareto-Front.

Die Selektion kann mittels verschiedener Operatoren durchgeführt werden, die unterschiedliche Eigenschaften bezüglich des Selektionsdrucks und der entstehenden Verteilung der Individuen im Zielfunktionsraum oder dem Definitionsraum haben. Voss et al. [165] und vanMoffert et al. [161] analysieren in diesem Zusammenhang die Nutzung von Skalarisierungsmethoden, Raquel et al. [136] untersucht den Einfluss der *Crowding Distance* und Mauldin [113] und Ursem [159] untersuchen Operatoren, die die *Genvielfalt* erhalten sollen.

#### 4.7.8 Überlebensselektion

Die Überlebensselektion stellt einen Operator dar, der auswählt, welche Individuen in der Population verbleiben und welche nicht. Es gibt verschiedene Versionen evolutionärer Algorithmen, die die Selektion auf unterschiedliche Weise durchführen. Klassisch kann man zwischen der Plus- und der Komma-Selektion unterscheiden. Bei der  $(\lambda, \mu)$ -Selektion werden aus einer Elternpopulation mit  $\lambda$  Individuen  $\mu$  Kinder erzeugt. Die nächste Generation wird dann ausschließlich aus den Kindern gebildet, während alle Eltern verworfen werden. Bei der  $(\lambda + \mu)$ -Selektion wird die nächste Generation aus den Eltern und den Kindern erzeugt.

Die Wahl der Individuen für die nächste Generation kann mit mehr oder weniger Selektionsdruck gestaltet werden. Je höher der Druck ist, desto wahrscheinlicher ist es, dass die guten Individuen selektiert werden. Bei geringerem Druck haben auch die schlechteren Individuen Chancen, in die nächste Generation aufgenommen zu werden. Dies ist unter anderem wichtig, um vorzeitige Konvergenz in ein lokales Optimum zu vermindern, welche Auftritt, wenn die Vielfalt unterschiedlicher Gene in den Chromosomen der Individuen gering ist.

Operatoren zur Selektion werden zum Beispiel in [161, 165, 113, 159, 136] vorgestellt und analysiert.

### 4.7.9 Bekannte Algorithmen

Bekannte Ansätze zur Modellierung von Evolutionären Algorithmen für mehrkriterielle Zielfunktionen basieren auf dem Ranking von Lösungen. Das Ranking wird erzeugt, indem die Anzahl der von einer Lösung dominierten anderen Lösungen bzw. die Anzahl der Lösungen, die diese Lösung dominieren, gezählt wird. Lösungen mit gleichem Rang sind entsprechend vergleichbar gut. Um die Pareto-Front möglichst gleichmäßig abzudecken, wird in der Selektion häufig zusätzlich zum Rang die Dichte der Lösungen in der Umgebung analysiert. Bekannte Verfahren, die auf diesen Konzepten basieren, sind zum Beispiel SPEA [181], SPEA2 [179], NSGA [150] oder NSGA-II [46]. Diese Algorithmen funktionieren vor allem bei geringer Dimensionalität sehr gut. Bei hochdimensionalen Problemen (mehr als drei Dimensionen) wird die Konvergenz dieser Methoden zunehmend schlechter. Dies liegt unter anderem daran, dass die relative Anzahl an nicht vergleichbaren Lösungen deutlich steigt und die Evaluation der Abdeckung des Lösungsraums sehr aufwendig wird, weswegen neuere Algorithmen wie NSGA-III vorgeschlagen wurden [45, 91]. Dieser Algorithmus beinhaltet weiterführende Dominanzbegriffe und ein Referenzpunktsystem.

### 4.7.10 Anwendung der Evolutionären Optimierung auf die Bahnstrukturen

Der oben vorgestellte Evolutionäre Algorithmus bildet das Grundkonzept für die Optimierung bei der Bahnplanung. Für das Erstellen eines konkreten Algorithmus basierend auf den oben beschriebenen Bahnstrukturen müssen einerseits eine konkrete Repräsentation der Bahn als Chromosom festgelegt und andererseits die am besten geeigneten Operatoren ausgewählt werden. Die Repräsentation ergibt sich für die drei Bahnstrukturen direkt aus deren Definition. Die Auswahl der Operatoren ist jedoch schwierig, da die Wirkung und das Zusammenwirken der Operatoren auf den Phänotyp nicht einfach vorhergesagt werden kann. Daher werden geeignete Operatoren auf Basis einer Analyse ihrer Wirkungsweise ausgewählt.

Im nächsten Abschnitt wird zunächst die Problemstruktur bei der Bahnplanung mit Hilfe von Vergleichen zu ähnlichen Problemen analysiert, bevor anschließend die Umsetzung des Evolutionären Algorithmus für die Punkt-zu-Punkt Bahnstruktur konkret beschrieben wird.

## 4.8 Problemstruktur

Um die Struktur des Problems der Bahnplanung besser zu verstehen, soll diese unter Zuhilfenahme von strukturell ähnlichen Problemen genauer untersucht werden. Durch den Vergleich zu diesen Problemen können Gemeinsamkeiten und Unterschiede aufgedeckt werden, die zu einem besseren Verständnis von der zu Grunde liegenden Problemstruktur



führen. Die Erkenntnisse können anschließend genutzt werden, um den Evolutionären Algorithmus so zu gestalten, dass er besonders gute, problemangepasste Methoden nutzt. Die betrachteten Probleme sind das Travelling Salesman Problem, das Rucksackproblem, das Travelling Thief Problem und das Scheduling.

#### 4.8.1 Zusammenhang mit dem TSP

Das Travelling Salesman Problem (TSP) ist ein sehr gut untersuchtes Optimierungsproblem, bei dem ein Weg für einen Handlungsreisenden gesucht wird. Konkret wird die kürzeste Rundreise basierend auf einer gegebenen Menge von Städten gesucht, so dass jede Stadt genau einmal besucht wird und der Handlungsreisende am Ende wieder an seinem Startpunkt ankommt.

Der Zusammenhang zwischen dem TSP und der Bahnplanung besteht in der Überdeckungseigenschaft beider Probleme. Wie oben erwähnt, müssen beim TSP alle Städte genau einmal besucht werden. Bei einer dichten Anordnung der Städte wird die zugrunde liegende Fläche von der Tour gewissermaßen überdeckt. Bei der Bahnplanung für thermisches Spritzen muss jeder Punkt der Oberfläche von der Bahn so überdeckt werden, dass die tatsächliche Beschichtung der Zielbeschichtung entspricht. Bei der Nutzung von Hilfspunkten auf der Oberfläche, die als Zielpunkte von Bahnsegmenten dienen, kann ein direkter Zusammenhang zwischen dem TSP und der Bahnplanung beim thermischen Spritzen hergestellt werden. Es ergeben sich allerdings auch wichtige Unterschiede, da die Forderung, jede Stadt genau einmal zu besuchen, nicht genutzt werden kann. Die Häufigkeit, mit der ein Zielpunkt angefahren werden muss, hängt von der Zielbeschichtung und den Parametern des Spritzprozesses ab, so dass einzelne Zielpunkte mehrfach angefahren werden können. Genauso gut kann es vorkommen, dass Punkte auf der Oberfläche gar nicht angefahren werden, da der Footprint ausgedehnt ist und damit einen größeren Teil der Fläche beeinflusst. Zwei Variationen des TSPs, das Close-Enough TSP [75] und das Covering Salesman Problem [41, 73] können diesen Umstand berücksichtigen, da nicht jede Stadt besucht werden muss, solange die Tour nahe genug an jede Stadt herankommt. Diese Generalisierungen des TSPs sind der Bahnplanung in diesem Punkt sehr ähnlich.

Die Komponente der Bahnplanung, die keine direkte Entsprechung in den oben genannten TSP-Varianten hat, ist die thermische Situation auf der Oberfläche. Diese ändert sich dynamisch mit der Zeit und wird vom Verlauf der Spritzpistole beeinflusst. Dynamische Veränderungen der Situation werden vom Time Dependent TSP, siehe [144], berücksichtigt, indem die Kosten für eine Reise zwischen den Städten dynamisch gestaltet werden. Die Kosten sind allerdings nur von den äußeren Umständen abhängig, nicht jedoch von der aktuellen Tour, wie es aber beim thermischen Spritzen der Fall ist.

Bei dem TSP handelt es sich um ein NP-vollständiges Problem. Dies bedeutet, dass es kein exaktes Lösungsverfahren mit polynomieller Worst Case Laufzeit gibt, wenn nicht  $P = NP$  gilt. Aus diesem Grund werden häufig Metaheuristiken eingesetzt, um in weniger Zeit eine approximative Lösung zu bestimmen. Unter anderem wurden viele



Lösungen auf Basis von Evolutionären Algorithmen vorgeschlagen. Diese unterscheiden sich vor allem in der Nutzung verschiedener Operatoren und Repräsentationen. Viele Möglichkeiten hierzu sind in [104] zusammengestellt.

Die Herausforderung beim Übertrag von Lösungsstrategien vom TSP auf die Bahnplanung liegt in der unterschiedlichen Repräsentationsform und den genutzten Operatoren. Da die Städte beim TSP genau einmal besucht werden müssen, werden häufig Strukturen benutzt, die dies erzwingen. So werden in der Repräsentation zum Beispiel Permutationen von Städteindizes verwendet, die von den Operatoren ebenfalls nur umsortiert werden. Mehrfachbesuche von Städten werden so unterbunden, was in der Bahnplanung jedoch möglich sein muss. Hieraus ergibt sich auch, dass die Länge der Chromosomen bei der Bahnplanung veränderlich sein muss, damit die Punkte der Oberfläche beliebig häufig besucht werden können. Es ist daher insgesamt notwendig, die Repräsentation und viele der vorgeschlagenen Operatoren (z. B.: [104, 97, 152, 156]) zu ersetzen, da sie für die Bahnplanung nicht geeignet sind.

Auf Grund der Popularität des TSPs sind viele unterschiedliche Analysen zur Untersuchung der Eigenschaften der Evolutionären Algorithmen für das Problem durchgeführt worden. Viele Arbeiten befassen sich mit der Auswahl oder der Konstruktion neuer Operatoren sowie der Wahl einer geeigneten Operatorkombination für die Rekombination, die Mutation und die Selektion. Tsai et al. [158] analysieren eine Menge Operatoren und wählen darauf aufbauend eine Kombination, die sehr gute Ergebnisse auf dem gegebenen Problem erzielt. Dieses Verfahren, dass die Operatoren auf Basis einer vorhergehenden Analyse auswählt, soll auch in dieser Arbeit genutzt werden, da im Vorhinein nicht klar ist, welche Operatoren und vor allem welche Kombination von Operatoren die besten Ergebnisse liefern können. Contreras et al. [40] untersuchen das Zusammenwirken von Operatoren und stellen fest, dass es sinnvoll ist, verschiedene Operatoren gleichzeitig zu nutzen, um Synergieeffekte der Operatoren auszunutzen. Basierend auf diesen Ergebnissen werden in dieser Arbeit die Mutationsoperatoren untersucht, so dass am Ende eine Menge der zur Verfügung stehenden Operatoren mit unterschiedlichen Gewichtungen genutzt werden, die zusammen die besten Ergebnisse liefern. Eine Untersuchung von Razali et al. [138] befasst sich mit Selektionsoperatoren, wobei die Tournament Selektion am besten abschneidet. Dieser Operator steht auch für die Bahnplanung zur Verfügung und kann bei entsprechend guten Resultaten bei der Analyse der Operatoren verwendet werden. Der Übertrag von Ergebnissen von einem auf ähnliche Probleme ist allerdings problematisch, wie Puljic et al. [135] feststellen. Die Autoren untersuchen die Qualität einiger Operatoren, die für das TSP vorgeschlagen wurden, auf dem ähnlichen Vehicle Routing Problem (VRP). Eine Erkenntnis ist, dass die besten Operatoren für das TSP nicht zugleich die besten Operatoren für das VRP darstellen. Diese Analyse zeigt demnach, dass Ergebnisse und Erkenntnisse im Bereich des TSPs trotz der Ähnlichkeit zur Bahnplanung nicht einfach übernommen werden sollten, sondern für das vorliegende Problem einzeln untersucht werden sollten. Dies gilt insbesondere für die Operatoren des Evolutionären Algorithmus.

### 4.8.2 Zusammenhang mit dem Rucksackproblem

Das Rucksackproblem ist ein kombinatorisches Optimierungsproblem, bei dem aus einer Menge von Elementen mit einem Wert und einem Gewicht diejenige Teilmenge ausgewählt werden soll, die den Wert maximiert und das Gewichtslimit, das vom Rucksack vorgegeben wird, nicht überschreitet. Das Problem kann einfach auf den mehrkriteriellen Fall ausgeweitet werden. In diesem Fall gibt es mehrere Rucksäcke und die Elemente haben jeweils einen Einfluss auf die verschiedenen Rucksäcke.

Der Zusammenhang des Rucksackproblems mit der Bahnplanung beim thermischen Spritzen kann vor allem in der Ablagerung der Partikel auf der Oberfläche sowie in gewissen Grenzen in dem Wärmeverhalten gesehen werden. Durch eine Diskretisierung der Oberfläche kann jedem (diskreten) Punkt auf der Oberfläche ein Rucksack zugeordnet werden. Jeder Rucksack kann mit Partikeln und Wärme befüllt werden. Das Ziel ist das Erreichen der Zielschichtdicke mit Hilfe der in dem Rucksack befindlichen Partikel und die Minimierung der Varianz der Wärme in den Rucksäcken. Das Minimieren der Varianz anstelle der Maximierung der Kapazitätsnutzung über die verschiedenen Rucksäcke hinweg stellt einen großen Unterschied zum originalen Rucksackproblem dar und verändert somit das originale Problem sehr stark, da die Rucksäcke nicht mehr unabhängig voneinander betrachtet werden können.

Auch das Rucksackproblem ist wie das TSP ein NP-schwieriges Problem. Daher werden häufig Metaheuristiken eingesetzt. Für die Analyse von Evolutionären Algorithmen wird das Rucksackproblem gerne verwendet, da es leicht zu formulieren ist und dennoch die Basis vieler praktischer Anwendungen bildet. Frühe Arbeiten [99, 180] befassen sich mit der Lösung des Rucksackproblems mit Hilfe von Evolutionären Algorithmen, insbesondere im mehrkriteriellen Fall.

Neuere Arbeiten befassen sich mit spezielleren Themen rund um die zu verwendenden Operatoren und Komponenten des Evolutionären Algorithmus für das Rucksackproblem. In [80] werden Initialisierungsmethoden analysiert und festgestellt, dass diese Einfluss auf das Ergebnis haben. Daher ist es sinnvoll, sich Gedanken über geeignete Initialisierungsmethoden zu machen, wie es auch in dieser Arbeit getan wird. Lu et al. [111] nutzen eine Quantum inspirierte Repräsentationsform für die Chromosomen. Diese ist sehr flexibel und nutzt reelle Werte anstelle der sonst üblichen Bitstrings. Die tatsächlichen Werte werden demnach wie in der Quantenphysik *unscharf*. Die Anwendbarkeit des Konzepts ist jedoch stark vom Problem bzw. dessen Repräsentation abhängig und zum Beispiel für die Punkt-zu-Punkt Repräsentation bei der Bahnplanung ungeeignet. Andere Autoren befassen sich mit der Anwendung von anderen Evolutionskonzepten, zum Beispiel der Differentiellen Evolution, auf das Rucksackproblem [153]. Auch hier muss die Repräsentation anders gestaltet werden, da reellwertige Gene verwendet werden.

Neben dem Erstellen und Anpassen vorhandener Operatoren werden Evolutionäre Algorithmen für das Rucksackproblem auch analysiert. Eine ausführliche Analyse verschiedener Evolutionärer Algorithmen, die zum Beispiel Pareto-Dominanz basiert, skalierungsbasiert und hypervolumenbasiert arbeiten, wird in [90] durchgeführt. Die

Analyse zeigt, dass die verschiedenen Algorithmientypen für unterschiedliche Problemvarianten mit unterschiedlicher Anzahl Dimensionen oder korrelierten Zielfunktionen sehr unterschiedliche Ergebnisse erzielen. Demnach gibt es nicht den einen besten Algorithmus, sondern jeweils auf unterschiedlichen Probleminstanzen andere, gut funktionierende Algorithmen. Dies ist ein Hinweis darauf, dass es notwendig ist, die Operatoren und die Repräsentation entsprechend des konkreten Problems bei der Bahnplanung anzupassen, um einen gut funktionierenden Algorithmus zu entwerfen.

### 4.8.3 Zusammenhang mit dem TTP

Das Travelling Thief Problem wurde erstmals 2013 von Bonyadi et al. [23] vorgeschlagen mit dem Hintergrund, dass viele reale Probleme aus mehreren (komplexen) Teilproblemen zusammengesetzt sind, welche wechselseitig abhängig sind. Beim TTP werden das Travelling Salesman Problem und das Rucksackproblem miteinander verbunden. Das kombinierte Problem besteht darin, dass ein Dieb eine Menge gegebener Städte besuchen muss, wobei in jeder Stadt ein oder mehrere Gegenstände vorhanden sind, die der Dieb stehlen und in einem Rucksack mit gegebener Kapazität transportieren kann. Jeder Gegenstand hat wie beim Rucksackproblem einen Wert und ein Gewicht. Je mehr Gewicht in dem Rucksack vorhanden ist, desto langsamer kann der Dieb reisen. Die Reisedauer hat jedoch einen direkten Einfluss auf den endgültigen Zielfunktionswert, da der Rucksack gemietet ist und der Mietpreis von der Dauer der Reise abhängig ist. Der Preis für die Miete wird am Ende einer Tour vom Gewinn abgezogen. Auf diese Weise hängen die beiden Teilprobleme zusammen, so dass die optimalen Lösungen der einzelnen Probleme nicht direkt zur optimalen Lösung des zusammengesetzten Problems führen. Benchmarkprobleme und eine Heuristik werden in [133] vorgeschlagen.

Der Zusammenhang des TTP mit der Bahnplanung ergibt sich größtenteils aus den Zusammenhängen der einzelnen Komponenten. Die TSP Komponente des TTP ergibt die Tour, die bei der Bahnplanung für die Überdeckung der Oberfläche verantwortlich ist. Bei entsprechender Problemformulierung können die Städte mit Punkten auf der Oberfläche in Verbindung gebracht werden, wie zum Beispiel bei der Punkt-zu-Punkt Bahnstruktur. Ein Unterschied in der Bahnplanung ist, dass die Anzahl der Besuche jedes Punkts uneingeschränkt ist, damit bei unterschiedlicher Prozesskonfiguration die geforderte Zielschichtdicke erreicht werden kann. Beim Rucksackanteil ist es wie beim Originalproblem sinnvoll, eine Verallgemeinerung mit mehreren Rucksäcken zu treffen. Dann kann pro (diskretisiertem) Oberflächenpunkt ein Rucksack betrachtet werden, der die Partikel und die Wärme aufnehmen kann. Das Ziel ist demnach wiederum das genaue Erreichen der Zielschichtdicke in jedem Rucksack und die Minimierung der Varianz der Wärmewerte in den Rucksäcken. Der letzte und nicht aus den einzelnen Komponenten stammende Zusammenhang kann aus der Semantik der Reisegeschwindigkeit gezogen werden. Diese sorgt bei dem TTP für die Verschränkung der beiden Probleme, da die Geschwindigkeit von dem Gewicht des Rucksacks bestimmt wird und damit eine geänderte Städtereihenfolge zu einer Verbesserung des Zielfunktionswertes führen kann, auch wenn die Länge des Weges zunimmt (wertvollere Gegenstände mit hohem Gewicht

sollten erst spät eingesammelt werden). Eine Analogie hierzu kann gezogen werden, da die thermische Varianz ganz entscheidend von der Reihenfolge der Besuche der einzelnen Punkte auf der Oberfläche abhängt. Daher kann eine Umsortierung der Reihenfolge auch hier den Zielfunktionswert deutlich verändern. Hierdurch wird die starke Verschränkung der Teilprobleme auch beim thermischen Spritzen deutlich.

Da es sich bei dem TTP um ein komplexes Problem handelt (NP-schwierig), werden auch hier viele heuristische Lösungen sowie Metaheuristiken vorgeschlagen. Dabei ist zu unterscheiden, ob die beiden Komponenten im Wesentlichen unabhängig voneinander optimiert werden oder beide Probleme gleichzeitig betrachtet werden. In die erste Gruppe fallen Untersuchungen von Polyakovskiy et al. [134], die mit Hilfe von Mixed Integer Programmierung zunächst das TSP lösen und anschließend das Rucksackproblem auf der fertigen TSP Lösung optimieren. Faulkner et al. [61] entwickeln verschiedene Heuristiken, die einzelne Teilkomponenten modifizieren, und untersuchen die Auswirkungen bei der Anwendung der Methoden. Bonyadi et al. [24] schlagen ebenfalls einfache Heuristiken vor, die die einzelnen Probleme getrennt lösen (zuerst das TSP und schließlich das Rucksackproblem auf gegebener Tour), jedoch ebenfalls eine Lösung basierend auf Koevolution. Letztere löst beide Problemtile gleichzeitig, was zu wesentlich besseren Lösungen führt. Lourenco et al. [110] greifen diese Lösungsmethode auf und entwickeln einen Evolutionären Algorithmus, der beide Teilprobleme gleichzeitig optimiert.

Mei et al. entwerfen ebenfalls Heuristiken, die im Zusammenhang mit Memetischen Algorithmen genutzt werden [118, 117, 119]. In diesen Algorithmen werden die beiden Komponenten mit Hilfe der Heuristiken parallel entwickelt. In [119] wird schließlich in einer mathematischen Analyse festgestellt, dass die beiden Komponenten untrennbar sind, so dass die Optimierung der einzelnen Komponenten nicht zu einer gesamtoptimalen Lösung führen. Insgesamt bedeuten diese Erkenntnisse, dass die Komponenten von komplexen zusammengesetzten Problemen nicht isoliert betrachtet werden können. Im Falle der Bahnplanung ist eine TSP-Komponente, eine Rucksack-Komponente und eine Scheduling-Komponente zu finden. Diese müssen als Gesamtproblem betrachtet werden und können nicht einzeln optimiert werden.

Eine Erweiterung des TTP Problems setzt mehrere Diebe ein [32], die auf großen Probleminstanzen mehrere Touren gleichzeitig erzeugen. Die Experimente deuten darauf hin, dass zwei bis drei Diebe die besten Ergebnisse liefern. Im Falle der Bahnplanung für eine einzelne Spritzpistole ist jedoch keine Entsprechung für mehrere Diebe vorhanden.

#### 4.8.4 Zusammenhang mit dem Scheduling

Das Job-Shop Scheduling Problem ist ein Optimierungsproblem, bei dem verschiedene Jobs auf eine gegebene Menge von Maschinen aufgeteilt werden müssen, so dass die Fertigstellung aller Jobs möglichst früh geschieht. Hierbei kann nicht jede Maschine jeden Job verarbeiten und die Verarbeitungszeiten der Jobs sind ungleich lang. Bei dieser Art von Problem handelt es sich um ein kombinatorisches Optimierungsproblem.

Eine Möglichkeit, um einen Zusammenhang zwischen der Bahnplanung und dem Scheduling zu ziehen, ist, dass das Auftragen von Partikeln an einer gewissen Stelle der Oberfläche zu einer gewissen Zeit geschehen soll. Dabei ist der Zeitpunkt des Auftrags von wesentlicher Bedeutung, um die thermischen Varianzen zu minimieren. Demnach entsprechen die einzelnen Punkte der (diskretisierten) Oberfläche den Jobs, während die Spritzpistole die Jobs abarbeiten kann, indem sie die Punkte der Oberfläche beschichtet. Die Bahnplanung ist jedoch etwas komplizierter, da die Wärmeentwicklung auf der Oberfläche durch Abkühlung und Diffusion dynamisch ist und ein Job, also ein Punkt auf der Oberfläche, nicht immer in einem Überlauf vollständig beschichtet werden muss. Auch ist zu beachten, dass benachbarte Punkte gleichzeitig beschichtet werden, da der Spritzkegel ausgedehnt ist. Dennoch erinnert die Optimierung der Zeitpunkte, zu denen die Bahn an einer gewissen Stelle der Oberfläche ankommt, an das Scheduling Problem.

Da es sich bei dem Job-Shop Scheduling um ein NP-schwieriges kombinatorisches Optimierungsproblem handelt, sind größere Instanzen des Problems nicht in akzeptabler Zeit exakt lösbar. Daher werden häufig Metaheuristiken wie Evolutionäre Algorithmen eingesetzt. Das Job-Shop Scheduling Problem steht auch heute noch im Fokus aktueller Forschung und es werden viele unterschiedliche Abwandlungen Evolutionärer Algorithmen vorgeschlagen. Vor allem die genutzten Operatoren oder die Fokussierung auf verschiedene Aspekte beim Scheduling werden viel untersucht. Eine Übersicht über den Stand der Forschung ist zum Beispiel in [68] zu finden. In [132] wird ein Algorithmus entworfen, dessen Fokus auf der Repräsentation des Problems und einer lokalen Suche mittels Scatter Search für das ressourcenbeschränkte Scheduling liegt. Chang et al. [34] betrachten *Building Blocks* und Chiang et al. [38] entwerfen einen Algorithmus der besonders einfach ist und dadurch mit nur wenigen Parametern auskommt.

Andere Arbeiten befassen sich mit spezielleren Themen wie dem stochastischen Job-Shop Scheduling [87] oder dem Rescheduling nach dem Ausfall von Maschinen [143]. Eine umfangreiche Analyse von Evolutionären Algorithmen auf verschiedenen Varianten des Job-Shop Scheduling wird in [124] durchgeführt.

Alle genannten Arbeiten beschreiben verschiedene Evolutionäre Algorithmen, die unterschiedliche Operatoren und Repräsentationen nutzen. Da selbst innerhalb des gleichen Problemgebiets kein einzelner Algorithmus für alle Problemvarianten gleich gute Ergebnisse liefert, siehe auch [143], muss davon ausgegangen werden, dass der Evolutionäre Algorithmus für die Bahnplanung auf Basis der gegebenen Aufgabenstellung angepasst werden muss. Dies schließt vor allem die Wahl der Repräsentation und der Operatoren ein.

## 4.9 Optimierung von Punkt-zu-Punkt Bahnen

Nachdem in den vorhergehenden Abschnitten Konzepte zur Bahngenerierung und zur Erstellung eines Evolutionären Algorithmus präsentiert und mögliche Zielfunktionskomponenten vorgestellt wurden, soll in diesem Abschnitt ein konkreter Algorithmus zur

Lösung des Bahnplanungsproblems erarbeitet werden. Dazu wird das Konzept des Evolutionären Algorithmus auf die Punkt-zu-Punkt Bahnerstellung angepasst. Im Einzelnen wird eine geeignete Repräsentation für das Chromosom des Evolutionären Algorithmus ausgewählt und mögliche Operatoren für die Rekombination, Mutation und Selektion werden bestimmt. Auch eine Auswahl der möglichen Zielfunktionskomponenten und eine Transformation in einen alternativen Zielfunktionsraum werden angegeben. Schließlich wird noch auf einige Beschleunigungstechniken, wie zum Beispiel Caching oder Parallelisierung eingegangen. Mit Hilfe verschiedener Techniken werden die Parameter des Algorithmus bestimmt, die für das Problem besonders geeignet sind, und einige Analysen zeigen die Stärken und Schwächen des Verfahrens auf.

### 4.9.1 Zielfunktion

Die Zielfunktion ist von entscheidender Bedeutung bei der Optimierung. Im Folgenden wird zunächst auf die Auswahl der Zielfunktionskomponenten eingegangen und anschließend auf mögliche Transformationen des Zielfunktionsraums.

#### Zielfunktionsauswahl

Ein Problem bei der mehrkriteriellen Optimierung ist die wachsende Zahl nicht vergleichbarer Individuen bei steigender Dimensionalität des Zielfunktionsraums. Die Anzahl der Individuen, die die aktuelle Pareto-Front bilden, steigt daher ebenfalls mit der Anzahl der Zielfunktionsdimensionen. Die Auswahl geeigneter Individuen zur Fortpflanzung oder zum Überleben mit Hilfe der Selektionsoperatoren wird demnach schwieriger. In der Literatur wird daher zwischen *Multi-* und *Many-objective Optimization* unterschieden, wobei ersteres Optimierungsprobleme mit bis zu drei oder vier Dimensionen, letzteres Probleme mit mehr als drei Dimensionen beschreibt. Gängige Algorithmen wie der NSGA-II [46] funktionieren gut für Probleme mit bis zu drei oder vier dimensional Zielfunktionen, werden jedoch für höherdimensionale Probleme zunehmend schlechter. Neuere Algorithmen, wie der NSGA-III [45] nutzen erweiterte Methoden, wie Referenzpunkte, um die Optimierung zielgerichtet durchzuführen.

Um dem Problem der stetig steigenden Schwierigkeit bei höheren Dimensionen zu begegnen, sollte die Anzahl der Zielfunktionskomponenten möglichst klein gehalten werden. In Abschnitt 4.6 wurden insgesamt neun mögliche Zielfunktionskomponenten vorgestellt. Hiervon werden im Folgenden jedoch nur vier verwendet, um eine geringere Dimensionalität zu erhalten. Im Einzelnen sind das:

- $E_l(\mathcal{X})$  zur Reduktion des Fehlers in der aufgetragenen Schicht,
- $E_t(\mathcal{X})$  zur Minimierung der thermischen Varianzen während des Prozesses,
- $E_m(\mathcal{X})$  zur Verringerung der Roboterachsbeschleunigungen,
- $E_o(\mathcal{X})$  zur Vermeidung von Overspray, der beim Spritzen neben das ROI entsteht.



Im Falle der Punkt-zu-Punkt Bahnen ist der Zielfunktionswert  $E_z(\mathcal{X})$  ohne Aussage, da keine Zwischenbahnen am Rand des EROIs erzeugt werden. Die Maximalwerte der Zielfunktionswerte, also  $E_{mi}(\mathcal{X})$ ,  $E_{mt}(\mathcal{X})$ ,  $E_{mz}(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$  werden zugunsten einer geringeren Dimensionalität nicht genutzt. Diese könnten für die schnellere Verringerung der maximalen Werte in den einzelnen Komponenten verwendet werden. Jedoch werden die maximalen Werte in den genutzten Zielfunktionskomponenten implizit berücksichtigt, so dass diese während der Optimierung reduziert werden. Die Erhöhung des Problems um weitere vier Dimensionen scheint daher ungerechtfertigt.

### Zielfunktionstransformationen und Desirability

Ein Problem, das sich bei mehrkriterieller Optimierung ergibt, ist, dass die verschiedenen Zielfunktionskomponenten durch völlig unterschiedliche Fehlermaße ausgedrückt werden können. Dadurch kommt es vor, dass die quantitativen Zielfunktionswerte unterschiedliche Größenordnungen haben. Dies kann ein Problem bei der Selektion darstellen, da die dort verwendeten Skalarisierungsmethoden zum Teil von den Größenordnungen der einzelnen Zielfunktionen abhängen.

Ein naheliegendes Konzept zur Transformation der Zielfunktionswerte auf eine gleiche Größenordnung ist die Nutzung von Gewichten. Jede Komponente wird mit einem Gewicht multipliziert und kann somit auf eine gleiche Größenordnung gebracht werden. Auch eine Fokussierung der Optimierung auf eine oder mehrere Komponenten ist auf diese Weise möglich, wenn die Gewichte einzelner, wichtiger Zielfunktionskomponenten vergrößert werden. Der wesentliche Nachteil des Verfahrens ist, dass die Festlegung entsprechender Gewichte durch den Nutzer vor der Optimierung durchgeführt werden muss. Dazu muss entsprechendes Problemwissen vorhanden sein.

Eine weitere Möglichkeit, neben der Nutzung von Gewichten, ist die Normierung aller Wertebereiche auf einen vorher festgelegten Bereich, zum Beispiel auf  $[0, 1]$ . Dazu kann folgende Formel verwendet werden:

$$d(y) = \frac{y - l}{u - l}. \quad (4.15)$$

Hierbei ist  $y$  der aktuelle Zielfunktionswert,  $l$  der kleinste mögliche Wert (zum Beispiel 0) und  $u$  der höchste mögliche Wert.  $d(y)$  ist dann der auf den neuen Bereich normierte Wert. Ein Problem ergibt sich, da bei vielen Komponenten, die in Abschnitt 4.6 vorgestellt wurden, der Wertebereich nach oben nicht beschränkt ist und demnach kein fester Wert für  $u$  gesetzt werden kann. In diesem Fall kann eine adaptive Methode den maximalen Wert in jeder Komponente im Verlauf der Optimierung automatisch festlegen. Dazu kann entweder der maximale Wert der Zielfunktionskomponente genutzt werden, der während der gesamten Optimierung gefunden wurde, oder der maximale Wert, der aktuell in der Population vorhanden ist. Nachteilhaft an diesem Vorgehen ist jedoch, dass sich die Werte einzelner Individuen im Verlauf der Optimierung schlagartig ändern können, wenn neue Werte für  $u$  bestimmt werden.

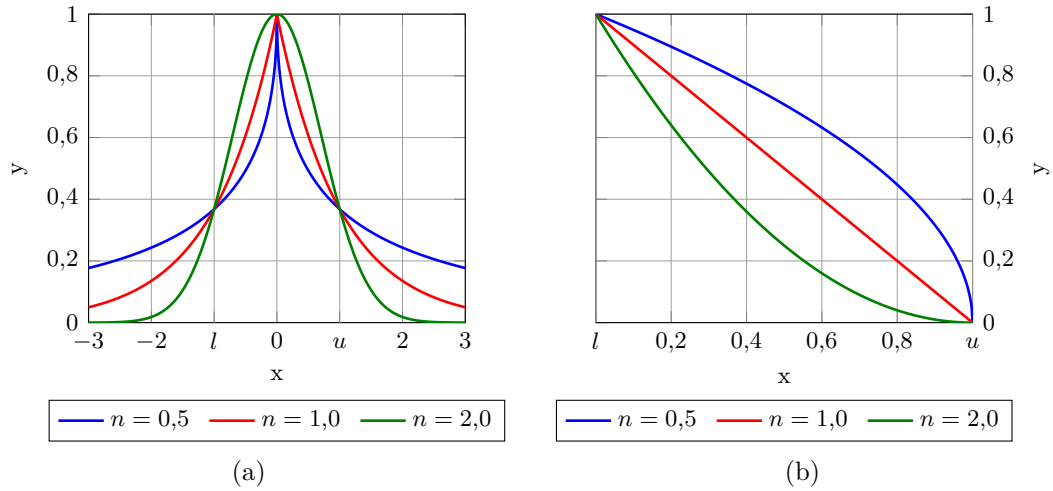


Abbildung 4.6: Verschiedene *Desirability Functions* nach (a) Harrington (siehe Gleichung (4.16)) und (b) Derringer/Suich (siehe Gleichung (4.17)) mit unterschiedlichen Werten für  $n$ .  $l$  und  $u$  geben die untere bzw. die obere Grenze an.

Eine andere Möglichkeit ist durch das Konzept der *Desirability Functions* gegeben. Dieses Konzept ist erstmals von Harrington [79] vorgestellt worden und von Derringer und Suich [50] überarbeitet worden. Neuere Arbeiten von Mehnen und Trautmann haben sich ebenfalls mit den *Desirability Functions* auseinandergesetzt [157, 116, 115].

Das Ziel der *Desirability Functions* ist, ebenso wie bei der Normierung, eine Transformation der Wertebereiche der Zielfunktionskomponenten. Im Gegensatz zu der Normierung werden jedoch flexibel anpassbare Funktionen genutzt, um die Funktionswerte auf den Bereich  $[0, 1]$  abzubilden. Auf Grund der Nicht-Linearität der *Desirability Functions* können durch die Parametrisierung der Funktionen unterschiedliche Präferenzen der einzelnen Zielfunktionskomponenten in der Optimierung ausgedrückt werden. Auf diese Weise kann Expertenwissen in der Optimierung genutzt werden.

In Abbildung 4.6 sind die von Harrington und Derringer/Suich vorgeschlagenen Funktionen für einige unterschiedliche Parametrisierungen dargestellt. Die entsprechenden Funktionsvorschriften sind in den Gleichungen (4.16) und (4.17) gegeben:

$$d(y)_H = \exp\left(-\left|\frac{2y - (u + l)}{u - l}\right|^n\right), \quad (4.16)$$

$$d(y)_{D/S} = \begin{cases} 1 & \text{wenn } y < l, \\ \left(\frac{u-y}{u-l}\right)^n & \text{wenn } l \leq y \leq u, \\ 0 & \text{wenn } y > u. \end{cases} \quad (4.17)$$

Bei den Funktionen von Harrington handelt es sich um symmetrische Funktionen, die den Mittelwert zwischen den beiden Grenzen  $u$  und  $l$  als wünschenswertesten Punkt beschreiben. Über den Parameter  $n$  kann die Form der Kurve angepasst werden, so dass



entweder ein größerer Bereich um den Zielwert hohe Werte für die *Desirability* liefert oder nur ein recht schmaler Bereich. Bei der Funktion von Derringer/Suich handelt es sich in der angegebenen Form um eine *Smaller-the-better* Funktion. Dies bedeutet, dass kleine Werte eine höhere *Desirability* ergeben. Über die Parameter  $u$  und  $l$  wird der Bereich der Funktion angegeben, in der sie von 0 bis 1 steigt und über  $n$  wird die Form der Kurve bestimmt. Hiermit kann wiederum sehr intuitiv festgelegt werden, ob ein eher breiter oder schmaler Bereich um den Zielwert akzeptabel ist. Von der Derringer/Suich Funktionsvorschrift gibt es entsprechende Varianten für *Larger-the-better* und *Nominal-the-best*.

Das Problem des nicht begrenzten Wertebereichs, das bei der Normierung bereits angesprochen worden ist, betrifft auch die *Desirability Functions*. Im Falle der vorliegenden Minimierungsfunktionen kann als untere Schranke für die Derringer/Suich Variante 0 angegeben werden. Die obere Schranke ist jedoch nicht ohne weiteres verfügbar. Wenn diese Grenze daher ohne Expertenwissen nicht gesetzt werden kann oder soll, bieten sich adaptive Methoden an. Hier könnte der Wert für  $u$  demnach auf den bisher größten während der Optimierung vorgekommenen Wert gesetzt werden. Da die Optimierung dafür sorgen sollte, dass vornehmlich kleinere Werte in allen Zielfunktionen generiert werden, kommen schlagartige Änderungen an den Werten aller Individuen bei neuen größten Werten gerade im späteren Verlauf der Optimierung selten vor.

Eine Alternative hierzu ist eine adaptive Variante, die die *Desirability Werte* in der Population kontinuierlich an die in der Population vorkommenden Werte anpasst, wie es in [151] vorgeschlagen wird. Dieses Verfahren in Kombination mit Steady-State Selektion sorgt jedoch für permanente Änderungen an den Fitnesswerten in der gesamten Population und sollte daher vermieden werden. Andernfalls erhöht sich die Gefahr, dass die Selektionsoperatoren annähernd zufällige Individuen auswählen.

Durch die Transformation mit den *Desirability Functions* wird auch die Art des Problems geändert, da die *Desirability* maximiert werden soll, während die Zielfunktionskomponenten bei der Bahnplanung allesamt minimiert werden sollen. Da alle Operatoren auf beiden Zielfunktionsräumen arbeiten können, wird auch das transformierte Problem in ein Minimierungsproblem umgewandelt, indem die Werte wie folgt umgerechnet werden:

$$E'_x(\mathcal{X}) = 1 - d(E_x(\mathcal{X})). \quad (4.18)$$

Hierbei ist  $d(\cdot)$  eine der *Desirability Functions* und  $E_x(\mathcal{X})$  eine der Zielfunktionskomponenten.

Die Optimierung kann wahlweise auf den originalen, gewichteten, oder transformierten Zielfunktionskomponenten arbeiten. Die Selektionsmethoden des Algorithmus nutzen die entsprechenden Werte, um die Individuen zur Fortpflanzung und zum Überleben auszusuchen.

### 4.9.2 Repräsentation

Die Auswahl einer Repräsentationsform für eine Bahn in Form eines Chromosoms ist im Falle der Nutzung von Punkt-zu-Punkt Bahnen sehr naheliegend. Da die Bahnsegmente bei dieser Konstruktionsform mit Hilfe der kürzesten geodätischen Verbindung zwischen zwei Knoten generiert werden (vgl. Abschnitt 4.5.3), kann das Chromosom mit Hilfe eines einfachen Indexstrings dargestellt werden. Die Werte in dem Indexstring geben die Indizes der Knoten des Oberflächennetzes des Werkstücks an. Je zwei aufeinanderfolgende Indizes generieren ein Bahnsegment und durch die Nutzung der Endpunkte vorhergehender Bahnsegmente als Startpunkte der nächsten Segmente entsteht eine stetige Bahn. Das konkrete Genotyp/Phänotyp Mapping zur Erstellung von Bahnsegmenten wird entsprechend der Methodik aus Abschnitt 4.5.3 durchgeführt.

Im Gegensatz zu vielen anderen Problemrepräsentationen, die zum Beispiel bei dem TSP genutzt werden, ist die Länge des Chromosoms im Verlauf der Optimierung nicht fest. Dies liegt daran, dass die Chromosomlänge nicht proportional zur Bahnlänge ist. Je nach Lage der aufeinanderfolgenden in dem Chromosom codierten Knoten ist die Länge der Bahnsegmente unterschiedlich. Diese ändert sich entsprechend mit jeder Modifikation des Chromosoms. Daher ist es notwendig, dass das Chromosom flexibel in der Länge veränderlich ist, um Bahnen mit unterschiedlich vielen und unterschiedlich langen Bahnsegmenten zu generieren.

Die variable Länge des Chromosoms hat jedoch auch Auswirkungen auf die Operatoren. In einigen Fällen, wie zum Beispiel bei manchen Rekombinationsoperatoren, wird von gleich langen Chromosomen ausgegangen. Die Operatoren, die dies voraussetzen, müssen entsprechend angepasst werden, um mit den variablen Längen umgehen zu können.

Ein weiteres Problem der Repräsentation über einen Indexstring ist durch das sehr komplexe Genotyp/Phänotyp Mapping gegeben. Eine kleine Veränderung an dem Chromosom kann unterschiedlich große Veränderungen an der Bahn hervorrufen. Dies ist vergleichbar mit der sogenannten *Hamming Cliff* [10, 31], die bei binärer Codierung beschreibt, dass eine kleine Änderung am Phänotyp in manchen Fällen nur durch große Änderungen am Genotyp erreicht werden können (Um aus dem Bitstring 0111, welcher eine 7 repräsentiert, den String 1000, welcher eine 8 repräsentiert, zu generieren, werden 4 Bitflips benötigt). Um diesem Problem Rechnung zu tragen, werden lokale Operatoren für die Mutation eingesetzt. Diese sind so gestaltet, dass das Chromosom so modifiziert wird, dass geringe, aber gezielte Veränderungen an der Bahn durchgeführt werden.

### 4.9.3 Initialisierung

Durch die Initialisierung werden die Individuen generiert, aus denen die erste Population zusammengesetzt wird. Um einen möglichst großen Bereich des Suchraums abzudecken, soll hier eine Variante des Latin Hypercube Samplings [114] eingesetzt werden.

Das Latin Hypercube Sampling kann verwendet werden, um Punkte in einem Raum so auszuwählen, dass sie im Gegensatz zu einer einfachen randomisierten Auswahl

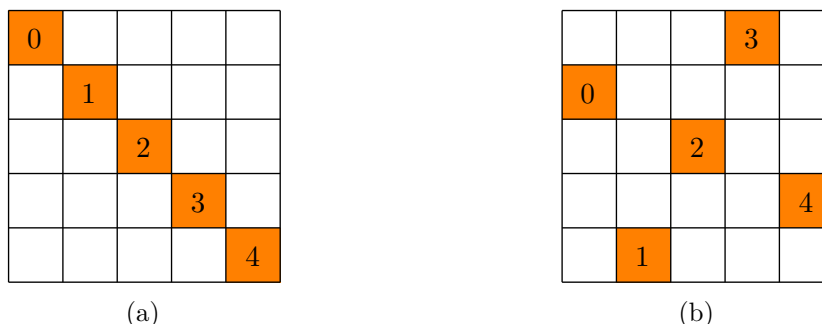


Abbildung 4.7: Beispiele für das Latin Hypercube Sampling. (a) Schlechte Verteilung mit minimaler Distanz von  $d = 1,41$  nach Gleichung (4.20). (b) Gute Verteilung mit minimaler Distanz von  $d = 2,24$  nach Gleichung (4.20).

besser über den Raum verteilt sind. Abbildung 4.7 zeigt zwei zweidimensionale Beispiele. Die Fläche wird durch ein Gitter in gleichgroße Quadrate unterteilt. Die Anzahl der Unterteilungen ergibt sich durch die Anzahl der auszuwählenden Punkte. Die eigentliche Auswahl der Quadrate geschieht so, dass in jeder Zeile und jeder Spalte genau ein Quadrat gewählt wird. In dem Beispiel in Abbildung 4.7b ergeben sich entsprechend 5 Punkte mit den Koordinaten  $(0, 3)$ ,  $(1, 0)$ ,  $(2, 2)$ ,  $(3, 4)$  und  $(4, 1)$ . Die Koordinaten können in Form einer Matrix  $\mathbf{M}$  aufgeschrieben werden, so dass sich in jeder Zeile die Koordinaten eines Elements befinden:

$$\mathbf{M} = \begin{pmatrix} 0 & 3 \\ 1 & 0 \\ 2 & 2 \\ 3 & 4 \\ 4 & 1 \end{pmatrix}. \quad (4.19)$$

Man erkennt, dass in jeder Spalte eine Permutation der Indizes vorhanden ist. Daher ist es auch in hochdimensionalen Räumen einfach, Koordinaten zu finden, um der Latin Hypercube Bedingung gerecht zu werden, indem in jeder Spalte entsprechende Indexpermutationen generiert werden.

Auf Grund der Tatsache, dass die Punkte, die über das Latin Hypercube Verfahren generiert wurden, zwar deutlich besser als randomisiert erzeugte Punkte, jedoch ebenfalls nicht vollkommen gleichmäßig über den Raum verteilt sind, kann das Verfahren noch etwas verbessert werden. Hierzu wird eine Metrik eingesetzt, die angibt, wie weit die beiden am nächsten zueinander liegenden Punkte voneinander entfernt sind. Dies kann mittels

$$d = \min_{i,j} \|\mathbf{M}_i - \mathbf{M}_j\|^2, \quad i \neq j \quad (4.20)$$

berechnet werden, wobei  $\mathbf{M}_i$  die  $i$ -te Zeile der Matrix angibt [98]. Ein einfaches Verfahren zur Verbesserung des Wertes erzeugt mehrere Samplings und nutzt dasjenige, das den

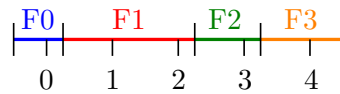


Abbildung 4.8: Beispiel zur Flächenumrechnung beim LHS. F0-F3 sind die gegebenen Flächen, wobei die Längen der Segmente den Beträgen der Flächeninhalte entsprechen. Die äquidistanten Abtaststellen 0-4 fallen jeweils in die Bereiche der Flächen.

größten minimalen Abstandswert liefert. Auf Grund der einfachen und schnellen Erzeugung der Samplings ist dieses Verfahren in der Initialisierungsphase des Evolutionären Algorithmus gut anwendbar.

Im konkreten Fall der Generierung initialer Chromosomen für die Bahnplanung kann das Verfahren leicht abgewandelt verwendet werden. Auf Grund der möglicherweise ungleichmäßigen Abtastung der Oberfläche des Werkstücks mit Knoten, wird eine zusätzliche Flächengewichtung genutzt. Jedem Knoten ist ein Anteil der Fläche des EROIs zugeordnet, siehe Abbildung 2.2. Wenn die Beträge der Größen der Flächen auf eine Achse aufgetragen werden, können die einzelnen Abschnitte mit den Knotenindizes beschriftet werden. In Abbildung 4.8 werden die Knotenindizes beispielhaft mit F0 bis F3 bezeichnet. In der Abbildung sind zusätzlich gleichverteilt die Positionen 0 bis 4 eingetragen. Diese stehen für die Koordinaten, die durch das Latin Hypercube ausgewählt werden (siehe Abbildung 4.7b). Die Koordinaten werden nun so umgerechnet, dass sich die Knotenindizes ergeben, welche direkt in den Chromosomen verwendet werden können. Die Koordinate (0,3) aus dem Beispiel oben würde im Chromosom (F0,F2) resultieren. Auf diese Art und Weise werden Knoten mit großen Flächen stärker berücksichtigt und die Oberfläche des Werkstücks wird gleichmäßiger überdeckt.

#### 4.9.4 Rekombination

Es sind viele unterschiedliche Rekombinationsoperatoren für verschiedene Evolutionäre Algorithmen vorgeschlagen worden. Im Folgenden werden einige dieser Operatoren beschrieben und genauer für die Eignung bei der Bahnplanung analysiert. Viele der Operatoren werden bei der Lösung des TSPs genutzt, siehe [104].

Beim *One-Point-Crossover* werden zwei Eltern verwendet. Zunächst wird ein Punkt in den Chromosomen ermittelt, an denen die beiden Elternchromosomen aufgetrennt werden. Die Teile werden anschließend über Kreuz neu zusammengesetzt, so dass der erste Teil des ersten Elternteils und der zweite Teil des zweiten Elternteils das neue Individuum ergibt. Ein zweites Individuum kann mit den anderen Teilen erzeugt werden. Da die Chromosomen nur an einer Stelle aufgetrennt werden, bleiben große Teile der Chromosomen unverändert erhalten, so dass die entstehenden Lösungen sehr ähnlich zu den Eltern sind.

Dieser Operator ist für die Bahnplanung uneingeschränkt nutzbar. Auf Grund der verwendeten Bahnrepräsentation werden der vordere Teil der ersten Bahn und der

hintere Teil der zweiten Bahn durch eine kurze Verbindungsbahn zusammengesetzt. Große Teile der Bahnstrukturen bleiben demnach erhalten.

Das *Two-Point-Crossover* ist sehr ähnlich zum One-Point Crossover, nutzt jedoch zwei Stellen zum Auftrennen der Chromosomen. An jeder Schnittstelle wird zwischen den beiden Eltern gewechselt. Auch hier können insgesamt zwei Kinder erzeugt werden.

Es ist ebenso möglich, noch mehr Schnittstellen zu erzeugen und häufiger zwischen den Eltern zu wechseln. Je größer jedoch die Anzahl der Schnittstellen ist, desto wahrscheinlicher ist es, dass gute Strukturen der Eltern zerstört werden. Dennoch kann der Operator mit beliebig vielen Schnittstellen ohne weitere Anpassungen genutzt werden.

Als Variante des *X-Point-Crossover* kann das *Uniform-Crossover* angesehen werden. Hierbei wird das Kind erzeugt, indem jedes Gen zufällig von dem ersten oder dem zweiten Elternteil genutzt wird. Die Wahrscheinlichkeit ein Gen vom ersten oder zweiten Elternteil zu nutzen, ist für beide Elternteile gleich, so dass in etwa die Hälfte der Gene beider Eltern genutzt wird. Dieser Operator eignet sich eher für Probleme, die keine Abhängigkeit in der Reihenfolge der Gene aufweisen. Bei der Bahnplanung kann der Operator zwar problemlos genutzt werden, jedoch werden die Bahnstrukturen der Eltern sehr stark verändert.

Die beim TSP genutzten Operatoren *Partially Mapped Crossover* und *Maximal Preservative Crossover* nutzen jeweils einen Block von Genen aus einem Elternteil und füllen den Rest des Chromosoms mit den Genen des zweiten Elternteils auf. Die Besonderheit dieses Operatortyps liegt darin, dass das Auffüllen der Gene der Vorschrift unterliegt, dass kein Gen doppelt in dem Chromosom vorkommen darf. Da jedoch genau diese Einschränkung im Falle der Bahnplanung nicht zutrifft, ist der Operator hier nicht geeignet. Ähnliches gilt für das *Cycle Crossover*, das die Auswahl der Gene so trifft, dass an jeder Stelle des Kindchromosoms entweder das entsprechende Gen des ersten oder das des zweiten Elternteils steht. Wenn die Bedingung der nicht doppelt vorkommenden Gene fallen gelassen wird, handelt es sich um einfaches uniformes Crossover.

Auch TSP-Operatoren, die die relative Reihenfolge von Teilen der Elternchromosomen unverändert lassen sollen, wie zum Beispiel *Order based Crossover*, *Position based Crossover* oder auch *Alternating-Position Crossover*, sind auf Grund der Möglichkeit mehrfach vorkommender Gene schwierig anzuwenden. Bei diesen Operatoren werden einige Gene eines Elternteils festgesetzt und die übrigen auf Grund der Reihenfolge ihrer Vorkommen in dem anderen Elternteil aufgefüllt. Hierbei werden bereits vorhandene Gene übersprungen. Wenn jedoch, wie im Falle der Bahnplanung, auch Mehrfachvorkommen von Genen vorgesehen sind, kann das Auffüllen der Chromosomen nicht mehr nach den vorgegebenen Schemata vonstatten gehen, so dass die Operatoren nicht angewendet werden können.

Das *Genetic Edge Recombination Crossover* nutzt die Kanten zwischen Städten beim TSP und wählt je eine der möglichen vier in den zwei Eltern vorkommenden Kanten, um die nächste Stadt zu erreichen. Danach werden alle Vorkommen der neu besuchten Stadt in der zu Grunde liegenden Tabelle aller Stadtverbindungen gelöscht. Dadurch

kann keine Stadt mehrfach besucht werden, so dass das entstehende Kind jede Stadt maximal einmal besuchen kann. Dieses Verhalten ist beim TSP erwünscht, nicht jedoch bei der vorliegenden Bahnplanung.

Operatoren, die auf den Kosten von einzelnen Kanten oder von Subtouren basieren, wie zum Beispiel *Heuristic Crossover* oder *Sorted Match Crossover* können bei der Bahnplanung nicht verwendet werden, da auf Grund der komplexen Zielfunktion keine Möglichkeit besteht, einzelne Kosten für Kanten oder Subtouren zu bestimmen.

Der *Voting Recombination Crossover* Operator untersucht mehr als zwei Eltern und nutzt eine Art Abstimmungsverfahren. Wenn an einer Position im Chromosom das gleiche Gen in mehr als  $x$  Eltern vorkommt, wird dieses Gen in das Kind übernommen. Die restlichen Gene werden durch Mutation aufgefüllt. Hierdurch sollen sich sehr gute Gene durchsetzen. Dieser Operator kann für die Bahnplanung uneingeschränkt genutzt werden.

Ein beliebter Operator beim TSP ist der *Edge Assembly Crossover* Operator [168]. Dieser basiert auf dem Konzept von *AB-Cycles*. Diese werden konstruiert, indem von einem Startknoten im TSP Graphen des ersten Elternteils eine adjazente Kante gewählt wird, danach von dem neu erreichten Knoten eine adjazente Kante vom zweiten Elternteil und so weiter. Sobald ein Kreis entstanden ist, werden die genutzten Kanten aus den Eltern entfernt und das Verfahren iteriert. Am Ende des Prozesses sind alle Kanten genutzt worden. Das Kind wird auf Basis eines der originalen Elternteile konstruiert, indem einige dieser Kreise verwendet werden. Dazu werden die Kanten, die sowohl in dem Elternteil und dem Kreis vorkommen, gelöscht und dafür die nur in dem Kreis vorkommenden Kanten, also die aus dem anderen Elternteil stammenden, eingefügt. Das entstandene Kind wird anschließend repariert, um wieder eine gültige TSP Lösung zu erhalten. Die Erzeugung von AB-Cycles ist im Falle der Bahnplanung nicht einfach möglich, da die Chromosomen nicht alle Gene genau einmal enthalten. Daher muss der Operator stark angepasst werden, um genutzt werden zu können.

Ein Operator, der eine Mischung aus Rekombination und Mutation darstellt, ist der *Inver-over* Operator [152]. Dieser wählt einen Bereich in dem Chromosom eines Individuums und invertiert die Gene in diesem Bereich. Mit einer gewissen Wahrscheinlichkeit wird der Bereich der Invertierung jedoch mit Hilfe eines anderen Individuums der Population so ausgewählt, dass nach der Invertierung zwei Indizes benachbart sind, die auch in dem zweiten Individuum benachbart sind. Auf diese Weise werden Informationen von einem Individuum auf ein anderes übertragen. Da auch dieser Operator darauf basiert, dass jedes Gen genau einmal in jedem Chromosom vorkommt, ist eine direkt Übertragung auf die Repräsentation bei der Bahnplanung schwierig.

Zusammenfassend lässt sich feststellen, dass es viele Operatoren gibt, die für unterschiedliche Probleme eingesetzt werden. Einige dieser Operatoren sind sehr allgemein gehalten und können problemlos auf das Bahnplanungsproblem angewendet werden. Andere Operatoren, vor allem solche, die auf das TSP zugeschnitten sind, müssen an die andere Repräsentation zunächst angepasst werden. Für die Bahnplanung wurden die Operatoren One-Point Crossover, Two-Point Crossover und Uniform Crossover implementiert, da

sich diese problemlos auf die Repräsentation der Punkt-zu-Punkt Bahnen anwenden lassen. Da jedoch nicht ersichtlich ist, welcher dieser Operator am besten geeignet ist und die besten Ergebnisse liefert, wird die Wahl des zu nutzenden Operators einem Parameter Tuning überlassen, siehe hierzu Abschnitt 4.9.12.

#### 4.9.5 Mutation

Mutationsoperatoren werden für die Veränderung von Genen genutzt, um neues Genmaterial in die Population einzufügen. Im Folgenden werden einige Operatoren vorgestellt, die auf Indexstrings verwendet werden können. Dabei funktionieren viele Operatoren problemübergreifend, da sie ausschließlich Änderungen an dem Chromosom vornehmen, ohne die Auswirkungen auf den Phänotyp zu berücksichtigen.

Die *Shift Mutation* ist ein problemspezifischer Operator, der lokale Veränderungen auf Bahnebene durchführt. Dazu wird ein Gen in dem Chromosom so verändert, dass anstelle des aktuellen Knotenindex ein Knotenindex eines Nachbarn genutzt wird. Die Nachbarschaft wird über den 1-Ring des ursprünglichen Knotens definiert. Durch diese Änderung verschiebt sich die Bahn lokal von einem Knoten auf einen Nachbarn und wird damit nur sehr begrenzt und lokal verändert.

Bei der *Reverse Mutation* handelt es sich um einen Operator, der Bahnsegmente umdreht. Dazu werden zwei benachbarte Gene vertauscht. Dies bewirkt, dass das Bahnsegment, das zwischen den zu den Genen gehörenden Oberflächenpunkten verläuft, in umgekehrter Richtung abgefahren wird. Als Seiteneffekt werden die Verbindungen zu diesen Oberflächenpunkten entsprechend verändert.

Die *Add Mutation* ist ein Operator, der eingesetzt werden kann, wenn, wie hier, die Länge des Chromosoms veränderlich ist. In diesem Fall wird zufällig gleichverteilt eine Stelle des Chromosoms ausgewählt, an der ein neues Gen eingefügt wird. Das neue Gen wird zufällig erzeugt, so dass neue Bereiche des Lösungsraums in der Umgebung der ursprünglichen Lösung untersucht werden.

Die *Delete Mutation* ist das Gegenstück zur Add Mutation und kann das Chromosom durch das Löschen von Genen verkürzen. Dazu wird ein Gen in dem Chromosom zufällig gleichverteilt ausgewählt und gelöscht. Eine untere Schranke für die Länge des Chromosoms sollte gegeben sein.

Bei der *Random Mutation* wird ein Gen zufällig verändert. Zunächst muss daher ein Gen des Chromosoms zufällig gleichverteilt ausgewählt werden, das modifiziert werden soll. Dieses Gen wird anschließend mit zufälligen Werten überschrieben.

Der *Exchange Operator* vertauscht zwei Gene in dem Chromosom. Die Gene, die getauscht werden sollen, werden zufällig gleichverteilt ausgewählt und anschließend in ihrer Position vertauscht. Dieser Operator hat vor allem dann eine große Wirkung, wenn das Problem auf besondere Weise von der Reihenfolge der Gene abhängt, wie bei der vorliegenden Bahnplanung.



Die *Displacement Mutation* ist ein Beispiel für einen Block-Operator. Block-Operatoren modifizieren ganze Blöcke anstelle von einzelnen Genen. Bei dem Displacement Operator wird ein Block in dem Chromosom zufällig gleichverteilt ausgewählt und dann innerhalb des Chromosoms verschoben. An der Reihenfolge der Gene wird ansonsten nichts verändert. Der Vorteil von Block Operatoren ist, dass bereits entwickelte Strukturen nur wenig verändert werden.

Der *Scramble Operator* ist ein blockorientierter Operator, der den Inhalt eines Blocks von Genen neu anordnet. Es wird demnach ein Block von Genen zufällig gleichverteilt ausgewählt und die Gene innerhalb dieses Blockes werden zufällig permutiert. Die neue Permutation hat insbesondere dann Auswirkungen, wenn die Reihenfolge der Gene in dem Problem eine große Bedeutung hat.

Die *Simple-Inversion Mutation* ist ein weiterer Block-Operator. Dieser wählt, wie die anderen Block-Operatoren, einen Block von Genen zufällig gleichverteilt und invertiert die Reihenfolge der Gene innerhalb dieses Blockes. Die Invertierung bedeutet eine Invertierung der Reihenfolge der Bahnsegmente und hat damit vor allem Auswirkungen auf die thermischen Varianzen.

Neben den oben aufgelisteten Operatoren, die mit Ausnahme der Shift- und Reverse-Mutation problemunabhängig die Gene des Chromosoms modifizieren, können auch noch weitere Operatoren eingeführt werden, die spezielles Problemwissen nutzen. Dies steht im Gegensatz zu dem ansonsten von dem Evolutionären Algorithmus genutzten Konzept der Blackbox Optimierung, bei der kein Problemwissen vorausgesetzt wird. Dennoch kann es von Vorteil sein, vorhandenes Problemwissen zu nutzen, wenn es zur Verfügung steht.

Auch wenn es sich bei der *Lin-Kernighan* Heuristik [108] eigentlich um ein eigenständiges Optimierungsverfahren handelt, kann das Grundkonzept auch als Mutationsoperator verstanden werden, der beim TSP genutzt werden kann. Die Grundidee besteht darin, dass die Tour lokal verändert wird, indem  $k$  Kanten verändert werden. Dabei werden ausgehend von einem Startpunkt Kanten getauscht, so dass sich der sogenannte *gain* verbessert. Dieser ergibt sich aus den Kosten der neu hinzugenommenen Kanten abzüglich der Kosten der gelöschten Kanten. Im Gegensatz zu den ähnlichen 2-Opt-, 3-Opt-, usw. Methoden ist der Lin-Kernighan Algorithmus flexibler, da er keine voreingestellte Anzahl an zu tauschenden Kanten pro Schritt benutzt. Der Parameter  $k$ , und damit die Anzahl der Kanten, wird dynamisch bestimmt. Dadurch ist es möglich, auch komplexe Operationen durchzuführen, um zum Beispiel lokalen Optima zu entkommen. Ein Problem bei der Nutzung des Lin-Kernighan Operators ist die häufige Berechnung verschiedener gain Werte, wenn Kanten für die Tauschoperatoren ausgewählt werden sollen. Für jede betrachtete Kante wird ein solcher gain Wert bestimmt. Beim TSP ist dies einfach möglich, da nur Kantenlängen betrachtet werden müssen. Bei der Bahnplanung jedoch müssen alle Zielfunktionskomponenten neu berechnet werden, was nur durch vollständige Simulation der Bahn geschehen kann. Entsprechend ergibt sich ein sehr großer Aufwand bei der Nutzung dieses Operators für die Bahnplanung.



Tabelle 4.2: Mutationsoperatoren

Problemspezifische Operatoren	Standard Operatoren	Blockorientierte Operatoren
Shift	Add	Displacement
Reverse	Delete	Scramble
–	Random	Simple-inversion
–	Exchange	–

Für die Optimierung bei der Bahnplanung stehen die in Tabelle 4.2 zusammengetragenen Operatoren zur Verfügung. Die Schwierigkeit besteht darin, die Anwendungswahrscheinlichkeiten so zu wählen, dass der Algorithmus gute Ergebnisse erzielt. Die Möglichkeiten hierzu werden in Abschnitt 4.9.12 beschrieben.

#### 4.9.6 Elternselektion

Die Elternselektion wählt zwei Eltern aus, die für die Rekombination und Mutation genutzt werden. Je nach Operator wird dabei mehr oder weniger Selektionsdruck erzeugt, so dass im Wesentlichen Individuen mit hoher Fitness oder beliebige Individuen gewählt werden. Im Folgenden werden verschiedene Operatoren für die Elternselektion vorgestellt.

#### Tournament basierte Operatoren

Das Tournament bildet die Basis verschiedener Operatoren. Dabei nutzt die Tournament Selektion das Konzept eines Turniers, bei dem verschiedene Individuen gegeneinander antreten müssen. Bei einem  $k$ -fachen Turnier werden  $k$  Individuen zufällig gleichverteilt ausgewählt und deren Zielfunktionswerte miteinander verglichen. Das Individuum mit dem besten Zielfunktionswert wird ausgewählt. Die Größe des Turniers ist entscheidend für den Selektionsdruck verantwortlich. Je größer das Turnier gewählt wird, desto wahrscheinlicher ist die Wahl des besten Individuums. Bei geringer Turniergröße ist der Selektionsdruck niedrig und auch geringer bewertete Individuen haben eine Chance ausgewählt zu werden. Bei mehrkriteriellen Problemen muss zunächst eine Skalarisierungsmethode verwendet werden, um einen skalaren Fitnesswert zu generieren.

In dieser Arbeit werden drei Möglichkeiten zur Skalarisierung eingesetzt:

1. Gewichtete Summe,
2. Tschebyscheff Skalarisierung,
3. Desirability Index.

Bei der *gewichteten Summe* werden die Zielfunktionskomponenten gewichtet aufsummiert:

$$E(\mathcal{X}) = \sum_{i=0}^{|\mathbf{E}(\mathcal{X})|-1} \lambda_i \mathbf{E}_i(\mathcal{X}). \quad (4.21)$$

Hierbei ist  $|\mathbf{E}(\mathcal{X})|$  die Anzahl der genutzten Zielfunktionskomponenten und  $\mathbf{E}_i(\mathcal{X})$  ist die  $i$ -te Zielfunktionskomponente.  $\lambda_i$  ist ein Gewicht, das für jede Komponente einzeln gewählt werden kann. Das Gewicht kann dazu verwendet werden, um alle Zielfunktionen auf die gleiche Größenordnung zu bringen oder einzelne Zielfunktionen zu fokussieren.

Die Grundlage bei der *Tschebyscheff-Selektion* ist die Tschebyscheff Skalarisierung aus [165]. Bei dieser Art der Skalarisierung wird aus den verschiedenen Zielfunktionskomponenten ein Fitnesswert abgeleitet. Um diesen Wert zu bestimmen, wird zunächst der ideale Punkt  $\mathbf{z}_i^*$  der aktuellen Population berechnet. Dieser ergibt sich zu

$$\mathbf{z}_i^* = \min_{\mathcal{X} \in P} \mathbf{E}_i(\mathcal{X}), \quad (4.22)$$

wobei  $\mathcal{X}$  die Bahn eines Individuums aus der Population  $P$  und  $\mathbf{E}_i(\mathcal{X})$  die  $i$ -te Zielfunktionskomponente des Individuums angibt. Aus diesem Punkt wird der utopische Punkt  $\boldsymbol{\mu}^*$  berechnet, der sich aus einer Verschiebung des idealen Punkts ergibt, zu

$$\boldsymbol{\mu}^* = \mathbf{z}^* - \boldsymbol{\epsilon}. \quad (4.23)$$

Der Vektor  $\boldsymbol{\epsilon}$  enthält in jeder Komponente eine kleine positive Zahl. Über die  $L_\infty$ -Metrik kann dann der Zielfunktionswert bestimmt werden mit

$$E(\mathcal{X}) = \max_i \lambda_i |\mathbf{E}_i(\mathcal{X}) - \boldsymbol{\mu}_i^*|, \quad (4.24)$$

wobei  $\lambda_i$  entsprechende Gewichte sind.

Der *Desirability Index* ist ein skalarer Wert, der sich aus den mit Hilfe der *Desirability Functions* transformierten Zielfunktionen berechnen lässt. Häufig wird die *geometric mean* Methode verwendet, die sich wie folgt berechnet [157]:

$$E(\mathcal{X}) = \left( \prod_{i=0}^{|\mathbf{E}(\mathcal{X})|-1} d(\mathbf{E}_i(\mathcal{X})) \right)^{1/|\mathbf{E}(\mathcal{X})|}. \quad (4.25)$$

Hierbei ist  $|\mathbf{E}(\mathcal{X})|$  die Anzahl der genutzten Zielfunktionskomponenten und  $\mathbf{E}_i(\mathcal{X})$  ist die  $i$ -te Zielfunktionskomponente. Die Zielfunktionskomponenten werden zunächst mit Hilfe der *Desirability Functions* transformiert, bevor alle Werte miteinander multipliziert werden und schließlich die *Gesamt-Desirability* über die  $|\mathbf{E}(\mathcal{X})|$ -te Wurzel bestimmt wird. Der *Desirability Index* hat die Eigenschaft, den Wert 0 zu liefern, wenn eine der Zielfunktionen 0 ist. Dies ist dann der Fall, wenn die Zielfunktion außerhalb des gewünschten Bereichs liegt. Dadurch werden Lösungen, die in einer Komponente völlig inakzeptabel sind, ausgeschlossen. Die *Desirability* kann im Anschluss von 1 subtrahiert werden, um anstelle einer Maximierung eine Minimierung zu erreichen.

## VAN

Ein weiterer Operator, der in [49] vorgestellt wurde, ist der *Vector Angle Neighborhood* Operator. Der Operator bestimmt zunächst einen beliebigen Kandidaten aus der Population. Mit einer gewissen Wahrscheinlichkeit wird als zweites Elternteil ein beliebiges anderes Individuum gewählt. Ansonsten wird als zweites Individuum einer der  $x$  nächsten Nachbarn gewählt. Die Nachbarschaft wird über die Winkel zwischen den Zielfunktionsvektoren bezüglich des idealen Punkts (siehe Gleichung (4.22)) bestimmt. Je geringer der Winkel ist, desto enger ist die Nachbarschaftsbeziehung. Der Winkel kann über das Skalarprodukt berechnet werden. Durch diese Selektion werden nahe beieinanderliegende Individuen bevorzugt gewählt, was zu einer besseren Ausnutzung bereits vorhandener Strukturen führt. Dieser Operator wird zusammen mit einem passenden Überlebensselektionsoperators vorgestellt (siehe Abschnitt 4.9.7).

### 4.9.7 Überlebensselektion

Bei der Überlebensselektion wird entschieden, welche Individuen in die Population aufgenommen bzw. welche aus der Population entfernt werden. Einige Operatoren für die Überlebensselektion werden im Folgenden kurz vorgestellt.

#### Pareto-Dominanz

Wie im Abschnitt 4.7.2 beschrieben, eignet sich im Bereich der mehrkriteriellen Optimierung die Pareto-Dominanz, um die Überlegenheit von Individuen gegenüber anderen zu ermitteln. Es ist daher naheliegend, die Pareto-Dominanz als Kriterium zur Ermittlung der in der Population verbleibenden Individuen zu wählen. Die Idee ist, dass diejenigen Individuen verworfen werden, die von anderen Individuen dominiert werden. Viele bekannte Algorithmen, wie zum Beispiel der NSGA-II, nutzen dieses oder das sehr ähnliche Prinzip des Rankings ebenfalls.

Für die Selektion in dieser Arbeit wird die Pareto-Dominanz wie folgt eingesetzt. Wenn die Population ihre geforderte Größe überschreitet, wird für jedes Individuum bestimmt, von wie vielen anderen Individuen der Population dieses dominiert wird. Die Individuen der aktuellen Pareto-Front werden nicht dominiert und stellen somit die besten Individuen dar. Solange die Population zu groß ist, wird jeweils eines der schlechtesten Individuen gemäß der Anzahl der dominierenden Individuen gelöscht. Dies wird so lange durchgeführt, bis entweder die Population ihre vorgegebene Größe erreicht oder es nur noch Individuen gibt, die nicht dominiert werden. Für den Fall, dass es nur noch nicht dominierte Individuen gibt, die Population aber noch immer zu groß ist, wird einer der unten aufgeführten Selektionsmechanismen ausgeführt, um die Population auf ihre geforderte Größe zu reduzieren.

### Survival of the fittest

Einer der einfachsten Selektionsoperatoren ist der Survival of the fittest Operator. Wie der Name bereits andeutet, überleben nur die fittesten Individuen. Dafür werden die  $n$  besten Individuen für den Verbleib in der Population gewählt, wobei  $n$  die geforderte Größe der Population darstellt. Dieser Operator zeichnet sich durch einen sehr hohen Selektionsdruck aus.

Im mehrkriteriellen Fall muss eine Skalarisierungsmethode genutzt werden, um die *besten* Individuen zu identifizieren. Eine einfache Methode ist die Nutzung der gewichteten Summe, wie es hier gemacht wird. Es können aber auch die Tschebyscheff oder die *Desirability* Skalarisierung genutzt werden, siehe Abschnitt 4.9.6.

### Tournament basierte Operatoren

Die Tournament basierten Operatoren für die Überlebensselektion funktionieren fast so wie die bei der Elternselektion, vergleiche Abschnitt 4.9.6. Es werden  $k$  verschiedene Individuen ausgewählt, die gegeneinander antreten. Der Verlierer des Turniers, also das Individuum mit der geringsten Fitness unter den gewählten Individuen, wird aus der Population entfernt. Über die Größe des Turniers wird bestimmt, wie groß der Selektionsdruck ist. Bei diesem Operator ist der Erhalt der besseren Individuen nicht garantiert, jedoch ist die Wahrscheinlichkeit des Erhalts mit hohem Selektionsdruck sehr groß. Bei mehrkriteriellen Problemen kann eine Skalarisierungstechnik einen einzelnen Zielfunktionswert liefern, der für das Turnier genutzt wird. Im Einzelnen können die gewichtete Summe, die Tschebyscheff Skalarisierung und der *Desirability Index* aus Abschnitt 4.9.6 genutzt werden.

### Crowding Distance

Ein weiterer Faktor bei mehrkriteriellen Evolutionären Algorithmen ist die genetische Vielfalt in der Population. Um diese zu erhalten oder zu vergrößern und damit vorzeitige Konvergenz zu mindern, können Operatoren auf Basis der *Crowding Distance* genutzt werden. Die Crowding Distance misst, wie stark eine gewisse Region des Zielfunktionsraums von den Individuen belegt ist.

Die Crowding Distance kann bestimmt werden, indem für alle Individuen die Entfernung zu seinen Nachbarn im Zielfunktionsraum pro Dimension berechnet wird:

$$c = \sum_{i=0}^{|\mathbf{E}(\mathcal{X})|-1} |\mathbf{E}_i(\mathcal{X}_1) - \mathbf{E}_i(\mathcal{X}_2)| + |\mathbf{E}_i(\mathcal{X}_3) - \mathbf{E}_i(\mathcal{X}_1)|. \quad (4.26)$$

Hierbei ist  $|\mathbf{E}(\mathcal{X})|$  die Anzahl der Zielfunktionskomponenten,  $\mathcal{X}_1$  die Bahn des betrachteten Individuums,  $\mathcal{X}_2$  die Bahn des Individuums, das den nächst kleineren, und  $\mathcal{X}_3$  die Bahn des Individuums, das den nächst größeren Zielfunktionswert in dieser Dimension

hat. Die am Rand liegenden Individuen haben nur einen Nachbarn und bekommen den Wert  $\infty$  als Crowding Distance.

Die Selektion basiert darauf, dass die nächste Population möglichst gut über die aktuelle Pareto-Front verteilt ist. Daher werden die Individuen mit geringer Crowding Distance verworfen, während die mit großer Crowding Distance erhalten bleiben.

## VAN

Als Gegenstück zum VAN Operator der Elternselektion aus Abschnitt 4.9.6 wird der VAN Operator für die Überlebensselektion in [49] beschrieben. Dieser nutzt wieder die Winkel zwischen den Zielfunktionsvektoren aus. Bei der Überlebensselektion werden die Individuen verworfen, die einen geringen Winkel zu einem anderen Individuum aufweisen. Genauer werden die beiden Individuen bestimmt, die den geringsten entsprechenden Winkel zueinander haben und dasjenige mit schlechterem skalarisiertem Fitnesswert wird verworfen.

### 4.9.8 Auswahlselektion

Die Auswahlselektion wird nur einmal nach Fertigstellung der Evolutionären Optimierung ausgeführt. Wenn die Optimierung endet, ist eine finale Population von Individuen erzeugt worden. Aus dieser Population muss nun ein einzelnes Individuum gewählt werden, das als *beste* für die weiteren Schritte wie die Bahnoptimierung und später zur Nutzung für den realen Prozess verwendet wird.

Prinzipiell kann ein Experte die Bahnen der finalen Population einzeln anschauen und die favorisierte Bahn selektieren. Es ist jedoch vorzuziehen, diesen Schritt zu automatisieren. Dazu werden die Verfahren angewendet, die zur Skalarisierung bei der Tournament Selektion verwendet werden. Jedes der drei Verfahren ist dabei in der Lage, jedem Individuum einen einzelnen Fitnesswert zuzuordnen. Dieser wird dann genutzt, um das beste Individuum bezüglich dieses Wertes zu selektieren.

### 4.9.9 Caching

Ein häufig auftretendes Problem bei Metaheuristiken wie dem Evolutionären Algorithmus ist der hohe Laufzeitbedarf. Bedingt durch die meist sehr großen und kompliziert strukturierten Suchräume sind sehr viele Iterationen notwendig, bevor sehr gute Lösungen gefunden werden. Des Weiteren wird die Zielfunktion häufig, wie hier bei der Bahnplanung, auf Basis einer komplexen Simulation ausgewertet, so dass dies eine der zeitaufwändigsten Komponenten der Optimierung ist. Aber auch andere Elemente des Algorithmus können große Laufzeiten aufweisen. Im Falle der Bahnplanung ist das Genotyp/Phänotyp Mapping ein weiterer zeitaufwändiger Schritt. Dies liegt daran, dass aus einfachen Indizes ganze Bahnsegmente aufgebaut werden müssen.

Bei dem Evolutionären Algorithmus werden viele Gene von den Eltern an die Kinder weitergereicht. Das Erstellen von Bahnsegmenten für diese Gene wird daher möglicherweise sehr häufig durchgeführt. Dies kann vermieden werden, indem vorher berechnete Bahnsegmente in einem Cache zwischengespeichert werden. Der Cache speichert die aus je zwei Genen generierten Bahnsegmente, so dass diese mit einem einfachen Look-Up wiederverwendet werden können. Der Cache ist dabei so aufgebaut, dass die Suche nach vorhandenen Elementen schnell geht, indem es je eine Liste für jeden möglichen Wert des ersten Gens gibt. Das Hinzufügen und Löschen von Segmenten ist durch die Nutzung von Listen ebenfalls schnell. Dies ist besonders deswegen wichtig, da eine Synchronisierung der Zugriffe aus verschiedenen Threads bei der Parallelisierung des Evolutionären Algorithmus notwendig ist, siehe Abschnitt 4.9.10.

#### 4.9.10 Parallelisierung und Generationenkonzept

Wie in Abschnitt 4.9.9 bereits beschrieben, sind Evolutionäre Algorithmen auf Grund vieler Zielfunktionsauswertungen relativ langsam. Eine weitere Möglichkeit, die Geschwindigkeit zu erhöhen, ist neben dem Caching die bessere Auslastung moderner Multicore Prozessoren. In den meisten Fällen ist die Auswertung der Zielfunktion mittels komplexer Simulationen der aufwändigste Teil des Algorithmus. Diese Auswertungen können jedoch völlig unabhängig von anderen Auswertungen durchgeführt werden. Daher bietet es sich an, die Auswertungen parallel auf allen zur Verfügung stehenden Prozessorkernen auszuführen, wie es in dieser Arbeit gemacht wird. Nur die Operationen auf der gemeinsamen Population müssen synchronisiert werden.

Da das so vorgestellte Konzept pro Thread bzw. Kern unterschiedliche Laufzeiten aufweist, wird zusätzlich das Konzept der Generationen aufgegeben, um auf diese Weise weitere Synchronisierungen und leerlaufende Prozessorkerne zu vermeiden. Das sogenannte Steady-State Prinzip [112] nutzt die neu erzeugten Kinder sofort und fügt sie mittels der Selektion direkt nach ihrer Erschaffung und Auswertung in die Population ein, sofern sie eine ausreichende Fitness haben.

Auch andere Möglichkeiten der Parallelisierung werden in der Literatur untersucht. So können auch verschiedene Populationen gleichzeitig entwickelt werden, deren Individuen in seltenem Austausch miteinander stehen. Übersichten über verschiedene Möglichkeiten sind zum Beispiel in [30, 3] zu finden.

#### 4.9.11 Parameterkontrolle

Die Operatoren, die beim Evolutionären Algorithmus genutzt werden, haben unterschiedlich viele Parameter, über die bestimmte Eigenschaften der Operatoren gesteuert werden. Zum Beispiel hat der häufig verwendete Tournament Operator einen Parameter, der bestimmt, wie viele Individuen an dem Turnier teilnehmen. Je nach Operator haben die Parameter erheblichen Einfluss auf die Konvergenz und das Verhalten der Optimierung.

Aus diesem Grund müssen die Parameter gut gewählt werden, so dass der Algorithmus in minimaler Zeit die bestmöglichen Ergebnisse liefern kann.

Es gibt verschiedene Möglichkeiten, um geeignete Parameterwerte für den Algorithmus zu finden [147]. Der klassische Ansatz ist das *Parameter Tuning*. Beim Parameter Tuning werden die Parameter vor dem Ausführen des Algorithmus gesetzt. Um die bestmöglichen Parameter für die genutzten Operatoren und das zu lösende Problem zu finden, müssen aufwändige Versuche durchgeführt werden. Häufig werden auch Faustregeln verwendet, die sich bei anderen Problemen als günstig erwiesen haben. Es ist jedoch nicht garantiert, dass sich Parameterwerte auf andere Algorithmen und andere Probleme einfach übertragen lassen. Die Alternative zum Parameter Tuning ist die *Parameter control*. Bei dieser werden die Parameter, während der Algorithmus ausgeführt wird, laufend angepasst. In der deterministischen Variante werden die Parameter nach festen Regeln verändert. Ein Beispiel wäre die Reduktion der Mutationsweite im Verlauf des Algorithmus, um zu späteren Zeitpunkten der Optimierung den Suchraum weniger zu explorieren und die gefundenen Extrema gründlicher zu erforschen. Adaptive Varianten passen die Parameter anhand der gerade vorliegenden Situation an. Beim Verlust von Genvielfalt in der Population kann zum Beispiel die Mutationsrate erhöht werden, um den Suchraum wieder mehr zu erforschen und vorzeitige Konvergenz zu mindern. Selbstadaptive Varianten codieren die Parameter selber als Gene und lassen diese über die Operatoren im Verlauf des Algorithmus entwickeln. Wenn die Parameter zuerst angepasst werden, erzeugen gute Parameter gute Individuen, welche überleben und ihre Gene (Parameter) an die nächsten Generationen weitergeben. Auf diese Weise wird der Benutzer von der Last der Parameterwahl befreit, jedoch dauert die Koevolution der Parameter Zeit, so dass tendenziell längere Laufzeiten des Algorithmus erwartet werden können.

Die verschiedenen Alternativen zur Parameterwahl sind in der Literatur gut untersucht worden. Nannen et al. [127] und Dobslaw [52] untersuchen das Parameter Tuning, Eiben et al. [55] untersucht und klassifiziert die Parameter control und viele andere Autoren untersuchen speziell die Selbstadaption [9, 71, 123, 147].

Der vorliegende Algorithmus nutzt eine Mischung aus den verschiedenen Parameterkontrolltechniken. So werden die Parameter für die Rekombination und Selektion per Tuning von einem externen Optimierungsprogramm festgelegt. Die Mutationsparameter können auf die gleiche Weise per Tuning gesetzt werden. Hier steht jedoch zusätzlich die Möglichkeit der Selbstadaption zur Verfügung. Im Abschnitt 4.9.12 werden die verwendeten Techniken detailliert beschrieben und auf ihre Eigenschaften untersucht.

#### 4.9.12 Parameteroptimierung: Rekombination, Mutation und Selektion

Für die Rekombination, Mutation und Selektion werden die Parameter über ein Parameter Tuning bestimmt. Da die Bestimmung guter Parameter zeitintensiv ist, wird

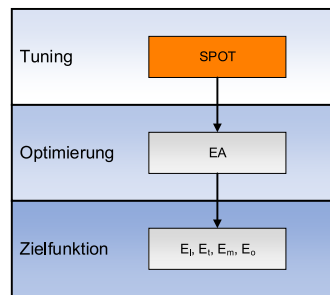


Abbildung 4.9: Grundprinzip der Parameteroptimierung mit SPOT nach [16].

hierzu ein Vorverarbeitungsschritt genutzt. Dadurch können einmal bestimmte Parameter auch für mehrere Optimierungsläufe wiederverwendet werden, so dass sich der Gesamtaufwand reduziert. Da feste Parameter auch Nachteile haben, wird im Bereich der Mutationswahrscheinlichkeiten auch die Selbstadaption untersucht.

Für die konkrete Realisierung des Parameter Tunings gibt es viele Konzepte. Im vorliegenden Fall ist die Auswertungszeit einer einzelnen Optimierung problematisch. Selbst mit reduzierten Auflösungen von Werkstück und Bahn, siehe Abschnitt 4.9.13, sowie wenigen Iterationen können Laufzeiten von einigen Minuten pro Optimierung entstehen. Bei einer Anzahl von 16 Parametern, die zum Teil kontinuierliche und zum Teil diskrete Wertebereiche aufweisen, müssen viele Optimierungen ausgeführt werden, um die optimalen Werte zu bestimmen.

Um mit einer möglichst geringen Anzahl an Optimierungen möglichst gute Ergebnisse zu erzielen, können statistische Methoden wie DACE (Design and Analysis of Computer Experiments, [93]) oder SPO (Sequential Parameter Optimization, [88]) verwendet werden. Die Idee dieser Verfahren ist, basierend auf Regressionsmodellen vielversprechende Parameterwerte zu finden, die das sogenannte *Expected Improvement* maximieren. Das Verfahren ist dadurch in der Lage, einerseits den noch unbekanntem Parameterraum so gut wie möglich zu erforschen und andererseits vielversprechende Parameterbereiche intensiver zu untersuchen. Durch die Auswertung von immer mehr Parametersätzen wird das Regressionsmodell sukzessive verfeinert und weitere Hypothesen für geeignete Parameter werden generiert.

Im Rahmen dieser Arbeit wird eine Toolbox namens SPOT genutzt [16, 13, 14, 175, 15, 17]. SPOT steht für *Sequential Parameter Optimization Toolbox* und implementiert die obige Idee, mittels Regressionsmodellen optimierte Parameter zu finden. Dabei ist die Toolbox explizit auf die Optimierung von Parametern von Algorithmen ausgelegt, die stochastischer Natur sind, also bei gleicher Eingabe auf Grund von randomisierten Komponenten unterschiedliche Ergebnisse liefern können. In Abbildung 4.9 ist das Konzept von SPOT grafisch dargestellt. Auf der untersten Ebene soll eine Zielfunktion optimiert werden. Hierzu wird ein Optimierungsalgorithmus verwendet. Bei diesem handelt es sich im vorliegenden Fall um den Evolutionären Algorithmus, der selbst von



verschiedenen Parametern abhängt. Diese Parameter werden von SPOT optimiert, so dass die eigentliche Optimierung der Zielfunktion möglichst schnell geht.

Im ersten Schritt baut SPOT ein initiales Design auf Basis eines Latin Hypercube Samplings auf. Für die entsprechenden Designpunkte im Parameterraum wird die Optimierung ausgeführt und die Ergebnisse ausgewertet, so dass eine erste Regressionsfunktion aufgebaut werden kann. Mit Hilfe dieser Funktion werden nun vielversprechende neue Punkte im Parameterraum ausgewählt, auf Basis derer die Optimierung ausgeführt und die Ergebnisse ausgewertet werden. Da es sich um einen stochastischen Prozess handelt, bei dem auf Grund des Verhaltens des Evolutionären Algorithmus verschiedene Ergebnisse bei gleichen Parametern entstehen, werden die Designpunkte mehrfach genutzt. Die genaue Anzahl wird innerhalb von SPOT bestimmt. Für die Modellbildung der Regression können verschiedene Modelle ausgewählt werden, jedoch wird das Random Forest Modell als relativ robustes Verfahren empfohlen [16], wenn kontinuierliche und diskrete Wertebereiche der Parameter vorliegen.

Für die Optimierung der Parameter mit SPOT werden die Zielfunktionskomponenten genau wie für die spätere Optimierung mit Hilfe der *Desirability Functions* transformiert, wobei die Funktion von Derringer/Suich genutzt wird. Die unteren Grenzen können auf 0 gesetzt werden, die oberen Grenzen werden durch Vorversuche so bestimmt, dass sie mit hoher Wahrscheinlichkeit nicht überschritten werden. Eine dynamische Anpassung der oberen Grenze ist nicht möglich, da sonst die Vergleichbarkeit der transformierten Werte zwischen verschiedenen Optimierungsläufen verloren geht. Die Potenz  $n$  in den Gleichungen (4.16) und (4.17) wird für die Gewichtung einzelner Komponenten verwendet. Hierbei wird die Temperatur mit einem Wert von 10 sehr hoch gewichtet und der Overspray mit einem Wert von 0,1 sehr gering. Die anderen Komponenten werden mit 1 gewichtet. Diese Gewichtung sorgt für einen starken Fokus auf die thermischen Varianzen, welche im weiteren Verlauf der Gesamtoptimierung (Bahnoptimierung [Kapitel 5] und Nutzung optimierter Bahnen [Kapitel 6]) nicht weiter verbessert werden können. Zur Optimierung wird das Werkstück A gewählt, das in Abschnitt 4.9.13 beschrieben wird und in Abbildung 4.13 dargestellt ist. Hierbei wird das grob aufgelöste Netz gewählt, um schnelle Optimierungen zu gewährleisten. Dies stellt kein Problem dar, da die relative Vergleichbarkeit der Ergebnisse gewährleistet ist.

Das Ergebnis von SPOT ist in erster Linie ein Parametersatz, der für die besten Ergebnisse des Optimierungsalgorithmus sorgt. In Tabelle 4.3 sind die optimierten Parameterwerte für zwei verschiedene Läufe zusammengestellt. Im ersten Lauf sind alle 16 Parameter der Rekombinations-, Mutations- Selektionsoperatoren gleichzeitig optimiert worden. Im zweiten Lauf sind die Mutationsoperatoren unberücksichtigt geblieben, da sie durch die Selbstadaptation während der Optimierung automatisch bestimmt werden. In beiden Fällen wurde die Anzahl der Evaluationen für SPOT auf 1000 und die Iterationen für den Evolutionären Algorithmus auf 20000 festgesetzt. Bei den Tournament Parametern handelt es sich um den prozentualen Anteil der Population, der an dem Turnier teilnimmt. Bei den Mutationswerten handelt es sich um die Mutationswahrscheinlichkeiten pro Operator pro Gen. Die erreichten Zielfunktionswerte, die durch die Nutzung der besten

Tabelle 4.3: Werte der Parameter des Evolutionären Algorithmus nach deren Optimierung durch SPOT mit und ohne Selbstadaption (SA).

Parameter	Wert ohne SA	Wert mit SA
Zielfunktionswert	0,233059	0,223784
Crossover Methode	Uniform	Two Point
Crossover Wahrscheinlichkeit	0,2330588	0,028968
Überlebensselektionsmethode	VAN	Tschebyscheff
Überlebensselektion: Tournament Parameter	0,2501812	0,030157
Elternselektionsmethode	Desirability	Desirability
Elternselektion: Tournament Parameter	0,9244317	0,641782
Auswahlselektionsmethode	Desirability	Gewichtete Summe
Shift Mutation	0,003847	SA
Add Mutation	0,002340	SA
Delete Mutation	0,000764	SA
Reverse Mutation	0,001258	SA
Random Mutation	0,000371	SA
Displacement Mutation	0,000256	SA
Exchange Mutation	0,001267	SA
Scramble Mutation	0,000713	SA
Simple Inversion Mutation	0,000707	SA

Tabelle 4.4: Indexzuordnung der Methoden des Evolutionären Algorithmus in SPOT und Abbildung 4.10.

Index	Rekombinationsmethode	Überlebensselektionsmethode	Elternselektionsmethode	Auswahlselektionsmethode
1	One Point	Tschebyscheff	Tschebyscheff	Gewichtete Summe
2	Two Point	Survival of the fittest	Gewichtete Summe	Desirability
3	Uniform	Gewichtete Summe	Desirability	Tschebyscheff
4	–	Desirability	VAN	–
5	–	Crowding Distance	–	–
6	–	VAN	–	–

Parameter entstehen, gibt SPOT mit 0,233059 bzw. 0,223784 an. Hierbei handelt es sich um die erzielten *Desirability Indizes*, siehe Abschnitte 4.9.1 und 4.9.6. Ein Wert von 0 wäre hierbei ideal.

In Abbildung 4.10 sind Visualisierungen der von SPOT zur Verfügung gestellten *Reporting Functions* für den Lauf mit Nutzung der Selbstadaption gegeben. Hierbei werden jeweils zwei Parameter gegenübergestellt, während alle weiteren Parameter auf ihren *besten* Wert gesetzt werden, siehe auch [17]. Per Farbcodierung werden die erreichten

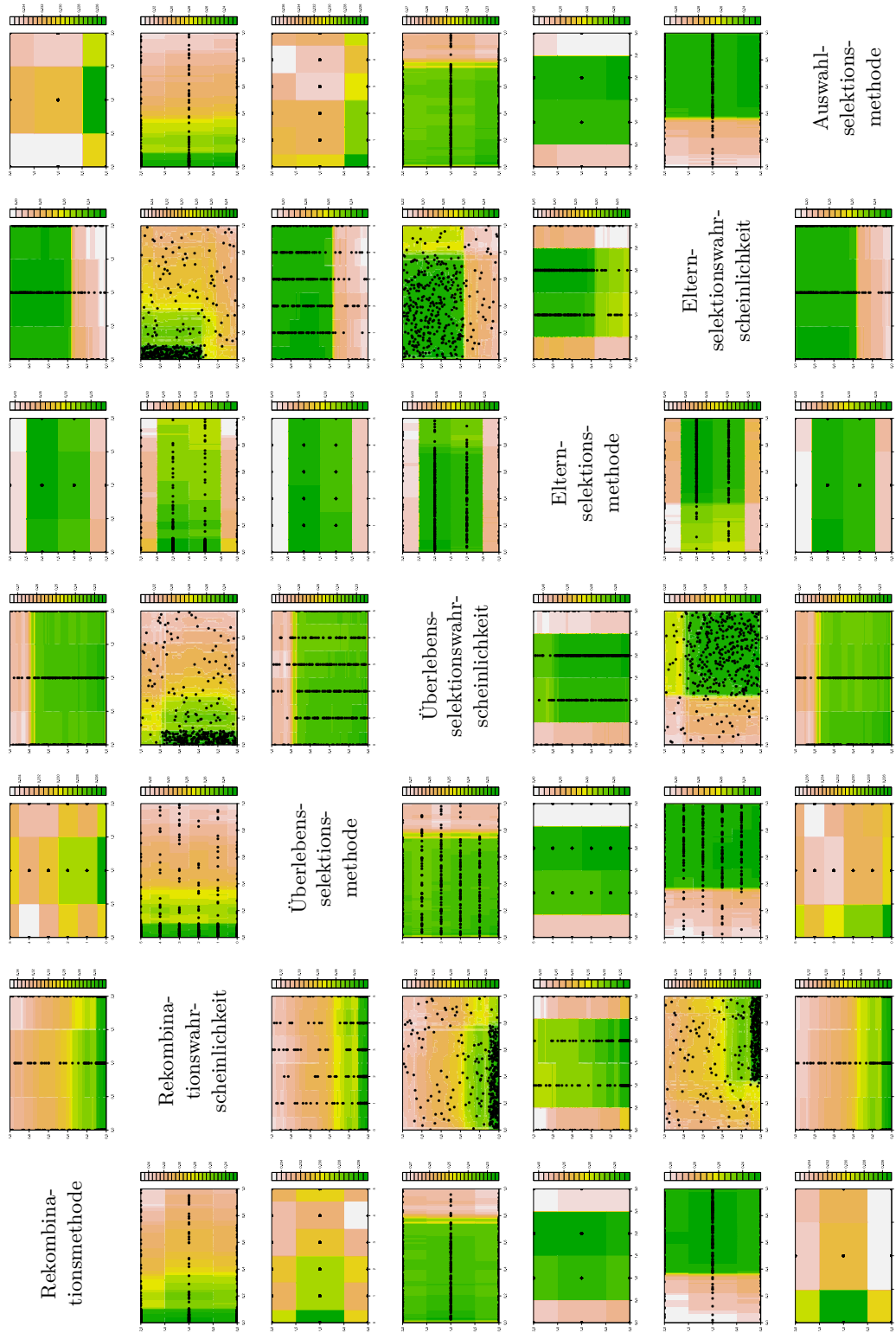


Abbildung 4.10: Ergebnisse der Parameteroptimierung mit SPOT bei Nutzung der Selbstadaptation. Bezeichner siehe Tabelle 4.4

Tabelle 4.5: Auswirkung verschiedener Parametersätze auf den Evolutionären Algorithmus. „Opt“: Optimierte Parameter. „Opt + SA“: Optimierte Parameter mit Nutzung der Selbstadaption. „Zufällig“: Zufällige Parameter. Alle Werte sind durch die *Desirability Function Derringer/Suich* transformiert, siehe Gleichung (4.18). Kleine Werte sind besser.

	Testfall	Desirability	Roboter	Schicht	Temperatur	Overspray
Mittelwert	Opt	0,2247	0,0669	0,0337	0,5713	0,0648
	Opt + SA	0,1998	0,0571	0,0315	0,5229	0,0586
	Zufällig	0,2678	0,0666	0,0283	0,6689	0,0360
Varianz	Opt	1,73E-04	6,69E-05	4,05E-05	3,17E-04	1,17E-04
	Opt + SA	6,55E-05	2,55E-06	3,16E-05	3,49E-04	4,70E-05
	Zufällig	5,17E-04	4,36E-06	1,65E-05	1,78E-03	7,35E-05

Zielfunktionswerte angegeben. Die schwarzen Sterne zeigen die von der Optimierung gewählten Punkte im Parameterraum an. Man kann gut erkennen, welche Parameterbereiche gute und welche weniger gute Ergebnisse hervorgebracht haben. So ist zum Beispiel die Nutzung der Tschebyscheff Selektion für die Eltern durchgehend schlechter als die Nutzung der *Desirability Selektion*, siehe dazu die fünfte Zeile/Spalte in Abbildung 4.10. Bei anderen Parametern kann man gut die Abhängigkeiten voneinander ablesen. So ist die Auswahlselektionsmethode nahezu unabhängig von der Elternselektionsmethode, jedoch stark abhängig von der Überlebensselektionsmethode (vgl. Abbildung 4.10 Zeile 7, Spalte 5 sowie Zeile 7, Spalte 3).

Ein recht auffälliges Merkmal der optimierten Parameter ist der sehr große Selektionsdruck, der durch die hohen Tournament Parameter ausgedrückt wird. Die hohen Werte sorgen für die Nutzung guter Individuen sowohl für die Vermehrung als auch für das Überleben. Dadurch ist eine schnellere Konvergenz möglich, jedoch erhöht sich die Gefahr der vorzeitigen Konvergenz in ein lokales Optimum.

Um die von SPOT ermittelten besten Zielfunktionswerte besser einordnen zu können, werden die entsprechenden Parametersätze für längere Optimierungen genutzt. Während die Optimierung mit dem Evolutionären Algorithmus in der Parameteroptimierung mit SPOT auf 20000 Iterationen begrenzt ist, sollen Optimierungen mit 100000 Iterationen dazu dienen, die Güte der optimierten Parameterwerte zu überprüfen. In Tabelle 4.5 sind die Ergebnisse für Testreihen zusammengestellt, in denen die optimierten Parameterwerte, die optimierten Parameterwerte mit Nutzung der Selbstadaption und beliebige unoptimierte Parameter genutzt wurden. Dabei sind jeweils drei unabhängige Optimierungen durchgeführt worden. Die Mittelwerte der Zielfunktionswerte sowie deren Varianz sind in der Tabelle zu finden. Da es sich wieder um Werte nach Umwandlung durch die *Desirability Functions* handelt, sind niedrige Werte besser. Wie man an den Werten erkennen kann ist die Nutzung der Selbstadaption leicht besser als die Nutzung fester Mutationswahrscheinlichkeiten. Deutlicher zeigt sich jedoch, dass die Optimie-

Die Wahl der Parameterwerte für erheblich bessere Ergebnisse sorgt als beliebige Werte. Dementsprechend ist es wichtig, dass eine Art der Parameteroptimierung durchgeführt wird.

#### 4.9.13 Analyse der Auflösungen

Ein wesentlicher Teil der Laufzeit bei der Optimierung wird für die Simulation des Auftrags und der thermischen Situation benötigt. Die Simulationszeiten sind in erster Linie von der Menge der Knoten des Werkstücknetzes sowie von der Anzahl der Bahnpunkte abhängig. Es ist daher erstrebenswert, möglichst gering aufgelöste Werkstücknetze und Bahnen mit wenigen Bahnpunkten zu nutzen. Die Kehrseite ist jedoch ein möglicher Qualitätsverlust der generierten Bahnen, da bei Nutzung zu grob aufgelöster Netze und Bahnen zu wenige Informationen für die Planung von guten Bahnen zur Verfügung stehen. Es muss daher eine Analyse der Auswirkungen bei der Reduktion der Auflösungen der Netze und Bahnen durchgeführt werden, um eine Grenzauflösung so zu bestimmen, dass die Optimierung möglichst performant läuft, ohne Fehler durch Unterabtastung einzuführen. In den folgenden Unterabschnitten werden daher zunächst entsprechende Auflösungsgrenzen bestimmt und daraufhin die Werkstücke in entsprechenden Auflösungsstufen vorgestellt.

##### Netz- und Bahnauflösung

In diesem Abschnitt wird der Einfluss der Netz- und der Bahnauflösung auf die Qualität der Simulation und der Bahnplanung untersucht. Das Ziel ist die Bestimmung einer Grenze für die Auflösungen, so dass die Ergebnisse der Simulationen unter besonderer Berücksichtigung der Nutzung der Simulationsergebnisse zur Bahnplanung nicht zu Qualitätsverlusten führen.

Zunächst soll die Netzauflösung untersucht werden. Das Simulationsmodell aus Abschnitt 3.2.2 ist so ausgelegt, dass die Schichtdicke an den Knoten des Netzes unabhängig von der Auflösung und Vernetzung exakt bestimmt werden. Da auch die Bahnplanung ausschließlich auf die Werte an den Knoten zurückgreift, entstehen keine Probleme durch Interpolation in das Innere der Dreiecke des Netzes. Problematisch ist die Netzauflösung dennoch, da Stellen des Netzes mit zu großem Abstand der Knoten zu wenig Informationen für eine geeignete Planung bereitstellen. So kann eine zu grobe Auflösung des Netzes im Extremfall dazu führen, dass ganze Bahnsegmente über die Oberfläche des Netzes verlaufen, ohne dass sie einen Einfluss auf die Beschichtung der vorhandenen Knoten haben. Solche Situationen müssen vermieden werden.

Der Einfluss eines Bahnpunkts auf einen Netzknoten ist vornehmlich von der relativen Lage der beiden Punkte zueinander, der Knotennormalen und der Spritzrichtung abhängig. Der kritischste Fall entsteht demnach, wenn die zentrale Spritzrichtung genau zwischen die Knoten des Netzes gerichtet ist, so dass der Punkt der TCP-Werkzeugbahn

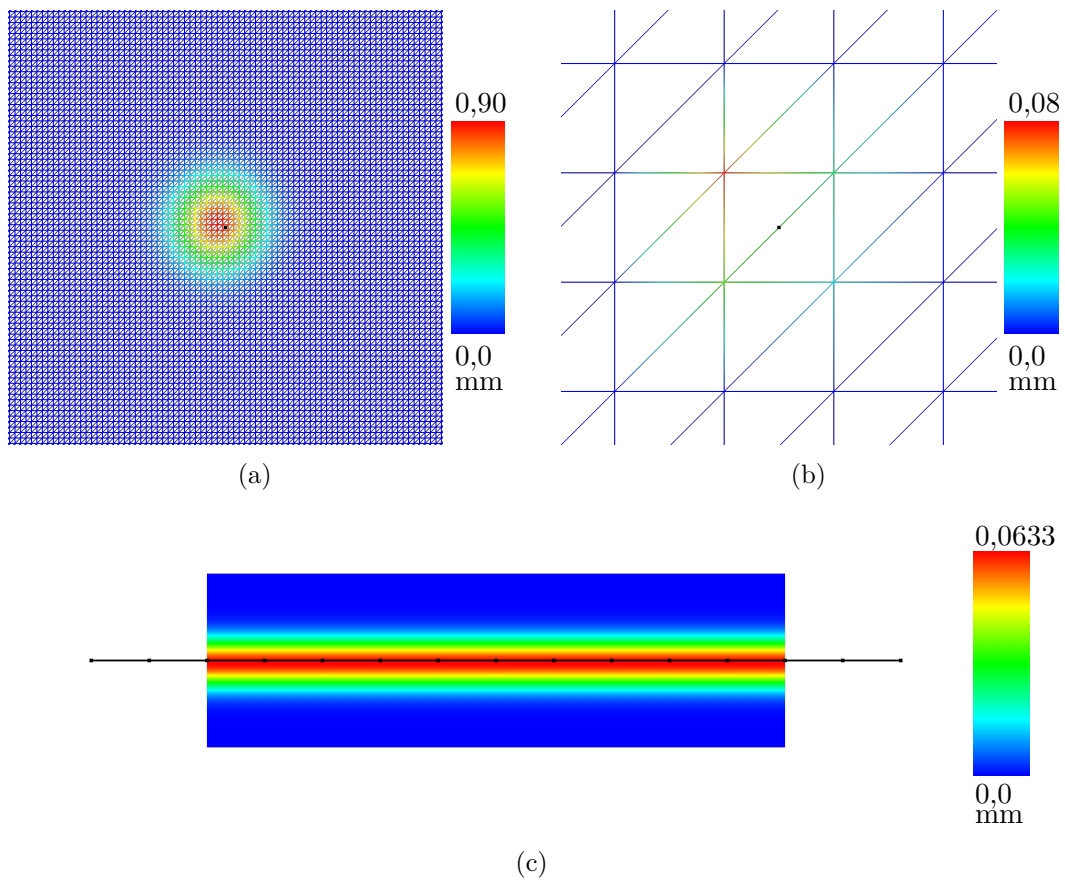


Abbildung 4.11: (a) Netz einer Flachprobe mit 1 mm Abstand zwischen den Netzknoten und (b) Netz derselben Flachprobe mit 20 mm Abstand zwischen den Netzknoten. (c) Bahn mit 20 mm Abstand zwischen den Bahnpunkten über einem Werkstück mit der gewünschten Zielbeschichtung. Durch die Farbcodierung wird die Schichtdicke dargestellt.

auf der Oberfläche eine maximale Entfernung von den Knoten des Netzes hat. Dieses Beispiel ist in den Abbildungen 4.11a und 4.11b dargestellt. Diese zeigen die Netze zweier Flachproben als Drahtgittermodell, wobei die erste Probe durch ein sehr fein aufgelöstes und die zweite durch ein sehr grob aufgelöstes Netz repräsentiert wird. Die Nutzung von Flachproben in den Experimenten ist günstig, da die Knoten aus Sicht der Spritzpistole in diesem Fall am weitesten voneinander entfernt liegen. Die Knoten auf gekrümmten Oberflächen mit gleichem geodätischen Abstand wie auf einer ebenen Oberfläche liegen in der Projektion auf die Footprint-Ebene dichter zusammen und sind daher weniger kritisch. Dies gilt sowohl für konvexe als auch für konkave Krümmungen. In beiden in den Abbildungen 4.11a und 4.11b dargestellten Fällen liegt der in Schwarz dargestellte Bahnpunkt der TCP-Werkzeugbahn genau auf der Oberfläche der Werkstücknetze. Der Bahnpunkt ist so positioniert, dass er exakt zwischen den Knoten des Netzes liegt, so dass er den maximal möglichen Abstand zu den Knoten aufweist. Die Farbcodierung

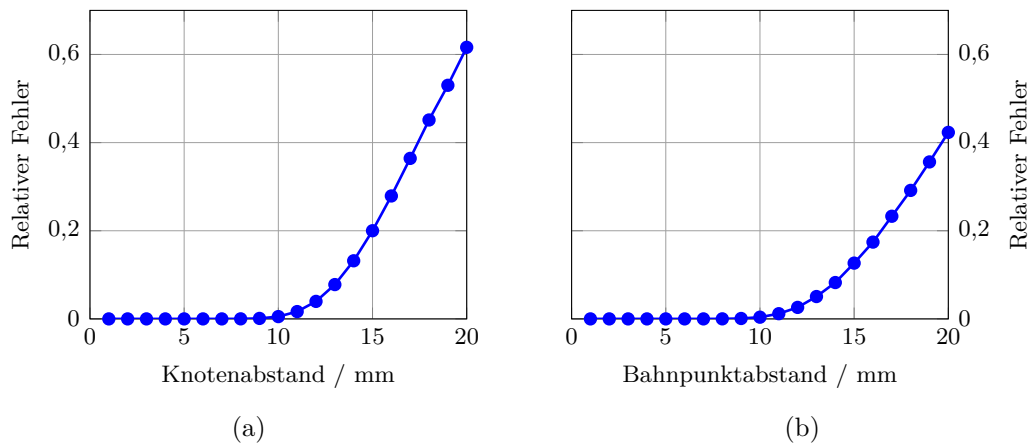


Abbildung 4.12: Abhängigkeit der Simulationsgüte von (a) der Netzauflösung und (b) der Bahnauflösung. Gezeigt ist der relative bzw. der maximale relative Beschichtungsfehler, der durch die zu grobe Auflösung erzeugt wird.

der Netze gibt die Beschichtung nach der Simulation des Bahnpunkts wieder, wobei die Skalen in beiden Abbildungen deutlich voneinander abweichen. Die Simulation nutzt in beiden Fällen das gleiche, über eine Gaußfunktion approximierte Modell eines realen Libo-Footprints, der leicht zur Symmetrieachse verschoben ist.

Obwohl die Beschichtung nur an den Knoten des Netzes bestimmt wird, sind in den Abbildungen die Interpolationen der Schichtdicken auf den Netzkanten zur besseren Vergleichbarkeit farblich dargestellt. In dem zweiten Bild ist deutlich zu erkennen, dass das meiste Material, das auf der Oberfläche zentral bezüglich der Spritzrichtung aufgetragen wird, auf Grund der Netzauflösung nicht berücksichtigt wird. Dies ist ein Problem, das aus der relativen Größe des Footprints und der Auflösung des Netzes resultiert. Um eine Aussage über das Ausmaß dieses Problems zu treffen, wird das auf dem Werkstück abgelagerte Materialvolumen betrachtet, das über die Schichtdicken an den Knoten und mittels der Flächen, die den Knoten zugeordnet sind, siehe Abschnitt 2.2, approximiert wird. Dieses müsste bei dem vorliegenden Experiment genau dem ausgegebenen Material aus der Spritzpistole entsprechen, da kein Overspray entsteht. Die Größe der Abweichung des abgelagerten Materialvolumens, das auf diese Weise als stückweise konstante Funktion bestimmt wird, von dem tatsächlich an der Spritzpistole ausgegebenen Materialvolumen, ist demnach ein Hinweis darauf, wie groß die Problematik durch die Nutzung zu grober Netzauflösungen ist. In der Abbildung 4.12a ist daher der relative Fehler zwischen dem aufgetragenen und dem ausgegebenen Materialvolumen  $V_s \tau_n$  für unterschiedliche Netzauflösungen angegeben, wobei das aufgetragene Materialvolumen über die Schichtdicke  $z_i$  an den Knoten multipliziert mit den jeweiligen Flächen  $A_i$  bestimmt wird. Die Formel für den Fehler ist dann

$$E = 1 - \frac{\sum_{i=0}^{m-1} z_i \cdot A_i}{V_s \tau_n}. \quad (4.27)$$



Man erkennt, dass der relative Fehler ab einer Netzauflösung von etwa 10 mm stark ansteigt. Dies ist ein Hinweis darauf, dass durch die Unterabtastung der Netzoberfläche die Ablagerung von Material an zu wenigen Stellen der Oberfläche bestimmt wird, so dass auch die Bahnplanung Probleme mit der Generierung guter Bahnen bekommen wird. Daher ist es besser, etwas höher aufgelöste Werkstücke zu nutzen, die solche Situationen vermeiden.

Da die Ergebnisse der Analyse über die Netzauflösung stark von der Größe des genutzten Footprints abhängen, muss eine entsprechende Analyse für Footprints anderer Größe neu ausgeführt werden.

Die zweite Analyse befasst sich mit der Auflösung der Bahn. Im Bereich der Bahnauflösung existiert prinzipiell das gleiche Problem wie bei der Netzauflösung, so dass bei zu grober Abtastung der Bahn mittels Bahnpunkten große Fehler in der Auftragssimulation auftreten. Daher muss auch in diesem Fall eine Grenze für die Auflösung der Bahn gefunden werden.

Für die Analyse der Bahnauflösung ist folgender Testfall durchgeführt worden. In Abbildung 4.11c ist ein hochaufgelöstes Werkstück und eine Linearbahn in geringer Auflösung dargestellt. Die strukturell gleiche Bahn wird in hochaufgelöster Variante simuliert, so dass der in der Abbildung farblich dargestellte Auftrag entsteht. Dieser wird als Zielbeschichtung angesehen. Nun wird die Bahn sukzessive vergrößert, indem sie mit einem immer größeren Abstand neu abgetastet wird. Die Bahngeschwindigkeit bleibt jedoch unverändert, so dass das gleiche Materialvolumen ausgegeben wird. Durch die geringere Abtastung der Bahnpunkte wird der Auftrag immer ungenauer und es entstehen bei relativ grober Abtastung anstelle einer gleichmäßigen Beschichtung einzelne Spritzflecken. Dies ist ein Fehler durch die Abtastung der Bahn und muss verhindert werden.

In Abbildung 4.12b ist der maximale relative Fehler in der Beschichtung bezüglich aller Netzknoten gezeigt. Man erkennt, dass die Auflösung ab einer Grenze von ca. 10 mm so grob wird, dass der Fehler rapide steigt. Daher sollte die Bahnauflösung so gewählt werden, dass sie den entsprechenden Grenzwert nicht überschreitet.

Insgesamt haben die beiden beschriebenen Testfälle gezeigt, dass sowohl die Netz- als auch die Bahnauflösung einen signifikanten Einfluss auf die Simulationsergebnisse haben können und als Konsequenz daraus auf die Bahnplanungsgüte. Da auf Grund der Rechenzeiten möglichst geringe Auflösungen erstrebenswert sind, kann über die gezeigten Testfälle eine ungefähre Grenze festgelegt werden, ab der die Auflösung nicht weiter reduziert werden sollte, um zu große Fehler zu vermeiden. Mit dem vorliegenden Footprint sollte sowohl die Netz- als auch die Bahnauflösung nicht unter 10 mm gesetzt werden.



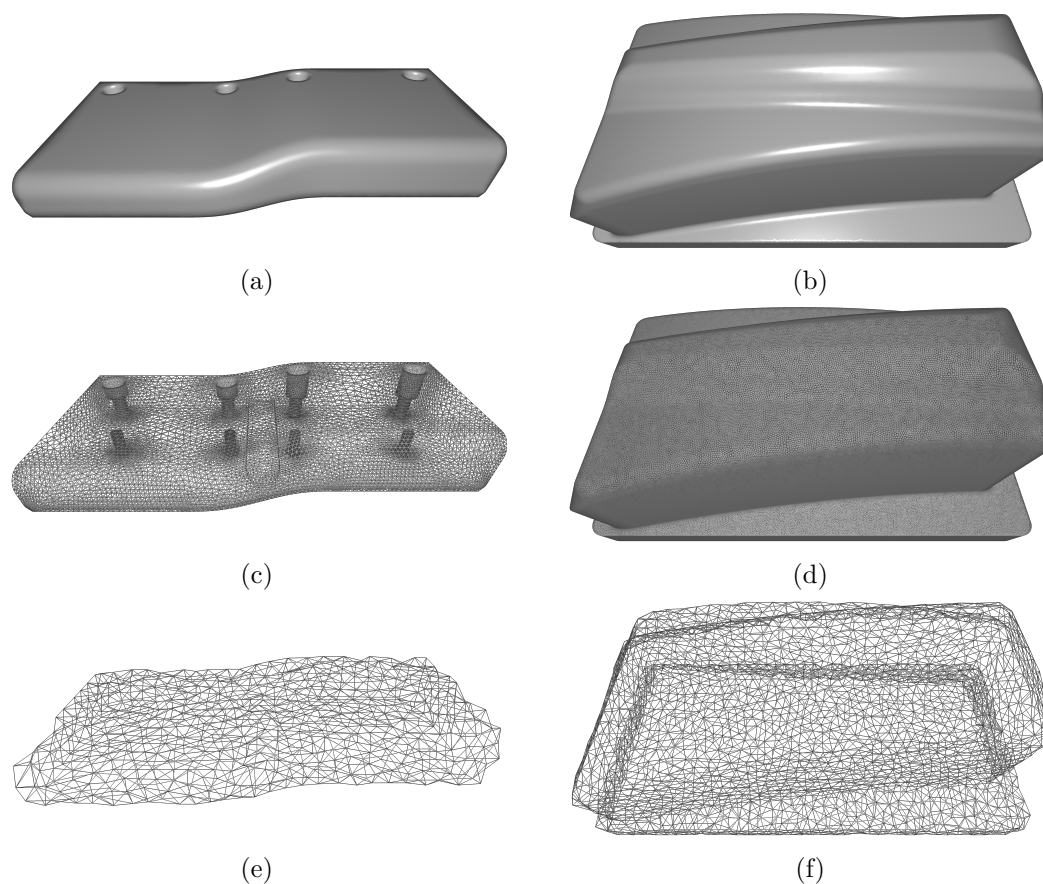


Abbildung 4.13: (a) Werkstück A. (b) Werkstück B. (c) Netz von Werkstück A in originaler Auflösung. (d) Netz von Werkstück B in originaler Auflösung. (e) Netz von Werkstück A in reduzierter Auflösung. (f) Netz von Werkstück B in reduzierter Auflösung.

### Testwerkstücke und Auflösungshierarchien

Als direkte Konsequenz aus den obigen Analysen zur Netz- und Bahnauflösung werden die für die Analysen und Evaluationen genutzten Werkstücke in entsprechenden Auflösungsstufen verwendet. In Abbildung 4.13 sind die beiden Testwerkstücke dargestellt. In beiden Fällen handelt es sich um Teile von real genutzten Tiefziehwerkzeugen. Das Werkstück A ist ein Ausschnitt einer segmentierten Matrize für das Tiefziehen von Demonstratorbauteilen und besteht im Wesentlichen aus einer leicht wellenförmigen Oberfläche und einem Radius von 10 mm an der Vorderseite. Die funktionalen Flächenanteile sind oben sowie vorne, da an diesen Stellen die Umformung des Blechs durchgeführt wird. Die Beschichtung kann demnach auf diese Stellen beschränkt werden. Bei den Strukturen an der Unterseite und am hinteren Ende des Werkstücks handelt es sich um Löcher zur Befestigung des Werkstücks in der Tiefziehmaschine. Das zweite Werkstück ist ein Stempel für Tiefziehprozesse, das ein Modell einer B-Säule im Automobilbereich

darstellt. Die funktionale Oberfläche ist in erster Linie der obere Teil des Werkstücks sowie die Seitenflächen, da an diesen Stellen das zu ziehende Blech Kontakt mit dem Werkstück hat.

In der untersten Zeile der Abbildung 4.13 sind die entsprechenden niedrig aufgelösten Werkstücke dargestellt. Dabei wurden die Netze so reduziert, dass keine Neuabtastung der Oberfläche stattgefunden hat, sondern nur bereits in den originalen Netzen vorhandene Knoten genutzt werden.

Ein Problem bei den genutzten Werkstücken ist das Nichtvorhandensein von Knotennormalen in den Originaldaten. Daher werden die Normalen auf den hochaufgelösten Werkstücken über die Methode aus Abschnitt 2.2 bestimmt. Da bei der Reduktion der Auflösung über das externe Tool *Meshlab*<sup>1</sup> ein komplett neues Meshing vorgenommen wird und nicht garantiert werden kann, dass die Dreiecke an allen Stellen der Oberfläche gleichmäßig geformt sind, würden sich die Normalen bei einer Neuberechnung auf dem reduzierten Netz teilweise deutlich von den Originalnormalen unterscheiden. Da die Beschichtungssimulation jedoch stark von der Normalenrichtung abhängig ist, werden die Normalen nicht neu berechnet, sondern direkt aus den Originalnetzen übernommen. Dies ist leicht möglich, da ausschließlich originale Knoten verwendet werden. Durch dieses Verfahren ist die Beschichtung auf den Knoten des grob aufgelösten Netzes gleich der auf dem originalen Netz und unabhängig von der neuen Vernetzung.

Bei den Werkstücken handelt es sich um real verwendete Bauteile, die mit einer Verschleißschutzbeschichtung versehen werden sollen. Bei dem genutzten Libo-Prozess ist der Footprint jedoch sehr groß, da der Spritzkegel einen großen Öffnungswinkel hat. Bei Werkstück A ist der kleinste Radius 10 mm. In der idealen Spritzdistanz von 110 mm fällt der Auftrag des approximierten Gaußmodells nach etwas mehr als 12 mm auf 10 % ab. Dementsprechend ist der Radius bezüglich der Footprintgröße bereits relativ klein. Verglichen mit Werkstück A sind die Radien von Werkstück B jedoch noch einmal wesentlich kleiner. Der kleinste Radius beträgt hier 4,53 mm. Der Libo-Prozess ist auf Grund des großen Footprints also eigentlich ungeeignet. Dennoch soll das Werkstück verwendet werden, da die Radien und Kanten besondere Herausforderungen darstellen und demnach Hinweise auf die Güte des Verfahrens geben.

#### 4.9.14 Analyse der Algorithmeigenschaften

In diesem Abschnitt werden einige Eigenschaften des Algorithmus untersucht, um herauszufinden, welche Komponenten des Evolutionären Algorithmus den größten Einfluss auf die Leistungsfähigkeit des Systems haben. Dazu zählt die benötigte Anzahl der Iterationen, um gute Ergebnisse hinsichtlich der auf die Initialbahnplanung folgenden Bahnoptimierung zu erlangen. Außerdem wird die Stabilität der Initialbahnplanung mit Hilfe einer Analyse der Streuung der Ergebnisse untersucht. Es folgt eine Betrachtung

<sup>1</sup><http://meshlab.sourceforge.net/> (07.02.2017)

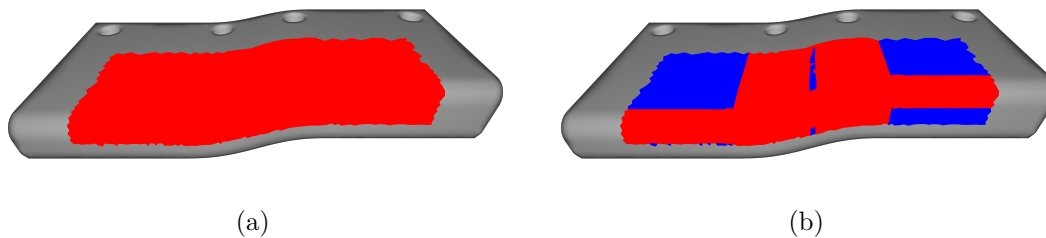


Abbildung 4.14: (a) Werkstück A mit Region of Interest in Rot. (b) Werkstück A mit Region of Interest in Blau/Rot, wobei die Stellen des Werkstücks auf dem Region of Interest mit einer Krümmung  $\geq 0,01$  in Rot dargestellt sind.

der Eigenschaften der Population von Individuen, die während des Evolutionären Algorithmus erzeugt werden. Außerdem werden die Einflüsse der Zielschichtdicke und der Bahngeschwindigkeit auf das Verfahren analysiert.

### Testdaten und Qualitätsmerkmale

Die Evaluationen in diesem Abschnitt werden auf Werkstück A ausgeführt. Hierbei wird das Region of Interest benutzt, das in Abbildung 4.14a dargestellt ist. Dieses ist so gewählt, dass die interessanten Stellen des Werkstücks von dem ROI abgedeckt werden und ein Randbereich bleibt, auf dem das EROI definiert werden kann.

Als Zielfunktionen werden die in Abschnitt 4.6 vorgestellten Maße genutzt. Hierbei handelt es sich um die Zielfunktionskomponenten für die Beschichtungsqualität ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ), für die thermischen Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), die Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$ ) sowie den Overspray ( $E_o(\mathcal{X})$ ).

Die kritischen Stellen des Werkstücks bezüglich des Schichtauftrags sind die Krümmungen an der Vorderseite und auf der Oberfläche. Die Bereiche des Werkstücks mit einer maximalen Hauptkrümmung (siehe [122]) von 0,01 oder größer sind in Abbildung 4.14b hervorgehoben. Die geforderte Beschichtung auf den ebenen Flächen zu erzeugen ist weniger schwierig als an den gekrümmten Stellen. Dies hat unter anderem die Auswirkung, dass der gemittelte Fehler  $E_l(\mathcal{X})$  bei Werkstücken mit großen ebenen Flächen besonders klein werden kann, obwohl die Stellen an den Krümmungen nur schlecht beschichtet werden. Es ist daher naheliegend, die gekrümmten Stellen gesondert zu betrachten oder höher zu gewichten. In dem vorliegenden Fall ist das Verhältnis von gekrümmten zu ebenen Flächenanteilen jedoch ausgeglichen, so dass die Fehler an den gekrümmten Stellen entsprechend berücksichtigt werden. Dennoch werden die Beschichtungsfehler  $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$  für einige der Evaluationen auch für den Fall angegeben, dass diese ausschließlich auf den gekrümmten Stellen des Werkstücks ausgewertet werden. Bezeichnet werden diese Fehler mit  $E_l^k(\mathcal{X})$  und  $E_{ml}^k(\mathcal{X})$ .

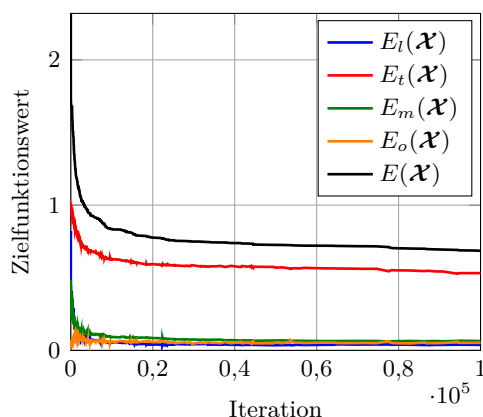


Abbildung 4.15: Verlauf der Zielfunktion sowie der Zielfunktionskomponenten bei der Initialbahnplanung einer Bahn auf Werkstück A. Gezeigt sind der Beschichtungsfehler ( $E_l(\mathcal{X})$ ), die Temperaturvarianzen ( $E_t(\mathcal{X})$ ), die Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ), der Overspray ( $E_o(\mathcal{X})$ ) sowie der summierte Fehler ( $E(\mathcal{X})$ ).

### Anzahl der Iterationen

Die Laufzeit des Evolutionären Algorithmus hängt direkt von der Anzahl der Iterationen ab, die dem Algorithmus zur Verfügung stehen. Je weniger Iterationen benötigt werden, desto schneller ist das Verfahren. Daher ist es erstrebenswert, die Initialbahnplanung möglichst früh abubrechen, sobald die generierten Lösungsbahnen dem möglichen Optimum nahe genug gekommen sind. Die Definition von *nahe genug* muss hier hinsichtlich der Weiteroptimierung durch die Bahnoptimierung verstanden werden. Es ist demnach nicht zwingend notwendig, in der Initialbahnplanung das Optimum zu erreichen, wenn die Lösungen gut genug sind, so dass sie von der nachfolgenden Bahnoptimierung (Kapitel 5) zu Lösungen mit der angestrebten Gesamtqualität weiterentwickelt werden können.

In der Gesamtoptimierung sind die Hauptzielkriterien die Minimierung des Beschichtungsfehlers, die Minimierung der thermischen Varianzen, die Optimierung der Roboterachsbeschleunigungen sowie die Minimierung des Oversprays. Da die thermischen Varianzen nur in der Initialbahnplanung optimiert werden können, weil sie auf der Umstrukturierung der Bahnsegmente basieren, muss dieser Aspekt in der Initialbahnplanung besonders stark gewichtet werden. Die Optimierung der Beschichtung kann zu großen Teilen von der anschließenden Bahnoptimierung durchgeführt werden. In den folgenden Analysen wird daher gezeigt, dass die Abbruchbedingung für die Evolutionäre Optimierung in erster Linie von den thermischen Varianzen abhängig gemacht werden kann.

Im ersten Testfall ist eine Initialbahnplanung auf dem niedrig aufgelösten Werkstück A, siehe Abbildung 4.13, für 100000 Iterationen durchgeführt worden. Auf einem PC mit einem Intel Core i7 4790K Prozessor mit 4 GHz und 4 physikalischen sowie 8 logischen Kernen, dauert die Initialbahnplanung mit 6 Threads für den Evolutionären Algorith-

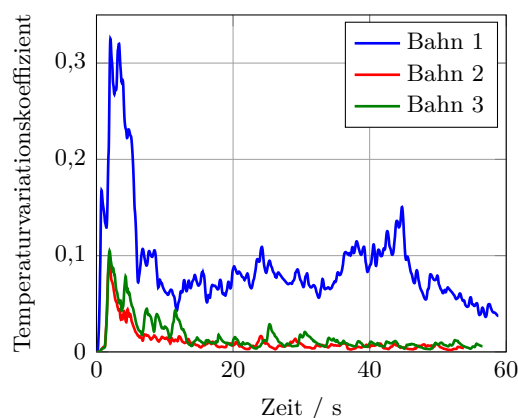


Abbildung 4.16: Thermische Varianzen einer durch die Initialisierung der Initialbahnplanung generierten Bahn („Bahn 1“), einer initialgeplanten Bahn („Bahn 2“) und einer initialgeplanten und anschließend durch die Bahnoptimierung optimierten Bahn („Bahn 3“) auf Werkstück A.

mus etwas weniger als 15 min. In Abbildung 4.15 sind die Zielfunktionswerte nach Transformation durch die *Desirability Functions*, siehe Gleichung (4.17), im Verlauf der Initialbahnplanung dargestellt. Es wird jeweils pro Iteration der Wert des aktuell *besten* Individuums in der Population aufgetragen. Das beste Individuum wird über die Auswahlsektion bestimmt, welche hier durch die gewichtete Summe gegeben ist, siehe Tabelle 4.3. Auf Grund der Gewichtungen ist der Wert für die thermischen Varianzen am größten, während die anderen Komponenten etwa gleich große Werte erzeugen. Die Werte fast aller Komponenten und insbesondere der thermischen Varianzen fallen anfangs stark ab, ändern sich jedoch gegen Ende der Initialbahnplanung nicht mehr besonders stark. Bei alleiniger Betrachtung der thermischen Varianzen könnte ein Abbruch nach ca. 20000 Iterationen sinnvoll erscheinen, da sich diese ab diesem Zeitpunkt nicht mehr so stark verbessern. Auf der anderen Seite finden leichte Verbesserungen bis zum willkürlichen Ende der Initialbahnplanung nach 100000 Iterationen statt, so dass nicht davon ausgegangen werden kann, dass das Optimum bereits gefunden wurde. Insbesondere fällt auf, dass es immer wieder Stellen gibt, an denen sich die thermischen Varianzen (und andere Komponenten) schlagartig verbessern. Dies liegt in der randomisierten Natur des Verfahrens begründet. Wenn per Mutation gute, neue Gene in ein Chromosom eingefügt werden, führt dies zu einer schlagartigen Verbesserung des Individuums und damit zu anschließenden Verbesserungen in dessen Kindern durch Weitergabe der Gene (mittels Rekombination oder weiterer Mutation). Es kann nicht abgesehen werden, ob auch nach längerer Stagnation des Verfahrens eine weitere Verbesserung stattfinden wird. Daher ist es nicht möglich, den idealen Abbruchzeitpunkt festzulegen.

Die thermischen Varianzen für eine komplette Bahn werden in Abbildung 4.15 als skalarer Wert ausgedrückt, der über die Gleichung (4.6) bestimmt wird. Da die thermischen Varianzen jedoch im Gegensatz zur Beschichtung während des gesamten Prozesses von

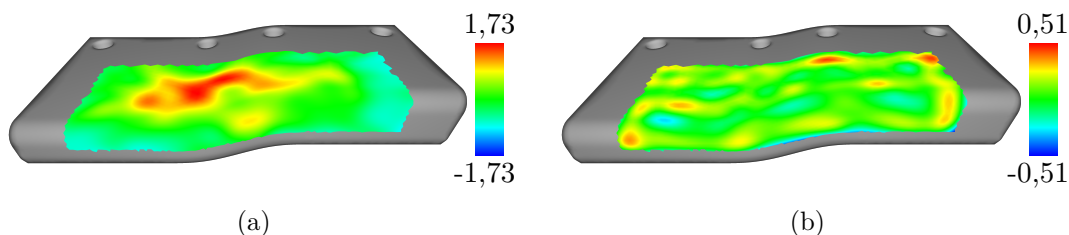


Abbildung 4.17: (a) Relativer Beschichtungsfehler einer durch die Initialisierung der Initialbahnplanung generierten Bahn und (b) einer durch die Initialbahnplanung geplanten Bahn. Die Skalen sind an die maximalen Fehler angepasst.

Bedeutung sind, werden diese in Abbildung 4.16 für verschiedene Bahnen im Verlauf des Prozesses dargestellt. Die erste Bahn ist durch die Initialisierung des Evolutionären Algorithmus mittels Latin Hypercube Sampling erzeugt worden und enthält keinerlei weitere Optimierungen. Diese Bahn entspricht dem ersten Individuum, das im Verlauf der Evolutionären Optimierung in Abbildung 4.15 zu sehen ist. Die zweite Bahn ist durch die Optimierung per Initialbahnplanung mit 100000 Iterationen erzeugt worden und entspricht dem besten Individuum der Optimierung, die in Abbildung 4.15 dargestellt ist. Die dritte Bahn entspricht ebenfalls der optimierten Bahn, welche jedoch durch die Bahnoptimierung aus Kapitel 5 weiter verbessert wurde. Des Weiteren wurden alle Bahnen vor deren Simulation zur Bestimmung der thermischen Varianzen neu abgetastet, um alle 0,02s einen Datenpunkt zu erhalten. Die durch die Bahnoptimierung verbesserte Bahn wurde vor der Bahnoptimierung, wie in Abschnitt 6.4 beschrieben, zusätzlich durch B-Splines geglättet. Die Darstellung zeigt die Varianzen, die auf der Werkstückoberfläche im Verlauf des eigentlichen Spritzprozesses entstehen. Es ist gut zu erkennen, dass die thermischen Varianzen der initial erzeugten Bahn durch die Initialbahnplanung deutlich verbessert werden, während die Bahnoptimierung die Varianzen kaum negativ beeinflusst. Die starken Varianzen zu Beginn des Spritzprozesses können nicht völlig vermieden werden, da zunächst auf das kalte Werkstück gespritzt werden muss und sich somit eine Stelle des initial kalten Werkstücks sehr stark erhitzt. Eine Verminderung dieses Effektes kann durch das Vorheizen des Werkstücks geschehen, wie es in manchen praktischen Anwendungen auch getan wird.

Da neben den thermischen Varianzen auch die Beschichtungsergebnisse von Interesse sind, sind in Abbildung 4.17 die Beschichtungsergebnisse der ersten, von der Initialisierung der Initialbahnplanung generierten, und der besten Bahn nach 100000 Iterationen dargestellt. Die Skalen sind jeweils an die maximalen Fehler angepasst. Es ist gut zu erkennen, wie stark sich die Fehler durch die Initialbahnplanung verbessert haben. Der maximale relative Fehler ist von 1,730828 auf 0,505608 gefallen. Der Fehlerwert  $E_l(\boldsymbol{x})$  hat sich entsprechend von 0,390516 auf 0,021372 verbessert und der Wert für  $E_{ml}(\boldsymbol{x})$  von 2,995770 auf 0,255640. Ebenso haben sich die Werte auf den Krümmungen für  $E_l^k(\boldsymbol{x})$  von 0,172908 auf 0,017720 reduziert und für  $E_{ml}^k(\boldsymbol{x})$  von 0,865925 auf 0,135638. Somit



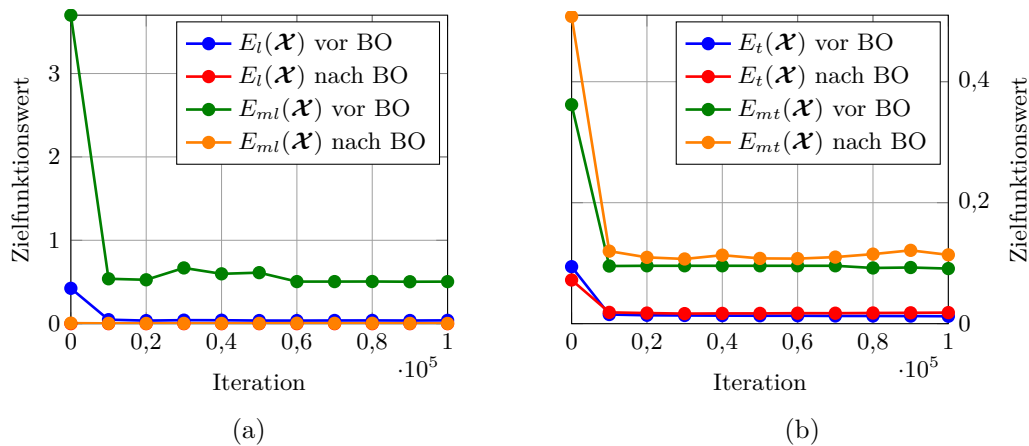


Abbildung 4.18: (a) Mittlerer und maximaler Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ) sowie (b) mittlere und maximale thermische Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ) der Zwischenbahnen, die bei einer Initialbahnplanung erzeugt werden, vor und nach der Optimierung mit der Bahnoptimierung („BO“).

liegen die Fehlerwerte auf den Krümmungen allein sogar leicht unter denen der gesamten Fläche.

Um die Möglichkeit eines frühzeitigen Abbruchs der Initialbahnplanung weiter zu untersuchen, sind in Abbildung 4.18 die Beschichtungsfehler und die thermischen Varianzen für die Bahnen aufgetragen, die als Zwischenergebnisse in der oben beschriebenen Initialbahnplanung erzeugt und anschließend durch die Bahnoptimierung optimiert wurden. Alle 10000 der insgesamt 100000 Schritte dauernden Initialbahnplanung wird die bis dahin beste Bahn gespeichert. Diese Bahnen werden, nach Neuabtastung und Glättung, von der Bahnoptimierung optimiert. Da die Güte der Bahn nach dem Optimieren mit der Bahnoptimierung und nicht nach der Initialbahnplanung von Bedeutung ist, sind diese Werte für den Gesamtprozess besonders aussagekräftig. In den Abbildungen ist deutlich zu erkennen, dass der Beschichtungsfehler beinahe unabhängig von der Ausgangsbahn durch die Bahnoptimierung sehr stark reduziert werden kann. Da die Bahnoptimierung die thermischen Varianzen nicht berücksichtigt, lassen sich diese in diesem Schritt nicht mehr systematisch minimieren. In Abbildung 4.18b ist allerdings gut zu erkennen, dass die bereits optimierten thermischen Varianzen von der Bahnoptimierung kaum verschlechtert werden, während die Bahn hinsichtlich des Beschichtungsfehlers deutlich optimiert wird. Dies entspricht der angestrebten Methodik der Trennung der Zielfunktionskomponenten, so dass die thermischen Varianzen in der Initialbahnplanung und der Großteil des Beschichtungsfehlers in der Bahnoptimierung optimiert werden. Für das Abbruchkriterium der Initialbahnplanung lässt sich hierdurch der Schluss ziehen, dass die Betrachtung der thermischen Varianzen ausreichend ist. Sobald diese nicht mehr wesentlich verbessert werden können, kann die Initialbahnplanung abgebrochen werden.

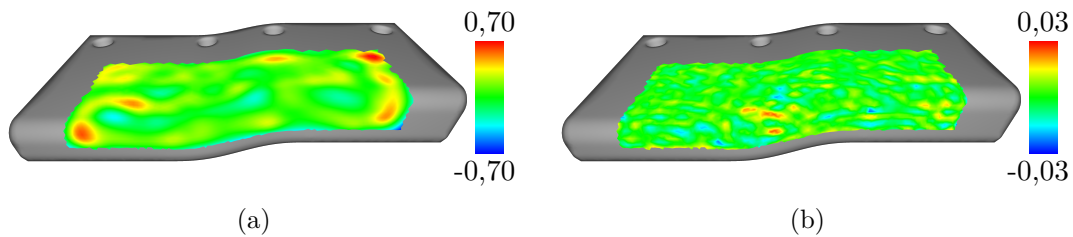


Abbildung 4.19: (a) Relativer Beschichtungsfehler bestimmt über die externe Simulation für die durch die Initialbahnplanung mit 100000 Iterationen generierte Bahn. (b) Relativer Beschichtungsfehler bestimmt über die externe Simulation für die durch die Initialbahnplanung mit 100000 Iterationen generierte und anschließend von der Bahnoptimierung optimierten Bahn. Die Skalen sind an die maximalen Fehler angepasst.

Auch wenn sich bezüglich des Abbruchkriteriums die thermischen Varianzen als ausschlaggebend herausgestellt haben, so ist die Optimierung des Schichtfehlers während der Initialbahnplanung wichtig. Durch diese wird sichergestellt, dass alle Stellen der Oberfläche einigermaßen gleichmäßig beschichtet werden und dass die Zielschicht überall grob eingehalten wird. Würde diese Komponente weggelassen und würden ausschließlich die thermischen Varianzen optimiert, würde schnell die triviale Lösung, nämlich die leere Bahn, generiert, die ideale thermische Varianzen erzeugt.

Ein Vergleich der Beschichtungsfehler vor und nach der Bahnoptimierung der Bahn nach 100000 Iterationen der Initialbahnplanung ist in Abbildung 4.19 dargestellt. Man erkennt die deutliche Verbesserung der Beschichtung, da der Fehler auf unter 3% fällt. Der Wert für  $E_l$  fällt von 0,039513 auf 0,000120 und der Wert für  $E_{ml}$  von 0,504230 auf 0,002981. Auch bei Betrachtung der Krümmungen allein ist die deutliche Verbesserung abzulesen. Hier fallen die Werte für  $E_l^k$  von 0,033136 auf 0,000197 und für  $E_{ml}^k$  von 0,367061 auf 0,002258.

### Streuung der Ergebnisse

Obwohl das übergeordnete Ziel des Gesamtverfahrens die Erzeugung einer Bahn mit sehr guten Eigenschaften nach der Bahnoptimierung ist, ist es dennoch wichtig, dass gute und auch reproduzierbare Ergebnisse durch die Initialbahnplanung erzielt werden. Daher soll in diesem Abschnitt analysiert werden, wie sich die Erzeugung und Optimierung durch die Initialbahnplanung unter gleichen Bedingungen verhält und ob reproduzierbare oder weit gestreute Ergebnisse auf Grund der randomisierten Natur des Verfahrens erzeugt werden.

Um zu ermitteln, wie sich der Algorithmus bei wiederholter Anwendung verhält, sind 100 Bahnen bei Nutzung gleicher Parameter, siehe Tabelle 4.3, erzeugt worden. Die Streuung der Ergebnisse der Initialbahnplanungen ist in Abbildung 4.20 zusammengefasst, wobei die gewichtete Summe der durch die *Desirability Functions* transformierten Zielfunktionswerte aufgetragen ist, so wie es durch die Auswahlselektionsmethode vorgesehen ist.



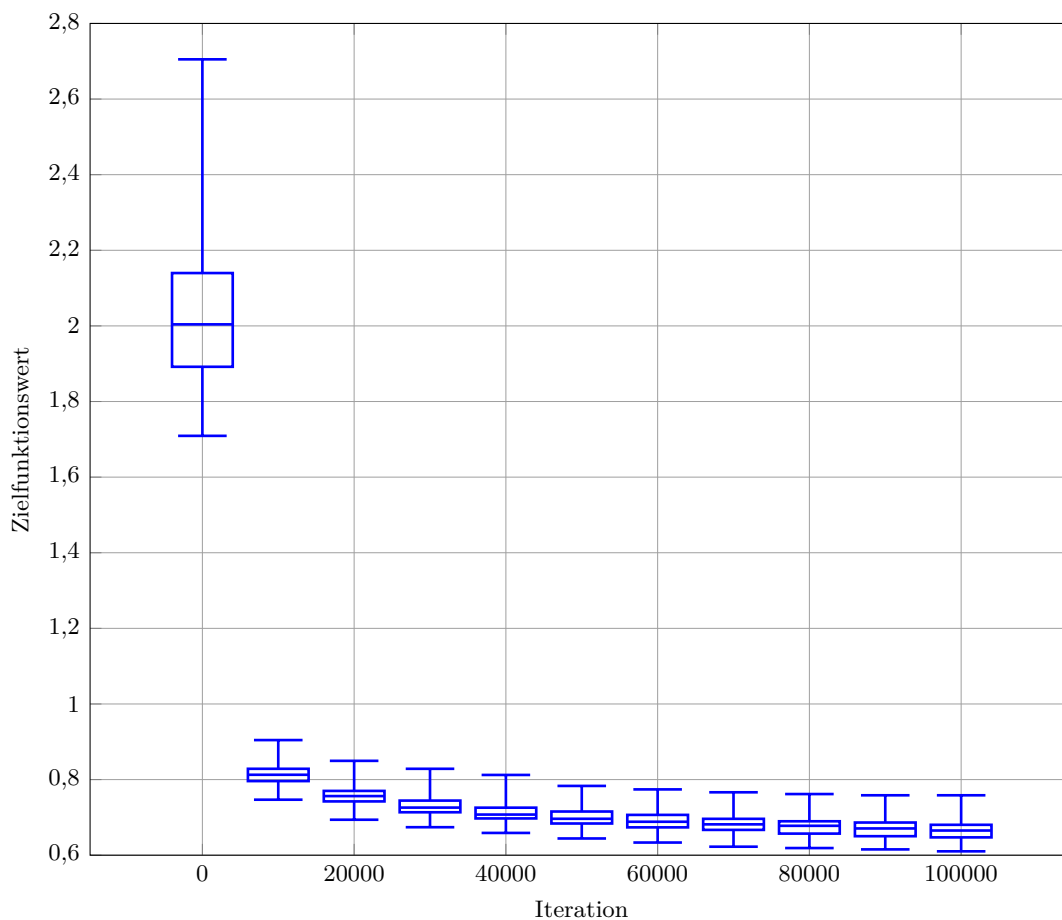


Abbildung 4.20: Streuung der Zielfunktionswerte von 100 Initialbahnplanungen mit gleicher Parametrisierung.

Man kann gut erkennen, dass sich die anfänglich hohe Streuung in den Fitnesswerten der Bahnen schnell stark verringert. Die geringe Streuung nach der Anfangsphase bedeutet, dass nach relativ kurzer Zeit ähnlich gute Lösungen in jedem Lauf gefunden werden.

Bei der Betrachtung der Lösungsbahnen fällt auf, dass die Bahnen alle voneinander verschieden sind. Da die Qualität der Lösungen jedoch vergleichbar gut ist, ist dies ein Hinweis auf die Struktur des Lösungsraumes. Offenbar gibt es viele weit entfernte Stellen im Lösungsraum, die eine ähnlich gute Qualität liefern. Eine solche Struktur ist für die Initialbahnplanung besonders schwierig, da lokale Änderungen an der Bahn nicht zum globalen Optimum führen. Vielmehr wird ein Individuum lange in einem lokalen Optimum verbleiben und selten durch eine Mutation zu anderen, eventuell nur leicht besseren Stellen in dem Lösungsraum springen. Das Finden des globalen Optimums, falls dieses eindeutig ist, ist demnach extrem schwierig.

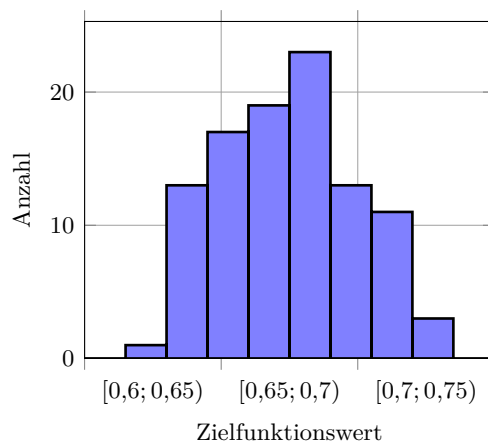


Abbildung 4.21: Histogramm der Zielfunktionswerte der besten Individuen von 100 Initialbahnplanungen mit gleicher Parametrisierung.

Obwohl die optimierten Bahnen alle unterschiedlich sind, lassen sich einige grundlegende Strukturen in den Bahnen immer wiederfinden. So lassen sich auf Werkstück A insbesondere viele lange Bahnsegmente entlang der langen Seite des Werkstücks finden, während die kürzeren, senkrecht hierzu stehenden Bahnsegmente tendenziell selten sind. Dies liegt unter anderem daran, dass die kurzen Bahnsegmente über den vorderen Radius verlaufen und auf Grund der Krümmung hier besonders starke Fehler in der Beschichtung sowie hohe thermische Varianzen erzeugen. Der Evolutionäre Algorithmus ist in der Lage, solche Strukturen erfolgreich zu reduzieren, um den Gesamtfehler zu minimieren.

In Abbildung 4.21 ist ein Histogramm der optimierten Zielfunktionswerte dargestellt. Das Histogramm zeigt die Streuung der gewichteten Summe der Zielfunktionskomponenten der besten Individuen der 100 Initialbahnplanungen nach jeweils 100000 Iterationen der Initialbahnplanung. Ein statistisches Merkmal ist die Schiefe der gegebenen Verteilung, die sich über die Formel

$$\hat{\nu}' = \frac{k}{(k-1)(k-2)} \sum_{i=1}^k \left( \frac{E_k - \bar{E}}{s} \right)^3 \quad (4.28)$$

berechnen lässt, wobei  $k$  die Anzahl der Experimente,  $E_k$  die entsprechenden Zielfunktionswerte,  $\bar{E}$  der Mittelwert der Zielfunktionswerte und  $s$  die Stichprobenstandardabweichung ist. Hier ergibt sich die Schiefe von ca. 0,1 und zeigt demnach eine rechtschiefe Verteilung an. Dies könnte ein Hinweis darauf sein, dass die Optimierung durch die Initialbahnplanung nahe an einem Optimum endet, da es nicht mehr möglich ist, bessere Werte zu finden. Jedoch ist die Schiefe so gering, dass es sich auch um ein zufälliges Ergebnis handeln kann.

### Analyse der Populationseigenschaften

Eine wichtige Metrik im Bereich Evolutionärer Algorithmen und generell von populationsbasierten Methoden ist die Genvielfalt in den Chromosomen der Individuen der Population. Die grundlegende Idee ist, dass eine vielfältige Population verschiedene Gene weiterentwickelt und über die Rekombination immer neue Individuen erzeugt, die mit Hilfe der Kombination dieser Gene neue, vielversprechende Bereiche des Lösungsraums untersuchen.

Aus diesem Grund besitzen viele bekannte Verfahren, wie zum Beispiel der NSGA-II, Methoden, die explizit dafür sorgen, dass die Vielfalt in der Population erhalten bleibt. Ein Operator, der in dieser Arbeit zum Erhalt bzw. zur Erhöhung der Vielfalt eingesetzt werden kann, ist der Crowding Distance Operator, der für die Überlebensselektion eingesetzt werden kann. In der Optimierung der Parameter in Abschnitt 4.9.12 hat sich dieser Operator jedoch nicht gegenüber den anderen Methoden durchsetzen können. Vielmehr sind Selektionsoperatoren ausgewählt worden, die einen sehr hohen Selektionsdruck erzeugen und weniger Vielfalt in der Population generieren. Der Vorteil solcher Operatoren ist die zum Teil deutlich schnellere Konvergenz auf Kosten der Vielfalt. Die Gefahr ist allerdings, dass sich die Individuen in der Population nach einigen Iterationen so weit ähneln, dass die Rekombination keinen Effekt mehr hat und neue Bereiche des Zielfunktionsraums nur noch durch Mutation erreicht werden können. Im Extremfall hat das Konzept einer Population keinerlei Bedeutung mehr, wenn alle Individuen in der Population Klone sind. Das Resultat ist häufig eine vorzeitige Konvergenz in lokale Optima, aus denen die Individuen der Population nicht oder nur schwer herauskommen.

Da durch die Parameteroptimierung Operatoren ausgewählt wurden, die wenig Vielfalt erzeugen und demnach die oben beschriebenen Nachteile mit sich bringen, soll die Entwicklung der Population hier genauer untersucht werden. In der Literatur wird als Maß für die Ähnlichkeit von Chromosomen häufig die Anzahl der benötigten (Mutations-) Operationen benutzt, um ein Individuum in ein anderes zu überführen. Da es sehr schwierig ist, dieses Maß bei der genutzten Menge an Operatoren exakt zu bestimmen, wird im Folgenden die Vielfalt der Individuen mit Hilfe der Levenshtein Distanz angegeben, die für den Vergleich von Strings entwickelt wurde und über

$$l_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{falls } \min(i,j) = 0, \\ \min \begin{cases} l_{a,b}(i-1,j) + 1 \\ l_{a,b}(i,j-1) + 1 \\ l_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{sonst,} \end{cases} \quad (4.29)$$

berechnet wird.  $a$  und  $b$  bezeichnen die Eingabestrings, welche in diesem Fall die Chromosomen der Individuen darstellen. Die einzelnen Gene werden über die Indizes  $i$  und  $j$  angesprochen.  $1_{(a_i \neq b_j)}$  ist genau dann gleich 1, wenn  $a_i \neq b_j$  ist. Die Distanz gibt demnach ein Maß für die Gleichheit von Strings bzw. in diesem Fall für die Chromosomen an. Es gilt jedoch zu beachten, dass Ähnlichkeiten in den Bahnen, die durch das komplexe

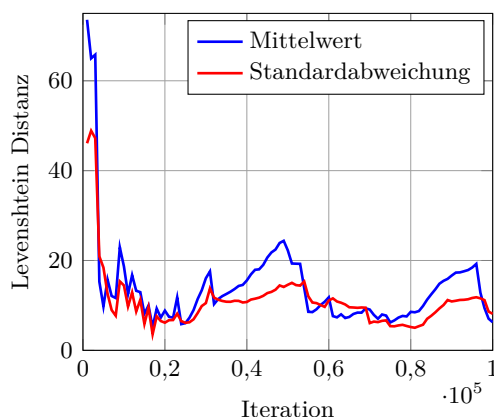


Abbildung 4.22: Populationsvielfalt im Verlauf einer Initialbahnplanung mit dem Evolutionären Algorithmus gemessen über die Levenshtein Distanz (siehe Gleichung (4.29)).

Genotyp/Phänotyp Mapping nicht in den Chromosomen zu erkennen sind, auch nicht durch dieses Maß abgebildet werden.

In Abbildung 4.22 sind der Mittelwert und die Standardabweichung der Levenshtein Distanz zwischen allen in der Population vorkommenden Individuen im Verlauf einer Initialbahnplanung dargestellt. Die Werte sind aus Rechenzeitgründen nur alle 1000 Iterationen bestimmt worden. Dies hat vor allem am Anfang der Initialbahnplanung Auswirkungen, da hier auf Grund der Initialisierung mit dem Latin Hypercube Verfahren zunächst recht hohe Distanzen vorkommen. Insgesamt ist jedoch ein schneller Abfall der Distanz zu beobachten, welche schließlich auf niedrigem Niveau verbleibt. Dennoch ist zu beobachten, dass die Distanz stets von Null verschieden ist und damit immer gewisse Unterschiede in den Chromosomen zu finden sind. Bei einer durchschnittlichen Chromosomlänge von ca. 134 Genen ist die Distanz zwischen je zwei Chromosomen, wie in der Abbildung 4.22 zu erkennen ist, meist im Bereich zwischen 5 und 20. Auch die Standardabweichung ist relativ groß, so dass es zwischen den Individuen immer einige Unterschiede gibt. Diese Unterschiede, wenn auch gering, sind wichtig für den Evolutionären Algorithmus und stellen sicher, dass das populationsbasierte Verfahren wie erwartet funktioniert.

### **Einfluss der Zielschichtdicke**

Im Folgenden wird untersucht, ob die Zielschichtdicke einen Einfluss auf die Qualität der von der Initialbahnplanung generierten Lösungen hat. Die Dicke der aufzutragenden Schicht wird von dem Anwendungsfall vorgegeben, für den die Beschichtung durchgeführt werden soll. Des Weiteren ist die mögliche Schichtdicke von dem Prozess abhängig, mit dem die Schicht erzeugt wird. So wird bei dem verwendeten Libo-Prozess relativ viel Masse pro Zeit von der Spritzpistole ausgegeben, so dass eher dickere Schichten erzeugt

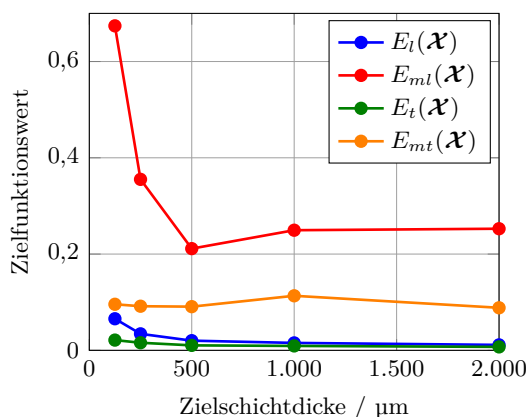


Abbildung 4.23: Einfluss der Zielschichtdicke auf die Initialbahnplanung. Dargestellt sind die mittleren und maximalen Beschichtungsfehler ( $E_t(\chi)$  und  $E_{mt}(\chi)$ ) sowie die mittleren und maximalen Temperaturvarianzen ( $E_t(\chi)$  und  $E_{mt}(\chi)$ ) für verschiedene Optimierungen mit unterschiedlichen Zielschichtdicken.

werden. Andere Prozesse (wie zum Beispiel der HVOF Prozess) geben wesentlich weniger Material aus.

Ein Ziel der Analyse ist es, herauszufinden, ob die Güte der Bahnen, die von dem Verfahren erzeugt werden, von der Zielschichtdicke abhängig sind und ob es eine minimale Schichtdicke gibt, die mit den gegebenen Bahnstrukturen noch sinnvoll erzeugt werden kann. Des Weiteren ist es bei einigen Prozessen möglich, eine kurze Bahn, die einen Bruchteil der geforderten Schicht erzeugt, einige Male zu wiederholen, bis die geforderte Schicht gebildet wird. Da die Simulationsdauer linear mit der Anzahl der Bahnpunkte skaliert und der Suchraum mit steigender Anzahl an Bahnsegmenten größer wird, ist die Optimierungsdauer und die Qualität der Optimierung bei der Initialbahnplanung von diesen Größen abhängig.

In Abbildung 4.23 sind die Beschichtungsfehler und die Temperaturvarianzen für das jeweils beste Individuum für verschiedene Initialbahnplanungen mit unterschiedlichen Zielschichtdicken aufgetragen. Da der Beschichtungsfehler relativ zur Zielschichtdicke angegeben wird (vgl. Gleichung (4.3)), sind die relativen Fehler bei geringer Zielschichtdicke bereits bei wenig absoluter Abweichung relativ groß. In Abbildung 4.23 ist gut zu erkennen, dass die Fehlerwerte für die Beschichtung für die Fälle mit geringer Zielschichtdicke deutlich größer sind als für höhere Zielschichtdicken, obwohl der Zielfunktionsraum für kurze Bahnen, die für wenig Schichtauftrag benötigt werden, deutlich kleiner ist und die Optimierung entsprechend einfacher sein sollte. Ein Grund für die schlechteren Werte ist, dass es schwierig ist, eine geeignete Bahnstruktur zu finden, wenn bereits einfache Kreuzungen von Bahnen auf der Oberfläche zu Überbeschichtungen führen. Es gibt entsprechend wenig Möglichkeiten, eine Bahn zu generieren, die in der Lage ist, den geforderten Schichtauftrag einzuhalten. Dies ist jedoch ein generelles Problem, das weder an die Bahnstruktur noch an die Optimierungsmethode gebunden ist, sondern immer

auftritt, wenn das Verhältnis von Materialabgabe pro Sekunde und Zielschichtdicke ungünstig ist. Daher ist es notwendig, eine minimale Zielschichtdicke abhängig von dem bei dem Prozess ausgegebenen Material pro Sekunde zu fordern.

Im Bereich der thermischen Varianzen gibt es ebenfalls Unterschiede, wenn die Zielschichtdicke verändert wird. Die Mittelwerte der thermischen Varianzen (siehe Gleichung (4.6)) zeigen ein ähnliches Verhalten wie bei der Beschichtung, so dass geringere Zielschichtdicken zu größeren Fehlern nach der Initialbahnplanung führen. Der Grund ist hier jedoch, dass die Mittelwerte auf Grund der kürzeren Bahnen schlechter werden, da es zu Beginn des Spritzprozesses zu den größten Varianzen kommt, siehe auch Abbildung 4.16. Ein langer Spritzprozess hat demnach immer den Vorteil einer langen Phase mit geringen Varianzen. Daher ist es besser, die Maximalwerte der thermischen Varianzen zu vergleichen. Die Ergebnisse in Abbildung 4.23 sind uneindeutig, da die maximalen Varianzen nach der Initialbahnplanung vergleichbar gut sind. Die Unterschiede können auch durch die normale Schwankung des Evolutionären Algorithmus auftreten. Es ist jedoch anzunehmen, dass die Optimierung der thermischen Varianzen für lange Bahnen schwieriger ist, da die Wahrscheinlichkeit, gute Mutationen und Rekombinationen für die kritischen Stellen zu Beginn einer Bahn zu finden, sinkt. Dies lässt sich jedoch aus den gewonnenen Daten nicht eindeutig herleiten.

### **Einfluss der Bahngeschwindigkeit**

Die Bahngeschwindigkeit hat ebenso wie die Zielschichtdicke Einfluss auf die Initialbahnplanung. Während der Initialbahnplanung ist die Bahngeschwindigkeit fixiert und kann erst später in der Bahnoptimierung geringfügig modifiziert werden. In diesem Abschnitt werden zwei Analysen durchgeführt. Erstens wird untersucht, was für ein Unterschied entsteht, wenn die Bahngeschwindigkeit für die TCP-Werkzeugbahn und wenn sie für die Werkzeugbahn konstant gehalten wird. Zweitens wird untersucht, welchen Einfluss die Änderung der Bahngeschwindigkeit auf die Gesamtoptimierung hat.

Die Bahngeschwindigkeit kann prinzipiell an verschiedenen Stellen gemessen werden. Bisher wurde die Bahngeschwindigkeit als TCP-Geschwindigkeit aufgefasst. Da die TCP-Werkzeugbahn auf Grund der Art und Weise, wie die Bahnstrukturen erzeugt werden, genau auf der Oberfläche des Werkstücks verläuft, handelt es sich demnach gleichzeitig um die Oberflächengeschwindigkeit. Es können jedoch neben der TCP-Werkzeugbahn auch die anderen in Abschnitt 2.1 beschriebenen Bahnen genutzt werden, um die Geschwindigkeit zu definieren. Eine TCP-Werkzeugbahn mit konstanter Geschwindigkeit sorgt für einen gleichmäßigen Auftrag und gleichmäßigen Wärmeeintrag pro Zeiteinheit in das Werkstück. Dafür sind die entsprechenden Geschwindigkeiten an der Werkzeugbahn sehr ungleichmäßig, was zu Problemen in der Roboterdynamik führen kann. Die Nutzung einer konstanten Geschwindigkeit auf der Werkzeugbahn sorgt für eine bessere Roboterdynamik, jedoch ist der Auftrag und der Wärmeeintrag an gekrümmten Stellen der Werkstückoberfläche kritisch.

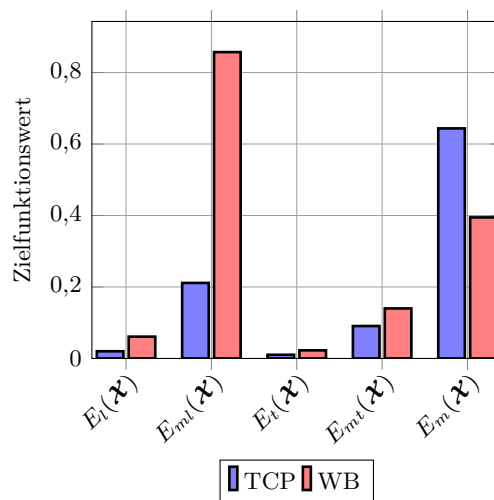


Abbildung 4.24: Einfluss der Definition der konstanten Bahngeschwindigkeit auf der TCP-Werkzeugbahn (TCP) bzw. der Werkzeugbahn (WB) bei der Initialbahnplanung. Gezeigt sind die mittleren und maximalen Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{mI}(\mathcal{X})$ ), die mittleren und maximalen Temperaturvarianzen ( $E_t(\mathcal{X})$  und  $E_{mT}(\mathcal{X})$ ) sowie die mittleren Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ). Die Werte für  $E_m(\mathcal{X})$  sind um den Faktor  $1 \cdot 10^{-5}$  skaliert.

In Abbildung 4.24 sind die Zielfunktionswerte für eine Bahnerstellung mit der konstanten Geschwindigkeit auf der TCP-Werkzeugbahn und auf der Werkzeugbahn gegenübergestellt. Es ist leicht zu erkennen, dass die Werte für die Beschichtung und die thermischen Varianzen für den Fall mit konstanter Bahngeschwindigkeit auf der TCP-Werkzeugbahn besser sind. Dies liegt an den Krümmungen des Werkstücks. An den gekrümmten Stellen muss die Spritzpistole stark umorientieren, so dass sie relativ lange auf die gleiche Stelle des Werkstücks spritzt. Dies führt zu einer zu hohen Schichtdicke und zu einer hohen thermischen Belastung an diesen Stellen. Im Bereich der Roboterachsbeschleunigungen ist zu sehen, dass diese im Falle der konstanten Geschwindigkeit auf der Werkzeugbahn besser sind als für den Fall mit konstanter TCP-Bahngeschwindigkeit. Für diesen Testfall mit einer Soll-Geschwindigkeit von 200 mm/s ist der Unterschied in den Achsbeschleunigungen bereits gut zu erkennen. Insgesamt zeigt die Analyse jedoch, dass die konstante Geschwindigkeit auf der Oberfläche des Werkstücks zu bevorzugen ist.

Die Bahngeschwindigkeit kann grundsätzlich auf einen beliebigen konstanten Wert gesetzt werden. Bei einer niedrigen Geschwindigkeit wird jedoch lokal sehr viel Wärme in das Werkstück eingetragen und auch der Auftrag ist sehr hoch. Daher ist es erstrebenswert, möglichst hohe Geschwindigkeiten zu erzielen. Auf der anderen Seite sind hohe Geschwindigkeiten nicht immer von dem Roboter abfahrbar, da die Roboterdynamik zu beachten ist.

In Abbildung 4.25 sind die Zielfunktionswerte für Initialbahnplanungen dargestellt, bei denen die Geschwindigkeit der TCP-Werkzeugbahn auf verschiedene konstante

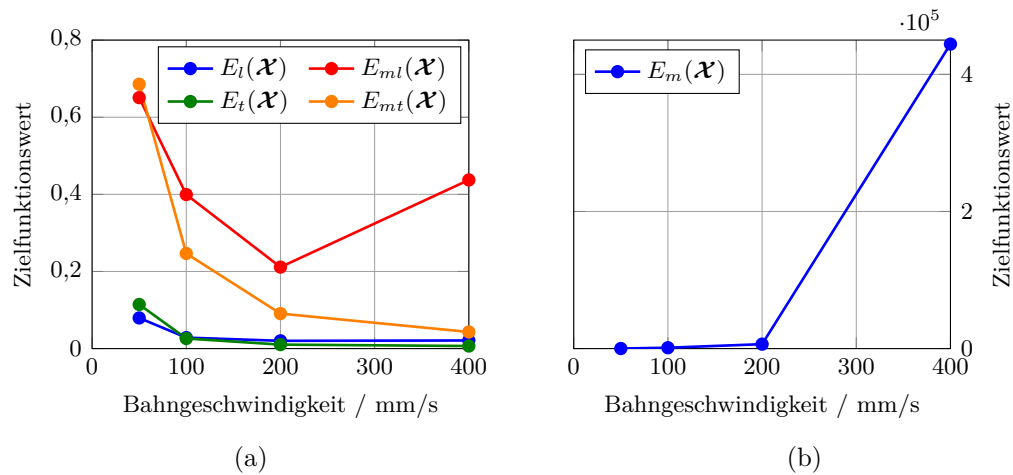


Abbildung 4.25: Einfluss der Bahngeschwindigkeit auf die Initialbahnplanung. Dargestellt sind (a) die mittleren und maximalen Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ), die mittleren und maximalen Temperaturvarianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ) sowie (b) die mittleren Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ) für verschiedene Initialbahnplanungen mit unterschiedlichen Bahngeschwindigkeiten.

Werte zwischen 50 und 400 mm/s gesetzt wurden. Wie gut zu erkennen ist, sind die verbleibenden mittleren Fehler in der Beschichtung und die thermischen Varianzen nach der Initialbahnplanung bei den Bahnen mit geringer Geschwindigkeit tendenziell schlechter als bei Bahnen mit hoher Geschwindigkeit. Dies lässt sich damit erklären, dass langsame Bahnen mehr Schicht und mehr Wärme lokal an eine Stelle der Oberfläche auf- bzw. eintragen. Je schneller die Bahn ist, desto weniger lokal ist der Auftrag bzw. der Eintrag, so dass die Zielfunktionswerte tendenziell geringer werden. Bei dem hohen maximalen Beschichtungsfehler bei der Bahn mit einer Geschwindigkeit von 400 mm/s handelt es sich um einen Ausreißer, der entsteht, da der maximale Wert nicht explizit optimiert wird. Ein Nachteil bei schnellen Bahnen sind die Roboterachsbeschleunigungen. Diese sind, wie in Abbildung 4.25b zu sehen ist, für die Bahn, die eine Geschwindigkeit von 400 mm/s hat, um ein Vielfaches größer als für die langsameren Bahnen. Es ist daher sehr unwahrscheinlich, dass die Bahn von einem realen Roboter gefahren werden kann. Daher ist es notwendig, niedrigere Geschwindigkeiten zu wählen. In Kapitel 6 wird zusätzlich ein Verfahren vorgestellt, das sicherstellt, dass der Roboter die Bahnen, die bereits mit einer vernünftigen Geschwindigkeit generiert wurden, fahren kann.

#### 4.9.15 Analyse der Laufzeit

In diesem Abschnitt wird der Laufzeitbedarf des Algorithmus untersucht. Hierzu werden die Laufzeiten der einzelnen Komponenten des Algorithmus betrachtet, so dass Aussagen darüber getroffen werden können, welche Komponenten besonders zeitaufwändig sind. Des Weiteren werden Möglichkeiten vorgestellt, wie die Laufzeit durch Nutzung von



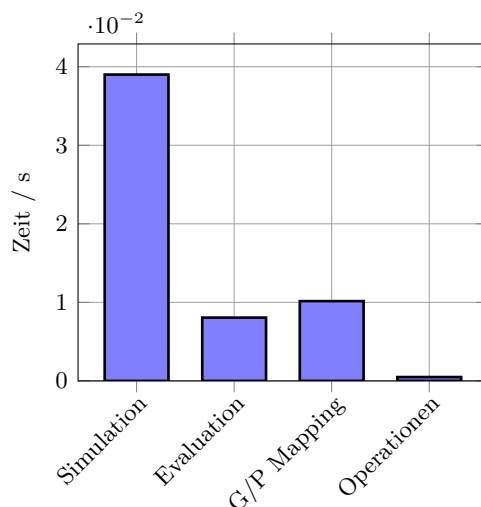


Abbildung 4.26: Zusammensetzung der Laufzeit der Initialbahnplanung. Gezeigt sind die durchschnittlichen Werte für eine Planung mit 100000 Iterationen für die wesentlichen Komponenten Simulation der Bahn, Bestimmung der Zielfunktionswerte, Genotyp/Phänotyp Mapping und Anwenden der genetischen Operationen. Zu beachten ist, dass beim Genotyp/Phänotyp Mapping das Caching aus Abschnitt 4.9.15 bereits genutzt wird.

weiteren Techniken wie Caching oder durch Nichtnutzung von Komponenten mit wenig Einfluss auf die Qualität der Ergebnisse reduziert werden kann.

### Zusammensetzung der Laufzeit

Um herauszufinden, welche Komponenten des Algorithmus die Laufzeit am meisten beeinflussen, werden die kritischen Komponenten im Einzelnen betrachtet. Hierzu werden in einem normalen Lauf des Evolutionären Algorithmus die benötigten Laufzeiten der Komponenten gemessen. Bei den betrachteten kritischen Komponenten handelt es sich um die Simulation der Bahn, der Bestimmung der Zielfunktionswerte, dem Genotyp/Phänotyp Mapping und dem Anwenden der genetischen Operationen.

In Abbildung 4.26 sind die durchschnittlichen Laufzeiten der kritischen Komponenten dargestellt. Man erkennt deutlich, dass die meiste Laufzeit von der Simulation der Bahn benötigt wird. Aber auch die anderen Komponenten haben Einfluss auf die Laufzeit. So ist die zweit aufwändigste Operation das Genotyp/Phänotyp Mapping, obwohl in dem Lauf bereits das Caching, siehe Abschnitt 4.9.9 genutzt wird. Die Bestimmung der Zielfunktionswerte benötigt Zeit, um aus den Rohdaten der Simulationen die entsprechenden skalaren Werte zu berechnen. Das Anwenden der genetischen Operationen hat kaum Auswirkungen auf die Laufzeit.

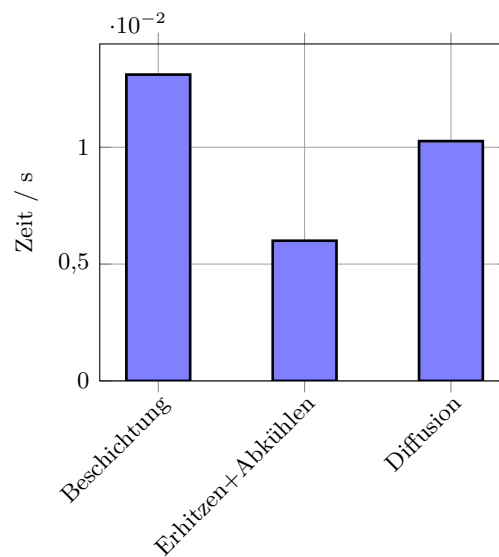


Abbildung 4.27: Zusammensetzung der Simulationslaufzeit. Gezeigt sind die summierten Laufzeiten der wesentlichen Komponenten Beschichtung, Erhitzen, Abkühlen und Diffusion bei der Simulation einer Bahn.

Da die Simulation selbst aus verschiedenen Komponenten besteht, soll der Laufzeitbedarf dieser Komponenten genauer untersucht werden. In Abbildung 4.27 ist die Zusammensetzung der Laufzeit für die Simulation einer einzelnen Bahn für die aufwändigsten Komponenten dargestellt. Bei diesen handelt es sich um die Beschichtungssimulation, die Berechnungen für den Wärmeeintrag und die Abkühlung sowie für die Diffusion. Des Weiteren wird noch Zeit für die Initialisierung, die Datenspeicherung und den Datentransfer benötigt, wobei diese Komponenten verglichen mit den angegebenen Komponenten recht wenig Zeit verbrauchen.

Die Abbildung 4.27 zeigt, dass der größte Teil der Laufzeit für die Beschichtungssimulation benötigt wird. Dies liegt vor allem daran, dass für jeden Bahnpunkt jeder Oberflächenknoten betrachtet werden muss, um den Auftrag zu bestimmen oder um festzustellen, dass keine Beschichtung stattfindet. Auf Grund der Tatsache, dass das geometrische Verhältnis von Spritzposition und Oberflächenknoten an dieser Stelle auf jeden Fall berechnet wird, werden zusätzlich die benötigten Daten für die thermische Simulation gespeichert, so dass auf eine wiederholte Berechnung verzichtet werden kann. Die dafür benötigte Zeit ist in der Zeit für die Beschichtung inbegriffen, fällt jedoch sehr gering aus. Den zweitgrößten Anteil an der Laufzeit hat die Diffusion. Diese ist aus zwei Gründen zeitintensiv. Zum Ersten wird für die Diffusion die Netzstruktur benötigt, so dass die Wärme über die Kanten von einem Knoten auf die Nachbarknoten übertragen werden kann. Hierzu müssen alle Kanten des Netzes betrachtet werden. Zum Zweiten darf die Diffusion nur jeweils geringe Energiemengen übertragen, um numerische Stabilität zu gewährleisten. Dies bedeutet, dass die Diffusion ungeachtet des Simulationszeitschritts nur sehr kleine Zeitschritte betrachten kann, wie es in Abschnitt 3.3.3 beschrieben

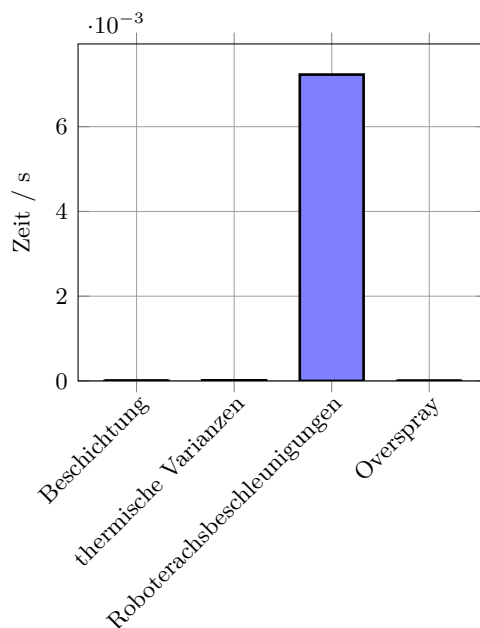


Abbildung 4.28: Zusammensetzung der Laufzeit zur Bestimmung der Zielfunktionswerte. Gezeigt sind die wesentlichen Komponenten Beschichtung, thermische Varianzen, Roboterachsbeschleunigungen und Overspray.

wird. Die Konsequenz daraus ist, dass die Diffusion möglicherweise sehr oft während eines Simulationszeitschritts berechnet wird und damit relativ viele Rechenoperationen benötigt werden. Die letzte Komponente ist der Wärmeeintrag bzw. der Wärmeverlust an der Oberfläche des Werkstücks. Da hier bereits einige Vorberechnungen aus der Beschichtungssimulation genutzt werden können, ist diese Komponente nicht besonders zeitaufwändig. Generell müssen nur pro Oberflächenknoten die entsprechenden Formeln aus Abschnitt 3.3.2 berechnet werden.

Auch die Bestimmung der Zielfunktionswerte besteht aus verschiedenen Komponenten. Hierzu zählen insbesondere die Bestimmung der Fehler in der Beschichtung, die Berechnung der thermischen Varianzen, der Roboterachsbeschleunigungen und des Oversprays. In Abbildung 4.28 sind die Laufzeiten für diese Komponenten dargestellt. Es ist leicht zu sehen, dass die Roboterachsbeschleunigungen um Größenordnungen langsamer ist als alle anderen Komponenten. Dies liegt daran, dass während der Simulation die inverse Kinematik nicht berechnet wird und für die Bestimmung der Achsbeschleunigungen diese Berechnungen nachgeholt werden müssen. Obwohl die inverse Kinematik relativ schnell zu berechnen ist, muss diese für viele Bahnpunkte bestimmt werden, so dass sie einen wesentlichen Anteil an der Laufzeit einnimmt. Alle anderen Berechnungen der Zielfunktionswerte sind dagegen sehr schnell.

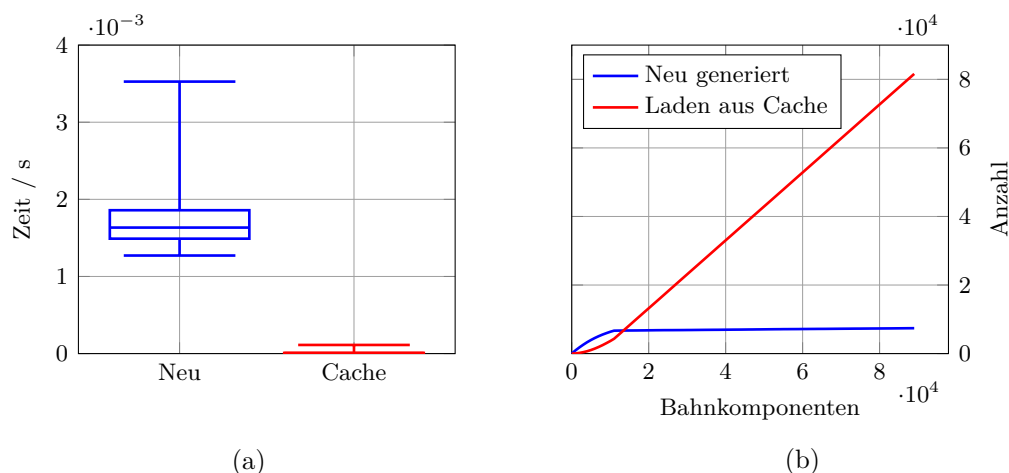


Abbildung 4.29: (a) Zeitbedarf beim Generieren eines Bahnsegments (Genotyp/Phänotyp Mapping) und beim Laden eines Segments aus dem Cache. (b) Anzahl der neu generierten Bahnsegmente und der Ladeoperationen aus dem Cache akkumuliert über den Verlauf der Initialbahnplanung mit 1000 Iterationen.

### Einfluss von Caching

Als besonders zeitaufwändiger Schritt in der Initialbahnplanung hat sich das Genotyp/Phänotyp Mapping (ohne Caching) herausgestellt. Dies liegt daran, dass ein Bahnsegment aus jeweils zwei Genen erzeugt wird, indem zunächst die geodätisch kürzeste Verbindung auf der Oberfläche zwischen den entsprechenden Netzknoten erzeugt wird und aus dieser Verbindung schließlich ein Bahnsegment gemäß dem Verfahren aus Abschnitt 4.5.6 generiert wird. Auf Grund der potentiell hohen Anzahl an verschiedenen Bahnsegmenten ist es nicht sinnvoll, alle möglichen Segmente vorzuberechnen. Jedoch liegt es in der Natur des Evolutionären Algorithmus, dass einige Bahnsegmente immer wieder in den Lösungen genutzt werden und solche Segmente daher nicht jedes Mal aufs Neue generiert werden sollten. Aus diesem Grund werden Bahnsegmente in einem Cache abgelegt und bei Bedarf aus diesem geladen, wie es in Abschnitt 4.9.9 beschrieben wird.

In Abbildung 4.29a ist der Zeitbedarf bei der Generierung eines Bahnsegments auf dem niedrig aufgelösten Werkstück A (Abbildung 4.13e), gemessen während 1000 Iterationen des Evolutionären Algorithmus, dargestellt. Weiterhin zeigt die Abbildung den Zeitbedarf beim Laden einer bereits vorhandenen Bahn aus dem Cache. Es ist deutlich zu erkennen, dass das Generieren eines neuen Bahnsegments wesentlich mehr Zeit benötigt als das einfache Laden. Der Aufwand zum Suchen eines Bahnsegments in der Cache Struktur ist demnach sehr gering.

Der Vorteil des Caches wird weiterhin dadurch verstärkt, dass es viele Bahnen gibt, die aus dem Cache geladen werden können und das Generieren neuer Bahnen nicht allzu häufig notwendig ist. In Abbildung 4.29b ist die akkumulierte Anzahl der Generierungsvorgänge und der Ladevorgänge aus dem Cache während 1000 Iterationen des Evolutionären

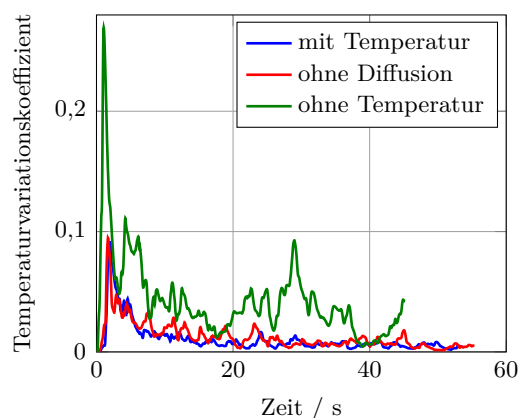


Abbildung 4.30: Thermische Varianzen von Bahnen, die mit bzw. ohne thermische Simulation geplant wurden, sowie von einer Bahn, die ohne Berechnung der Diffusion erzeugt worden ist.

Algorithmus dargestellt. Die Initialisierung der Population dauert 100 Iterationen, so dass in dieser Zeit fast ausschließlich neue Bahnsegmente erzeugt werden. Nach der Initialisierung werden die Nachkommen der aktuellen Population erzeugt, so dass viele Gene erhalten bleiben und dadurch viele Bahnsegmente bereits im Cache vorhanden sind. Dies ist leicht an dem steilen Anstieg der roten Kurve in Abbildung 4.29b zu erkennen. Die Anzahl neu generierter Bahnsegmente nimmt hingegen nur noch leicht zu (blaue Kurve).

Insgesamt ist das Caching demnach eine sehr effiziente Methode um Rechenzeit ohne Qualitätsverlust zu sparen.

### **Einfluss von Simulationskomponenten**

Die Simulation ist ein integraler Bestandteil des Evolutionären Algorithmus, da auf Grundlage der simulierten Daten die Zielfunktionswerte berechnet werden. In diesem Abschnitt wird analysiert, welche Simulationskomponenten keinen ausschlaggebenden Einfluss auf das Gesamtergebnis der Initialbahnplanung haben.

In Abbildung 4.30 sind die thermischen Varianzen dreier Bahnen aufgetragen, die mit unterschiedlichen Optionen durch die Initialbahnplanung generiert wurden. Im ersten Fall sind alle Simulationskomponenten aktiviert, im zweiten Fall ist die Diffusion deaktiviert und im letzten Fall sind keine thermischen Varianzen berechnet worden. Die Verläufe der thermischen Varianzen in der Abbildung sind durch erneute Simulation der neu abgetasteten und geglätteten Bahnen entstanden. Wie zu erwarten ist, ist die Nutzung der thermischen Simulation während der Optimierung der Bahnen mit der Initialbahnplanung von entscheidender Bedeutung, um Bahnen mit geringen thermischen Varianzen zu generieren. Dies ist leicht an den hohen Varianzen in der Bahn zu erkennen, die ohne

die thermische Simulation generiert wurde. Interessanterweise ist die Berechnung der Diffusion jedoch nicht von besonders großer Bedeutung, wie an den ähnlichen Varianzen der Bahnen zu erkennen ist, die mit bzw. ohne Diffusion generiert wurden. Dies liegt daran, dass die Varianzen durch den Wärmeeintrag und die Abkühlung entstehen und die Diffusion nur eine Art Glättung (Tiefpass) der Wärmeverteilung durchführt, was zu tendenziell weniger Varianz führt. Da bei der Initialbahnplanung der konkrete Wert in der Zielfunktion nicht entscheidend ist, kann die Initialbahnplanung in dem Fall sowohl mit wie auch ohne Diffusion zu den gleichen Ergebnissen kommen. Es ist demnach möglich, auf die Diffusion zu verzichten und nur mit dem Wärmeeintrag und der Abkühlung zu arbeiten.

### Theoretische Laufzeitbetrachtung

Neben der Messung der Zeit, die der Algorithmus benötigt, um eine Bahn zu generieren, können auch theoretische Überlegungen über den Laufzeitbedarf angestellt werden. Hier soll in Kürze zusammengestellt werden, wie die zu erwartende Laufzeit aussieht.

Für die Beschichtungssimulation müssen einige Operationen zur Bestimmung der geometrischen Lage zwischen je allen Bahnpunkten und allen Oberflächenknoten bestimmt werden, so dass die Laufzeit hierfür  $O(n \cdot m)$  beträgt. Hierbei ist wie üblich  $n$  die Anzahl der Bahnpunkte und  $m$  die Anzahl der Oberflächenknoten des ROIs. Bei der Temperatursimulation ist die Laufzeit nicht ganz so einfach anzugeben. Diese hängt von der Anzahl der Bahnpunkte und der Oberflächenknoten ab. Hinzu kommt, dass für jeden Bahnpunkt eine unbekannt Anzahl  $s$  an Diffusionsiterationen ausgeführt wird, welche wiederum eine Laufzeit abhängig von der Anzahl der Kanten und Knoten des Netzes hat. Die Laufzeit kann damit mit  $O(n \cdot (m + s \cdot (m_e + m)))$  angegeben werden. Das Generieren eines Bahnsegments geschieht in erster Linie durch die Bestimmung des geodätisch kürzesten Weges auf der Oberfläche des Netzes und wird von der Bibliothek *geodesic*<sup>2</sup> geleistet, welche nach [125] arbeitet. Die Bibliothek gibt die Generierung eines Weges mit  $O(m^2 \log m)$  an. Diese Laufzeit muss mit der Anzahl der zu generierenden Bahnsegmente multipliziert werden, die wiederum durch die Länge des Chromosoms abzüglich der durch das Caching bereits vorhandenen Bahnsegmente gegeben ist. Die durch diese Komponenten bestimmten Laufzeiten müssen schließlich noch mit der Anzahl der Iterationen multipliziert werden.

## 4.10 Optimierung von Rand-zu-Rand Bahnen

Die Initialbahnplanung nutzt eine gegebene Bahnstruktur (Abschnitt 4.5), um mit Hilfe des Prinzips des Evolutionären Algorithmus (Abschnitt 4.7) Bahnen zu generieren, die gute Eigenschaften gemessen an den Zielfunktionskomponenten (Abschnitt 4.6) besitzen. Im vorherigen Abschnitt 4.9 ist der Evolutionäre Algorithmus im Detail für die

<sup>2</sup><https://code.google.com/archive/p/geodesic/> (07.02.2017)

Punkt-zu-Punkt Bahnstruktur konkretisiert worden. Auf die gleiche Art und Weise kann eine Variante des Evolutionären Algorithmus auch für die Rand-zu-Rand Bahnstruktur aufgebaut werden. Da die Gestaltung des Algorithmus für die Rand-zu-Rand Bahnen sehr ähnlich dem in Abschnitt 4.9 beschriebenen Verfahren ist, wird in diesem Abschnitt nur in aller Kürze auf die wichtigsten Eigenschaften des Evolutionären Algorithmus eingegangen.

Die Zielfunktion setzt sich aus den in Abschnitt 4.6 genannten Zielfunktionskomponenten zusammen. Im Gegensatz zu der Punkt-zu-Punkt Variante des Algorithmus wird jedoch zusätzlich die Komponente  $E_z(\mathcal{X})$  genutzt. Diese misst die Länge von Zwischenbahnen, die außen an dem Werkstück entlanglaufen, um von den aktuellen Endpunkten der Bahnsegmente zu den Startpunkten der jeweils nächsten zu gelangen. Diese Zwischenbahnen erzeugen viel Overspray und sollten so gut wie möglich vermieden werden. Die Transformation der Zielfunktionen mit den *Desirability Functions* kann genau wie bei den Punkt-zu-Punkt Bahnen eingesetzt werden.

Die Repräsentation der Chromosomen geschieht über fünf Werte, um die Konstruktion der Bahnsegmente nach dem Verfahren aus Abschnitt 4.5.2 zu ermöglichen. Die drei ersten Werte,  $\text{index}_a$ ,  $\text{index}_b$  und  $\text{index}_c$ , sind drei Indizes, die die drei Eckpunkte eines Dreiecks des Werkstücknetzes identifizieren. In Abbildung 4.4b entsprechen diese den Knoten  $\mathbf{v}_0$ ,  $\mathbf{v}_a$  und  $\mathbf{v}_b$ . Mit Hilfe des vierten Wertes der Repräsentation, dem Wert  $\text{ratio}$ , wird ein Interpolationsfaktor zwischen 0 und 1 beschrieben, der die Lage des Punkts  $\mathbf{v}_1$  in Abbildung 4.4b bestimmt. Der letzte Wert der Repräsentation ist die  $\text{direction}$ , die die Richtung der Bahn als booleschen Wert angibt.

Die Initialisierung der Bahnen geschieht randomisiert, so dass die Werte für die Chromosomen der ersten Individuen zufällig gewählt werden.

Als Operatoren für den Evolutionären Algorithmus stehen die gleichen Operatoren zur Verfügung, die auch bei den Punkt-zu-Punkt Bahnen genutzt werden können. Konkret sind das die One-Point-, Two-Point- und Uniforme Rekombination sowie die Mutationsoperatoren, die in Tabelle 4.2 aufgelistet sind. Der *Shift* Operator ist hier jedoch dafür verantwortlich, den Index  $\text{index}_a$  auf den Index eines Nachbarknotens von  $\mathbf{v}_0$  zu verschieben. Zusätzlich kann der Punkt  $\mathbf{v}_1$  mit einem Operator namens *Rotation* auf dem 1-Ring des Knotens mit Index  $\text{index}_a$  verschoben werden.

Die Selektion funktioniert bei diesem Evolutionären Algorithmus genau wie bei dem für die Punkt-zu-Punkt Bahnen. Dies bedeutet, dass alle Operatoren für die Eltern-, Überlebens- und Auswahlselektion zur Verfügung stehen, die auch für den oben beschriebenen Algorithmus genutzt werden können.

Für den Evolutionären Algorithmus bieten sich, wie oben beschrieben, Caching Methoden an, um das wiederholte Generieren von gleichen Bahnsegmenten zu vermeiden, wobei für die Rand-zu-Rand Bahnen neben einem Cache auch das Vorberechnen von Bahnsegmenten genutzt wird. Das Caching funktioniert ähnlich wie oben, indem bereits generierte Bahnen in eine von mehreren Listen abhängig vom  $\text{index}_a$  abgelegt werden. Es gibt demnach für jeden Knoten des Oberflächennetzes eine Liste, so dass bei der

Suche nach einem Bahnsegment in dem Cache für jedes Gen mit dem Index  $\text{index}_a$  nur die entsprechende Liste durchsucht werden muss. Neben diesem Caching wird eine Reihe von Zwischenbahnen zwischen allen Paaren von Knoten auf dem äußeren Rand des EROIs vorberechnet, so dass diese schnell zur Verfügung stehen. Zusammen sorgen diese Maßnahmen für eine deutlich geringere Laufzeit des Algorithmus.

Das Konzept der Parallelisierung, das mit der Aufteilung des Algorithmus in einen Master und viele Threads einhergeht, siehe Abbildung 4.1, wird zur Beschleunigung des Verfahrens wie oben beschrieben verwendet. Das Konzept der Generationen wird hierdurch verworfen und das Steady-State Prinzip zur Selektion wird genutzt. Dafür skaliert der Algorithmus mit der Anzahl der zur Verfügung stehenden Prozessorkerne.

Die Parameter zur Steuerung des Evolutionären Algorithmus sind auf Grund der Ähnlichkeit zu dem oben beschriebenen Algorithmus identisch. Dies bedeutet insbesondere, dass die oben genannten Überlegungen für die Parameter sowie die aufgezeigten Parameteroptimierungen auch hier angewendet werden können.

Insgesamt unterscheidet sich der Algorithmus für die Rand-zu-Rand Bahnen kaum von dem für die Punkt-zu-Punkt Bahnen. Die geänderte Bahnstruktur erfordert leichte Anpassungen an der Repräsentation, den spezifischen Operatoren und dem Caching. Das komplette Framework und die Logik des Algorithmus bleiben jedoch erhalten.

## 4.11 Optimierung von quasi-konturparallelen Bahnen

Die Evolutionären Algorithmen für die beiden Bahnstrukturen Punkt-zu-Punkt und Rand-zu-Rand sind sehr ähnlich konstruiert und unterscheiden sich nur an wenigen Stellen, um die leicht unterschiedliche Bahnrepräsentation zu berücksichtigen. Für die quasi-konturparallelen Bahnen wird zwar auf das grundlegende Konzept der bisherigen Algorithmen zurückgegriffen, jedoch erfordert die deutlich andere Bahnstruktur eine größere Menge an Änderungen an dem Evolutionären Algorithmus. Im Folgenden wird kurz zusammengefasst, welche wesentlichen Gemeinsamkeiten und Unterschiede zu dem Algorithmus für die Punkt-zu-Punkt Bahnen bestehen.

Im Gegensatz zu den anderen beiden Bahnstrukturen sind die quasi-konturparallelen Bahnen sehr unflexibel. Die Bahnsegmente verlaufen quasi-parallel zu einer Werkstückkontur und haben jeweils einen festen Abstand zueinander. Der Auftrag auf der Oberfläche wird in erster Linie über diese Abstände bestimmt, so dass im Voraus der Abstand berechnet werden kann, mit dem die Zielschichtdicke möglichst exakt eingehalten wird. Auf planaren Werkstücken kann der geforderte Auftrag mit dieser Bahnstruktur sehr genau erzielt werden, während Beschichtungsfehler an den Krümmungen komplexer Werkstücke auftreten. An diesen sind jedoch vor allem die Roboterachsbeschleunigungen sehr hoch, was zu Problemen in der Praxis führen kann. Da durch den Evolutionären Algorithmus an der grundlegenden Bahnstruktur nichts verändert werden kann, beschränkt sich die Optimierung auf die Veränderung der Reihenfolge der Bahnsegmente, so dass zumindest



die thermischen Varianzen reduziert werden können. Die Richtung der Bahnsegmente wird nicht angepasst, da diese für aufeinanderfolgende Bahnsegmente alternierend ist.

Die Zielfunktionen für den hier genutzten Algorithmus sind die gleichen wie für den Algorithmus, der für die Punkt-zu-Punkt Bahnen genutzt wird. Dies bedeutet, dass die Komponenten  $E_l(\boldsymbol{x})$ ,  $E_t(\boldsymbol{x})$ ,  $E_m(\boldsymbol{x})$ ,  $E_o(\boldsymbol{x})$  zur Verfügung stehen. Die Zielfunktionskomponente  $E_z(\boldsymbol{x})$ , welche für die Rand-zu-Rand Bahnen genutzt wird, wird für die quasi-konturparallelen Bahnen nicht verwendet. Die *Desirability Functions* zur Zielfunktionsraumtransformation können auch für diesen Fall genutzt werden.

Die Repräsentation der Bahnen beruht auf einem Chromosom, das im Wesentlichen aus einem Vektor von Indizes besteht. Die Indizes stehen für die Bahnsegmente (Scans), die abgefahren werden sollen. Damit bestimmt die Reihenfolge der Indizes die Reihenfolge, in der die Bahnsegmente abgefahren werden.

Die Initialisierung zur Generierung der ersten Individuen geschieht randomisiert, so dass die Indexreihenfolge durch zufällige Permutation der Indizes erzeugt wird.

Als Rekombinationsoperator wird der Operator OX1 aus [104] genutzt. Dieser ist für das TSP entwickelt worden und kann mit der Bedingung umgehen, dass jeder Index in dem Chromosom genau einmal vorkommen muss. Als Mutation wird ein einfaches Vertauschen zweier Gene in dem Chromosom genutzt.

Die Selektionsoperatoren entsprechen denen der anderen Algorithmen. Sowohl für die Eltern-, Überlebens- als auch für die Auswahlselektion stehen alle genannte Operatoren zur Verfügung.

Für diese Variante des Evolutionären Algorithmus mit quasi-konturparallelen Bahnen ist ein Verfahren für das Caching von Bahnen nicht notwendig. Die Bahnsegmente können alle im Voraus berechnet und in einem Vektor abgelegt werden. Der Zugriff geschieht per Index, so dass auch keine Suche in der Datenstruktur erforderlich ist.

Die Parallelisierung gemäß dem Schema aus Abbildung 4.1 ist auch in diesem Algorithmus enthalten. Die Threads können die restrukturierten Bahnen unabhängig voneinander konstruieren, simulieren und evaluieren.

Die meisten Parameter des Algorithmus sind identisch zu denen der anderen Algorithmen. Es werden zusätzlich nur ein paar weitere Parameter zur Initialisierung der Bahnsegmente benötigt. Diese haben auf den Ablauf des Algorithmus jedoch keine Auswirkungen.

Obwohl der Algorithmus zur Planung quasi-konturparalleler Bahnen strukturell denen für Punkt-zu-Punkt und Rand-zu-Rand Bahnen ähnlich ist, gibt es einige Änderungen, die sich aus den Eigenschaften der Bahnstruktur ergeben. Vor allem die geringe Flexibilität in der Bahnstruktur ist ein wesentlicher Unterschied zu den anderen Algorithmen. Bei den folgenden Evaluationen der Algorithmen werden die quasi-konturparallelen Bahnen mit den anderen Bahnstrukturen verglichen.



Abbildung 4.31: Werkstück für die Evaluation. Das ROI des Werkstücks ist in Rot gekennzeichnet.

## 4.12 Evaluation

In diesem Abschnitt wird die Initialbahnplanung für die Punkt-zu-Punkt und die quasi-konturparallele Bahnstruktur miteinander verglichen sowie herkömmlichen quasi-konturparallelen Bahnen auf einem komplexen Werkstück gegenübergestellt. Ziel der Untersuchung ist es, herauszufinden, welche Eigenschaften der Bahnstrukturen auf komplexen Werkstücken besonders gut geeignet sind und in welchen Belangen sie jeweils Vorteile gegenüber den anderen Strukturen haben. Die Rand-zu-Rand Bahnen werden am Ende des Abschnitts kurz behandelt.

### 4.12.1 Testdaten und Qualitätsmerkmale

Als Testwerkstück soll das bereits vorgestellte Werkstück B, siehe Abschnitt 4.9.13, verwendet werden. In Abbildung 4.31 ist dieses noch einmal mit dem entsprechenden ROI dargestellt. Dieses ist auf die wesentlichen und funktionalen Stellen des Werkstücks reduziert. Das EROI lässt sich einfach durch Vergrößerung des ROIs nach unten erstellen.

Das Werkstück B wird auf Grund der vielfältigen konvexen und konkaven Radien ausgewählt, die im Allgemeinen schwierig zu beschichten sind. Entgegen dem Aufbau des Werkstücks A sind die vorhandenen Krümmungen mit kleinen Radien in beiden Dimensionen zu finden, so dass keine eindeutige Vorzugsrichtung für parallele Bahnen gegeben ist. Die Radien auf diesem Werkstück sind mit unter 5 mm für den genutzten Libo-Beschichtungsprozess eigentlich zu klein, so dass es sich hier um ein besonders anspruchsvolles Werkstück handelt. Es kann demnach nicht erwartet werden, dass die engen Radien fehlerfrei beschichtet werden können. Dennoch soll dieses Werkstück als Grundlage der Evaluation dienen, um die Möglichkeiten aber auch die Grenzen des Verfahrens zu ermitteln.

Die Initialbahnplanung wird auf dem Werkstück mit den unterschiedlichen Bahnstrukturen ausgeführt. Die Ergebnisbahnen werden anschließend mit den in Abschnitt 4.6 vorgestellten Zielfunktionen bewertet und miteinander verglichen. Die Kriterien sind demnach der Fehler in der Beschichtung  $E_l(\mathcal{X})$ , die thermischen Varianzen  $E_t(\mathcal{X})$ , die Roboterachsbeschleunigungen  $E_m(\mathcal{X})$  sowie der Overspray  $E_o(\mathcal{X})$ . Auf Grund der

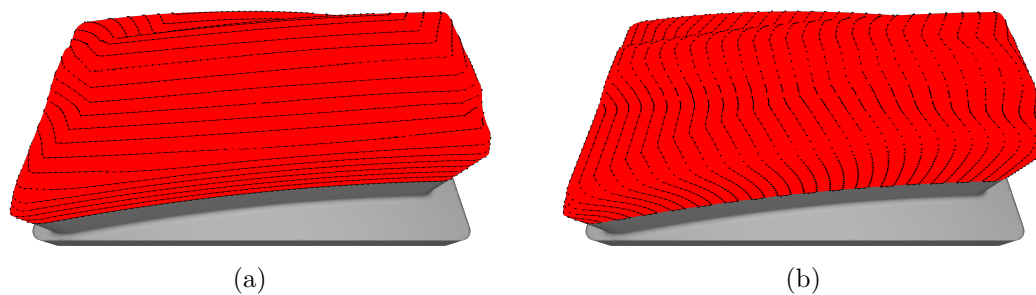


Abbildung 4.32: Visualisierung von quasi-konturparallelen Bahnen auf dem Werkstück B. In (a) sind die Bahnen quasi-konturparallel zur vorderen, unteren Kontur des EROIs erzeugt worden und in (b) zur linken unteren Kontur. Für die Übersichtlichkeit sind die Bahnsegmente in dieser Abbildung im festen Abstand von 9 mm respektive 9,1 mm zueinander erzeugt worden, was jeder 5. Bahn entspricht. Auf Grund der Krümmungen wirken die Bahnen vor allem weit von der Quellkontur entfernt nicht parallel.

Tatsache, dass die Initialbahnplanung und die Bahnoptimierung, siehe Kapitel 5, als Einheit für den gesamten Prozess gesehen werden sollen, werden die Zielfunktionen auch nach der Optimierung der Bahnen mit der Bahnoptimierung betrachtet.

#### 4.12.2 Vergleich der Verfahren

Die Evaluation betrachtet herkömmliche quasi-konturparallele Bahnen sowie die durch die Initialbahnplanung generierten quasi-konturparallelen und Punkt-zu-Punkt Bahnen. Dabei werden sowohl die Stärken als auch die Schwächen der jeweiligen Verfahren betrachtet, um zu entscheiden, in welchen Situationen welche der Bahnstrukturen besonders gut geeignet sind.

Zunächst werden herkömmliche quasi-konturparallele Bahnen betrachtet. Diese sind in den Abbildungen 4.32a und 4.32b für das Werkstück B dargestellt. Die Bahnen unterscheiden sich in der Richtung der quasi-parallel Bahensegmente und sind erzeugt worden, indem eine der Konturen als Quelle für die Erstellung der Bahnsegmente genutzt wurde. Im ersten Fall dient die lange, im zweiten Fall die kurze Kontur als Quelle. In beiden Fällen wirken die Bahnen auf Grund der Krümmungen auf der Oberfläche mit zunehmendem Abstand zur Quelle immer weniger parallel. Dies ist ein normales Verhalten der quasi-konturparallelen Bahnen, da der Abstand zwischen zwei Bahnsegmenten konstant gehalten wird.

Die Beschichtungen, die von den herkömmlichen quasi-konturparallelen Bahnen generiert werden, haben einen maximalen Fehler von unter 24 %, was einen guten Wert für eine initialgeplante Bahn darstellt. Die thermischen Varianzen sind jedoch sehr schlecht. Diese sind in Abbildung 4.33 für den Verlauf des Beschichtungsprozesses dargestellt. In der gleichen Abbildung sind zum Vergleich auch die Varianzen von optimierten Bahnen abgebildet. Es ist leicht zu erkennen, dass die Varianzen bei den herkömmlichen quasi-

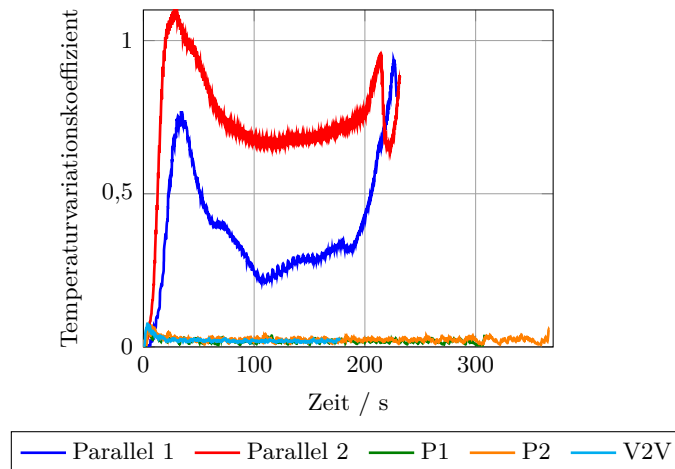


Abbildung 4.33: Thermische Varianzen verschiedener Bahnen auf Werkstück B. „Parallel 1“ und „Parallel 2“ bezeichnen herkömmliche quasi-konturparallele Bahnen und „P1“ und „P2“ durch die Initialbahnplanung generierte quasi-konturparallele Bahnen, wobei die Bahnen mit der Ziffer 1 entlang der langen und die mit Ziffer 2 entlang der kurzen Kontur erzeugt wurden. „V2V“ bezeichnet die durch die Initialbahnplanung generierte Bahn mit Punkt-zu-Punkt Bahnstruktur.

konturparallelen Bahnen nicht annähernd das Niveau der optimierten Bahnen erreichen, sondern um ca. eine Größenordnung schlechter sind.

Um das Problem der hohen thermischen Varianzen zu lösen, wird der Evolutionäre Algorithmus für die quasi-konturparallelen Bahnen genutzt. Dieser sortiert die Bahnsegmente derart um, dass die Varianzen reduziert werden, ohne dass sich die Qualität der Beschichtung merklich verändert. Das Resultat der Initialbahnplanung ist eine neue Bahn, die die gleiche Beschichtungsqualität bei geringerer thermischer Varianz erreicht. Die thermischen Varianzen der optimierten Bahnen sind in Abbildung 4.33 dargestellt. Wie im Vergleich mit den herkömmlichen quasi-konturparallelen Bahnen gut zu erkennen ist, ist die maximale thermische Varianz wesentlich besser.

Ein bisher nicht betrachtetes Problem der umsortierten quasi-konturparallelen Bahnen ist der Overspray, der beim Abarbeiten der Bahn erzeugt wird. Dieser fällt sehr hoch aus, da durch die Umsortierung viele aufeinanderfolgende Bahnsegmente sehr weit auseinander liegen und der Roboter daher große Strecken entlang des Randes des Werkstücks abfahren muss. Dies ist ein großes Problem in der Praxis, da das Verschenden von Material nicht nur sehr hohe Kosten verursacht, sondern auch sehr schädlich für die Umwelt ist.

Um das Problem des hohen Oversprays bei gleichbleibender Schichtqualität zu lösen, sollen die Punkt-zu-Punkt Bahnen betrachtet werden. Die Punkt-zu-Punkt Bahnen werden mit dem in Abschnitt 4.9 vorgestellten Verfahren erzeugt. Zur Planung wird, wie bei den quasi-konturparallelen Bahnen auch, die gering aufgelöste Version des Werkstücks verwendet. Die Werte der Zielfunktionen der entstehenden Bahn (ermittelt auf einer hö-

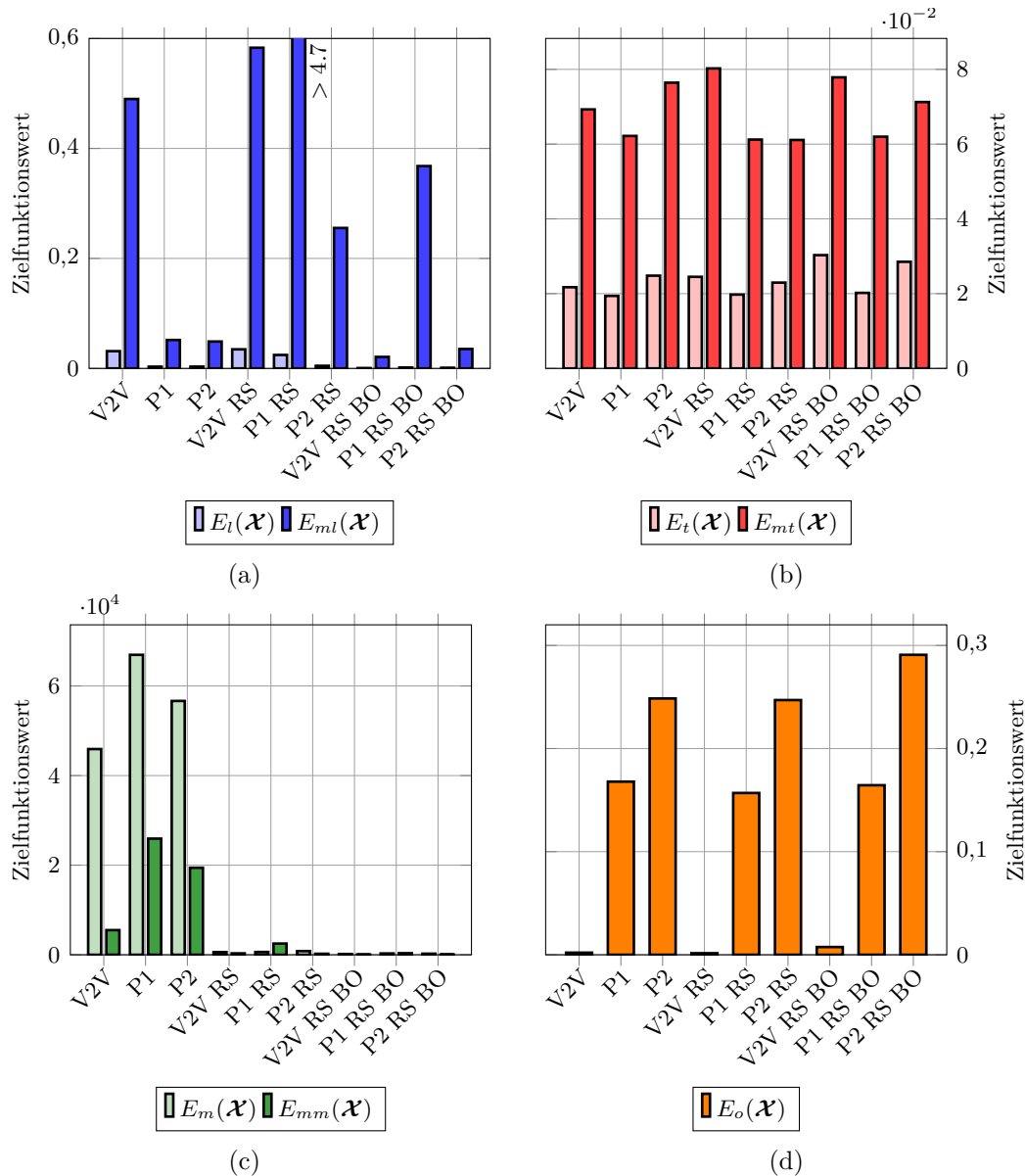


Abbildung 4.34: Zielfunktionswerte von quasi-konturparallelen („P1“, „P2“) und Punkt-zu-Punkt Bahnen („V2V“) nach der Initialbahnplanung (linkes Drittel), wobei „P1“ entlang der langen und „P2“ entlang der kurzen Kontur erstellt wurde. Des Weiteren sind die Zielfunktionswerte nach dem Neuabtaben und Glätten der Bahnen („RS“ [„Resampling“], mittleres Drittel) sowie nach der Optimierung dieser Bahnen mit der Bahnoptimierung („BO“, rechtes Drittel, Kapitel 5) aufgetragen. (a) zeigt die mittleren und maximalen Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ), (b) die mittleren und maximalen thermischen Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), (c) die mittleren und maximalen Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$ ) und (d) den Overspray ( $E_o(\mathcal{X})$ ).

her aufgelösten Version des Werkstücks, wie bei allen Auswertungen in diesem Abschnitt) sind in Abbildung 4.34 unter dem Kürzel „V2V“ zusammengestellt. In den gleichen Abbildungen sind im linken Drittel auch die Werte der beiden quasi-konturparallelen Bahnen, die sich in der Quellmenge für die Erzeugung der quasi-konturparallelen Bahnsegmente unterscheiden, dargestellt. Die Werte für die Beschichtung, siehe Abbildung 4.34a, sind für die Punkt-zu-Punkt Bahnen zunächst deutlich schlechter als für die quasi-konturparallelen Bahnen. Dies liegt einerseits an der relativ geringen Zielschichtdicke, so dass Kreuzungen von Bahnen bereits kritisch sind, und andererseits daran, dass alle Bahnen mit konstanter Oberflächengeschwindigkeit erzeugt werden. Dadurch ist die Beschichtung für quasi-konturparallele Bahnen sehr gut. Das Problem bei diesem Vorgehen ist jedoch, dass eine konstante Oberflächengeschwindigkeit an Krümmungen, die durch die parallele Struktur der Bahnen nicht umgangen werden können, zu sehr starken Umorientierungen des Roboters führt. Dies ist in Abbildung 4.34c daran zu erkennen, dass die maximalen Roboterachsbeschleunigungen der quasi-konturparallelen Bahnen deutlich höher sind, als die der Punkt-zu-Punkt Bahnen. Im Falle der Roboterachsbeschleunigungen sind vor allem die maximalen Werte kritisch, da diese auf Stellen in der Bahn hindeuten, die der Roboter mit der geforderten Geschwindigkeit nicht abfahren kann. Die thermischen Varianzen beider Verfahren sind vergleichbar, wie in Abbildung 4.34b zu sehen ist, so dass hier keine der Bahnstrukturen einen wesentlichen Vorteil hat. Der Overspray, siehe Abbildung 4.34d, ist jedoch deutlich besser für die Punkt-zu-Punkt Bahnen. In diesem Punkt liegt ein großer Vorteil dieser Bahnstruktur.

Als Zwischenfazit kann festgehalten werden, dass die durch die Initialbahnplanung erzeugten quasi-konturparallelen Bahnen einen besseren Auftrag erzeugen als die Punkt-zu-Punkt Bahnen, die jedoch ihrerseits bessere Roboterachsbeschleunigungen und wesentlich geringeren Overspray aufweisen. Tatsächlich sind die Roboterachsbeschleunigungen für die quasi-konturparallelen Bahnen so hoch, dass der reale Roboter mit hoher Wahrscheinlichkeit Schwierigkeiten hat, die Bahnen wie gefordert zu fahren. Da die Beschichtungsergebnisse der initialgeplanten Bahnen für beide Bahnstrukturen nicht zufriedenstellend sind, muss der Vergleich noch weitergeführt werden, so dass die Bahnen im Folgenden nach der Optimierung durch die Bahnoptimierung (Kapitel 5) betrachtet werden. Daher werden in den nächsten Schritten die Bahnen zunächst neu abgetastet und durch B-Splines geglättet, siehe Abschnitt 6.4, und anschließend durch die Bahnoptimierung optimiert.

Die Bahnoptimierung erhält als Eingabe die Bahnen, die durch die Initialbahnplanung erzeugt wurden, und optimiert die Bahnen bezüglich ihres Auftrags und des Roboterhaltens, siehe Kapitel 5. In Abbildung 4.34a sind im rechten Drittel die Zielfunktionswerte für den endgültigen Auftrag für die Bahnen nach der Optimierung im Vergleich zu den Werten für die Bahnen nach der Initialbahnplanung (linkes Drittel) und dem Glätten (mittleres Drittel) dargestellt. Es ist deutlich zu erkennen, dass die Beschichtungsqualität für die Punkt-zu-Punkt Bahnen nach der Bahnoptimierung deutlich gestiegen ist. Sowohl die durchschnittlichen als auch die maximalen Fehler sind stark gesunken. Bei den quasi-konturparallelen Bahnen ist dies nicht der Fall. Die Beschichtung kann nicht stark verbessert werden, da das Roboterverhalten bezüglich der Achsbeschleunigungen

sehr schlecht ist. Bereits durch die Glättung der Bahn wird diese so stark verändert, dass sehr hohe Schichtfehler entstehen. Auf das Glätten kann jedoch nicht verzichtet werden, wenn die Bahn später auf einen echten Roboter übertragen werden soll, siehe auch Abschnitt 5.7.6. Die Alternative, die Bahnoptimierung direkt auf die initialgeplante Bahn anzuwenden, da diese die Roboterachsbeschleunigungen ebenfalls reduzieren kann, ist nicht zu empfehlen. Wenn die Beschleunigungen sehr hoch sind, werden diese bei dem Gradientenabstieg stark fokussiert, so dass die Bahnoptimierung die Beschichtungsqualität kaum verbessert oder sogar verschlechtert. Ebenso führen hohe Fehler in der Eingabe zu allgemein schlechteren Ergebnissen nach der Bahnoptimierung, da diese mit einer gegebenen Anzahl an Iterationen nur beschränkt starke Verbesserungen erreichen kann. Die Betrachtung der thermischen Varianzen in Abbildung 4.34b zeigt, dass diese für beide Bahnstrukturen auch nach der Bahnoptimierung auf dem gleichen Niveau verbleiben. Sowohl die Glättung als auch die Bahnoptimierung sorgen nicht für wesentliche Verschlechterungen in den Varianzen, obwohl sie dieses Zielkriterium nicht explizit berücksichtigen. Auch der direkte Vergleich der thermischen Varianzen zwischen den beiden Bahnstrukturen zeigt, dass diese untereinander konkurrenzfähig sind. Die Differenzen der konkreten Werte liegen durchaus im Bereich der üblichen Variationen der Evolutionären Algorithmen. Der Overspray (Abbildung 4.34d) wird durch die Glättung und Optimierung der Bahnen nur geringfügig beeinflusst, wie es auch zu erwarten war. Damit sind die Werte für den Overspray der Punkt-zu-Punkt Bahnen deutlich besser, als die der quasi-konturparallelen Bahnen.

Insgesamt lässt sich festhalten, dass beide Bahnstrukturen Vor- und Nachteile besitzen. Bei alleiniger Betrachtung der Ergebnisse nach der Initialbahnplanung schneiden die quasi-konturparallelen Bahnen im Bereich der Beschichtungsqualität besser ab. Dafür besitzen die Bahnen dieses Typs zu hohe Roboterachsbeschleunigungen und deutlich zu hohen Overspray. Nach der Optimierung sind die Beschichtungsergebnisse der Bahnstrukturen ähnlich und sogar leicht besser für die Punkt-zu-Punkt Bahnen. Auch die Roboterachsbeschleunigungen sind leicht besser für diese Bahnstruktur. Der Overspray ändert sich durch die Optimierung kaum, so dass die Werte für die Punkt-zu-Punkt Bahnen deutlich besser sind. Die resultierenden thermischen Varianzen sind für alle Verfahren vergleichbar und in jedem Fall wesentlich besser als bei herkömmlichen, nicht umsortierten quasi-konturparallelen Bahnen.

In Abbildung 4.35 sind die Werkzeugbahnen nach der Initialbahnplanung, nach dem Neuabtasten und der Glättung sowie nach der Bahnoptimierung dargestellt. Bei genauer Betrachtung können die unterschiedlichen Strukturen der Bahnen erkannt werden. Zusätzlich ist die Beschichtung auf der Werkstückoberfläche farbcodiert dargestellt, wobei zur Ermittlung der Fehlerwerte in diesem Fall auf die externe Simulation *SimSpray* zurückgegriffen wurde, die die Beschichtung etwas exakter vorhersagt und für die Optimierung mit der Bahnoptimierung verwendet wird. Das beschriebene Verhalten nach den einzelnen Schritten kann an den Abbildungen in horizontaler Richtung nachvollzogen werden. Die Skala ist hier für alle Bilder gleich gewählt worden, auch wenn die maximalen Beschichtungsfehler nach den ersten Schritten deutlich höher sind. In Abbildung 4.36 sind die Stellen mit den größten Beschichtungsfehlern für die optimierten Bahnen noch



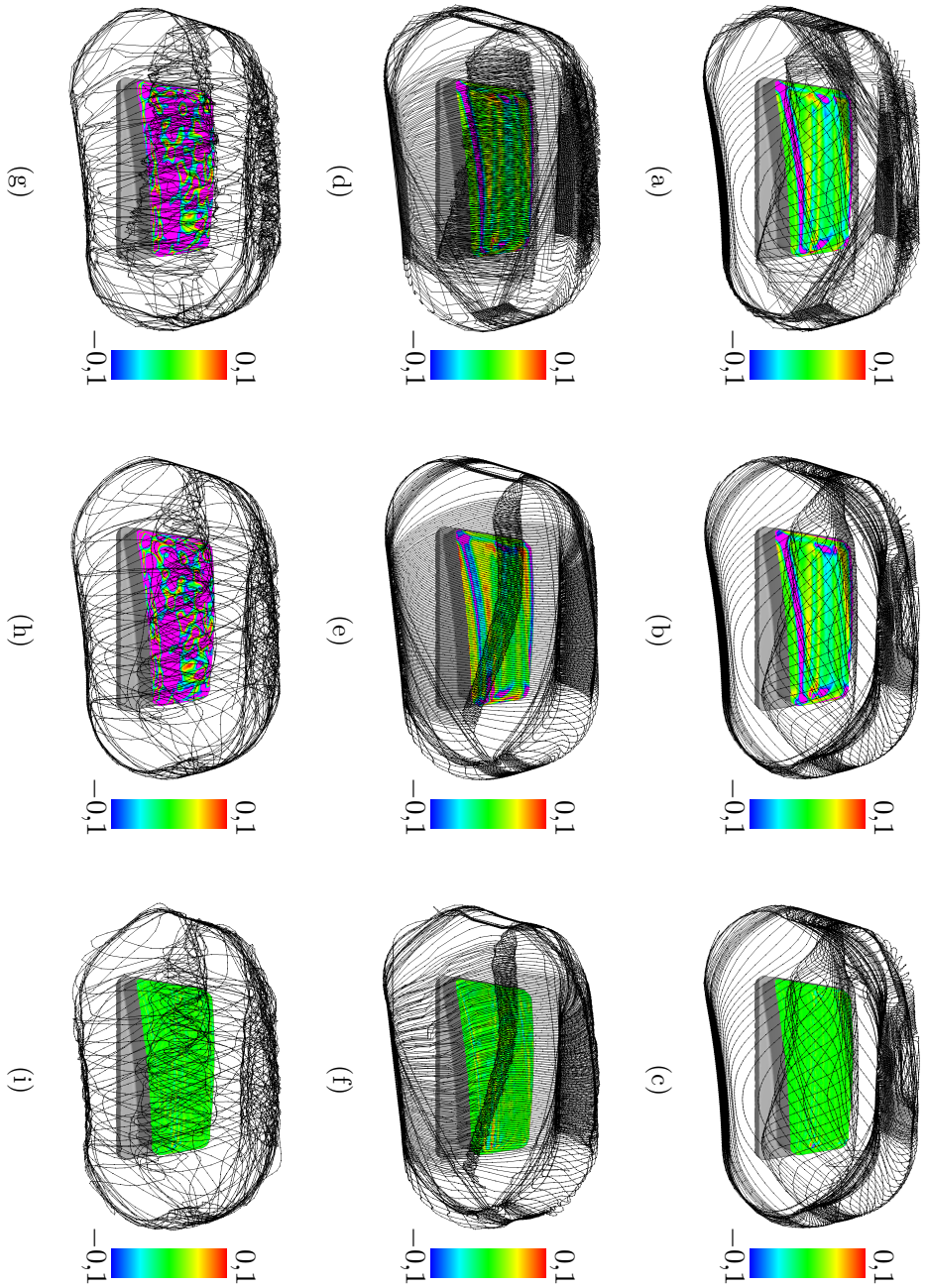


Abbildung 4.35: Visualisierung der Bahnen verschiedener Bahnstrukturen nach den Optimierungsschritten. Von links nach rechts: Initialgeplante Bahn; neu abgetastete und geglättete Bahn; mit der Bahnoptimierung optimierte Bahn. Von oben nach unten: quasi-konturparallele Bahn an langer Kontur; quasi-konturparallele Bahn an kurzer Kontur; Punkt-zu-Punkt Bahn. Rosa Bereiche auf der Oberfläche beschreiben Fehler außerhalb der angegebenen Skala. Die Ergebnisse sind mit der externen Simulation *SimSpray* erzeugt worden.



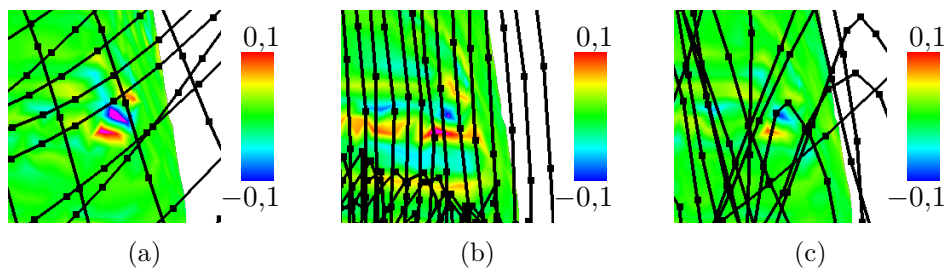


Abbildung 4.36: Ausschnitte aus den Abbildungen 4.35c, 4.35f und 4.35i an der vorderen rechten Stelle des Werkstücks, welche die größten Beschichtungsfehler aufweisen.

einmal vergrößert dargestellt. Man erkennt, dass die Fehler für die Punkt-zu-Punkt Bahnen am geringsten sind.

In der obigen Analyse sind die Rand-zu-Rand Bahnen nicht betrachtet worden. Die Rand-zu-Rand Bahnen können als Zwischenstruktur zwischen quasi-konturparallelen und Punkt-zu-Punkt Bahnen betrachtet werden. Die Rand-zu-Rand Bahnen verlaufen genau wie die quasi-konturparallelen Bahnen von einer Kontur zu einer anderen, sind jedoch, wie die Punkt-zu-Punkt Bahnen, nicht darauf beschränkt parallel zueinander zu verlaufen. Das größte Problem der Rand-zu-Rand Bahnen bei der Generierung und Optimierung ist, dass viele Zwischenbahnen erzeugt werden, die wiederum hohen Overspray erzeugen. Des Weiteren sind die Punkt-zu-Punkt Bahnen strukturell deutlich flexibler als die Rand-zu-Rand Bahnen, so dass die Punkt-zu-Punkt Bahnen die Rand-zu-Rand Bahnen abbilden können, wenn für die Konstruktion der Punkt-zu-Punkt Bahnen das EROI verwendet wird, wie es hier der Fall ist. Daher ist kein Vorteil durch die Nutzung der Rand-zu-Rand Bahnen gegenüber den Punkt-zu-Punkt Bahnen zu erwarten.



# Bahnoptimierung

Die in diesem Kapitel vorgestellte Bahnoptimierung stellt ein neuartiges Konzept zur Verbesserung bereits vorhandener Bahnen im Bereich des thermischen Spritzens dar. Die Eingabe besteht aus einer Bahn und einem zu beschichtenden Werkstück. Das Ziel ist die Optimierung der Bahn bezüglich des Beschichtungsfehlers und der Erhalt oder die Verbesserung der geometrischen Eigenschaften der Bahn (Beschleunigung, Winkelbeschleunigung, Krümmung) oder der Roboterkinematik (Achsbeschleunigungen), so dass eine problemlose Ausführbarkeit der Bahn durch einen Roboter gewährleistet wird. Da die Qualität der Optimierung von der Güte der Vorhersagen der durch die Bahn entstehenden Schichtdicken abhängt, wird eine externe Simulation verwendet, die gegen einen realen Prozess kalibriert ist. Auch Messdaten nach Testläufen könnten verwendet werden. Um trotz der Nutzung der externen Simulation effizient optimieren zu können, wird ein analytisches Auftragsmodell genutzt, um Gradientenrichtungen für ein iteratives nichtlineares konjugiertes Gradientenverfahren zu erzeugen. Der Fehler in der Beschichtung, der in der Zielfunktion benötigt wird, wird aber ausschließlich mit Hilfe der externen Simulation bestimmt.

Zunächst wird in Abschnitt 5.1 das grundlegende Konzept der Bahnoptimierung erläutert. Anschließend wird in Abschnitt 5.2 eine Übersicht über den Stand der Forschung im Bereich der Bahnoptimierung gegeben. Der Aufbau des Optimierungsverfahrens mit einer kurzen Erläuterung der Komponenten des Systems wird in Abschnitt 5.3 dargestellt. Es folgt die Definition der Zielfunktion des Optimierungsproblems in Abschnitt 5.4. Das Lösungsverfahren verwendet nichtlineare konjugierte Gradienten und wird in Abschnitt 5.5 vorgestellt. Die Optimierung wird in Abschnitt 5.6 für zwei Varianten des Verfahrens aufgestellt, nämlich für die bahnbasierte und die roboterbasierte Optimierung. Als Letztes folgt in Abschnitt 5.7 eine umfangreiche Evaluation der Bahnoptimierung, die sowohl die einzelnen Komponenten als auch Fallbeispiele berücksichtigt.

## 5.1 Einleitung

Das Ziel der Bahnoptimierung ist die Verbesserung gegebener Bahnen. Dazu zählt das Verringern der Abweichungen der Schichtdicke von der Zielschichtdicke sowie die Verbesserung der Bahngeometrie bezüglich der Ausführbarkeit der Bahn mit einem Roboter. Die Bahnen, die als Eingabe der Bahnoptimierung dienen, können sowohl von einer beliebigen Software generiert als auch durch Teaching erzeugt worden sein. Auch vorhandene Bahnen, die für einen anderen Prozess oder ein anderes, jedoch ähnliches Werkstück erstellt worden sind, können genutzt werden. Insbesondere kann die Bahnoptimierung als Nachverarbeitungsschritt der in Kapitel 4 beschriebenen Initialbahnplanung eingesetzt werden.

In vielen Fällen werden nach dem initialen Erstellen einer Bahn nicht alle Zielkriterien vollständig erfüllt. Dies liegt einerseits an der Komplexität des Bahnplanungsproblems, die es schwierig macht, eine Lösung zu finden, die allen Gütekriterien gerecht wird. Andererseits können Änderungen der Prozessparameter zu Fehlern führen, wobei die Änderungen sowohl gewollt als auch ungewollt, zum Beispiel durch Verschleiß, entstehen können. Wenn die Fehler zu groß sind, muss eine neue Bahn erzeugt werden, was für komplexe Freiformgeometrien ein langwieriger und kostspieliger Prozess ist. Die Anpassung vorhandener Bahnen durch die Bahnoptimierung kann das Erzeugen neuer Bahnen in vielen Fällen überflüssig machen.

Ein weiterer Vorteil, der durch die Optimierung von Bahnen entsteht, ist, dass der vollständige Bahnplanungsprozess in eine globale und eine lokale Planungsphase aufgeteilt werden kann. Die globale Planung kann daher auf die Nutzung exakter, jedoch zeitintensiver Simulationsmodelle verzichten und damit bei vergleichbarem Aufwand einen größeren Bereich des Suchraums betrachten. Die Fehler durch die Nutzung von inexakten Modellen werden dann von der Bahnoptimierung korrigiert. Auf diese Weise können insgesamt bessere Bahnen bei gleichem Zeiteinsatz erzeugt werden, da nur wenige Bahnen mit den exakten Simulationsmodellen ausgewertet werden müssen.

Im Bereich der hier eingesetzten Bahnoptimierung werden zwei Optimierungsziele verfolgt. Das erste ist die Minimierung der Abweichung der Schicht von der Zielschichtdicke an jedem Punkt auf der zu beschichtenden Oberfläche. Das zweite Ziel ist die Erhaltung oder Verbesserung der Bahngeometrie bezüglich der Roboterkinematik. Das zweite Ziel kann über direkte Anpassungen der Bahnparameter erfolgen, so dass die geometrische Krümmung der Bahn, die (Winkel-)Geschwindigkeiten zwischen aufeinanderfolgenden Werkzeugposen sowie die Beschleunigungen verringert werden. Bei einer Bahnparametrisierung über die Roboterachsstellungen können alternativ auch direkt die Winkelgeschwindigkeiten und -beschleunigungen der Roboterachsen optimiert werden.

## 5.2 Stand der Forschung

Bahnoptimierung und im Speziellen simulationsbasierte Bahnoptimierung beim thermischen Spritzen ist ein wenig erforschtes Gebiet. Die meisten Arbeiten beschränken sich darauf, die Bahngenerierung anzupassen, indem die Basisparameter der Bahnstruktur verändert oder optimiert werden. Bei den häufig eingesetzten Mäanderbahnen sind das der Mäanderbahnabstand, der Spritzwinkel oder die Geschwindigkeit.

Allgemeine Bahngenerierungsmethoden und allgemeine Arbeiten zur Bahnplanung beim thermischen Spritzen sind bereits im Abschnitt 4.2 zusammengestellt worden. An dieser Stelle werden daher ergänzend einige Arbeiten mit direktem Bezug zur simulationsbasierten Bahnoptimierung aufgeführt.

Candel und Gadow [29, 66] beschreiben eine Möglichkeit zur Optimierung von Bahnen mit Hilfe von Simulationen. Die Basis bilden Werkstücke, die entweder als CAD-Daten vorliegen oder mittels Sensoren eingescannt werden. Auf der Oberfläche werden dann die Normalen bestimmt, die zur Bestimmung der Spritzrichtungen genutzt werden. Die Bahn wird als Mäanderbahn erzeugt, die eine feste Spritzdistanz, einen festen Mäanderbahnabstand und eine feste Geschwindigkeit aufweist. Die Bahn wird simuliert, indem die Bahndaten als Grundlage für ein Finite Elemente Modell genutzt werden, das die Wärmeverteilung in dem Werkstück berechnet. Diese Daten sollen für die Bahnoptimierung genutzt werden. Wie dies jedoch konkret geleistet werden soll, wird nicht beschrieben.

Fasching et al. [59] benutzen einen Ansatz zur Optimierung von Mäanderbahnen, um eine gleichmäßige Schichtdicke zu generieren. Dazu wird der Anstellwinkel der Spritzpistole so angepasst, dass eine möglichst gaußförmige Verteilung auf der Oberfläche entsteht. Schließlich wird auf dieser Grundlage der Bahnabstand so bestimmt, dass bei mehreren Überläufen möglichst wenige Abweichungen von der Zielschichtdicke entstehen. Diese Art der Vorberechnung der Bahnparameter erspart die Simulation der vollständigen Bahnen, was in relativ wenig Rechenaufwand resultiert. Allerdings ist das Verfahren auf Mäanderbahnen beschränkt und kann keine lokalen Anpassungen an der Bahn vornehmen.

Ein Bahnoptimierungsansatz für Fräsen wird in [21] und [141] beschrieben. Dieser nutzt eine Raumverzerrung auf Basis von B-Splines, um die CAM-Bahnen zu modifizieren. Die B-Spline Kurven werden mit Hilfe von Kontrollpunkten eines Gitters beschrieben, welche bei der Optimierung ebenfalls verzerrt werden. Die Verzerrung des Raums wird hier von den Abweichungen der Ist-Form von der Soll-Form des Werkstücks bestimmt. Durch Simulation werden die Abweichungen iterativ bestimmt und entsprechend der Methode korrigiert. Im Gegensatz zu diesem Vorgehen arbeitet das hier vorgestellte Verfahren direkt auf den Bahndaten und nicht auf einer B-Spline Repräsentation. Ein Grund dafür ist, dass beim thermischen Spritzen ein weitaus größerer Raum betrachtet werden muss als beim Fräsen, was zu einer größeren Menge an Kontrollpunkten führen würde. Außerdem werden viele Kontrollpunkte benötigt, wenn die Bahn auch feine Strukturen aufweisen soll.

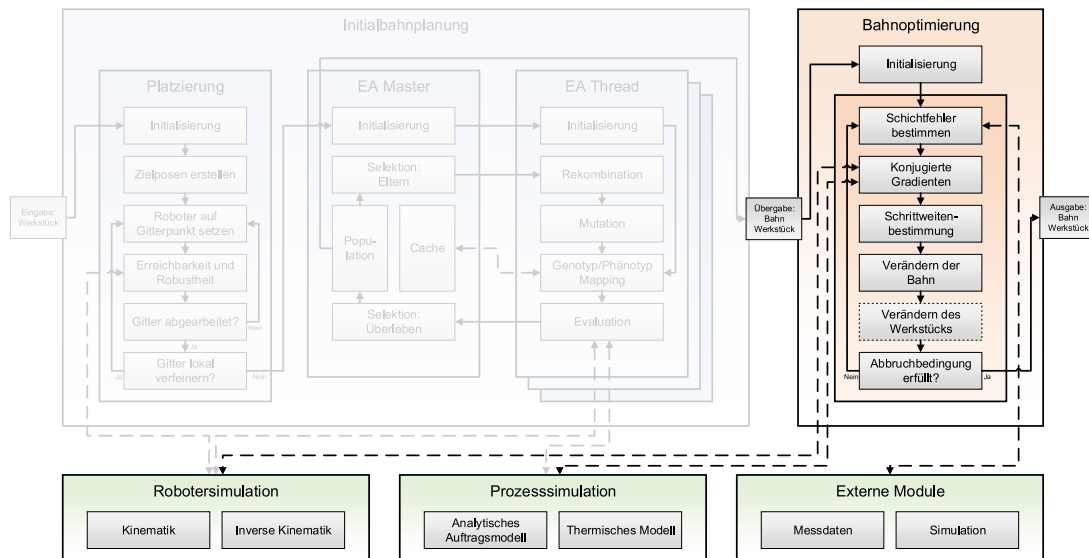


Abbildung 5.1: Konzeptioneller Aufbau der Bahnoptimierung. Die Komponenten des Systems sind durch graue Rechtecke dargestellt. Optionale Komponenten haben einen gepunkteten Rand. Die Kontrollflüsse sind durch durchgezogene Pfeile gekennzeichnet, die Datenkommunikation über gestrichelte Pfeile.

### 5.3 Übersicht

In Abbildung 5.1 sind die einzelnen Schritte, die das Verfahren ausführt, sowie die Optimierungsschleife dargestellt. Nach einer Initialisierung iteriert das Verfahren die Optimierung startend mit der Bestimmung des Beschichtungsfehlers, der durch die Eingabebahn entsteht. Hierzu wird eine externe Simulation genutzt oder alternativ reale Messdaten. Die Optimierung benutzt ein nichtlineares konjugiertes Gradientenverfahren, dessen Gradienten mit Hilfe eines analytischen Auftragsmodells bestimmt werden. Anschließend wird eine geeignete Schrittweite berechnet und die Bahn sowie, falls gewünscht, das Werkstück basierend auf diesen Daten manipuliert. Das Verfahren kann beliebig viele Schritte ausführen, bis ein Abbruchkriterium erfüllt ist. Dies könnte das Erreichen der gewünschten Qualität der Schicht, das Erreichen einer vorgegebenen Anzahl von Schritten oder zu geringfügiger Fortschritt in den Iterationen sein.

Im Gegensatz zur Initialbahnplanung benötigt die Bahnoptimierung sehr präzise Vorhersagen oder Messungen der Schichtdicke, die durch eine Bahn erzeugt wird. Die weitere Verbesserung der aktuellen Bahn basiert auf diesen Fehlerwerten und kann nur dann richtig funktionieren, wenn die Schichtfehler genau bekannt sind. Prinzipiell können die Fehler in der Schichtdicke sowohl mittels realer Messdaten als auch mit einer Simulation bestimmt werden. Wenn reale Messungen genutzt werden, steht eine geringe Anzahl an Iterationen im Vordergrund, während eine Simulation recht häufig ausgeführt werden

kann. Die Nutzung einer Simulation setzt jedoch eine Kalibrierung am realen Prozess voraus, um möglichst realitätsnahe Ergebnisse zu erzielen.

Die in Kapitel 3 vorgestellte Simulation ist prinzipiell in der Lage, die Schichtdicken mit hoher Präzision vorherzusagen. Dennoch basiert das Auftragsmodell auf einer Überlagerung mehrerer Gaußfunktionen, die zusammengenommen den Massefluss des Prozesses modellieren. Auch wenn die Präzision des Modells mit der Anzahl der Gaußfunktionen steigt, sind Simulationsmodelle, die zum Beispiel gemessene Footprintprofile verwenden, genauer. Die Simulation aus [173] und [172] nutzt diesen Ansatz und berücksichtigt noch einige weitere Effekte, die die Beschichtung beeinflussen, und nutzt fortgeschrittene GPGPU Techniken für schnelle Simulationsläufe. Eine Netzwerkschnittstelle und ein Netzwerkprotokoll zur Anbindung der Simulation an die Bahnoptimierung sind daher entwickelt worden, so dass die Simulation automatisch aufgerufen und die Ergebnisse für die Optimierung genutzt werden können.

Die hier entwickelte Bahnoptimierung nutzt ein nichtlineares konjugiertes Gradientenverfahren um die Bahn zu optimieren. Dazu ist die Bestimmung von partiellen Ableitungen der entsprechenden Zielfunktionsgrößen nach den veränderlichen Bahnparametern notwendig. Da der Fehler in dem Schichtauftrag eine der Zielfunktionsgrößen ist, muss auch diese Größe abgeleitet werden. Aus Sicht der Bahnoptimierung handelt es sich bei der externen Simulation jedoch um eine Blackbox, so dass die benötigten Ableitungen mit Hilfe von Finiten-Differenzen bestimmt werden müssten. Dies wiederum würde eine große Anzahl an Simulationsläufen benötigen, was selbst mit der schnellen Simulation nicht praktikabel wäre. Durch die Nutzung des analytischen Auftragsmodells aus Abschnitt 3.2.2 können jedoch geeignete Gradientenrichtungen analytisch bestimmt werden, indem das Auftragsmodell anhand der externen Simulation kalibriert und anschließend abgeleitet wird. Durch die Nutzung dieses neuartigen, hybriden Systems aus externer Simulation und analytischem Modell stehen sowohl exakte Fehlerwerte in der Beschichtung als auch geeignete und einfach zu berechnende Gradienten für die Optimierung zur Verfügung. Wie in der Evaluation dieses Kapitels gezeigt wird, ist die Güte der Gradienten vollkommen ausreichend, um sehr gute Ergebnisse zu erzielen.

## 5.4 Zielfunktion

Die Eingabe für das Optimierungsverfahren ist eine Bahn  $\boldsymbol{x} = (\boldsymbol{x}_0, \dots, \boldsymbol{x}_n)^\top$  und ein Werkstück, das als Dreiecksnetz mit den Knoten  $\boldsymbol{v}_i$ ,  $i = 0, \dots, m$ , repräsentiert wird. Außerdem ist für jeden Knoten des Netzes entweder eine Zielschichtdicke  $Z_i$  gegeben oder es wird ein zweites Netz zur Verfügung gestellt, das die Zielkontur repräsentiert, so dass hieraus die Werte für  $Z_i$  automatisch bestimmt werden können. Die Zielschichtdicke wird häufig konstant für alle Punkte der Oberfläche gewählt. In einigen Anwendungsfällen ist es jedoch nötig, unterschiedliche Zielschichtdicken zu definieren, zum Beispiel an Radien von komplexen Freiformflächen. Falls die roboterbasierte Variante des Verfahrens verwendet werden soll, müssen zusätzlich ein Roboter und ein Werkzeug gegeben sein.

Insbesondere müssen die Initialtransformation  $\mathbf{T}^r$  des Roboters und die Transformation  $\mathbf{T}^t$  vom Roboterflansch zum TCP sowie die Roboterkonfiguration  $\mathbf{T}_i(\Phi)$ , welche über die Denavit-Hartenberg Matrizen beschrieben werden kann, zur Verfügung stehen.

Das primäre Zielkriterium ist die Minimierung des Beschichtungsfehlers auf dem Werkstück. Der Beschichtungsfehler auf der Oberfläche kann mittels

$$E_l(\boldsymbol{\mathcal{X}}) = \frac{1}{A} \sum_{i=0}^m A_i \left( \frac{z_i(\boldsymbol{\mathcal{X}}) - Z_i}{Z_i} \right)^2 \quad (5.1)$$

bestimmt werden.  $E_l(\boldsymbol{\mathcal{X}})$  gibt demnach die flächengewichtete Summe der quadratischen Fehler in der Beschichtungsdicke an jedem Oberflächenpunkt an. Dieser berechnet sich aus der quadratischen Differenz der simulierten Schichtdicke  $z_i(\boldsymbol{\mathcal{X}})$  und der an dieser Stelle geforderten Zielschichtdicke  $Z_i$ , die beide bezüglich der Normalenrichtung der Oberfläche an den Knoten ausgedrückt werden, dividiert durch die quadratische Zielschichtdicke.

Der Beschichtungsfehler hängt primär von den Eigenschaften der Bahn  $\boldsymbol{\mathcal{X}}$  ab. Die Optimierung ist daher in der Lage, die Parameter der Bahn abhängig von den bestimmten Fehlern zu verändern. Dies schließt alle vorhandenen Parameter der Bahn mit ein, so dass sich als Parameterraum

$$\boldsymbol{\mathcal{X}} = (\boldsymbol{\chi}_0, \dots, \boldsymbol{\chi}_n)^\top, \quad \boldsymbol{\chi}_j = (\boldsymbol{\rho}_j, \hat{t}_j)^\top, \quad j = 0, \dots, n \quad (5.2)$$

mit

$$\boldsymbol{\rho}_j = (\mathbf{x}_j, \phi_j)^\top \quad (5.3)$$

oder

$$\boldsymbol{\rho}_j = \Phi_j \quad (5.4)$$

ergibt. Die Unterscheidung für die Definition von  $\boldsymbol{\rho}_j$  ergibt sich durch die beiden Möglichkeiten der Bahnrepräsentation mittels der Position und Ausrichtung der Spritzpistole oder über die Achsstellungen des Roboters. In beiden Fällen bietet  $\boldsymbol{\rho}_j$  für jeden Bahnpunkt genau sechs Freiheitsgrade, nämlich entweder drei für die Position und drei für die Ausrichtung der Spritzpistole oder jeweils einen für jeden der sechs Roboterachswinkel. Die Uneindeutigkeiten der Achsstellungen in der Roboterkonfiguration werden jeweils so aufgelöst, dass sich die Achsstellungen aufeinanderfolgender Bahnpunkte möglichst wenig verändern. Als weiterer Parameter dient die Spritzdauer an jedem Bahnpunkt, die indirekt auch die Geschwindigkeit der Spritzpistole beschreibt. Insgesamt ergeben sich demnach sieben Freiheitsgrade pro Bahnpunkt.

Eine Erweiterung des Parameterraums lässt sich dadurch erreichen, dass zusätzlich zur Bahnmodifikation auch das Werkstück verändert werden kann. Das primäre Ziel der Optimierung ist normalerweise das Erreichen der Zielschichtdicke an allen Punkten der Oberfläche des Werkstücks. Es kommt jedoch vor, dass nicht das Erzeugen der genauen



Schichtdicke, sondern das Erreichen einer vorgegebenen Endkontur entscheidend ist. In dem Fall kann die Schichtdicke in gewissen Grenzen variieren, solange sie gegebene Minimal- und Maximalwerte nicht überschreitet. Daher kann alternativ zur Erhöhung bzw. Verringerung der Schichtdicke an gewissen Stellen des Werkstücks auch die Werkstückgeometrie verändert werden, um die gewünschte Kontur zu erreichen. Dies ist in der Praxis jedoch nur dann möglich, wenn das Werkstück noch nicht fertiggestellt wurde und zum Beispiel der Fräsprozess zur Erzeugung des Werkstücks vor der Beschichtung noch angepasst werden kann.

Um diesen Freiheitsgrad auszunutzen, kann die Optimierung zusätzlich die Knoten des Werkstücknetzes beeinflussen. Dies geschieht, indem die Knoten entlang ihrer Normalenrichtung verschoben werden. Um die Verschiebung zu bestimmen, wird der Beschichtungsfehler aus Gleichung (5.1) betrachtet. Dabei wird ausgenutzt, dass die Schichtdicke als Summe von Beschichtung und Knotenverschiebung angesehen werden kann, so dass eine Verschiebung in Normalenrichtung zu einer Reduktion der benötigten Schichtdicke führt. Für die neue Schichtdicke ergibt sich damit der Zusammenhang

$$\bar{z}_i = z_i + v_i. \quad (5.5)$$

Hierbei ist  $z_i$  die Schichtdicke, die durch das Beschichten entsteht und  $v_i$  ist die Länge der Verschiebung des  $i$ -ten Knotens entlang seiner Normalen.

#### 5.4.1 Bahnbasierte Zielfunktionskomponenten

Bei der bahnbasierten Variante der Optimierung soll neben der Verringerung des Beschichtungsfehlers auch eine Bahnglättung durchgeführt werden. Glätten bedeutet in diesem Zusammenhang das Verringern der geometrischen Krümmung, der Beschleunigungen des Werkzeugs sowie die Verringerung der Winkelbeschleunigungen bei Umorientierungen des Werkzeugs. Durch die Optimierung dieser drei Parameter soll die Bahn soweit geglättet werden, dass auch der im realen Prozess genutzte Roboter keine Schwierigkeiten hat, die Bahn abzufahren. Auch wenn der genaue Zusammenhang zwischen der Bahn und den Roboterachsstellungen des Roboters bis auf die genannten Uneindeutigkeiten der Achskonfigurationen über die inverse Kinematik gegeben ist, sind die Auswirkungen bei Veränderungen der Bahn nicht intuitiv fassbar. In vielen Fällen werden durch die Glättung einer bereits von einem Roboter problemlos abfahrbaren Bahn keine neuen Probleme bezüglich der Ausführbarkeit der Bahn erzeugt. Jedoch muss ebenfalls angemerkt werden, dass auch geringe Änderungen an der Bahn, vor allem im Bereich der Orientierung des Werkzeugs, zu größeren Roboterachsbeschleunigungen führen können. Daher ist es sinnvoll, bei Nutzung dieses Verfahrens die Ausführbarkeit mit dem Roboter nach der Bahnmodifikation zu testen. Alternativ kann das unten beschriebene Verfahren genutzt werden, das die Beschleunigungen in den Roboterachsen anstelle der geometrischen Bahneigenschaften als Optimierungskriterien nutzt.

Die Kriterien, die im Folgenden berücksichtigt werden, sind die Beschleunigung des Endeffektors  $E_a(\mathcal{X})$ , die Winkelbeschleunigung bei der Umorientierung des Werkzeugs

$E_{aa}(\boldsymbol{\mathcal{X}})$  sowie die geometrische Bahnkrümmung  $E_c(\boldsymbol{\mathcal{X}})$ . Die Werte für diese Kriterien können über

$$E_a(\boldsymbol{\mathcal{X}}) = \frac{1}{n-1} \sum_{j=1}^{n-1} \|e_{a,j}(\boldsymbol{\mathcal{X}})\|^2 = \frac{1}{n-1} \sum_{j=1}^{n-1} \left\| \frac{\mathbf{x}_{j+1} - \mathbf{x}_j}{t_j} - \frac{\mathbf{x}_j - \mathbf{x}_{j-1}}{t_{j-1}} \right\|^2, \quad (5.6)$$

$$E_{aa}(\boldsymbol{\mathcal{X}}) = \frac{1}{n-1} \sum_{j=1}^{n-1} \left( \|e_{aa,q,j}(\boldsymbol{\mathcal{X}})\|^2 + \|e_{aa,r,j}(\boldsymbol{\mathcal{X}})\|^2 + \|e_{aa,s,j}(\boldsymbol{\mathcal{X}})\|^2 \right), \quad (5.7)$$

$$E_c(\boldsymbol{\mathcal{X}}) = \frac{1}{n-1} \sum_{j=1}^{n-1} \|e_{c,j}(\boldsymbol{\mathcal{X}})\|^2 = \frac{1}{n-1} \sum_{j=1}^{n-1} \left\| \frac{\mathbf{x}_{j+1} - \mathbf{x}_j}{\|\mathbf{x}_{j+1} - \mathbf{x}_j\|} - \frac{\mathbf{x}_j - \mathbf{x}_{j-1}}{\|\mathbf{x}_j - \mathbf{x}_{j-1}\|} \right\|^2, \quad (5.8)$$

mit

$$e_{aa,q,j}(\boldsymbol{\mathcal{X}}) = \frac{\mathbf{q}_{j+1} - \mathbf{q}_j}{t_j} - \frac{\mathbf{q}_j - \mathbf{q}_{j-1}}{t_{j-1}}, \quad (5.9)$$

und äquivalent für  $e_{aa,r,j}(\boldsymbol{\mathcal{X}})$  und  $e_{aa,s,j}(\boldsymbol{\mathcal{X}})$  ausgerechnet werden. Hierin sind  $\mathbf{x}_j$  sowie  $\mathbf{q}_j$ ,  $\mathbf{r}_j$ , und  $\mathbf{s}_j$ , die Position sowie die aufspannenden Vektoren der Pose  $\boldsymbol{\rho}_j$ ,  $j = 0, \dots, n$ , und  $t_j$  die Zeit, die das Werkzeug benötigt, um von der Pose  $\boldsymbol{\rho}_j$  zur Pose  $\boldsymbol{\rho}_{j+1}$  zu gelangen.

In Gleichung (5.6) wird anstelle der Beschleunigung die Differenz der Geschwindigkeiten bestimmt. Dies ist eine Vereinfachung, die die Komplexität der Ableitungen der Funktion deutlich reduziert, ohne die Kernaussage des Kriteriums zu stark zu verfälschen. Die gleiche Vorgehensweise wurde auch bei den anderen Kriterien genutzt.

Da die Gesamtziel­funktion aus einer gewichteten Summe aller Zielfunktionskomponenten besteht, werden die einzelnen Kriterien entsprechend zusammengefasst zu

$$E_m(\boldsymbol{\mathcal{X}}) = \lambda_a E_a(\boldsymbol{\mathcal{X}}) + \lambda_{aa} E_{aa}(\boldsymbol{\mathcal{X}}) + \lambda_c E_c(\boldsymbol{\mathcal{X}}). \quad (5.10)$$

#### 5.4.2 Roboterbasierte Zielfunktionskomponenten

Die Alternative zu den bahnbasierten Kriterien, die für eine Glättung der Bahn sorgen sollen, ist, den Roboter direkt zu nutzen. Die Glattheit der Bahn ist nur ein indirektes Indiz für die Ausführbarkeit mit einem Roboter. Die tatsächliche Ausführbarkeit hängt in erster Linie von den Achsgeschwindigkeiten und -beschleunigungen ab. Daher kann die Roboterachsbeschleunigung direkt als Zielfunktionskomponente verwendet werden. Die Gleichung zur Bestimmung der Roboterachsbeschleunigungen wird mit

$$\begin{aligned} E_m(\boldsymbol{\mathcal{X}}) &= \frac{1}{6(n-1)} \sum_{j=1}^{n-1} \sum_{k=1}^6 (e_{aa,k,j}(\boldsymbol{\mathcal{X}}))^2 \\ &= \frac{1}{6(n-1)} \sum_{j=1}^{n-1} \sum_{k=1}^6 \left( \frac{\Phi_{j+1,k} - \Phi_{j,k}}{t_j t_{j-1}} - \frac{\Phi_{j,k} - \Phi_{j-1,k}}{t_{j-1}^2} \right)^2 \end{aligned} \quad (5.11)$$

angegeben. Hierbei ist  $\Phi_{j,k}$  der Winkel der  $k$ -ten Achse beim Erreichen des  $j$ -ten Bahnpunkts. Eine Reduktion des Wertes  $E_m(\boldsymbol{x})$  reduziert die Achsbeschleunigungen, was zu einer ruhigeren Bewegung des Roboters führt. Der Einfluss auf die Bahn ist dabei über die Vorwärtskinematik (Abschnitt 2.3.2) gegeben.

### 5.4.3 Werkstückbasierte Zielfunktionskomponenten

Für den Fall, dass auch das Werkstück modifiziert werden kann, muss sichergestellt werden, dass durch die Verschiebung der Werkstückknoten die Beschichtung nicht zu gering wird. Das Verändern der Werkstückgeometrie soll ausschließlich zur Unterstützung der Bahnmodifikationen dienen, wenn durch die Bahnveränderungen allein an einigen Stellen die Zielkontur nicht erreicht werden kann. Daher werden zwei Kriterien eingeführt, die dafür sorgen, dass das Werkstück nicht übermäßig verändert wird. Zum einen wird dafür gesorgt, dass das Werkstück seine originale Form so gut wie möglich beibehält, zum anderen werden die Unterschiede in den Veränderungen benachbarter Knoten gering gehalten. Dies wird mittels der beiden Gleichungen

$$E_v(\boldsymbol{x}) = \frac{1}{m+1} \sum_{i=0}^m v_i^2 \quad (5.12)$$

und

$$E_{vc}(\boldsymbol{x}) = \frac{1}{m+1} \sum_{i=0}^m \sum_{j \in N_1(i)} (v_i - v_j)^2 \quad (5.13)$$

erreicht, wobei  $v_i$  die Länge der Verschiebung des  $i$ -ten Knotens entlang der Normalen beschreibt. Die erste Gleichung summiert die Abweichungen vom originalen Werkstück pro Knoten, während die zweite ungleichmäßige Änderungen in der Nachbarschaft der Knoten beschreibt, wobei die Nachbarschaft  $N_1(i)$  die Knoten des 1-Ringes um den Knoten mit Index  $i$  bezeichnet. Dies soll sicherstellen, dass lokal keine allzu großen Unterschiede entstehen, die bei der Herstellung des Werkstücks nicht exakt eingehalten werden können.

### 5.4.4 Gesamtzielfunktion

Die oben beschriebenen Zielfunktionskomponenten können in einer Gesamtzielfunktion zusammengefasst werden. Dazu werden alle Zielfunktionskomponenten gewichtet aufsummiert. Es ergibt sich

$$E(\boldsymbol{x}) = E_l(\boldsymbol{x}) + \lambda (E_m(\boldsymbol{x}) + \lambda_v E_v(\boldsymbol{x}) + \lambda_{vc} E_{vc}(\boldsymbol{x})). \quad (5.14)$$

Die letzten beiden Komponenten sind dabei optional, da die Knotenverschiebung nicht zwingend ausgeführt wird.

### 5.4.5 Einschränkungen

Die Bahnoptimierung basiert auf der Zielfunktion  $E(\mathcal{X})$ . Diese wiederum besteht aus einer Summe von Zielfunktionskomponenten, die bestimmte Kriterien beschreiben, die es zu optimieren gilt. Prinzipiell kann die Zielfunktion noch erweitert werden, indem neue Komponenten hinzugefügt werden. Zwei Komponenten, die bewusst nicht integriert wurden, sind die Roboterdynamik und die Wärmevarianz. Ersteres ist ein sehr kompliziertes Problem, da die Dynamik des Roboters von vielen Faktoren, wie der Arbeitsweise der Gelenkmotoren, dem Gewicht des Werkzeugs und des Schlauchpaketes sowie dessen Montage abhängt. Das Optimierungsmodul ist jedoch darauf ausgelegt, dass dieses Verhalten in einer externen Simulation bestimmt werden kann und die Ergebnisse, ähnlich wie die der Beschichtungssimulation, in der Optimierung genutzt werden können, siehe [155, 85]. Die Wärmevarianz wurde an dieser Stelle nicht weiter beachtet, da sie durch die Optimierung kaum beeinflusst werden kann. Die Wärmevarianz kann entweder durch starke Veränderungen in der Bahnstruktur verbessert werden, was durch die Initialbahnplanung und nicht durch die Bahnoptimierung geleistet werden soll oder durch eine starke Erhöhung der Bahngeschwindigkeit, was zu Problemen mit der Roboterdynamik führt.

## 5.5 Nichtlineare konjugierte Gradienten

Das Optimierungsproblem wird mittels der Zielfunktion  $E(\mathcal{X})$  aus Gleichung (5.14) beschrieben, die die gewichtete Summe der einzelnen Teilziele beschreibt. Auf Grund der großen Menge an Freiheitsgraden sind ableitungsfreie Optimierungsmethoden wie das Nelder-Mead Verfahren [129] nicht geeignet. Zur Optimierung aller Zielkomponenten wird daher ein nichtlineares konjugiertes Gradientenverfahren genutzt. Dieses setzt das Vorhandensein von Gradienten der Zielfunktion bezüglich der Optimierungsparameter, welche in Abschnitt 5.4 beschrieben wurden, voraus. Während die für die Bildung der Gradienten notwendigen partiellen Ableitungen der bahn- und roboterbezogenen Zielfunktionskomponenten direkt gebildet werden können, ist eine Ableitung von  $E_l(\mathcal{X})$  auf Grund der Nutzung einer externen Simulation (oder realer Messwerte) nicht direkt möglich. Eine Alternative zur Bestimmung der Gradienten wäre die Nutzung finiter Differenzen durch häufige Simulation mit jeweils geringer Modifikation der Bahn an einem einzelnen Freiheitsgrad. Da es jedoch sehr viele Freiheitsgrade gibt (sieben pro Bahnpunkt und meist tausende Bahnpunkte), ist dieses Verfahren nicht praktikabel. Auch die ausschließliche Nutzung eines einfacheren Ersatzmodells ist auf Grund des zu erwartenden Qualitätsverlustes nicht sinnvoll. Daher soll ein neuartiges Verfahren, das beide Möglichkeiten vereint, verwendet werden. Die externe Simulation wird genutzt, um die Schichtabweichungen exakt vorhersagen zu können. Die Gradienten werden jedoch gebildet, indem diese aus einem an dieser Simulation kalibrierten, analytischen Ersatzmodell bestimmt werden. Als Modell soll das in Abschnitt 3.2.2 beschriebene

Auftragsmodell genutzt werden, wobei die Parameter des Modells mittels des in Abschnitt 3.2.3 beschriebenen Verfahrens an die externe Simulation angepasst werden.

### 5.5.1 Richtungen

Zur Optimierung wird das nichtlineare konjugierte Gradientenverfahren genutzt, welches die Gradienten der Zielfunktion verwendet, um die Veränderungsrichtung für die freien Parameter der Bahn zu bestimmen. Dies bedeutet, dass in jedem Schritt zunächst die Gradienten  $\nabla E(\boldsymbol{\mathcal{X}}_k)$  der Zielfunktion bezüglich aller Freiheitsgrade bestimmt werden müssen.  $\boldsymbol{\mathcal{X}}_k$  beschreibt hierbei die Bahn im  $k$ -ten Optimierungsschritt. Das Bilden der Gradienten erzeugt einen Gradientenvektor, der alle partiellen Ableitungen bezüglich der Freiheitsgrade beinhaltet. Dieser Vektor zeigt in die Richtung des steilsten Anstieges der Zielfunktion bezüglich der Freiheitsgrade. Das Verfolgen der negativen Richtung des Vektors reduziert den Wert von  $E(\boldsymbol{\mathcal{X}}_k)$  und verbessert demnach die Zielfunktion.

Das nichtlineare konjugierte Gradientenverfahren unterscheidet sich vom einfachen Gradientenabstieg, indem Gradientenrichtungen aus vorherigen Schritten zusätzlich in Betracht gezogen werden. Dies ermöglicht die schnellere Konvergenz in ein möglicherweise lokales Optimum. Sei  $\mathbf{d}_k$  die zu ermittelnde Suchrichtung und  $\mathbf{g}_k = -\nabla E(\boldsymbol{\mathcal{X}}_k)$  die aktuelle negative Gradientenrichtung, dann wird die neue Suchrichtung mittels [145, 77, 128, 130]

$$\mathbf{d}_k = \mathbf{g}_k + \beta_k \mathbf{d}_{k-1} \quad (5.15)$$

beschrieben, wobei  $\mathbf{d}_0 = \mathbf{0}$  ist.

Es gibt verschiedene Möglichkeiten zur Auswahl von  $\beta_k$ . Einige bekannte Varianten sind [76, 77, 177, 69]

$$\beta_k^{\text{FR}} = \frac{\mathbf{g}_k^\top \mathbf{g}_k}{\mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}}, \quad (5.16)$$

$$\beta_k^{\text{PRP}} = \frac{\mathbf{g}_k^\top \mathbf{y}_k}{\mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}}, \quad (5.17)$$

$$\beta_k^{\text{HS}} = -\frac{\mathbf{g}_k^\top \mathbf{y}_k}{\mathbf{d}_{k-1}^\top \mathbf{y}_k}, \quad (5.18)$$

$$\beta_k^{\text{N}} = \left( \mathbf{y}_k - 2\mathbf{d}_{k-1} \frac{\mathbf{y}_k^\top \mathbf{y}_k}{\mathbf{d}_{k-1}^\top \mathbf{y}_k} \right)^\top \frac{\mathbf{g}_k}{\mathbf{d}_{k-1}^\top \mathbf{y}_k}, \quad (5.19)$$

mit  $\mathbf{y}_k = \mathbf{g}_k - \mathbf{g}_{k-1}$ . Eine andere Variante mit

$$\mathbf{d}_k = \mathbf{g}_k + \beta_k^{\text{MPRP}} \mathbf{d}_{k-1} - \xi_k \mathbf{y}_k \quad (5.20)$$

und

$$\beta_k^{\text{MPRP}} = \frac{\mathbf{g}_k^\top \mathbf{y}_k}{\mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}}, \quad (5.21)$$

$$\xi_k = \frac{\mathbf{g}_k^\top \mathbf{d}_{k-1}}{\mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}} \quad (5.22)$$

ist ebenfalls vorgeschlagen worden [178]. Bei den Varianten  $\beta_k^{\text{PRP}}$  und  $\beta_k^{\text{MPRP}}$  werden die Werte für  $\beta_k$  auf positive Zahlen beschränkt. Sollte  $\beta_k$  negativ werden, wird es auf Null gesetzt, womit ein Neustart des Verfahrens durchgeführt wird. Bei dem Verfahren von *Hager* wird ein Neustart dynamisch über die Bedingung

$$\bar{\beta}_k^{\text{N}} = \max\{\beta_k^{\text{N}}, \nu_k\}, \quad (5.23)$$

$$\nu_k = \frac{-1}{\mathbf{d}_{k-1}^\top \mathbf{d}_{k-1} \min\{\nu, \mathbf{g}_{k-1}^\top \mathbf{g}_{k-1}\}} \quad (5.24)$$

gesteuert, wobei  $\nu$  eine kleine Konstante ist [76].

Die Verfahren haben jeweils unterschiedliche Eigenschaften bezüglich des Konvergenzverhaltens, besonders wenn die optimale Schrittweite (siehe Abschnitt 5.5.2) nicht exakt bestimmt werden kann, wie im Kontext dieser Arbeit. Das Verfahren von *Hager* bietet allerdings auch in diesem Fall noch einige gute Eigenschaften [77]. Grundsätzlich bieten jedoch alle Varianten eine schnellere Konvergenz im Vergleich zum einfachen Gradientenabstieg. Im nächsten Abschnitt wird ein Verfahren zur Schrittweitenbestimmung vorgestellt, das mit nur zwei Simulationsschritten auskommt.

## 5.5.2 Schrittweite

Die Optimierungsschrittweite für nichtlineare konjugierte Gradientenverfahren wird meist mittels eines Line-Search Algorithmus bestimmt (siehe [162]). Dieser sucht in der gegebenen Suchrichtung die Schrittweite, so dass die Zielfunktion auf dieser Linie minimal wird. Die optimale Schrittweite hat die zusätzliche Eigenschaft, dass der Gradient an der Stelle auf der Linie in  $90^\circ$  zu der aktuellen Suchrichtung liegt.

Das Suchen der optimalen Schrittweite kann mittels der folgenden zu minimierenden Funktion beschrieben werden:

$$\min_{\mu} E(\mathcal{X} - \mu \mathbf{d}_k). \quad (5.25)$$

Die oben genannte Bedingung der senkrecht auf der aktuellen Suchrichtung stehenden Gradienten kann dann ausgenutzt werden mittels

$$f_k(\mu) = \nabla_{\mu} E(\mathcal{X}_k - \mu \cdot \mathbf{d}_k) = -\nabla E(\mathcal{X}_k - \mu \cdot \mathbf{d}_k) \cdot \mathbf{d}_k \stackrel{!}{=} 0, \quad (5.26)$$

wobei  $\nabla_{\mu}$  die partielle Ableitung nach  $\mu$  ausdrückt.

Die Suche nach einem optimalen  $\mu$  wird häufig durch eine Menge von Auswertungen an verschiedenen Stellen der Linie durchgeführt. Um die Anzahl benötigter Simulationsläufe zu reduzieren, wird hier, im Gegensatz zu komplexeren Methoden (siehe [162]), ein Verfahren angewendet, das mit nur zwei Simulationsläufen auskommt. Dazu wird folgender Ablauf genutzt, der  $\mu$  durch eine lineare Abschätzung bestimmt und den Gradienten von  $E(\boldsymbol{\mathcal{X}}_k)$  durch den Gradienten  $\nabla \tilde{E}(\boldsymbol{\mathcal{X}}_k)$  ersetzt, der die analytische Ersatzfunktion nutzt. Dieser Ersatz wird auf Grund der Nutzung der externen Simulation benötigt, die keine Gradienten liefert, und wird in Abschnitt 5.6 genauer beschrieben. Das Verfahren ist dann:

1. Setze  $\Delta\mu$  auf eine kleine Konstante.
2. Bestimme  $f_k(0) = -\nabla \tilde{E}(\boldsymbol{\mathcal{X}}_k) \cdot \mathbf{d}_k$ .
3. Bestimme  $f_k(\Delta\mu) = -\nabla \tilde{E}(\boldsymbol{\mathcal{X}}_k - \Delta\mu \cdot \mathbf{d}_k) \cdot \mathbf{d}_k$ .
4. Löse die Gleichung  $f_k(\Delta\mu) + \mu \cdot \frac{f_k(\Delta\mu) - f_k(0)}{\Delta\mu} = 0$  bezüglich  $\mu$ .

Damit sind sowohl die Suchrichtung als auch die Schrittweite bestimmt und das Verfahren kann iterativ angewendet werden. Da das Verfahren auf die Ableitungen der Zielfunktion  $E(\boldsymbol{\mathcal{X}})$  nach den zu verändernden Parametern angewiesen ist, werden diese in den nächsten Abschnitten vorgestellt.

## 5.6 Optimierung

Im oben vorgestellten nichtlinearen konjugierten Gradientenverfahren werden die Gradienten  $\nabla \tilde{E}(\boldsymbol{\mathcal{X}}_k)$  der Zielfunktion  $E(\boldsymbol{\mathcal{X}}_k)$  benötigt, um diese zu minimieren. Die Gradienten werden je Freiheitsgrad bestimmt und ergeben zusammengenommen einen Gradientenvektor, der die Suchrichtungen vorgibt. Für die ursprüngliche Zielfunktionskomponente  $E_l(\boldsymbol{\mathcal{X}})$ , welche über

$$E_l(\boldsymbol{\mathcal{X}}) = \frac{1}{A} \sum_{i=0}^m A_i \left( \frac{z_i(\boldsymbol{\mathcal{X}}) - Z_i}{Z_i} \right)^2 \quad (5.27)$$

definiert ist, ergibt sich demnach eine Ableitung nach einer der Komponenten durch

$$\nabla_p E_l(\boldsymbol{\mathcal{X}}) = \frac{1}{A} \sum_{i=0}^m \frac{2A_i}{Z_i^2} \cdot \Delta z_i \cdot \nabla_p z_i \quad (5.28)$$

mit  $\Delta z_i = z_i - Z_i$ . Hierbei steht  $\nabla_p$  für eine partielle Ableitung nach einem Parameter  $p$ . In Frage kommende Parameter sind der Ort  $\mathbf{x}_j$ , die Orientierung  $\phi_j$ , die Roboterachsstellungen  $\Phi_j$  sowie die Zeitdifferenz zwischen zwei Bahnpunkten  $t_j$ . Durch die Nutzung des analytischen Modells ändert sich diese Formel zu

$$\nabla_p \tilde{E}_l(\boldsymbol{\mathcal{X}}) = \frac{1}{A} \sum_{i=0}^m \frac{2A_i}{Z_i^2} \cdot \Delta z_i \cdot \nabla_p \tilde{z}_i, \quad (5.29)$$

wobei  $\tilde{z}_i$  die Beschichtung ermittelt über das analytische Ersatzmodell aus Gleichung (3.1) angibt. Man beachte jedoch, dass sich der Wert  $\Delta z_i$  weiterhin aus den Ergebnissen der externen Simulation ergibt.

### 5.6.1 Bahnbasierte Optimierung

Im Falle der bahnbasierten Optimierung wird die Bahn mittels der Position und der Ausrichtung der Spritzpistole parametrisiert, siehe Abschnitt 2.1. Die Parameter für die Optimierung sind demnach  $\mathbf{x}_j$ ,  $\phi_j$  und  $t_j$ , so dass die Position der Bahnpunkte im Raum verschoben, die Spritzrichtung verändert und die Geschwindigkeiten auf der Bahn modifiziert werden können. Durch Bestimmung der Gradienten der Zielfunktion  $E_l(\mathcal{X})$  nach diesen Parametern und Anwendung des nichtlinearen konjugierten Gradientenverfahrens verringert sich der Fehler in der Beschichtungsstärke. Um gleichzeitig dafür zu sorgen, dass die Bahn für einen Roboter ausführbar bleibt, soll auch die Funktion  $E_m(\mathcal{X})$  aus Abschnitt 5.4.1 optimiert werden. Die entsprechenden Ableitungen werden in Abschnitt A.1 angegeben.

### 5.6.2 Roboterbasierte Optimierung

Bei der roboterbasierten Optimierung wird die Bahn direkt über die Achsstellungen des Roboters parametrisiert, so dass die Optimierung direkt auf diesen Werten arbeiten kann. Die Parameter zur Ableitung sind demnach  $\Phi_j$  und  $t_j$ . Wie beim bahnbasierten Verfahren müssen alle Gradienten der Funktionen  $E_l(\mathcal{X})$  und  $E_m(\mathcal{X})$  nach diesen Parametern bestimmt werden. Die Gradienten nach  $E_l(\mathcal{X})$  können für die Verbesserung der Beschichtung genutzt werden, während die Gradienten nach  $E_m(\mathcal{X})$  zur Reduktion der Roboterachsbeschleunigungen verwendet werden können. Die Ausführbarkeit der Bahn ist demnach durch die Reduktion der Achsbeschleunigungen direkt gegeben. Die entsprechenden Ableitungen werden in Abschnitt A.2 angegeben.

### 5.6.3 Werkstückbasierte Optimierung

Wenn zusätzlich zu den Bahnmodifikationen auch das Werkstück verändert werden soll, müssen die Gradienten der Zielfunktion bezüglich der Knotenverschiebung bestimmt werden. Entsprechend wird die Funktion  $E_l(\mathcal{X})$  nach der Verschiebung abgeleitet. Außerdem sollen die Bedingungen  $E_v(\mathcal{X})$  und  $E_{vc}(\mathcal{X})$  aus Abschnitt 5.4.3 berücksichtigt werden. Die entsprechenden Ableitungen werden in Abschnitt A.3 angegeben.

## 5.7 Evaluation

In diesem Abschnitt wird die Bahnoptimierung evaluiert. Dazu werden einerseits die verschiedenen Bestandteile des Algorithmus und andererseits der Einfluss der dazugehörigen



Parameter untersucht. Als Erstes werden die beiden Varianten des Algorithmus mit und ohne Nutzung des Roboters verglichen und danach die zur Verfügung stehenden nichtlinearen konjugierten Gradientenfunktionen. Anschließend werden die Einflüsse verschiedener Parameter evaluiert, wie zum Beispiel die Glättungsparameter, die Modellparameter oder die Parameter der Eingabebahn. Abschließend werden die Anwendungsbereiche der Bahnoptimierung mittels einiger Fallbeispiele vorgestellt.

### 5.7.1 Testdaten und Qualitätsmerkmale

Die Evaluation soll auf einem komplexen Werkstück durchgeführt werden, um die Stärken und Schwächen des Verfahrens aufzuzeigen. Dazu wird, wie bei der Evaluation der Initialbahnplanung, erneut auf das Werkstück B zurückgegriffen. Dieses ist inklusive des Region of Interest in Abbildung 4.31 dargestellt. Das Werkstück bietet sich an, da die vorhandenen Radien sehr klein im Vergleich zu dem eher großen Footprint des genutzten Libo-Prozesses sind und es demnach sehr schwierig ist, die Oberfläche an den Stellen der kleinen Radien exakt zu beschichten. Daher wird das genannte Werkstück verwendet, um durch die Evaluation auch die Grenzen des Verfahrens zu ermitteln.

Die Zielfunktion setzt sich aus den Komponenten zusammen, die in Abschnitt 5.4 vorgestellt wurden. Zur Bestimmung des Beschichtungsfehlers wird demnach  $E_l(\boldsymbol{\mathcal{X}})$  verwendet. Für das bahnbasierte Modell werden zusätzlich die Zielfunktionskomponenten  $E_a(\boldsymbol{\mathcal{X}})$ ,  $E_{aa}(\boldsymbol{\mathcal{X}})$  und  $E_c(\boldsymbol{\mathcal{X}})$  genutzt und für das roboterbasierte Verfahren die Komponente  $E_m(\boldsymbol{\mathcal{X}})$ . In beiden Fällen ist es möglich, die werkstückbasierten Komponenten  $E_v(\boldsymbol{\mathcal{X}})$  und  $E_{vc}(\boldsymbol{\mathcal{X}})$  zu nutzen.

Die Bahnoptimierung ist ein iteratives Verfahren, so dass die Zielfunktionswerte durch sukzessive Schritte in die durch die nichtlinearen konjugierten Gradienten bestimmten Richtungen reduziert werden. Auf Grund der Schrittweitenbestimmung ist es pro Iteration notwendig, die Simulation zweimal auszuführen. Im Folgenden wird als Iteration ein einzelner Simulationslauf bezeichnet. Dies erklärt die in den Graphen der Zielfunktionsverläufe zu sehenden Stufen, da zunächst die Schrittweite mittels kleiner Testschritte bestimmt wird ( $\Delta\mu$  in Abschnitt 5.5.2).

Die Bahnoptimierung soll in erster Linie als Nachverarbeitungsschritt der Initialbahnplanung genutzt werden. Da die thermischen Varianzen in diesem Schritt nicht mehr optimiert werden, sollen diese zumindest nicht verschlechtert werden. Deswegen wird das Verfahren so parametrisiert, dass Geschwindigkeitsänderungen auf der Bahn nur begrenzt durchgeführt werden. Diese haben deutlich stärkeren Einfluss auf die thermischen Varianzen als Änderungen an den Positionen der Bahnpunkte oder der Spritzwinkel, so dass die letztgenannten Änderungen favorisiert durchgeführt werden sollen.

### 5.7.2 Vergleich der Verfahren und Gradientenfunktionen

Für die Bahnoptimierung stehen verschiedene Varianten des nichtlinearen konjugierten Gradientenverfahrens zur Verfügung. Diese werden untereinander verglichen, um zu evaluieren, welches Verfahren im Kontext der Bahnoptimierung die besten Ergebnisse erzielt. Zum einen werden hierzu die bahnbasierte und die roboterbasierte Variante des Verfahrens betrachtet, zum anderen werden die unterschiedlichen Gradientenfunktionen verglichen.

In Abbildung 5.2 sind die Optimierungsverläufe für verschiedene Testszenarien dargestellt. In den oberen beiden Grafiken sind die Zielfunktionsverläufe der roboterbasierten Variante des Algorithmus für alle in Abschnitt 5.5.1 vorgestellten Gradientenfunktionen abgebildet und in den unteren beiden Darstellungen die für die bahnbasierte Variante. Zusätzlich zur Gesamtzielfunktion in den linken Abbildungen sind auch die Zielfunktionskomponenten für  $E_l(\mathcal{X})$  in den rechten Abbildungen aufgeführt.

Die Gradientenfunktionen für die roboterbasierte Variante des Verfahrens bilden zwei Gruppen mit ähnlichem Verhalten. Die Funktionen *Fletcher Reeves*, *Polak Ribière Polyak* und *Hager* schneiden sowohl in der Gesamtzielfunktion als auch in der Zielfunktionskomponente für die Beschichtung ( $E_l(\mathcal{X})$ ) deutlich besser ab als die anderen beiden Funktionen, sind jedoch untereinander sehr ähnlich. Die Methode von *Fletcher Reeves* erzielt den geringsten Zielfunktionswert in der Gesamtzielfunktion und die Methode von *Hager* erzeugt die beste Schichtqualität. Auf Grund der geringen Unterschiede können daher problemlos alle drei Verfahren angewendet werden.

Verglichen mit den Verläufen für die roboterbasierte Variante des Algorithmus zeigen die Graphen für die bahnbasierte Variante ein etwas homogeneres Bild. Alle Gradientenfunktionen erzeugen ähnliche Ergebnisse sowohl bei der Betrachtung der Gesamtzielfunktion als auch bei der Beschichtung. Wieder zeigen die Funktionen *Polak Ribière Polyak* und *Hager* die besten Ergebnisse, können sich jedoch nicht wesentlich von den anderen Funktionen absetzen. Die Graphen für die Beschichtung zeigen einen nicht monotonen Optimierungsverlauf. Dies liegt an dem relativ instabilen Verhalten der Optimierung mit den bahnbasierten Zielfunktionen. Einige Parametrisierungen führen hier zur Divergenz des Verfahrens oder zu einem frühen Erreichen eines Plateaus, das die Optimierung beinahe vollständig ausbremst. Hier könnte auch die nicht optimale Schrittweitenbestimmung ein Problem darstellen, so dass keine wesentliche Verbesserung in einem oder mehreren aufeinanderfolgenden Schritten erzielt werden kann.

Insgesamt zeigen die meisten Verfahren ein recht ähnliches Verhalten. Im Bereich der roboterbasierten Optimierung haben sich drei Funktionen als gleichwertig herausgestellt. In der bahnbasierten Optimierung zeigen alle Funktionen ein recht homogenes Verhalten. Da die Optimierung mit den bahnbasierten Zielfunktionen bei Nutzung ungeeigneter Parameter dazu neigt zu divergieren oder zu stagnieren, wird insgesamt das roboterbasierte Verfahren empfohlen. Dies hat auch den Vorteil, dass die Zielfunktion betrachtet wird, die eine direkte Aussage über die Ausführbarkeit einer Bahn auf dem echten Roboter liefert. Außerdem wird die Methode von *Hager* empfohlen, da diese für beide Varianten

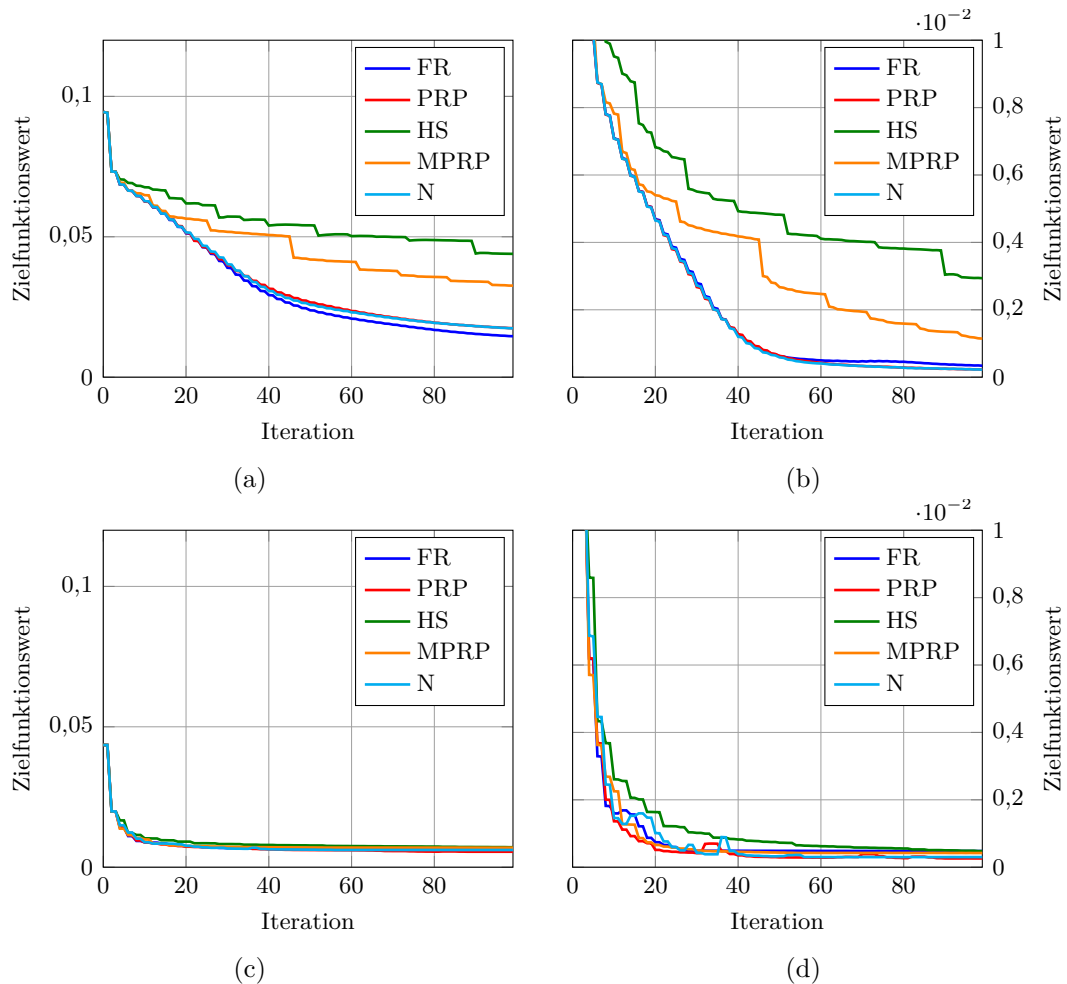


Abbildung 5.2: Zielfunktionsverlauf bei der Bahnoptimierung mit verschiedenen Verfahren. (a) und (b) geben die Gesamtzielfunktion respektive die Zielfunktionskomponente für den Beschichtungsfehler ( $E_l(\mathcal{X})$ ) für das roboterbasierte Verfahren und (c) und (d) die entsprechenden Werte für das bahnbasierte Verfahren für die unterschiedlichen Gradientenfunktionen an. Die Abkürzungen stehen für: Fletcher Reeves („FR“), Polak Ribière Polyak („PRP“), Hestenes Stiefel („HS“), Modified Polak Ribière Polyak („MPRP“), Hager („N“).

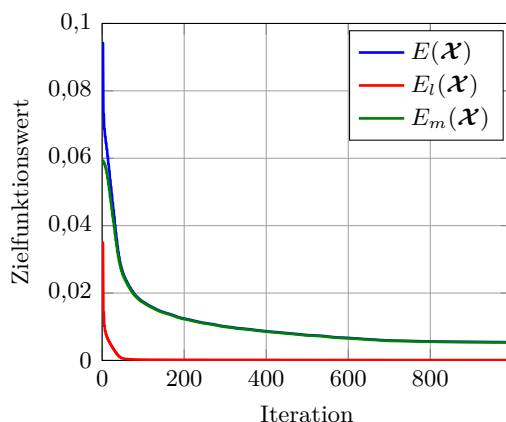


Abbildung 5.3: Zielfunktionsverlauf bei der Bahnoptimierung auf Werkstück B mit der roboterbasierten Variante und der Methode von Hager. Gezeigt sind die Zielfunktionsverläufe für den summierten Fehler ( $E(\mathcal{X})$ ), den Beschichtungsfehler ( $E_l(\mathcal{X})$ ) und die Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ).

des Algorithmus sehr stabil ist und auch in beiden Fällen sehr gute Ergebnisse im Vergleich zu den anderen Funktionen erzielt.

### 5.7.3 Anzahl der Iterationen

Die Bahnoptimierung ist ein iteratives Verfahren, so dass die Anzahl der Iterationen die Qualität der Ergebnisse bestimmt. Theoretisch wird das Optimum bei Nutzung von nichtlinearen konjugierten Gradienten mit exakter Schrittweitenbestimmung für quadratische Funktionen nach  $n$  Iterationen erreicht, wenn  $n$  die Anzahl der zu optimierenden Variablen darstellt [42, 139]. Da die Zielfunktion jedoch nicht rein quadratisch ist und die Schrittweite nur approximiert wird, ist zu erwarten, dass das Verfahren länger bis zum Erreichen des Optimums benötigt.

In Abbildung 5.3 sind die Zielfunktionsverläufe für die Bahnoptimierung auf einer initialgeplanten Bahn mit dem Verfahren von Hager und der roboterbasierten Variante für 1000 Iterationen dargestellt. Wie gut zu erkennen ist, wird der größte Fortschritt zu Beginn des Verfahrens erzielt. Im weiteren Verlauf werden die Fehler jedoch immer geringer, so dass auch nach 1000 Iterationen immer noch kleine Verbesserungen erzielt werden. Dies gilt sowohl für die Roboterachsbeschleunigungen als auch für die Beschichtungsfehler. Demnach ist es durchaus sinnvoll, die Optimierung länger laufen zu lassen, wenn die Zeit hierfür zur Verfügung steht.

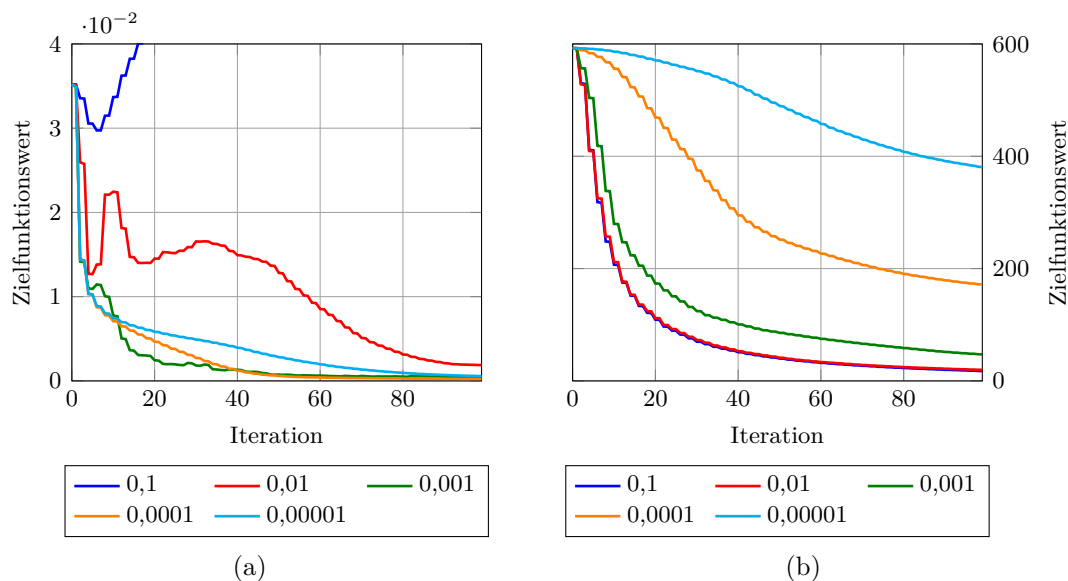


Abbildung 5.4: Bahnoptimierung mit unterschiedlichen Werten für  $\lambda$ . In (a) sind die Zielfunktionswerte für den Beschichtungsfehler ( $E_l(\mathcal{X})$ ) dargestellt und in (b) sind die Zielfunktionswerte für die Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ) skaliert um den entsprechenden Wert für  $\lambda$  abgebildet.

#### 5.7.4 Einfluss der Parameter

Ein wesentlicher Parameter des Verfahrens ist das Gewicht  $\lambda$ , das für die Bahnglättung genutzt wird. Dieser Parameter bestimmt, wie hoch die Bahnglättung gegenüber der Schichtqualität gewichtet werden soll, siehe Gleichung (5.14). Ein hoher Wert für  $\lambda$  sorgt für eine Fokussierung auf die Bahnglättung zum Nachteil der Reduktion der Beschichtungsfehler. Daher sollte das Gewicht mit Bedacht gewählt werden, damit sowohl die Bahnglättung als auch die Beschichtungsqualität gleichermaßen berücksichtigt werden.

In Abbildung 5.4 sind beispielhaft die Zielfunktionsverläufe für die Bahnoptimierung mit Nutzung unterschiedlicher Werte für  $\lambda$  dargestellt. Dabei werden sowohl die Roboterachsbeschleunigungen als auch die Beschichtungsfehler abgebildet. Man erkennt sehr gut, dass mit zunehmendem Wert für  $\lambda$  die Roboterachsbeschleunigungen stärker optimiert werden. Gleichzeitig tritt die Optimierung der Beschichtung mit höherem Wert von  $\lambda$  immer weiter in den Hintergrund. Bei zu hohem Gewicht für die Glättung kann dies soweit gehen, dass der Beschichtungsfehler nicht mehr verkleinert, sondern zunächst immer weiter vergrößert wird. Dieses Verhalten ist zum Beispiel bei der Optimierung mit  $\lambda = 0,1$  zu erkennen. Hier wird die Beschichtung im Verlauf der Optimierung zunächst verschlechtert, da der Beschichtungsfehler auf die Summe der Zielfunktionskomponenten zu geringen Einfluss hat. Erst wenn der Zielfunktionswert der Roboterachsbeschleunigungen einen sehr kleinen Fehlerwert erreicht, beginnt das Verfahren, auch die Beschichtungsfehler

zu reduzieren. In diesem Fall reichen die gegebenen 100 Iterationen nicht aus, um die ursprüngliche Schichtqualität zu erreichen.

Bei der bahnbasierten Optimierung kann mittels des Parameters  $\lambda$  ebenfalls eine Gewichtung der Glättungskomponenten erfolgen. Hier gibt es jedoch drei unterschiedliche Komponenten, die jeweils andere Eigenschaften der Bahn beeinflussen. Die Komponente  $E_a(\mathcal{X})$  berücksichtigt die Beschleunigungen der Spritzpistole auf der Bahn und wird durch die Positionen der Bahnpunkte sowie den Zeitpunkten, zu denen diese Punkte erreicht werden, bestimmt.  $E_{aa}(\mathcal{X})$  berücksichtigt die Winkelbeschleunigungen beim Umorientieren der Spritzpistole, die von der Orientierung der Spritzkoordinatensysteme an jedem Punkt und den Zeitpunkten abhängig sind, zu denen die Punkte erreicht werden.  $E_c(\mathcal{X})$  berücksichtigt die geometrische Krümmung der Bahn und ist ausschließlich von den Positionen der Bahnpunkte abhängig. Dies bedeutet, dass neben der Variation des Parameters  $\lambda$  auch die Parameter  $\lambda_a$ ,  $\lambda_{aa}$  und  $\lambda_c$  genutzt werden können, um die einzelnen Komponenten feingranular zu gewichten. In allen Fällen führt eine hohe Gewichtung einer Komponente dazu, dass diese besonders stark optimiert wird, während alle anderen Komponenten weniger Einfluss auf die Optimierung haben. Das Verhalten des Algorithmus ist demnach dem oben beschriebenen Verhalten bei der Variation der Gewichtung zwischen der Beschichtungsqualität und den Roboterachsbeschleunigungen ähnlich.

Die konkreten Werte für die Gewichte können frei gewählt werden, so dass entsprechende Eigenschaften der Bahn bevorzugt werden können. Es bietet sich jedoch an, die Gewichte so zu wählen, dass die Zielfunktionskomponenten etwa die gleiche Größenordnung aufweisen. Somit wird gewährleistet, dass alle Komponenten einen ausreichenden Einfluss auf das Gesamtergebnis haben.

Eine weitere Möglichkeit, um Einfluss auf das Verhalten des Algorithmus zu nehmen, besteht darin, nicht alle zur Modifikation zur Verfügung stehenden Bahnparameter gleich stark zu verändern. Der Parameterraum besteht bei der roboterbasierten Optimierung aus den Roboterachsstellungen  $\Phi_j$  und den Spritzdauern  $\hat{t}_j$ , die in direktem Zusammenhang mit der Geschwindigkeit der Spritzpistole stehen. Da jedoch die Geschwindigkeiten für die thermischen Varianzen von besonderem Interesse sind, werden diese Werte nur geringfügig modifiziert, so dass die Bahn hauptsächlich über die anderen Komponenten (Roboterachsstellungen) verändert wird.

Um diese Entscheidung zu begründen, sind in Abbildung 5.5 die thermischen Varianzen einer originalen initialgeplanten Bahn sowie die Varianzen nach der Optimierung mit bzw. ohne die Nutzung der vollen Modifikation der Spritzzeiten/Geschwindigkeiten dargestellt. Es kann gut nachvollzogen werden, dass sich die thermischen Varianzen für die Bahn mit reduzierter Modifikation der Zeiten kaum verändern, die für die Bahn mit voller Modifikation jedoch relativ stark. Dies ist der Grund für die entsprechende Parameterwahl. Nicht dargestellt sind die anderen Zielfunktionswerte. Während sich bei der Variante mit reduzierter Modifikation auch geringe Beschichtungsfehler ergeben, werden etwas schlechtere Roboterachsbeschleunigungen verglichen mit dem anderen Verfahren erreicht. Dies liegt an der starken Abhängigkeit der Roboterachsbeschleunigungen von

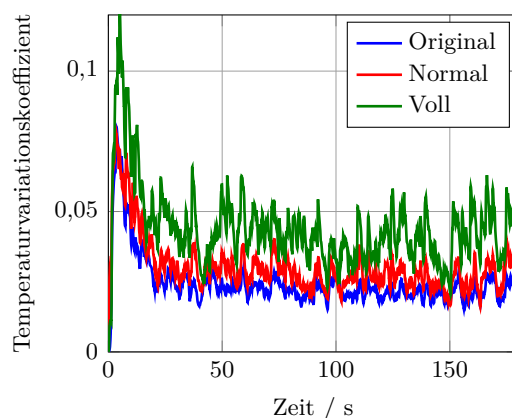


Abbildung 5.5: Darstellung der thermischen Varianzen der originalen initialgeplanten Bahn (nach Neuabastung und Glättung) („Original“), der normal optimierten Bahn mit reduzierter Modifikation der Geschwindigkeiten („Normal“) sowie einer optimierten Bahn mit voller Modifikation der Geschwindigkeiten („Voll“).

den Zeiten, zu denen die Spritzpistole die Bahnpunkte erreicht. Wenn diese jedoch weniger stark geändert werden können, ist die Optimierung der Roboterachsbeschleunigungen entsprechend schwieriger.

Zusammenfassend ist die Reduktion der Modifikation der Zeiten daher sinnvoll, um die Bahngeschwindigkeiten weniger stark zu beeinflussen. Dadurch können die durch die Initialbahnplanung generierten geringen thermischen Varianzen besser erhalten bleiben. Die Konsequenz hiervon ist eine etwas langsamere Reduktion der Roboterachsbeschleunigungen während der Optimierung.

### 5.7.5 Einfluss des Modells

Die Bahnoptimierung nutzt eine externe Simulation, um die Schichtdicke exakt vorhersagen zu können. Die Gradienten werden jedoch durch ein vorher kalibriertes Auftragsmodell ermittelt, so dass gute Gradientenrichtungen ohne viele Simulationsläufe für finite Differenzen bestimmt werden können. Dadurch ist das Verfahren jedoch abhängig von der Kalibrierung der Parameter. Dies soll in diesem Abschnitt genauer untersucht werden.

Für die Analyse werden vier verschieden parametrisierte Auftragsmodelle genutzt, um zu bestimmen, inwiefern eine Abhängigkeit der Ergebnisse von diesen Modellen besteht. Zwei der Modelle sind gegen den in der externen Simulation genutzten Libo-Footprint kalibriert. Das erste dieser Modelle nutzt nur eine einzelne Gaußfunktion („Libo“), das zweite Modell („Libo20“) nutzt 20 überlagerte Gaußfunktionen, die jeweils mit dem Verfahren aus Abschnitt 3.2.3 gegen den Footprint der externen Simulation kalibriert werden. Das dritte Modell („Gauß“) ist gegen einen synthetischen, gaußförmigen Footprint

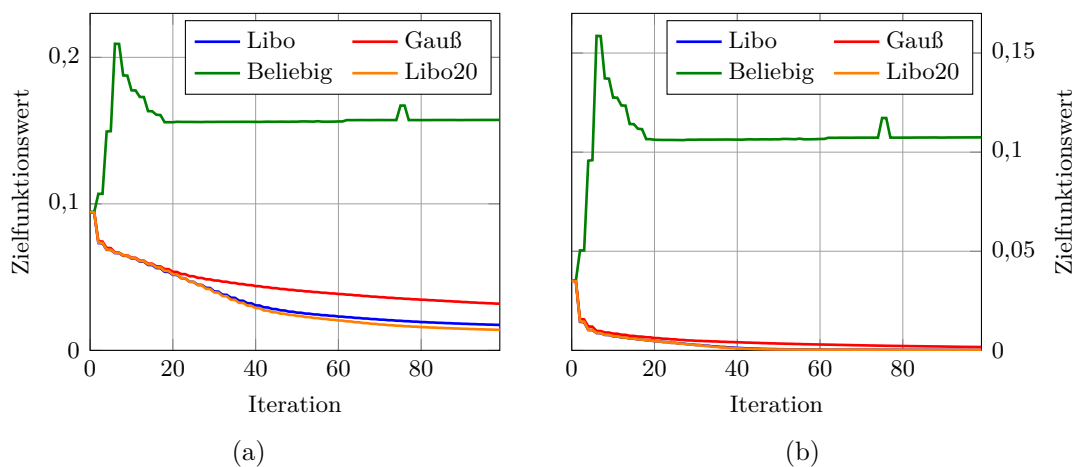


Abbildung 5.6: Bahnoptimierung mit unterschiedlichen Modellen. (a) zeigt den Zielfunktionswert für die Gesamtzielfunktion ( $E(\mathcal{X})$ ) und (b) zeigt die Zielfunktionswerte für den Beschichtungsfehler ( $E_l(\mathcal{X})$ ). „Libo“ beschreibt ein Modell mit einer und „Libo20“ mit 20 Gaußfunktionen kalibriert gegen den realen Libo-Footprint. „Gauß“ bezeichnet ein Modell, das gegen einen Gauß-Footprint kalibriert ist und „Beliebig“ bezeichnet ein nicht kalibriertes Modell.

kalibriert, der relativ ähnliche Charakteristiken wie der Libo-Footprint hat. Das letzte Modell ist mit beliebigen Parametern erstellt worden und hat keinen Bezug zu dem tatsächlich genutzten Footprint.

In der Abbildung 5.6 sind die Verläufe der Zielfunktionen für die Bahnoptimierung mit den oben vorgestellten Modellen dargestellt. Wie gut zu erkennen ist, ist die Qualität des Modells ein entscheidender Faktor für die Qualität der Optimierung. Bei Nutzung eines nicht kalibrierten, beliebigen Modells divergiert das Verfahren und verschlechtert die Bahn sogar. Bei Nutzung eines relativ guten Modells („Gauß“) kann das Verfahren die Zielfunktionen deutlich verbessern. Die Nutzung von kalibrierten Modellen („Libo“, „Libo20“) liefert die besten Ergebnisse. Dabei ist das Modell mit den überlagerten Gaußfunktionen noch einmal leicht besser als das Modell mit nur einer Gaußfunktion, jedoch unterscheiden sich die Ergebnisse nur noch minimal.

Zusammenfassend kann demnach festgehalten werden, dass die Nutzung von kalibrierten Modellen sehr wichtig ist, um gute Ergebnisse durch die Optimierung zu erreichen. Daher sollte für jeden genutzten Footprint das Modell entsprechend der Methode aus Abschnitt 3.2.3 kalibriert werden. Jedoch ist es nicht entscheidend, den Footprint vollständig perfekt abzubilden, wie an den geringen Unterschieden zwischen den Modellen („Libo“) und („Libo20“) festgestellt werden kann. Es ist demnach möglich, zu Gunsten der Einfachheit ein weniger komplexes Ersatzmodell zu nutzen, ohne starke Qualitätsverluste hinnehmen zu müssen.



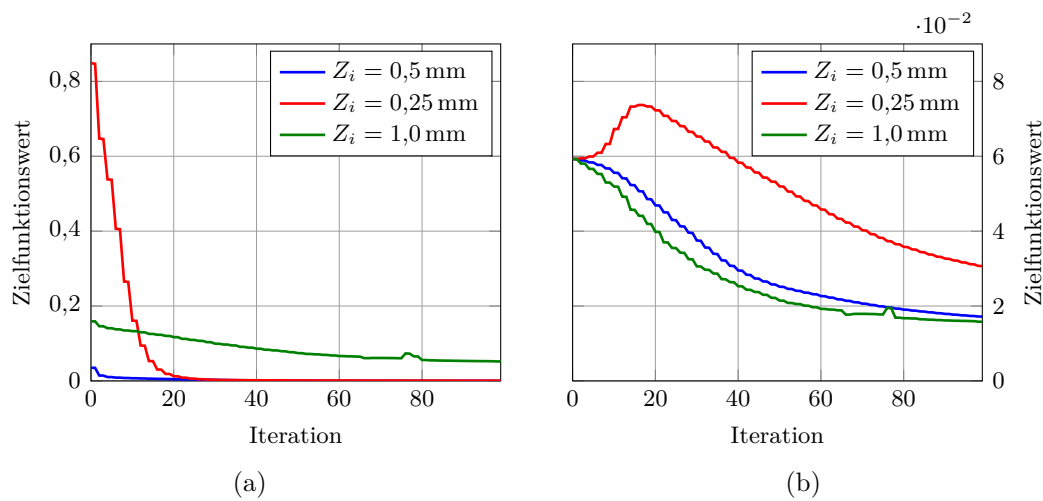


Abbildung 5.7: Verlauf der Zielfunktionskomponenten für (a) den Beschichtungsfehler ( $E_l(\mathcal{X})$ ) sowie (b) die Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ) bei der Bahnoptimierung für eine Bahn mit einer ungefähren durchschnittlichen Beschichtung von 0,5 mm für verschiedene Zielschichtdicken  $Z_i$ .

### 5.7.6 Einfluss der Eingabe

Einen nicht unbedeutenden Einfluss auf das Ergebnis der Bahnoptimierung hat die Qualität der Eingabe. Daher wird an einigen Beispielen gezeigt, wie sich Eingabedaten mit schlechter Qualität auf die Bahnoptimierung auswirken.

Eine Voraussetzung für die Eingabebahn ist, dass sie die Zielschichtdicke durchschnittlich schon einigermaßen erreicht, so dass die Bahnoptimierung die Bahn lokal nur noch so zu ändern braucht, dass die Fehler in der Beschichtung reduziert werden. In diesem ersten Testfall werden jedoch Zielschichtdicken gefordert, die wesentlich von der durchschnittlich erzeugten Schichtdicke der Bahn abweichen. Hierzu wird eine initialgeplante Bahn genutzt, die in etwa eine durchschnittliche Schichtdicke von 0,5 mm erzeugt. Die Optimierung wird dann für die Zielschichtdicken 0,5 mm, 1 mm sowie 0,25 mm durchgeführt. Die Verläufe der Zielfunktionskomponenten der Optimierungen sind in Abbildung 5.7 dargestellt. Es kann gut nachvollzogen werden, dass im Falle der zu hohen geforderten Schichtdicke die Beschichtung auch nach der Optimierung nicht gut ist. Dies liegt daran, dass eine höhere Schichtdicke nur dadurch erzeugt werden kann, dass die Bahn langsamer abgefahren wird. Die Änderung der Bahngeschwindigkeit ist jedoch in der Optimierung gedämpft, wie oben beschrieben wurde, damit die thermischen Varianzen nicht verschlechtert werden. Daher wird der Fehler in der Beschichtung nur langsam reduziert, während die Roboterachsbeschleunigungen stark verbessert werden. Im Falle einer wesentlich zu gering geforderten Zielschichtdicke zeigt sich ein anderes Verhalten. Die Beschichtungsfehler werden schnell deutlich reduziert, während die Roboterachsbeschleunigungen zunächst etwas schlechter, später jedoch besser werden. Dies liegt daran, dass die Bahn durch die

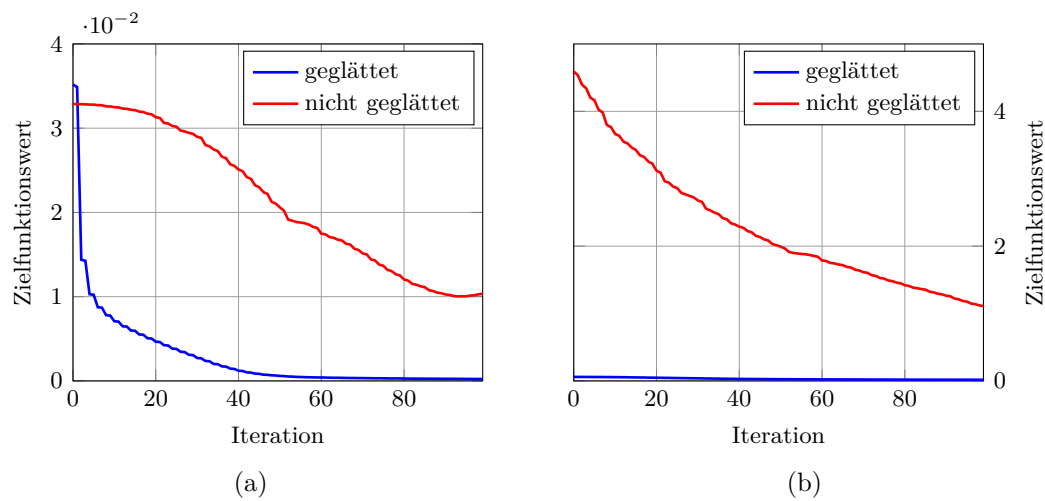


Abbildung 5.8: Verlauf der Zielfunktionskomponenten für (a) den Beschichtungsfehler ( $E_l(\mathcal{X})$ ) sowie (b) die Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$ ) bei der Bahnoptimierung für eine geglättete und eine nicht geglättete Bahn.

Optimierung so verschoben wird, dass ein Großteil des Materials neben das Werkstück gespritzt wird. Damit wird der Auftrag reduziert, ohne die Bahngeschwindigkeit deutlich anzupassen. Auch wenn die Beschichtungsergebnisse relativ gut sind, ist dies kein gewünschtes Verhalten auf Grund des hierdurch entstehenden Oversprays. Insgesamt ist es demnach wichtig, dass die Eingabebahn bereits ungefähr die Zielschichtdicke erreicht, die bei der Optimierung gefordert ist. Im Zusammenhang mit der Initialbahnplanung ist dies kein Problem, da die Bahnen die geforderte Zielschichtdicke bereits ungefähr erreichen.

Als zweiter Testfall wird das Verhalten der Optimierung bei Nutzung nicht geglätteter Bahnen untersucht. Dazu wird eine initialgeplante Bahn in einer jeweils geglätteten und nicht geglätteten Form genutzt. In beiden Fällen wird die Bahn neu abgetastet, damit die Bahnpunkte etwa äquidistant verteilt sind. In Abbildung 5.8 sind die Zielfunktionsverläufe während der Optimierung für die beiden Versuche dargestellt. An den Startwerten für die Zielfunktion  $E_m(\mathcal{X})$  ist deutlich zu erkennen, dass die vorherige Glättung der Bahn die Roboterachsbeschleunigungen stark reduziert. In dem nicht geglätteten Fall werden die Roboterachsbeschleunigungen durch die Bahnoptimierung stark verringert, erreichen jedoch auch nach 100 Iterationen nicht die gleiche Qualität, die die geglättete Bahn bereits zu Anfang hat. Aber auch der Beschichtungsfehler bleibt im Verlauf der Optimierung deutlich schlechter, wenn die Bahn nicht geglättet ist, obwohl er ganz zu Anfang sogar leicht besser ist. Dies liegt daran, dass die Zielfunktionskomponente für die Glättung die Gesamtzielfunktion dominiert und diese dadurch bevorzugt optimiert wird. Insgesamt zeigt sich daher, dass die vorherige Glättung der Bahn nicht nur für die Roboterachsbeschleunigungen vorteilhaft ist, sondern auch die Beschichtungsqualität nach der Optimierung erhöht.

0.0	88.0	99.7	85.3	83.9	92.3	91.5	86.6	90.6	91.3	90.6	90.1	91.8	88.5	85.2	95.1	95.5	85.6	87.1	90.3	88.1	90.4	92.9	90.2	88.5
88.0	0.0	87.0	90.2	87.9	89.2	92.0	89.9	85.9	92.8	89.5	91.1	88.2	86.0	92.9	90.2	92.1	92.4	82.3	87.5	93.4	88.2	91.9	93.0	85.1
99.7	87.0	0.0	91.5	92.3	87.9	88.2	89.8	92.8	85.6	94.4	88.2	83.6	90.3	93.0	92.9	85.0	86.4	94.7	84.8	85.8	97.6	87.6	82.7	93.0
85.3	90.2	91.5	0.0	91.1	90.3	85.1	93.7	86.8	95.6	85.0	85.8	90.3	92.4	88.8	83.6	92.4	89.8	83.5	96.1	90.8	78.2	90.0	94.2	85.3
83.9	87.9	92.3	91.1	0.0	90.3	96.1	81.9	98.8	85.9	88.0	87.2	95.5	86.9	83.7	92.4	86.6	88.6	94.8	86.0	88.3	90.5	83.3	87.2	94.2
92.3	89.2	87.9	90.3	90.3	0.0	86.3	102.4	77.3	88.9	89.0	98.9	82.1	87.1	91.8	84.4	90.6	91.1	86.6	89.8	85.7	88.4	91.2	83.8	86.5
91.5	92.0	88.2	85.1	96.1	86.3	0.0	83.7	94.4	84.1	101.9	82.0	88.0	85.9	89.1	92.3	86.7	89.0	86.5	84.1	92.7	85.4	87.1	98.4	87.2
86.6	89.9	89.8	93.7	81.9	102.4	83.7	0.0	90.2	104.2	77.6	89.9	86.8	94.4	84.6	89.0	94.6	79.9	85.4	92.4	81.6	93.6	95.5	81.3	87.6
90.6	85.9	92.8	86.8	98.8	77.3	94.4	90.2	0.0	84.3	94.8	82.0	95.6	82.5	98.7	89.4	77.3	95.0	88.8	81.2	97.3	89.1	83.2	93.8	88.8
91.3	92.8	85.6	95.6	85.9	88.9	84.1	104.2	84.3	0.0	91.8	98.9	81.5	101.2	83.5	84.3	92.6	82.7	90.9	95.9	85.1	86.0	89.7	86.5	88.3
90.6	89.5	94.4	85.0	88.0	89.0	101.9	77.6	94.8	91.8	0.0	89.7	104.6	77.7	92.3	88.7	86.0	93.9	87.5	90.1	92.3	85.3	85.5	88.6	90.3
90.1	91.1	88.2	85.8	87.2	98.9	82.0	89.9	82.0	98.9	89.7	0.0	90.0	94.5	81.1	95.2	87.6	89.2	96.4	81.6	89.4	94.2	82.1	89.2	95.1
91.8	88.2	83.6	90.3	95.5	82.1	88.0	86.8	95.6	81.5	104.6	90.0	0.0	94.6	99.3	79.0	102.4	88.0	80.0	100.3	80.3	83.1	103.9	85.7	77.7
88.5	86.0	90.3	92.4	86.9	87.1	85.9	94.4	82.5	101.2	77.7	94.5	94.6	0.0	89.5	114.0	72.6	97.5	98.0	70.0	100.0	96.4	76.6	98.0	95.6
85.2	92.9	93.0	88.8	83.7	91.8	89.1	84.6	98.7	83.5	92.3	81.1	99.3	89.5	0.0	79.3	112.8	76.8	85.3	101.0	80.5	91.1	95.3	77.3	87.1
95.1	90.2	92.9	83.6	92.4	84.4	92.3	89.0	89.4	84.3	88.7	95.2	79.0	114.0	79.3	0.0	86.3	105.9	79.9	99.7	93.2	76.3	93.5	90.3	79.8
95.5	92.1	85.0	92.4	86.6	90.6	86.7	94.6	77.3	92.6	86.0	87.6	102.4	72.6	112.8	86.3	0.0	87.9	118.7	73.0	94.6	97.9	72.7	94.4	100.5
85.6	92.4	86.4	89.8	88.6	91.1	89.0	79.9	95.0	82.7	93.9	89.2	88.0	97.5	76.8	105.9	87.9	0.0	80.7	113.3	75.2	89.1	102.3	76.0	84.6
87.1	82.3	94.7	83.5	94.8	86.6	86.5	85.4	88.8	90.9	87.5	96.4	80.0	98.0	85.3	79.9	118.7	80.7	0.0	89.4	107.9	77.7	95.0	98.7	78.8
90.3	87.5	84.8	96.1	86.0	89.8	84.1	92.4	81.2	95.9	90.1	81.6	100.3	70.0	101.0	99.7	73.0	113.3	89.4	0.0	91.0	116.7	74.1	93.6	105.7
88.1	93.4	85.8	90.8	88.3	85.7	92.7	81.6	97.3	85.1	92.3	89.4	80.3	100.0	80.5	93.2	94.6	75.2	107.9	91.0	0.0	84.6	119.0	75.8	78.8
90.4	88.2	97.6	78.2	90.5	88.4	85.4	93.6	89.1	86.0	85.3	94.2	83.1	96.4	91.1	76.3	97.9	89.1	77.7	116.7	84.6	0.0	86.0	114.6	81.6
92.9	91.9	87.6	90.0	83.3	91.2	87.1	95.5	83.2	89.7	85.5	82.1	103.9	76.6	95.3	93.5	72.7	102.3	95.0	74.1	119.0	86.0	0.0	85.3	125.2
90.2	93.0	82.7	94.2	87.2	83.8	98.4	81.3	93.8	86.5	88.6	89.2	85.7	98.0	77.3	90.3	94.4	76.0	98.7	93.6	75.8	114.6	85.3	0.0	77.9
88.5	85.1	93.0	85.3	94.2	86.5	87.2	87.6	88.8	88.3	90.3	95.1	77.7	95.6	87.1	79.8	100.5	84.6	78.8	105.7	78.8	81.6	125.2	77.9	0.0

Abbildung 5.9: Winkel zwischen den Gradienten, die während der ersten 50 Iterationen des Algorithmus erzeugt werden. Der Wert in der Zelle in der  $i$ -ten Zeile und der  $j$ -ten Spalte der Tabelle gibt den Winkel zwischen dem  $i$ -ten und  $j$ -ten Gradienten an. Bei den Grautönen entspricht weiß einem Winkel von  $90^\circ$  und schwarz einem Winkel von  $0^\circ$  sowie  $180^\circ$ . Je dunkler ein Feld ist, desto stärker weicht der Winkel vom gewünschten Wert ab.

### 5.7.7 Winkel der Gradienten

Einen großen Einfluss auf das nichtlineare konjugierte Gradientenverfahren hat die Qualität der generierten Gradientenrichtungen. Im Gegensatz zum herkömmlichen Gradientenabstieg werden die vorhergehenden Gradientenrichtungen in den nachfolgenden Schritten weiter verwendet, um die Konvergenz des Verfahrens zu beschleunigen. In dieser Analyse soll festgestellt werden, wie sich die Gradientenrichtungen im Verlauf des Algorithmus verhalten.

In Abbildung 5.9 sind die Winkel zwischen den Gradientenrichtungen der ersten 50 Iterationen (entspricht 25 Gradientenrichtungen) des Algorithmus dargestellt. Man

erkennt sehr gut, dass viele Richtungen einen Winkel von nahezu  $90^\circ$  gegenüber allen anderen Gradientenrichtungen haben, was dem idealen Wert entspricht. Erst nach einigen Iterationen werden Gradientenrichtungen erzeugt, die bezüglich einiger Nachbarn im Bereich von  $\pm 30^\circ$  von den idealen  $90^\circ$  abweichen. Diese Abweichungen können hauptsächlich unten rechts, also zu einem späten Zeitpunkt der Optimierung, in der Abbildung gefunden werden. Ursächlich für die nicht idealen Winkel ist unter anderem die nicht ideale Schrittweitenbestimmung, so dass nicht in jedem Schritt das Minimum entlang der Suchrichtung gefunden wird. Daher müssen die nachfolgenden Schritte dieses Defizit korrigieren, wodurch sich die nicht idealen Winkel im späteren Verlauf erklären lassen.

Insgesamt kann jedoch festgestellt werden, dass zumindest zu Beginn des Verfahrens sehr gute Gradientenrichtungen erzeugt werden. Dies ist ein Hinweis darauf, dass das Verfahren seine Stärken gegenüber dem herkömmlichen Gradientenabstieg ausnutzen kann.

### 5.7.8 Einfluss der Knotenverschiebung

In diesem Abschnitt wird die Nutzung der Knotenverschiebung untersucht. Der Grundgedanke ist, dass durch eine geringfügige Modifikation des Werkstücknetzes die Zielkontur auch an den Stellen des Werkstücks erreicht werden kann, die allein durch die Modifikation der Bahn nicht vollständig exakt mit der Zielschichtdicke beschichtet werden können. Dabei ist es natürlich wichtig, dass die Endkontur nicht allein durch die Knotenverschiebung entsteht, sondern eine gewisse Mindestschichtdicke abgelagert wird.

Um zu ermitteln, wie sich die Nutzung der Knotenverschiebung auswirkt, ist die roboterbasierte Optimierung ohne sowie mit Verwendung der Knotenverschiebung mit unterschiedlichen Parametern ausgeführt worden. Für die Evaluationen mit der Knotenverschiebung wird im ersten Versuch der Gradient für die Verschiebung unverändert genutzt, im zweiten Versuch werden die entsprechenden Gradientenanteile um den Faktor 10 höher gewichtet. In allen Fällen werden die Glättungsfunktionen aus den Gleichungen (5.12) und (5.13) genutzt. In Abbildung 5.10 sind die Zielfunktionsverläufe der unterschiedlichen Komponenten für die verschiedenen Testfälle dargestellt. Man erkennt die Unterschiede vor allem zwischen den beiden Versuchen ohne und mit Knotenverschiebung bei Nutzung des verstärkten Gradienten. Für den Fall mit der Knotenverschiebung verringert sich der Beschichtungsfehler schneller und ist am Ende der Optimierung nochmals deutlich geringer als für den Fall ohne die Knotenverschiebung. Im Gegenzug verringert sich die Roboterachsbeschleunigung etwas langsamer, erreicht jedoch nach 100 Iterationen einen sehr ähnlichen Zielfunktionswert und ist nur geringfügig schlechter. Die Zielfunktionswerte für die Glättung steigen für den Versuch mit Knotenverschiebung recht schnell an, nehmen jedoch im späteren Verlauf wieder ab, so dass die anfänglich stärkere Knotenverschiebung wieder rückgängig gemacht wird. Dies deutet darauf hin, dass durch die Nutzung dieser Zielfunktionskomponenten erreicht wird, dass die Knoten-

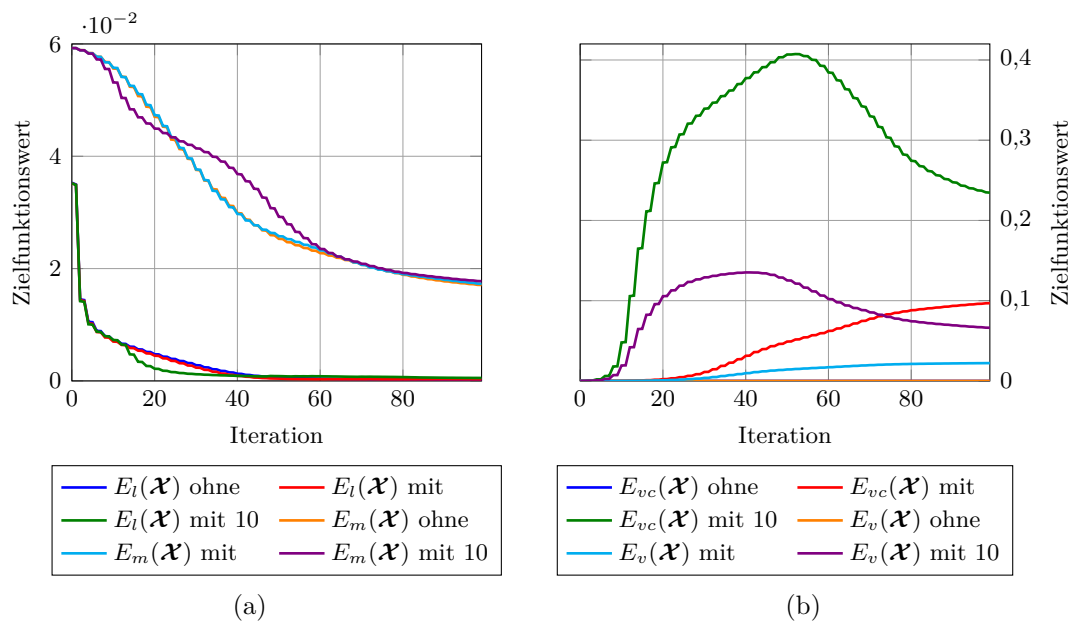


Abbildung 5.10: Bahnoptimierung mit unterschiedlichen Parametrisierungen für die Knotenverschiebung. (a) zeigt die Zielfunktionswerte für den Beschichtungsfehler ( $E_l(\mathcal{X})$ ) und die Roboterausbeschleunigungen ( $E_m(\mathcal{X})$ ) und (b) zeigt die Zielfunktionswerte für die werkstückbasierten Komponenten ( $E_v(\mathcal{X})$  und  $E_{vc}(\mathcal{X})$ ).

verschiebung nur sekundär genutzt wird, und Stellen, die über die Bahnmodifikation verbessert werden können, diese auch nutzen.

Insgesamt kann festgestellt werden, dass die Modifikation des Werkstücknetzes zu einer Verbesserung der Beschichtungsergebnisse führt, wenn allein das Erreichen der Zielkontur entscheidend ist und die Schichtdicke in einem gegebenen Bereich variieren kann. Wenn das Werkstück zum Beispiel durch Fräsen noch erzeugt werden muss und Veränderungen noch möglich sind, ist dieses Verfahren demnach sehr gut geeignet, um unmögliche Bahnanpassungen an schwierigen Stellen auszugleichen.

### 5.7.9 Zusammensetzung der Laufzeit

Auf Grund der iterativen Natur des nichtlinearen konjugierten Gradientenverfahrens und der rechenintensiven Einzelschritte benötigt die Optimierung einige Zeit zur Ausführung. In diesem Abschnitt soll der Rechenzeitbedarf daher genauer untersucht werden.

Die Laufzeit der Bahnoptimierung ist hauptsächlich von zwei Komponenten abhängig. Zum einen ist die Simulation der Bahn mit der externen Simulation zeitaufwändig, zum anderen ist die Bestimmung der Gradienten sehr rechenintensiv. In Abbildung 5.11 sind die Rechenzeiten der beiden Hauptkomponenten gegenübergestellt. Hierzu ist eine Optimierung auf einer relativ hochaufgelösten Version des Werkstücks B für 100

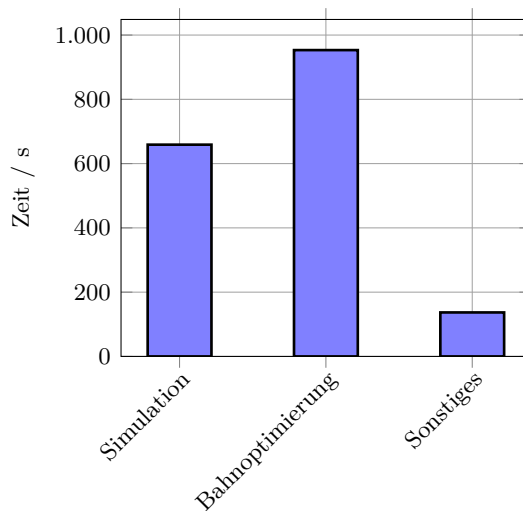


Abbildung 5.11: Zusammensetzung der Laufzeit der Bahnoptimierung. Gezeigt sind die summierten Werte für eine Planung mit 100 Iterationen für die wesentlichen Komponenten externe Simulation, Bahnoptimierung und Sonstiges für eine initialgeplante Bahn auf Werkstück B.

Iterationen ausgeführt worden, so dass 100 Simulationen und 99 Gradientenbestimmungen ausgeführt wurden. Man erkennt, dass die Simulation über das externe Programm den zweitgrößten Anteil an der Gesamtlaufzeit hat. Die externe Simulation nutzt effiziente GPGPU Berechnungen und ist daher sehr effizient. Für den Testlauf wurde eine GeForce GTX 760 genutzt. Die Bahnoptimierung, die den größten Anteil an der Laufzeit der Optimierung hat, findet auf der CPU statt. Hierfür stand ein Core i7 4790K mit 4 GHz zur Verfügung. Ebenfalls in Abbildung 5.11 angegeben ist die restliche Rechenzeit, die unter anderem für die Kommunikation der Optimierung mit der externen Simulation über Netzwerk, die Aufbereitung der Daten und zur Visualisierung verbraucht wird. Auf Grund des iterativen Charakters des Algorithmus skaliert die Laufzeit linear mit der Anzahl der Optimierungsschritte.

Insgesamt setzt sich die Laufzeit der Optimierung demnach hauptsächlich aus der Simulation und der Bestimmung der Gradienten zusammen. Die restlichen Komponenten haben nur einen geringfügigen Anteil an der Gesamtlaufzeit.

### 5.7.10 Fallbeispiele

In diesem Abschnitt sollen Fallbeispiele für die Anwendung der Bahnoptimierung aufgeführt werden. Der Kernanwendungsfall ist die Nutzung der Bahnoptimierung als Nachverarbeitungsschritt im Anschluss an die Initialbahnplanung. Es ist jedoch auch möglich, die Bahnplanung dazu zu nutzen, bereits vorhandene Bahnen an andere Gegebenheiten anzupassen. Zwei Beispiele hierfür sind die Nutzung einer Bahn für ein anderes,

jedoch ähnliches Werkstück sowie für einen anderen Spritzprozess. In beiden Fällen ist das Ziel, die vorhandene Bahn anzupassen und keine neue Bahn zu erstellen. Dadurch kann sehr viel Rechenzeit gespart werden, da auf eine erneute Initialbahnplanung verzichtet werden kann.

### **Optimieren von Initialbahnen**

Die Optimierung einer durch die Initialbahnplanung erstellten Bahn ist in den vorangegangenen Abschnitten bereits evaluiert worden. In Abbildung 5.12a ist eine Bahn für das Werkstück A noch einmal dargestellt. Die Bahn wurde für 100000 Iterationen mit der Initialbahnplanung generiert, anschließend geglättet und für 100 Iterationen durch die Bahnoptimierung nachbearbeitet. Das Ergebnis ist eine glatte Bahn mit geringen Roboterachsbeschleunigungen und einem relativ geringen maximalen Beschichtungsfehler von knapp über 3%. Dieser Fehler könnte durch eine längere Bahnoptimierung noch weiter reduziert werden. Die gezeigte Bahn soll für die nachfolgenden Fallbeispiele weiter verwendet werden.

### **Ändern der Werkstückgeometrie**

Im Verlauf von realen Produktionsprozessen kann es vorkommen, dass ein Werkstück leicht verändert werden soll. Wenn diese Veränderungen nicht besonders ausgeprägt sind, ist es möglich, die vorhandene Bahn weiterzunutzen und auf die Veränderungen anzupassen, so dass keine Bahn von Grund auf neu erzeugt werden muss. Dies erspart sehr viel Rechenzeit, Aufwand und Kosten.

In Abbildung 5.12b und Abbildung 5.12c ist ein im Gegensatz zu Werkstück A leicht verändertes Werkstück dargestellt, das keine Wellenform auf der Oberfläche besitzt. Die Auswirkungen auf die Beschichtung sind trotz der geringfügigen Änderungen dennoch sehr stark, wenn die originale Bahn weiter verwendet wird, wie in Abbildung 5.12b zu sehen ist. An den Stellen, an denen das Werkstück verändert wurde, ist die Beschichtung sehr schlecht. Die Optimierung der vorhandenen Bahn mit der Bahnoptimierung für 100 Iterationen kann diesen Fehler beheben, wie in Abbildung 5.12c zu sehen ist. Die Beschichtung erreicht die gleiche Qualität, die sie auch für das ursprüngliche Werkstück hatte. Daher ist es durchaus sinnvoll, wenn möglich, vorhandene Bahnen weiter zu verwenden und neu zu optimieren, um den Aufwand der vollständigen Neuerstellung einer Bahn zu vermeiden.

### **Ändern des Spritzprozesses**

Ein weiterer in der Praxis vorkommender Fall ist, dass sich zum Beispiel durch Verschleiß die Prozessparameter verändern. Dadurch verändert sich das Auftragsverhalten und die ursprünglich geplante und optimierte Bahn muss angepasst werden, um die so



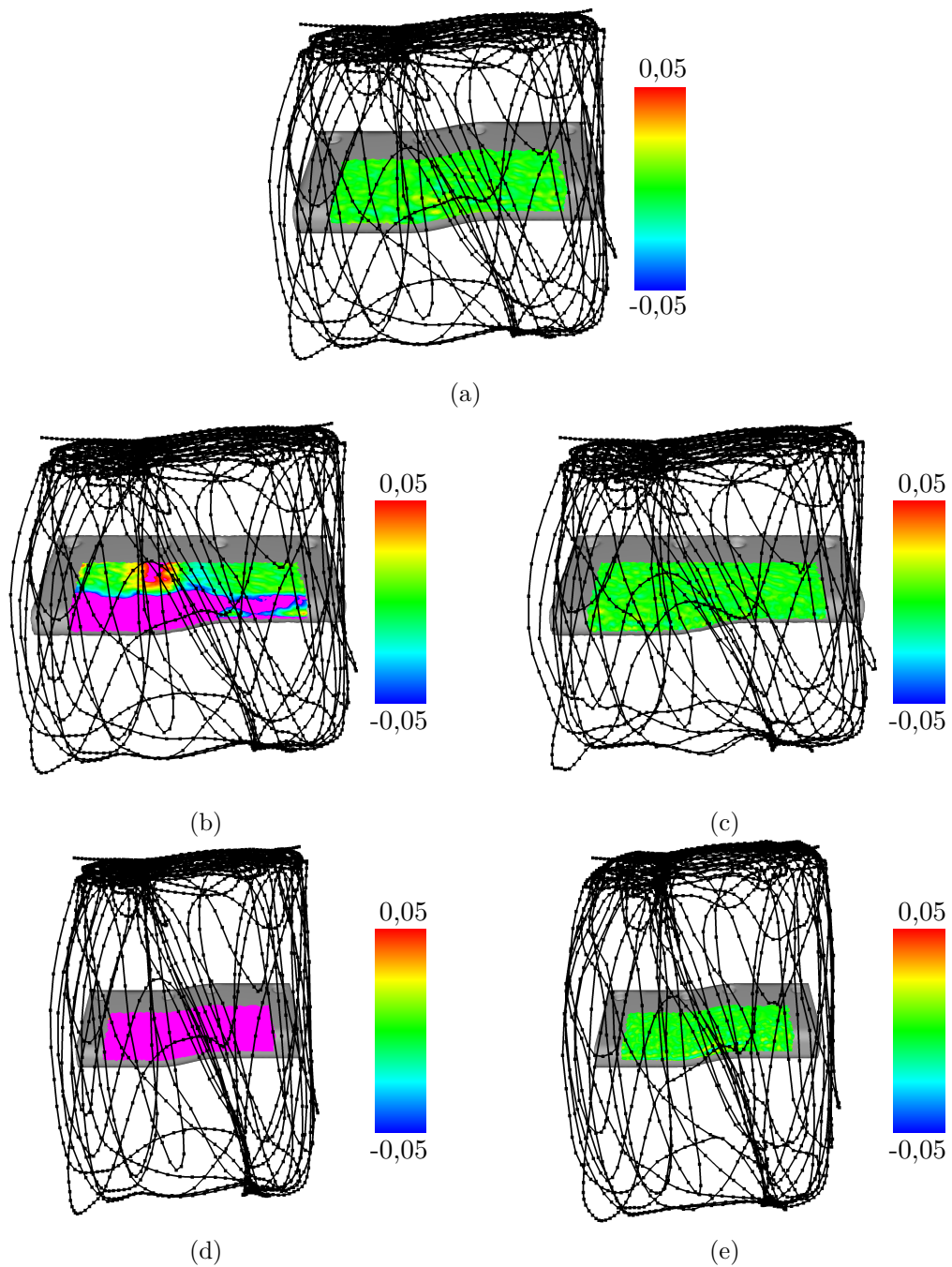


Abbildung 5.12: Fallbeispiele für die Bahnoptimierung. (a) Beispiel für eine initialgeplante Bahn nach der Optimierung mit der Bahnoptimierung (Basisbahn). (b) Simulation der Basisbahn auf einem anderen Werkstück. (c) Simulation der neu optimierten Basisbahn auf einem anderen Werkstück. (d) Simulation der Basisbahn mit einem anderen Prozess. (e) Simulation der neu optimierten Basisbahn mit einem anderen Prozess. Rosa Bereiche auf dem Werkstück liegen außerhalb der angegebenen Skala.



entstehenden Fehler zu korrigieren. Ähnlich hierzu ist die vollständige Änderung des Spritzprozesses, was ebenfalls in einer Änderung der Prozessparameter resultiert. Hier wird der Fall untersucht, dass von einem Libo- zu einem HVOF-Prozess gewechselt werden soll.

Da der HVOF-Prozess deutlich andere Eigenschaften hat als der Libo-Prozess, werden einige Parameter von Hand angepasst. Erstens ist die Materialabgabe beim HVOF-Prozess deutlich geringer als beim Libo-Prozess, so dass der Footprint um den Faktor 20 skaliert wird. Dies simuliert das Beschichten beim 20-fachen Abfahren der Bahn, wie es in der Praxis getan wird. Zweitens liegt der ideale Spritzabstand bei 170 mm anstelle von 110 mm, so dass der Bahnabstand auf diesen Wert angepasst wird. Auf die so modifizierte Bahn wird dann die Bahnoptimierung angewendet.

In [Abbildung 5.12d](#) ist die Beschichtung vor und in [Abbildung 5.12e](#) nach der Optimierung dargestellt. Die sehr hohen Fehler werden durch die Optimierung korrigiert und es wird wieder eine ähnliche Schichtqualität wie bei der Originalbahn beim ursprünglichen Prozess erreicht. Daher kann auch hier der Aufwand des Erstellens einer neuen Bahn gespart werden, wenn die vorhandene Bahn durch die Bahnoptimierung angepasst werden kann.

Insgesamt zeigen die Fallbeispiele, dass die Bahnoptimierung nicht nur zur Nachverarbeitung der initialgeplanten Bahnen, sondern auch zur Anpassung von Bahnen bei der Änderung der Werkstückgeometrie oder der Veränderung der Prozessparameter dienen kann.



# Nutzung optimierter Bahnen

In diesem Kapitel wird ein Verfahren zur Sicherstellung der Ausführbarkeit der Bahnen vorgestellt, die durch die in den vorherigen Kapiteln vorgestellten Methoden generiert und optimiert wurden. Die bisher erstellten Bahnen zeichnen sich durch eine hohe Qualität bezüglich der in das Werkstück eingebrachten thermischen Varianzen, durch geringen Overspray und durch die Erzeugung einer Schicht aus, die nur geringe Abweichungen von der Zielschichtdicke aufweist. Die Kinematik des Roboters wurde durch Nutzung von entsprechenden Zielfunktionen berücksichtigt, so dass die Erreichbarkeit aller Bahnpunkte durch den Roboter gewährleistet ist. Ein mögliches Problem der Bahnen ist die in dem Erstellungsprozess unberücksichtigte Roboterdynamik, was dazu führen kann, dass die Bahnen von dem eingesetzten Robotersystem nicht exakt nach den Vorgaben abgefahren werden können. Daher wird in diesem Kapitel ein Verfahren vorgestellt, das dieses Defizit behebt, so dass die Bahnen im realen Prozess uneingeschränkt eingesetzt werden können. Zur Verifikation der Ausführbarkeit der Bahnen wird beispielhaft auf die Steuerungssoftware eines Roboterherstellers zurückgegriffen, um die Bewegungen des Roboters bei Ausführung der durch die oben beschriebenen Verfahren generierten Bahnen zu analysieren. Nach einer Konvertierung der Bahnen in entsprechende Roboterprogramme ist die Herstellersoftware in der Lage, die genauen Bewegungen, die vom echten Roboter zu erwarten sind, auch unter Berücksichtigung des Dynamikverhaltens zu simulieren. Die Ergebnisse der Simulation werden dazu genutzt, die Bahnen entsprechend der durch die Dynamik entstehenden Abweichungen anzupassen, so dass der geforderte Schichtauftrag unter Einhaltung der Dynamikgrenzen erreicht wird.

In Abschnitt 6.1 wird in einer kurzen Einführung die Idee des Verfahrens erläutert. Abschnitt 6.2 gibt eine Übersicht über das Konzept des Verfahrens und stellt die einzelnen Komponenten vor. Anschließend wird in Abschnitt 6.3 die Robotersteuerungssoftware RobotStudio vorgestellt, mit der die Bahnen realitätsnah simuliert werden können. In

Abschnitt 6.4 werden die Details zu dem Verfahren beschrieben und Abschnitt 6.5 schließt das Kapitel mit einer Evaluation des Verfahrens ab.

## 6.1 Einleitung

In den vorangegangenen Bahnplanungsschritten sind Bahnen erzeugt worden, die bezüglich der Beschichtung, der thermischen Varianzen, der Roboterkinematik sowie dem Overspray optimiert wurden. Die Ausführbarkeit der Bahn ist demnach nur über die Kinematik, nicht jedoch bezüglich der Dynamik berücksichtigt worden. Um dieses Defizit zu beheben, wird ein Verfahren zur Bahnkorrektur in die Optimierungsschleife eingefügt.

Im Gegensatz zur Roboterkinematik, die mit den Verfahren aus Abschnitt 2.3.2 beschrieben werden kann, ist die Roboterdynamik ein sehr komplexes Gebiet. Die Dynamik befasst sich mit dynamischen Effekten bei der Bewegung des Roboterarms. Hierzu wird die Bewegung des Roboters als Resultat der Kräfte betrachtet, die unter anderem von den Gelenkmotoren erzeugt werden. Aber auch die Masse, die die Glieder des Roboterarms sowie die Spritzpistole besitzen, trägt wesentlich zum dynamischen Verhalten des Roboters bei. Eine naheliegende Konsequenz ist, dass der Roboter aus dem Stillstand eine gewisse Beschleunigungsphase hat, bevor er die gewünschte TCP-Geschwindigkeit erreicht.

Die Dynamik soll mit Hilfe einer Simulations- und Steuerungssoftware eines Roboterherstellers berücksichtigt werden. Hier wird RobotStudio (siehe [1]) von ABB verwendet. Die Software ist in der Lage, bei Eingabe eines Roboterprogramms die Bewegungen des Roboters unter Berücksichtigung der dynamischen Effekte zu simulieren. Dabei nutzt die Software dieselben Steuersysteme, die auch den realen Roboter steuern, so dass ein hohes Maß an Realitätsbezug entsteht. Im Rahmen dieser Arbeit wird davon ausgegangen, dass die Simulation exakt den Bewegungen des realen Roboters entspricht. Abweichungen könnten dadurch entstehen, dass das Gewicht des Schlauchpakets oder der Rückstoß der Spritzpistole Einfluss auf die Bewegungen des echten Roboters hat.

Die Robotersimulation soll durch die Integration in die Bahnoptimierung in das Gesamtkonzept eingebunden werden, wie es in Abschnitt 6.2 beschrieben wird. Die Idee ist, dass während der Optimierung der Bahn die Dynamik überprüft und die Bahn entsprechend korrigiert wird, so dass mit der dynamisch zulässigen Bahn anschließend weiter optimiert werden kann.

## 6.2 Übersicht

In Abbildung 6.1 ist der konzeptionelle Aufbau des Verfahrens bzw. die Integration der neuen Komponenten in das Gesamtsystem dargestellt. Eine initiale Bahn, die zum Beispiel aus der Initialbahnplanung stammen kann, dient als Eingabe. Der erste Schritt ist eine optionale Bahnglättung. Dieser Schritt kann die Qualität der Bahn bezüglich

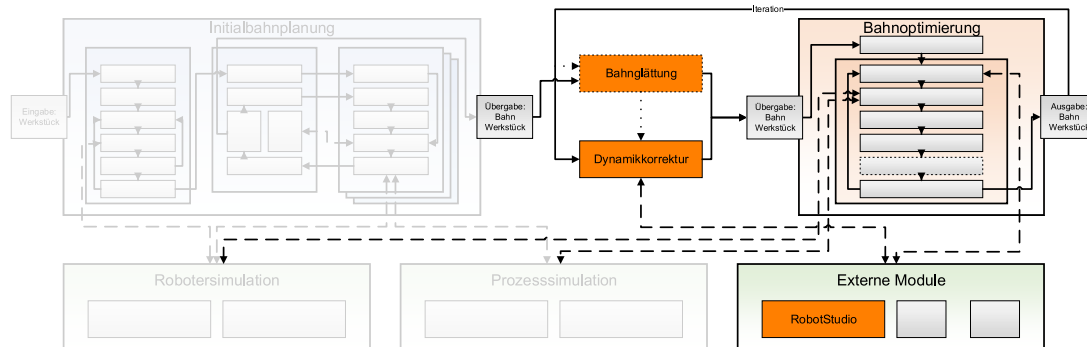


Abbildung 6.1: Konzeptioneller Aufbau der Dynamikkorrektur. Die Komponenten des Systems sind durch graue Rechtecke dargestellt. Optionale Komponenten haben einen gepunkteten Rand. Die Kontrollflüsse sind durch durchgezogene Pfeile gekennzeichnet, die Datenkommunikation über gestrichelte Pfeile. Optionale Kontrollflüsse sind durch gepunktete Pfeile gekennzeichnet. Die neuen Komponenten haben einen orangefarbenen Hintergrund.

der Dynamik deutlich verbessern, da die dynamisch schwierigen Stellen meist an starken Umlagerungen des Werkzeugs oder an Stellen mit großen Beschleunigungen auftreten, die durch eine Glättung deutlich vermindert werden können.

Die möglicherweise durch die Glättung modifizierte Bahn wird anschließend von der Bahnoptimierung, wie in Kapitel 5 beschrieben, bezüglich der Beschichtung sowie der Roboterachsbeschleunigungen optimiert.

Nach dem initialen Erstellen der Bahn und der Bahnoptimierung kann die resultierende Bahn außerhalb des dynamisch zulässigen Bereichs liegen. Daher soll die Dynamikkorrektur genutzt werden, um die Bahn in den zulässigen Bereich zu projizieren. Dazu wird die Software RobotStudio genutzt, die in der Lage ist, die Bahn zu bestimmen, die bei Eingabe des Roboterprogramms tatsächlich vom Roboter auch unter Berücksichtigung der Dynamik ausgeführt wird (siehe Abschnitt 6.3). Die Bewegungen des Roboters können während der Simulation gespeichert und in eine neue Bahn konvertiert werden. Die neue Bahn kann schließlich in die Bahnplanung zurückgeführt werden.

Die Korrektur der Dynamik erzeugt eine neue Bahn, die auf Grund der durchgeführten Modifikationen entsprechende Abweichungen bezüglich der Schichtdicke hervorruft. Daher wird die Bahnoptimierung, die Dynamikkorrektur und wahlweise die Bahnglättung iterativ ausgeführt. Nach einigen Iterationen konvergiert das Verfahren, so dass eine Bahn entsteht, die dynamisch zulässig ist und eine gute Beschichtung erzeugt.

### 6.3 RobotStudio

Bei RobotStudio handelt es sich um eine Software zur Steuerung und Simulation von Robotern der Firma ABB Asea Brown Boveri Ltd. Mit der Software können Arbeitsstationen entworfen und die Abläufe einzelner oder kooperierender Roboter geplant werden. Zur Steuerung der Roboter werden Rapid Programme genutzt. Diese werden in einer eigenen Programmiersprache geschrieben und beschreiben, welche Bewegungen die Roboter ausführen sollen. Die Roboterprogramme können sowohl von den realen Robotern ausgeführt als auch innerhalb von RobotStudio simuliert werden. Da in beiden Fällen die gleiche Robotersteuerung genutzt wird, ermöglicht die Simulation eine sehr exakte Abbildung der real durchgeführten Bewegung.

Die von RobotStudio genutzte Programmiersprache Rapid beschreibt die Bewegungen, die der Roboter ausführen soll, mit Hilfe verschiedener Kommandos. Ein einfaches Programm besteht aus einer Folge von Bewegungsbefehlen, die nacheinander ausgeführt werden. Ein solcher Befehl besteht aus verschiedenen Komponenten, wie

1. den Koordinaten des Punkts, der angefahren werden soll,
2. den Quadranten der Roboterglieder,
3. der Geschwindigkeit der Bewegung gemessen am TCP,
4. der Überschleifzone,
5. dem Werkzeug, gegenüber dem die Bewegung gemessen wird.

Alternativ können anstelle der Koordinaten des Zielpunkts für den TCP auch die Winkel der Gelenke für die Zielpose angegeben werden. Dies hat den Vorteil, dass es keine Uneindeutigkeiten mit den Quadranten der Roboterglieder gibt und dass keine relative Positionierung des Roboters bezüglich der Bahn in RobotStudio stattfinden muss.

Die Geschwindigkeiten, mit denen der TCP bewegt werden soll, wird in Rapid pro Bahnpunkt angegeben. Dabei gilt es jedoch zu beachten, dass die Geschwindigkeit, die an einem Punkt angegeben wird, auf der Strecke von dem Vorgänger- bis zu dem aktuellen Punkt gefahren wird. Dazu kommt, dass es eine Beschleunigungsphase gibt, bevor der Roboter die gewünschte Geschwindigkeit erreicht und eventuell eine Endschleunigungsphase, bevor der Zielpunkt erreicht wird. Die Ausprägung der beiden Beschleunigungsphasen sind insbesondere von den Überschleifzonen abhängig.

Bei den Überschleifzonen, siehe Abbildung 6.2, handelt es sich um Parameter, die angeben, wie exakt Bahnpunkte erreicht werden müssen. Wenn die Zone sehr klein gewählt wird, dann wird der Bahnpunkt exakt angefahren. Der Vorteil beim Überschleifen mit größeren Radien ist jedoch, dass der Roboter seine Geschwindigkeit weniger stark regulieren muss, um den Punkt zu erreichen und die Richtungsänderung für den nächsten Bahnabschnitt durchzuführen. Daher ist es gerade im Bereich des thermischen Spritzens durchaus sinnvoll, eine etwas größere Überschleifzone zu wählen, da sich eine Reduktion

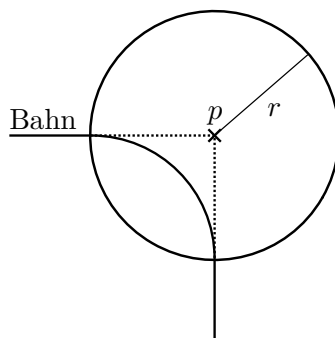


Abbildung 6.2: Visualisierung einer Überschleifzone am Punkt  $p$  mit Radius  $r$  nach [2]. Die tatsächlich abgefahrte Bahn (durchgezogene Linie) nutzt die Überschleifzone um Punkt  $p$  zur Reorientierung.

der Geschwindigkeit deutlich stärker auswirkt als eine leicht veränderte Position der Spritzpistole.

Mit Hilfe der in RobotStudio eingebauten Simulation ist es möglich, eine sehr exakte Vorhersage der tatsächlichen Roboterbewegung zu erhalten, siehe [1]. Der aktuelle Status des Roboters wird während des Simulierens in kurzen Zeitintervallen mit Hilfe eines Signalanalyse Systems festgehalten, so dass verschiedene Metriken des Roboters gespeichert werden können. Hier sind vor allem die Achsstellungen des Roboters sowie die Geschwindigkeiten des TCPs von Interesse, so dass diese Werte gespeichert werden. Die gespeicherten Werte können in das Bahnplanungsprogramm zurückgeführt und in eine Bahn konvertiert werden, so dass die tatsächlich ausgeführte Bahn dort simuliert werden kann, um die Beschichtung, die durch diese Bahn generiert wird, zu untersuchen.

## 6.4 Iterative Anpassung der Bahnen

Wie in Abschnitt 6.2 beschrieben, wird ein iteratives Verfahren angewendet, das mit Hilfe der Herstellersoftware RobotStudio eine Dynamikkorrektur der Bahn vornimmt. Die einzelnen Schritte werden im Folgenden genauer erläutert.

Die optionale Bahnglättung soll dafür sorgen, dass die Bahn keine starken Knicke oder Beschleunigungen aufweist. Diese sind besonders kritisch und sorgen dafür, dass der Roboter die Bahn nicht wie gefordert ausführen kann. Das normale Verhalten des Roboters in solchen Fällen ist die Verlangsamung der TCP-Geschwindigkeit, so dass dem Bahnverlauf im Rahmen der geforderten Überschleifgenauigkeiten gefolgt werden kann. Dieses Verhalten ist für das thermische Spritzen sehr unvorteilhaft, da eine Verlangsamung des Roboters zu sehr starken Überbeschichtungen und starken thermischen Varianzen führt. Durch die Bahnglättung werden die Knicke und Beschleunigungsspitzen reduziert und der Roboter ist eher in der Lage, dem Bahnverlauf ohne signifikante Anpassung der Geschwindigkeit zu folgen.

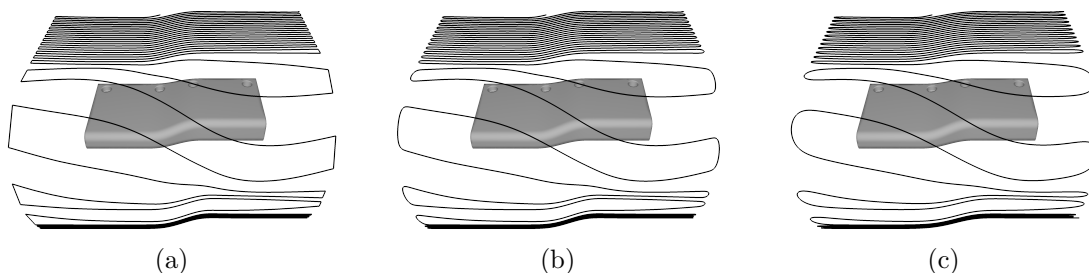


Abbildung 6.3: Vergleich verschiedener Glättungsstufen. (a) Originalbahn. (b) Geglättete Bahn mit B-Spline Grad 10. (c) Geglättete Bahn mit B-Spline Grad 100.

Zur Bahnglättung wird hier auf B-Splines (siehe [44]) zurückgegriffen. Diese sind in der Lage, ausgehend von einer Menge von Kontrollpunkten, eine approximierende, jedoch glatte Kurve zu erzeugen. Als Kontrollpunkte können die Bahnpunkte der ursprünglichen Bahn verwendet werden. Mit Hilfe des De Boor Algorithmus wird dann eine entsprechende approximierende Bahn erzeugt, indem neue Bahnpunkte generiert werden. Gesteuert wird das Verfahren außer über die Kontrollpunkte auch über den Grad des Splines. Über diesen wird der Bereich des ursprünglichen Kontrollpolygons angegeben, der Einfluss auf einen Punkt der resultierenden Kurve hat. Je höher der Grad ist, desto glatter wird die Kurve im Allgemeinen. In Abbildung 6.3 sind eine originale Bahn und zwei Glättungsstufen der Bahn mit unterschiedlichen B-Spline Graden dargestellt.

Ein wesentlicher Nachteil der Bahnglättung ist das approximative Verhalten von Splines. Die resultierende Kurve liegt in der Vereinigung der konvexen Hüllen von je Grad+1 aufeinanderfolgenden Punkten des Kontrollpolygons und *shrumpft* mit zunehmendem Grad. Dadurch weicht die Bahn bei hohem Grad relativ stark von der originalen Bahn ab und es ergeben sich Fehler in der Beschichtung, die korrigiert werden müssen. Dennoch ist es sinnvoll, eine gewisse Glättung durchzuführen, da die Bahnqualität dadurch generell, jedoch vor allem im Bereich der Roboterachsbeschleunigungen, steigt.

Die möglicherweise geglättete Bahn wird mit Hilfe der Bahnoptimierung für einige Schritte optimiert. Die Ausgabe der Bahnoptimierung ist eine Bahn in Form von dicht liegenden Bahnpunkten, die im nächsten Schritt durch die Dynamikkorrektur angepasst werden soll.

Auf Grund der Tatsache, dass in RobotStudio Roboterprogramme in Form von Rapid Programmen vorliegen müssen, wird die Bahn zunächst in ein Rapid Programm konvertiert. Das Rapid Programm besteht im Wesentlichen aus einer Folge von Bewegungsanweisungen. Diese werden durch eine Folge von Zielpunkten angegeben, die durch den Roboter nach und nach abgefahren werden. Auf Grund der Möglichkeit, dass einzelne Zielposen für den TCP mit mehreren Konfigurationen des Roboterarms erreicht werden können, wird darauf zurückgegriffen, anstelle der Koordinaten und Orientierungen des TCPs die Achsstellungen des Roboters für die Zielposen anzugeben. Daher werden mit Hilfe der inversen Kinematik für jeden Bahnpunkt die Achsstellungen des Roboters ermittelt. Diese werden anschließend im Rapid Programm gespeichert. Dies



hat ebenfalls den Vorteil, dass in der RobotStudio Station die Bahn nicht gegenüber dem Roboter positioniert werden muss. Zusätzlich zu den Achsstellungen müssen noch die Geschwindigkeiten der Bahnabschnitte angegeben werden. Diese werden aus den Zeiten berechnet, zu denen der Roboter einen Bahnpunkt erreichen soll. Es ist zu beachten, dass RobotStudio die Geschwindigkeit so interpretiert, dass diese auf dem Bahnstück vor dem angegebenen Punkt gefahren werden soll. Für jeden Punkt wird schließlich noch eine Überschleifzone angegeben. Diese wird großzügig gewählt, was den Vorteil hat, dass es ausreicht, den Punkt nur ungefähr zu erreichen. Dies ermöglicht dem Roboter, die Richtungsänderung für den nächsten Bahnabschnitt in einem relativ großen Bereich um den eigentlichen Bahnpunkt durchzuführen. Das Resultat hiervon ist, dass der Roboter seltener seine Geschwindigkeit reduziert, um die Richtungsänderung in der vorgegebenen Zone durchzuführen. Der Nachteil ist, dass die geforderten Bahnpunkte nicht völlig exakt angefahren werden. Für das thermische Spritzen ist jedoch eine ungewollte Änderung der Geschwindigkeit wesentlich unvorteilhafter als das Nichteinhalten der genauen Bahnposition. Als Letztes wird das Werkzeug gewählt, auf dass sich die Angaben der Bewegung beziehen. Hierbei handelt es sich einfach um die in der Station bereitgestellte Spritzpistole. Mit Hilfe dieser Angaben ist das Roboterprogramm vollständig und kann in RobotStudio zur Simulation genutzt werden.

Das geladene Roboterprogramm wird mit Hilfe der Simulation von RobotStudio ausgeführt, so dass die Bewegungen des Roboters durch die Robotersteuerung bestimmt werden. Mit Hilfe der zur Verfügung stehenden Signalanalyse in RobotStudio können die Bewegungsdaten des Roboters gespeichert werden. In diesem Fall werden die Achsstellungen und die TCP-Geschwindigkeiten festgehalten. Die Simulation speichert diese Werte in kleinen Zeitschritten (minimal 12 ms).

Die mitgeschnittenen Daten des Roboters können dazu genutzt werden, wieder eine Bahn nach der Definition aus Abschnitt 2.1 zu generieren. Dazu werden die Achsstellungen aus RobotStudio mit Hilfe der Vorwärtskinematik in Bahnpunkte konvertiert. Die Zeitpunkte des Erreichens der Bahnpunkte können aus den Daten direkt abgelesen werden.

Mit Hilfe des oben beschriebenen Verfahrens kann eine Bahn durch RobotStudio in den dynamisch zulässigen Bereich projiziert werden. Die Eingabebahn in RobotStudio wird von dem Roboter so gut wie möglich abgefahren und die tatsächlich ausgeführte Bahn wird zurück in die Bahnplanung überführt. Dort kann die tatsächlich ausgeführte Bahn mit Hilfe der Beschichtungssimulation überprüft werden. Wenn die Bahn so stark verändert wurde, dass der Schichtauftrag nicht akzeptabel ist, wird das gesamte Verfahren iteriert. Dadurch ergeben sich nach und nach ausführbare Bahnen, die eine gute Beschichtung liefern.

## 6.5 Evaluation

In diesem Abschnitt wird das oben beschriebene Verfahren evaluiert. Ziel hierbei ist es, eine Aussage über die Güte der endgültigen Bahnen treffen zu können. Da diese bereits

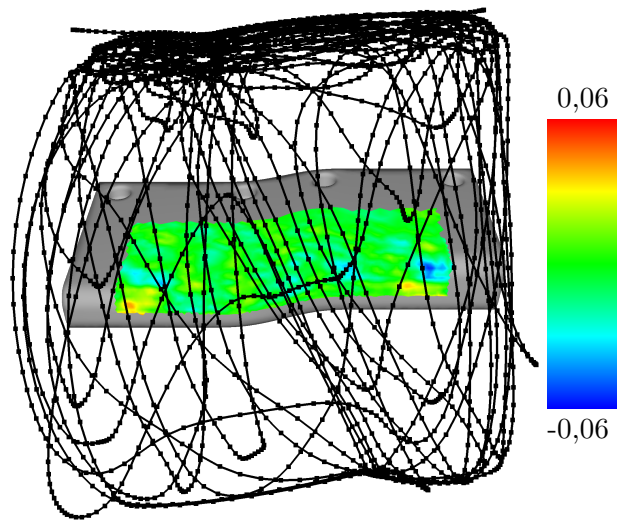


Abbildung 6.4: Visualisierung der Bahn und der Beschichtung auf dem Werkstück A nach 10 Iterationen des Optimierungsverfahrens mit der Dynamikkorrektur. Die Skala ist an den maximalen Fehler angepasst.

als Roboterprogramme in RobotStudio vorliegen, können sie direkt verwendet werden und stellen somit das Gesamtergebnis des Verfahrens dar.

### 6.5.1 Testdaten und Qualitätsmerkmale

Die Evaluation wird mit jeweils einer durch die Initialbahnplanung erstellten Bahn auf den beiden Werkstücken A und B (siehe Abbildung 4.13) durchgeführt. Dies stellt den normalen Anwendungsfall des Gesamtprozesses dar. Als Qualitätsmerkmale werden die üblichen Zielfunktionen verwendet, so dass der mittlere und maximale Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ), die mittleren und maximalen thermischen Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), die mittleren und maximalen Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$ ) und der Overspray ( $E_o(\mathcal{X})$ ) betrachtet werden.

### 6.5.2 Evaluation auf Werkstück A

Für den ersten Testfall wird eine durch die Initialbahnplanung generierte Bahn für das Werkstück A genutzt. Diese wird als Eingabe für das oben beschriebene Verfahren verwendet, so dass die Schritte der Bahnoptimierung und der Dynamikkorrektur mittels RobotStudio iteriert werden. Insgesamt werden auf diese Weise 10 Iterationen ausgeführt. In Abbildung 6.4 ist die Ergebnisbahn sowie die von der Bahn generierte Beschichtung auf dem Werkstück ermittelt durch die externe Simulation *SimSpray* dargestellt. Man erkennt den glatten Bahnverlauf und die gute Beschichtung mit einem maximalen

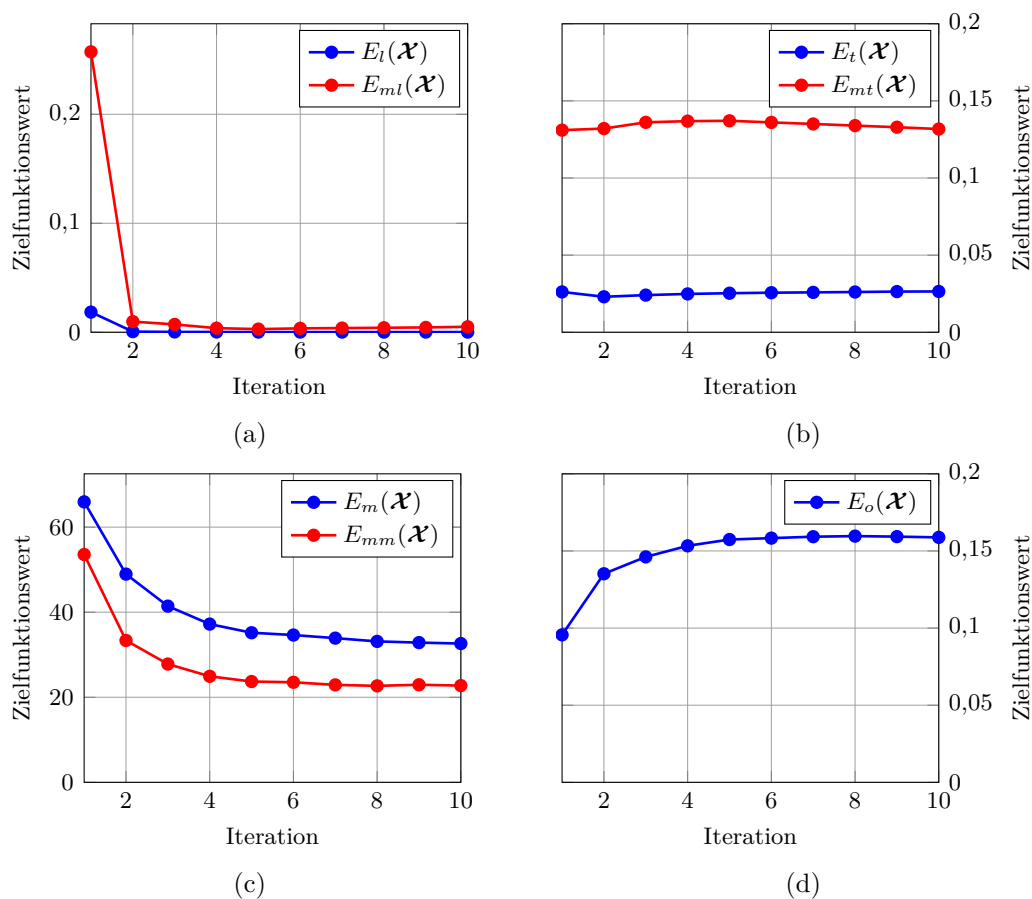


Abbildung 6.5: Entwicklung der Zielfunktionswerte während der Optimierung mit der Dynamikkorrektur auf Werkstück A. Die Werte werden jeweils nach der Dynamikkorrektur ermittelt. (a) zeigt die mittleren und maximalen Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ), (b) die mittleren und maximalen thermischen Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), (c) die mittleren und maximalen Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$ ) und (d) den Overspray ( $E_o(\mathcal{X})$ ).

Fehler von 5,5 %. Dies ist ein sehr gutes Ergebnis, da die initialgeplante Bahn nach der ersten Bahnoptimierung und der ersten Dynamikkorrektur einen maximalen Fehler von über 51 % aufwies. In Abbildung 6.5a ist der Verlauf der mittleren sowie maximalen Beschichtungsfehler während der Optimierung dargestellt. Man erkennt die rapide Verbesserung nach der ersten Iteration. Im weiteren Verlauf verbessern sich die Fehler bis zu dem Erreichen eines relativ stabilen Wertes. Der maximale Fehler liegt im Bereich der Krümmung des Werkstücks. Dies ist nicht verwunderlich, da hier die Stellen mit den stärksten Achsbeschleunigungen liegen. Die Verbesserungen durch die Bahnoptimierung werden demnach durch die Dynamikkorrektur zum Teil zunichte gemacht.

Die Roboterachsbeschleunigungen nehmen im Verlauf des Verfahrens immer weiter ab, wie in Abbildung 6.5c gut zu erkennen ist. Hieraus folgt, dass die Dynamik im Verlaufe des Verfahrens immer weiter verbessert wird. Dies liegt vor allem an der Projektion in den dynamisch zulässigen Bereich durch die Dynamikkorrektur, aber zum Teil auch an der Bahnoptimierung, die die Roboterachsbeschleunigungen ebenfalls verbessern kann. Gegen Ende des Verfahrens nehmen die Werte nicht mehr stark ab, so dass das Verfahren schon nach dieser geringen Anzahl an Iterationen eine starke Konvergenz zeigt.

Die thermischen Varianzen im Verlauf der Optimierung sind in Abbildung 6.5b dargestellt. Man erkennt sehr gut, dass die Varianzen recht stabil sind und sich kaum verändern. Dies ist das gewünschte Verhalten, so dass sich trotz der Nichtberücksichtigung dieser Zielfunktionskomponente in der Optimierungsschleife keine Verschlechterung ergibt.

In Abbildung 6.5d sind die Werte für den Overspray nach jeder Iteration dargestellt. Man erkennt den Anstieg des Oversprays in den ersten Iterationen. Dieser Anstieg resultiert daraus, dass die Dynamikkorrektur für eine Verlangsamung der Bahn sorgt, was zu einer systematischen Überbeschichtung führt. Als Korrekturmaßnahme werden die Bahnpunkte von der Bahnoptimierung von der funktionalen Oberfläche weggeschoben, so dass mehr Overspray entsteht. Die Vergrößerung des Oversprays ist im Wesentlichen jedoch auf die ersten Iterationen beschränkt, so dass am Ende insgesamt ein nicht übermäßiger Overspray entsteht.

Insgesamt sorgt das Verfahren dafür, dass die initialgeplante Bahn nach einigen Schritten der Bahnoptimierung und Dynamikkorrektur zu sehr guten Ergebnissen führt. Der Beschichtungsfehler ist mit unter 6 % auf einer dynamisch zulässigen Bahn gut, die thermischen Varianzen sind gerade im Vergleich zu herkömmlichen quasi-konturparallelen Bahnen sehr gering und der Overspray ist ebenfalls klein.

### 6.5.3 Evaluation auf Werkstück B

Der zweite Testfall wird auf dem komplexeren Werkstück B ausgeführt. Dieses hat deutlich kleinere Radien in den Krümmungen der Freiformoberfläche. Diese sind bezüglich der Größe des Footprints bei dem Libo-Prozess sehr klein, so dass das Verfahren für dieses Werkstück eigentlich ungeeignet ist. Daher stellt dieser Testfall eine Art Extremfall dar.

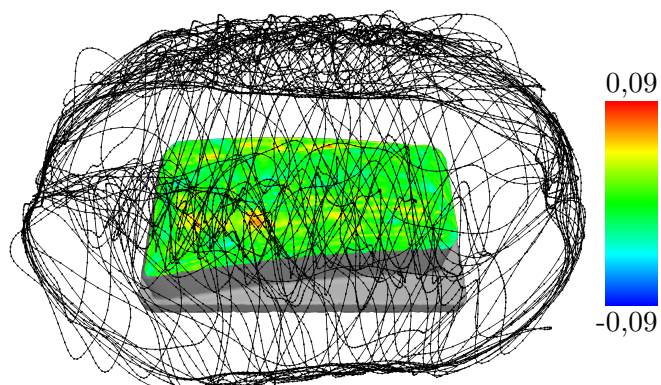


Abbildung 6.6: Visualisierung der Bahn und der Beschichtung auf dem Werkstück B nach 10 Iterationen des Optimierungsverfahrens mit der Dynamikkorrektur. Die Skala ist an den maximalen Fehler angepasst.

In Abbildung 6.6 ist die optimierte Bahn und die von dieser erzeugten Beschichtung auf der Oberfläche nach insgesamt 10 Iterationen der Optimierung dargestellt. Auch hier ist das Ergebnis, wie bei dem Testfall mit Werkstück A, eine glatte Bahn. Die Beschichtung ist insgesamt gut, weist jedoch einige Fehler im Bereich der starken Krümmungen der Oberfläche auf. Der maximale Fehler ist dennoch kleiner als 8,4 %. Der Großteil der Oberfläche weist geringere Fehler auf. Insgesamt hat sich der Fehler sehr stark reduziert, da die ursprüngliche initialgeplante, optimierte und anschließend in den dynamisch zulässigen Bereich projizierte Bahn einen maximalen Fehler von über 56 % aufwies. In Abbildung 6.7a ist der Verlauf der Beschichtungsfehler über die Iterationen aufgeführt. Der rapide Abfall der Fehler am Anfang des Verfahrens ist gut zu erkennen. Der mittlere Fehler nimmt auch im späteren Verlauf immer weiter ab, während der maximale Fehler sein Minimum in der vierten Iteration annimmt und danach leicht steigt. Dies liegt daran, dass der mittlere Fehler minimiert wird und der maximale Fehler nicht explizit in der Optimierung berücksichtigt wird. Einzelne Ausreißer fallen demnach nicht stark ins Gewicht der Zielfunktion.

Die in Abbildung 6.7c dargestellten Roboterachsbeschleunigungen nehmen im Verlauf des Verfahrens immer weiter ab. Das Verhalten ist das gleiche wie für den Testfall mit Werkstück A und resultiert aus der Projektion der Bahn in den dynamisch zulässigen Bereich sowie aus der Optimierung dieser Werte durch die Bahnoptimierung.

Die thermischen Varianzen nehmen sowohl im Mittel als auch im Maximum im Verlauf des Verfahrens leicht zu, wie in Abbildung 6.7b zu sehen ist. Dies liegt daran, dass durch die starken Krümmungen auf der Oberfläche auch starke Bewegungen des Roboters notwendig sind. Durch die Dynamikkorrektur werden die Stellen mit hohem Bewegungsanspruch an den Roboter langsamer abgefahren, wenn diese ansonsten außerhalb des dynamisch zulässigen Bereichs lägen. Diese Verlangsamungen sorgen für die Zunahme der thermischen Varianzen.

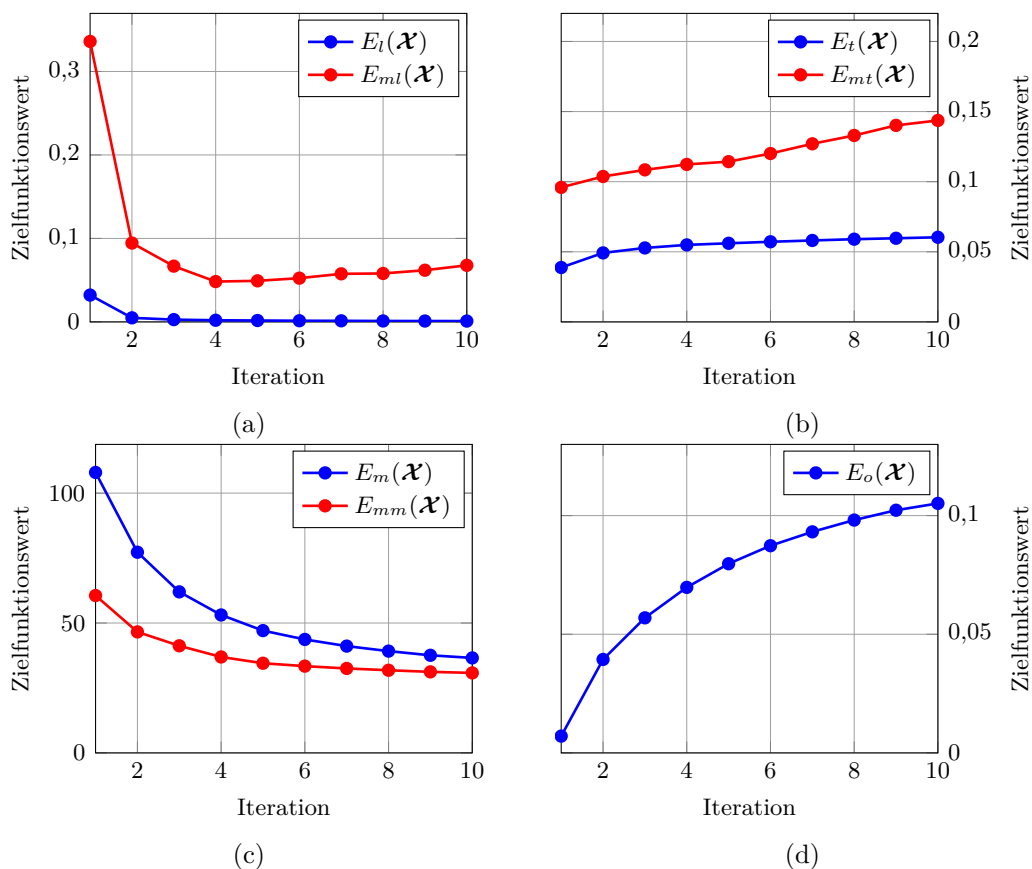


Abbildung 6.7: Entwicklung der Zielfunktionswerte während der Optimierung mit der Dynamikkorrektur auf Werkstück B. Die Werte werden jeweils nach der Dynamikkorrektur ermittelt. (a) zeigt die mittleren und maximalen Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{ml}(\mathcal{X})$ ), (b) die mittleren und maximalen thermischen Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), (c) die mittleren und maximalen Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$ ) und (d) den Overspray ( $E_o(\mathcal{X})$ ).

Der Overspray nimmt, wie in Abbildung 6.7d dargestellt ist, ebenso wie die thermischen Varianzen im Verlauf des Verfahrens zu. Dies liegt wie im Testfall mit Werkstück A daran, dass die Bahn durch die Dynamikkorrektur grundsätzlich verlangsamt wird. Dies führt zu der bereits angesprochenen systematischen Überbeschichtung auf der Oberfläche. Diese wird durch das Verschieben der Bahnpunkte weg von der funktionalen Oberfläche korrigiert, so dass die Schichtdicke erreicht wird und im Gegenzug ein größerer Overspray erzeugt wird. Dieses Verhalten der Optimierung ließe sich dadurch reduzieren, dass schon die initiale Bahn mit einer geringeren Grundgeschwindigkeit erzeugt wird.

Auf dem komplexeren Werkstück B sind die Ergebnisse erwartungsgemäß etwas schlechter als auf dem einfacher aufgebauten Werkstück A, jedoch immer noch sehr gut. Um dies zu untermauern, soll das Ergebnis mit einer herkömmlichen quasi-konturparallelen Bahn

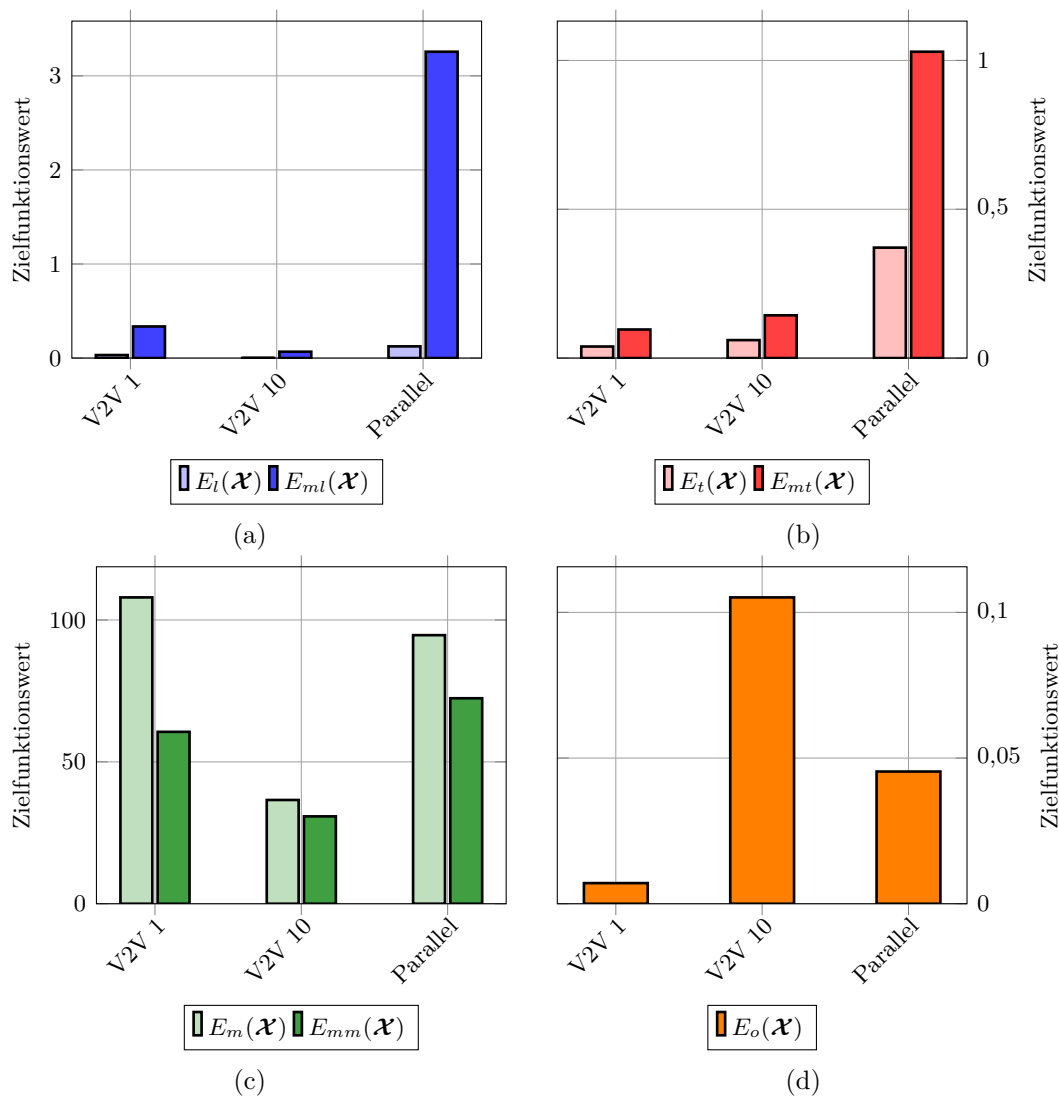


Abbildung 6.8: Zielfunktionswerte einer Punkt-zu-Punkt Bahn (V2V 1) sowie einer herkömmlichen quasi-konturparallelen Bahn (Parallel) nach der Dynamikkorrektur. Außerdem werden die Zielfunktionswerte der Punkt-zu-Punkt Bahn nach 10 Iterationen der Optimierung gezeigt (V2V 10). (a) zeigt die mittleren und maximalen Beschichtungsfehler ( $E_l(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), (b) die mittleren und maximalen thermischen Varianzen ( $E_t(\mathcal{X})$  und  $E_{mt}(\mathcal{X})$ ), (c) die mittleren und maximalen Roboterachsbeschleunigungen ( $E_m(\mathcal{X})$  und  $E_{mm}(\mathcal{X})$ ) und (d) den Overspray ( $E_o(\mathcal{X})$ ).

verglichen werden, die nicht optimiert, sondern mit einem herkömmlichen Verfahren erstellt wurde. Hierzu wird eine quasi-konturparallele Bahn entlang der langen Kante des Werkstücks erzeugt, siehe Abbildung 4.32a. Diese wird schließlich mit Hilfe von RobotStudio in den dynamisch zulässigen Bereich projiziert. In Abbildung 6.8 sind die Zielfunktionswerte für die Bahnen nach der Projektion dargestellt. Wie leicht zu sehen ist, sind fast alle Zielfunktionswerte für die Punkt-zu-Punkt Bahn besser als die der quasi-konturparallelen Bahn. Vor allem der Beschichtungsfehler sowie die thermischen Varianzen sind für die quasi-konturparallele Bahn deutlich schlechter. Dies liegt daran, dass durch die Projektion in den dynamisch zulässigen Bereich die Bahngeschwindigkeiten recht stark reduziert werden müssen. Dadurch reduzieren sich die Roboterachsbeschleunigungen und sind auf einem vergleichbaren Niveau wie die der Punkt-zu-Punkt Bahn. Der Overspray ist ebenfalls deutlich schlechter für die quasi-konturparallele Bahn. Zum Vergleich sind auch die Werte der optimierten Punkt-zu-Punkt Bahn dargestellt. Man erkennt, dass im Vergleich zur nicht optimierten Punkt-zu-Punkt Bahn die Beschichtung und die Roboterachsbeschleunigungen nochmals besser sind, während die thermischen Varianzen minimal und der Overspray deutlich schlechter geworden sind. Würde die quasi-konturparallele Bahn ebenfalls optimiert werden, um die Beschichtungsfehler zu reduzieren, würde der Overspray und eventuell die thermischen Varianzen ebenfalls schlechter werden.

Zusammenfassend kann festgestellt werden, dass die Optimierung auch auf dem komplexen Werkstück sehr gute Ergebnisse liefert. Diese sind auf Grund der Komplexität der Freiformoberfläche nicht ganz so gut wie bei Werkstück A, jedoch ist das Gesamtergebnis verglichen mit einer Bahn, die auf herkömmliche Weise erstellt und in den dynamisch zulässigen Bereich projiziert wird, deutlich besser.



# Zusammenfassung und Ausblick

Zum Abschluss der Arbeit soll noch eine Zusammenfassung der wichtigsten in dieser Arbeit präsentierten Methoden und Ergebnisse sowie ein Ausblick auf Möglichkeiten zur weiteren Forschung gegeben werden.

## 7.1 Zusammenfassung

Diese Arbeit befasst sich mit der Erzeugung und Optimierung von neuartigen Bahnen für Industrieroboter beim thermischen Spritzen auf komplexen, d. h. insbesondere verschiedentlich gekrümmten Freiformoberflächen unter besonderer Berücksichtigung der thermischen Verhältnisse in dem Werkstück. Ein Alleinstellungsmerkmal des präsentierten Systems ist der modulare Aufbau, der vor allem eine in diesem Bereich unübliche Trennung zwischen der Initialbahnplanung und der Bahnoptimierung vorsieht. Die Basis des Gesamtsystems bilden verschiedene Simulationskomponenten, wie die Beschichtungssimulation, die thermische Simulation und die Robotersimulation. Auch die Nutzung externer Simulationstools ist in dem System vorgesehen. Durch die Anbindung einer Robotersteuerungssoftware können die entworfenen Bahnen in den dynamisch zulässigen Bereich projiziert, darauf aufbauend optimiert und anschließend im realen Prozess verwendet werden.

In Kapitel 3 wurde eine Simulationsumgebung für thermisches Spritzen mit verschiedenen Simulationskomponenten vorgestellt. Die Auftragssimulation befasst sich mit der Vorhersage der Schichtdicke auf dem Werkstück bei der Ausführung einer Bahn. Hierzu wurde ein Simulationskonzept auf Makroebene genutzt, das die wesentlichen Komponenten wie die Materialverteilung im Spritzstrahl oder die Abhängigkeit der Beschichtung von dem Einschlagwinkel der Partikel berücksichtigt. Als konkrete Materialverteilungsfunktionen

wurden einfache, optional auch überlagerte Gaußfunktionen verwendet. Das Auftragsmodell ist unter anderem darauf ausgelegt, die für die Bahnoptimierung benötigten Gradienten durch die Ableitung der Funktionen bestimmen zu können. Eine Methode zur Kalibrierung des Modells wurde vorgestellt und genutzt, um die Modellparameter gegenüber realen Footprints zu kalibrieren. Das Modell der thermischen Simulation bildet viele physikalische Prozesse des Wärmetransfers ab. Hierzu zählen die Konvektion, die Diffusion und die Strahlung. Auf ein dreidimensionales Diffusionsmodell für den Wärmetransfer ins Innere des Werkstücks wurde zu Gunsten geringerer Berechnungszeiten verzichtet und ein einfacher Wärmeausgleich genutzt. Das Modell wurde an einer komplexen Simulation kalibriert und die Ergebnisse zeigen eine qualitativ sehr gute Übereinstimmung. Mit Hilfe dieses Modells ist es möglich, die thermischen Varianzen mit geringem Berechnungsaufwand zu bestimmen.

Die Initialbahnplanung aus Kapitel 4 erzeugt flächenüberdeckende Bahnen auf einem Werkstück unter Berücksichtigung verschiedener Qualitätsmerkmale. Dazu wurden die Bahnen über flexible Bahnstrukturen repräsentiert, darunter eine an herkömmliche Verfahren angelehnte, quasi-konturparallele Bahnstruktur, jedoch insbesondere neuartige Strukturen, wie die Rand-zu-Rand Bahnen und die Punkt-zu-Punkt Bahnen. Die Qualität der Bahnen wurde über verschiedene Zielfunktionen bewertet, die neben der Schichtqualität vor allem die thermischen Varianzen berücksichtigen, welche bisher nur selten in Betracht gezogen wurden, obwohl sie großen Einfluss auf die endgültige Schichtqualität haben. Weitere praxisrelevante Zielkriterien, wie die Roboterachsbeschleunigungen und der Overspray, welcher das Material beschreibt, das nicht auf der funktionalen Fläche abgelagert wird, wurden ebenfalls beachtet. Das Problem der Initialbahnplanung wurde als mehrkriterielles Optimierungsproblem formuliert und mit Hilfe eines Evolutionären Algorithmus optimiert. Verschiedene Varianten für die Operatoren des Evolutionären Algorithmus wurden verwendet und gegeneinander evaluiert. Hieraus wurde die Kombination von Operatoren bestimmt, mit der der Algorithmus mit hoher Konvergenzgeschwindigkeit strukturell gute Bahnen für den anschließenden Bahnoptimierungsprozess erzeugt.

Die Bahnoptimierung aus Kapitel 5 wird für die Verbesserung vorhandener Bahnen bezüglich der Beschichtungsfehler und der Ausführbarkeit mit Robotern verwendet. Ein neuartiges Konzept zur kombinierten Anwendung des in der Arbeit entwickelten, analytischen Auftragsmodells mit einer externen Blackbox Simulation wurde verwendet, um die Bahnen mit Hilfe des Verfahrens der nichtlinearen konjugierten Gradienten zu optimieren. Die Fehler wurden hierbei über die externe Simulation und die Gradienten über das analytische Auftragsmodell bestimmt. Um Simulationsläufe zu sparen, wurde eine heuristische Schrittweitenbestimmung durchgeführt. In der Evaluation wurden verschiedene Gradientenfunktionen verglichen und Fallbeispiele aufgeführt. Die Verwendung der Bahnoptimierung beschränkt sich nicht nur auf die Optimierung der Bahnen, die von der Initialbahnplanung erstellt worden sind, sondern kann ebenfalls genutzt werden, um bereits erstellte Bahnen an andere Spritzprozesse oder ähnliche Werkstückgeometrien anzupassen. Hierdurch lässt sich der erhebliche Aufwand zur Generierung neuer Bahnen stark reduzieren.

Zum Abschluss der Arbeit wurde in Kapitel 6 ein Verfahren vorgestellt, das die bisher unberücksichtigte Roboterdynamik in das System miteinbezieht. Dazu wurde eine Dynamikkorrektur präsentiert, die die Bahnen mit Hilfe einer Roboterherstellersoftware in den dynamisch zulässigen Bereich projiziert. Diese Projektion wurde in einer weiteren Optimierungsschleife alternierend mit der Bahnoptimierung genutzt, um eine dynamisch zulässige Bahn zu erzeugen, die sehr gute Ergebnisse bezüglich der Qualitätsmaße liefert.

Insgesamt wurde ein vollständiges System zur Erstellung, Optimierung und Simulation von Roboterbahnen für das thermische Spritzen vorgestellt. Durch das neuartige Konzept, das eine Trennung der Bahnplanung in eine Initialbahnplanung und eine Bahnoptimierung vorsieht, können sehr gute Bahnen auch für komplexe Werkstücke erzeugt werden. Durch die Nutzung flexibler Bahnstrukturen und die Optimierung der Bahnen können die verschiedenen Qualitätsmerkmale und insbesondere die meist nicht beachteten thermischen Varianzen sehr gut erfüllt werden.

## 7.2 Ausblick

Die Erkenntnisse, die in der vorliegenden Arbeit gewonnen wurden, können als Grundlage für weitere Forschungen dienen.

Im Bereich der Beschichtungssimulation kann die Auswirkung bei der Nutzung komplexerer Modelle untersucht werden. Hierzu bieten sich vor allem footprintbasierte Modelle an. Die Bestimmung der Gradienten für die Bahnoptimierung müsste dann entsprechend angepasst werden. Die thermische Simulation stützt sich auf physikalische Prozesse, nutzt jedoch Vereinfachungen um Rechenzeit zu sparen. Auch wenn die Abweichungen der Simulationsergebnisse zu komplexeren Modellen gering sind, wie durch die Evaluation gezeigt wurde, könnten weitere Verfahren zur Verbesserung der Vorhersagequalität sowie deren Auswirkungen auf die Bahnplanung untersucht werden.

Das Konzept der Initialbahnplanung kann in der angegebenen Form auf beliebige Bahnstrukturen übertragen werden. Drei unterschiedliche Bahnstrukturen wurden in dieser Arbeit untersucht und miteinander verglichen. Die Analyse hat gezeigt, dass jede Bahnstruktur eigene Vor- und Nachteile hat. Daher könnten weitere flächenüberdeckende Strukturen, wie zum Beispiel Spiralbahnen, untersucht werden, um einen Vergleich zu den vorhandenen Strukturen zu ziehen. Die Nutzung eines evolutionären Algorithmus hat sich als gute Lösung herausgestellt. Die Untersuchung verschiedener Operatoren hat Einblicke in die Funktionsweise der Optimierung geliefert. Andere Methoden, vor allem weitere Metaheuristiken, könnten jedoch in Betracht gezogen werden, um zu untersuchen, ob mit diesen Verfahren Bahnen gleicher Qualität mit weniger Iterationen erzeugt werden können. Des Weiteren sind durch eine Evaluation Grenzwerte für die minimalen Auflösungen der Bahnen und Werkstücknetze gefunden worden. Eine Methodik zur adaptiven Anpassung der Auflösungen könnte entwickelt werden, die in der Lage ist, die Auflösung im Verlauf der Optimierung anzupassen, so dass schnell eine grobe

Bahnstruktur und anschließend feinere Verbesserungen erzeugt werden können. Dadurch könnten vor allem weitere Laufzeitvorteile erreicht werden.

Die Bahnoptimierung nutzt nichtlineare konjugierte Gradienten als Basis für die Optimierung der Bahnen. Der Einsatz einer heuristischen Schrittweitenbestimmung hat sich als praktikabel herausgestellt. Dennoch birgt der Einsatz dieses Verfahrens das Problem, dass eine Konvergenz nicht garantiert ist. Hier könnten entsprechend weitere Verfahren zur Schrittweitenbestimmung mit wenigen Simulationen untersucht werden, um die Konvergenz des Verfahrens zu verbessern. Des Weiteren ist die Bahnoptimierung nicht in der Lage, die thermischen Varianzen explizit zu berücksichtigen, da die externe Simulation keine entsprechenden Daten liefert und auch keine Gradienten zur Verfügung stehen. Hierin liegt die Gefahr einer Verringerung der Qualität der Bahnen in diesem Bereich. Auch wenn durch die explizite Nutzung dieses Kriteriums keine wesentliche Verbesserung der Varianzen zu erwarten ist, da nur lokale Anpassungen an der Bahn durchgeführt werden, könnte die Gefahr einer Verschlechterung der thermischen Varianzen reduziert werden. Eine andere mögliche Erweiterung der Bahnoptimierung ist die Hinzunahme der Roboterplatzierung in die Optimierung. Der Vorteil wäre, dass durch eine simultane Platzierung des Roboters bessere Roboterachsbeschleunigungen erreicht und gleichzeitig eine bessere Platzierung bezüglich der konkreten Bahn erzeugt werden könnte. Der Nachteil wäre die deutlich vergrößerte Komplexität der Optimierung, da eine Menge weiterer Parameter in die Optimierung aufgenommen werden müssten.

Die Nutzung einer Herstellersoftware von Robotersystemen zur Projektion der Bahnen in den dynamisch zulässigen Bereich hat sich als praktikabel erwiesen. Als Alternative zur Projektion wäre die Nutzung von Gradienten denkbar. Hierzu müssten die entsprechenden Gradienten der Bahnparameter bezüglich der Roboterdynamik bestimmt werden, welche die Bahn nicht nur in den dynamisch zulässigen Bereich bringen, sondern die Roboterdynamik darüber hinaus noch weiter verbessern können. Ein weiterer Vorteil dieses Verfahrens wäre die einfache Integration der neuen Gradienten in das vorhandene Bahnoptimierungssystem. Ein weiterer Ansatzpunkt zur Verbesserung des Verfahrens wäre die Nutzung von alternativen Glättungsmethoden, die nicht zum Schrumpfen der Bahnen führen. Dadurch würde die Bahnstruktur weniger verändert, so dass die Qualitätsverluste durch die Bahnglättung vor allem im Bereich der Beschichtung verringert würden.



# Ableitungen

## A.1 Bahnbasierte Optimierung

Im Folgenden werden die Ableitungen für die bahnbasierte Optimierung angegeben.

### A.1.1 Ableitungen der Zielfunktion

Zunächst sollen die partiellen Ableitungen von  $\tilde{z}_i$  aus Gleichung (5.29) gebildet werden. Nach Definition von  $\tilde{z}_i$  aus Gleichung (3.1) ergibt dies:

$$\begin{aligned}\nabla_{\mathbf{x}_j} \tilde{z}_i &= \left( \nabla_{\mathbf{x}_j} a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_M(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \right. \\ &\quad + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,q}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\mathbf{x}_j} q(\mathbf{v}_i, \boldsymbol{\rho}_j) \\ &\quad + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,r}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\mathbf{x}_j} r(\mathbf{v}_i, \boldsymbol{\rho}_j) \\ &\quad \left. + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,d}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\mathbf{x}_j} d(\mathbf{v}_i, \boldsymbol{\rho}_j) \right) \cdot \frac{t_{j-1} + t_j}{2},\end{aligned}\tag{A.1}$$

sowie

$$\begin{aligned}\nabla_{\phi_j} \tilde{z}_i &= \left( a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,q}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\phi_j} q(\mathbf{v}_i, \boldsymbol{\rho}_j) \right. \\ &\quad \left. + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,r}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\phi_j} r(\mathbf{v}_i, \boldsymbol{\rho}_j) \right) \cdot \frac{t_{j-1} + t_j}{2},\end{aligned}\tag{A.2}$$

und

$$\begin{aligned}\nabla_{t_j} \tilde{z}_i &= \frac{1}{2} \cdot a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_M(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \\ &\quad + \frac{1}{2} \cdot a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_{j+1}) \cdot z_M(q(\mathbf{v}_i, \boldsymbol{\rho}_{j+1}), r(\mathbf{v}_i, \boldsymbol{\rho}_{j+1}), d(\mathbf{v}_i, \boldsymbol{\rho}_{j+1})).\end{aligned}\tag{A.3}$$

In diesen Gleichungen kommen weitere partielle Ableitungen vor, die bestimmt werden müssen. Zunächst sollen die Ableitungen nach  $\mathbf{x}_j$  betrachtet werden. Für die Ableitung von  $a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j)$  nach  $\mathbf{x}_j$  ergibt sich

$$\nabla_{\mathbf{x}_j} a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) = \frac{\|\mathbf{x}_j(\boldsymbol{\rho}_j) - \mathbf{v}_i\| \cdot \mathbf{n}_i - a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j)(\mathbf{x}_j(\boldsymbol{\rho}_j) - \mathbf{v}_i)}{d(\mathbf{v}_i, \mathbf{x}_j(\boldsymbol{\rho}_j))^2}. \quad (\text{A.4})$$

Die beiden Ableitungen von  $q(\mathbf{v}_i, \boldsymbol{\rho}_j)$  und  $r(\mathbf{v}_i, \boldsymbol{\rho}_j)$  nach  $\mathbf{x}_j$  können über

$$\begin{aligned} \nabla_{\mathbf{x}_j} q(\mathbf{v}_i, \boldsymbol{\rho}_j) = & - \frac{\left(-\mathbf{R}(\boldsymbol{\rho}_j)_q^\top\right)^\top \left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s}{\left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s^2} \\ & - \frac{\left(\mathbf{R}(\boldsymbol{\rho}_j)_s^\top\right)^\top \left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_q}{\left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s^2} \end{aligned} \quad (\text{A.5})$$

und

$$\begin{aligned} \nabla_{\mathbf{x}_j} r(\mathbf{v}_i, \boldsymbol{\rho}_j) = & - \frac{\left(-\mathbf{R}(\boldsymbol{\rho}_j)_r^\top\right)^\top \left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s}{\left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s^2} \\ & - \frac{\left(\mathbf{R}(\boldsymbol{\rho}_j)_s^\top\right)^\top \left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_r}{\left(\mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j))\right)_s^2} \end{aligned} \quad (\text{A.6})$$

bestimmt werden, wobei  $\mathbf{R}(\boldsymbol{\rho}_j)$  die Rotationsmatrix ist, die in Gleichung (3.7) eingeführt wurde. Die Indizes  $q$ ,  $r$  und  $s$ , welche an den Matrizen genutzt werden, stehen für die erste, zweite und dritte Zeile der jeweiligen Matrix nach der Ausführung einer eventuellen Transponation. Schließlich fehlt noch die Ableitung von  $d(\mathbf{v}_i, \boldsymbol{\rho}_j)$  nach  $\mathbf{x}_j$ , welche über

$$\nabla_{\mathbf{x}_j} d(\mathbf{v}_i, \boldsymbol{\rho}_j) = - \frac{\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)}{\|\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)\|} \quad (\text{A.7})$$

gegeben ist.

Nun werden die Ableitungen nach  $\phi_j$  bestimmt, wobei die Winkel der drei Rotationen einzeln betrachtet werden. Für  $q(\mathbf{v}_i, \boldsymbol{\rho}_j)$  und  $r(\mathbf{v}_i, \boldsymbol{\rho}_j)$  ergeben sich daher die Gleichungen

$$\begin{aligned} \nabla_{\phi_{j,1}} q(\mathbf{v}_i, \boldsymbol{\rho}_j) = & - \left( \left( \nabla_{\phi_{j,1}} \mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_q \cdot \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s - \right. \\ & \left. \left( \nabla_{\phi_{j,1}} \mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s \cdot \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_q \right) \cdot \frac{1}{\left( \mathbf{R}(\boldsymbol{\rho}_j)^\top(\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s^2} \end{aligned} \quad (\text{A.8})$$

und

$$\begin{aligned} \nabla_{\phi_{j,1}} r(\mathbf{v}_i, \boldsymbol{\rho}_j) = & - \left( \left( \nabla_{\phi_{j,1}} \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_r \cdot \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s - \right. \\ & \left. \left( \nabla_{\phi_{j,1}} \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s \cdot \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_r \right) \cdot \frac{1}{\left( \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s^2} \end{aligned} \quad (\text{A.9})$$

wobei die Berechnungen für  $\nabla_{\phi_{j,2}}$  und  $\nabla_{\phi_{j,3}}$  äquivalent zu der Berechnung von  $\nabla_{\phi_{j,1}}$  sind. Da es sich bei der Matrix  $\mathbf{R}(\boldsymbol{\rho}_j)$  um eine einfache Rotationsmatrix handelt, die hauptsächlich Sinus- und Kosinusfunktionen in den einzelnen Komponenten benutzt, sind die Ableitungen nach den Winkeln  $\phi_{j,1}$ ,  $\phi_{j,2}$  und  $\phi_{j,3}$  sehr einfach und werden hier nicht einzeln aufgeführt.

Als Letztes sollen die verbleibenden Ableitungen für  $z_{M,q}$ ,  $z_{M,r}$  und  $z_{M,d}$  bestimmt werden, die durch das konkrete Beschichtungsmodell gegeben sind. Diese ergeben sich dann zu

$$z_{M,q}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) = \sum_{k=0}^K z_{M,k,q}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)), \quad (\text{A.10})$$

$$z_{M,r}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) = \sum_{k=0}^K z_{M,k,r}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)), \quad (\text{A.11})$$

$$z_{M,d}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) = \sum_{k=0}^K z_{M,k,d}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \quad (\text{A.12})$$

mit

$$\begin{aligned} z_{M,k,q}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) = & - \left( \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{1,k}^2} + \frac{2r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{3,k}} \right) \\ & \cdot z_{M,k}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)), \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} z_{M,k,r}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) = & - \left( \frac{2r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{2,k}^2} + \frac{2q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)}{\sigma_{3,k}} \right) \\ & \cdot z_{M,k}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)), \end{aligned} \quad (\text{A.14})$$

$$\begin{aligned} z_{M,k,d}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) = & - \frac{2}{d(\mathbf{v}_i, \boldsymbol{\rho}_j)} \\ & \cdot z_{M,k}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)). \end{aligned} \quad (\text{A.15})$$

Damit sind alle notwendigen Ableitungen von  $E_l(\boldsymbol{\mathcal{X}})$  bestimmt. Im nächsten Abschnitt wird die Funktion  $E_m(\boldsymbol{\mathcal{X}})$ , die zur Bahnglättung verwendet wird, betrachtet.

### A.1.2 Ableitungen der Glättungsterme

In diesem Abschnitt werden die Ableitungen der Funktion  $E_m(\boldsymbol{x})$  aus Gleichung (5.10) für das bahnbasierte Verfahren angegeben. Dazu werden zunächst die Ableitungen nach  $\mathbf{x}_j$  bestimmt. Dies ergibt

$$\nabla_{\mathbf{x}_j} E_m(\boldsymbol{x}) = \frac{\lambda_a}{n-1} \cdot \nabla_{\mathbf{x}_j} \sum_{k=1}^{n-1} \|e_{a,k}(\boldsymbol{x})\|^2 + \frac{\lambda_c}{n-1} \cdot \nabla_{\mathbf{x}_j} \sum_{k=1}^{n-1} \|e_{c,k}(\boldsymbol{x})\|^2 \quad (\text{A.16})$$

mit

$$\nabla_{\mathbf{x}_j} \sum_{k=1}^{n-1} \|e_{a,k}(\boldsymbol{x})\|^2 = 2e_{a,j-1}(\boldsymbol{x}) \frac{1}{t_{j-1}} + 2e_{a,j}(\boldsymbol{x}) \left( \frac{-1}{t_j} - \frac{1}{t_{j-1}} \right) + 2e_{a,j+1}(\boldsymbol{x}) \frac{1}{t_j}. \quad (\text{A.17})$$

Die Ableitungen von  $\nabla_{\mathbf{x}_j} \sum_{k=1}^{n-1} \|e_{c,k}(\boldsymbol{x})\|^2$  werden einzeln gebildet. Dabei wird als abkürzende Schreibweise  $\Delta \mathbf{x}_j := \mathbf{x}_j - \mathbf{x}_{j-1}$  benutzt.

$$\begin{aligned} \frac{\partial \|e_{c,j-1}(\boldsymbol{x})\|^2}{\partial \mathbf{x}_{j,x}} &= 2 \left( \frac{1}{\|\Delta \mathbf{x}_j\|} - \frac{(\Delta \mathbf{x}_{j,x})^2}{\|\Delta \mathbf{x}_j\|^2} \right) \left( \frac{\Delta \mathbf{x}_{j,x}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,x}}{\|\Delta \mathbf{x}_{j-1}\|} \right) \\ &\quad - \frac{2(\Delta \mathbf{x}_{j,x})(\Delta \mathbf{x}_{j,y}) \left( \frac{\Delta \mathbf{x}_{j,y}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,y}}{\|\Delta \mathbf{x}_{j-1}\|} \right)}{\|\Delta \mathbf{x}_j\|^2} \\ &\quad - \frac{2(\Delta \mathbf{x}_{j,x})(\Delta \mathbf{x}_{j,z}) \left( \frac{\Delta \mathbf{x}_{j,z}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,z}}{\|\Delta \mathbf{x}_{j-1}\|} \right)}{\|\Delta \mathbf{x}_j\|^2}, \end{aligned} \quad (\text{A.18})$$

$$\begin{aligned} \frac{\partial \|e_{c,j-1}(\boldsymbol{x})\|^2}{\partial \mathbf{x}_{j,y}} &= 2 \left( \frac{1}{\|\Delta \mathbf{x}_j\|} - \frac{(\Delta \mathbf{x}_{j,y})^2}{\|\Delta \mathbf{x}_j\|^2} \right) \left( \frac{\Delta \mathbf{x}_{j,y}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,y}}{\|\Delta \mathbf{x}_{j-1}\|} \right) \\ &\quad - \frac{2(\Delta \mathbf{x}_{j,y})(\Delta \mathbf{x}_{j,x}) \left( \frac{\Delta \mathbf{x}_{j,x}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,x}}{\|\Delta \mathbf{x}_{j-1}\|} \right)}{\|\Delta \mathbf{x}_j\|^2} \\ &\quad - \frac{2(\Delta \mathbf{x}_{j,y})(\Delta \mathbf{x}_{j,z}) \left( \frac{\Delta \mathbf{x}_{j,z}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,z}}{\|\Delta \mathbf{x}_{j-1}\|} \right)}{\|\Delta \mathbf{x}_j\|^2}, \end{aligned} \quad (\text{A.19})$$

$$\begin{aligned} \frac{\partial \|e_{c,j-1}(\boldsymbol{x})\|^2}{\partial \mathbf{x}_{j,z}} &= 2 \left( \frac{1}{\|\Delta \mathbf{x}_j\|} - \frac{(\Delta \mathbf{x}_{j,z})^2}{\|\Delta \mathbf{x}_j\|^2} \right) \left( \frac{\Delta \mathbf{x}_{j,z}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,z}}{\|\Delta \mathbf{x}_{j-1}\|} \right) \\ &\quad - \frac{2(\Delta \mathbf{x}_{j,z})(\Delta \mathbf{x}_{j,x}) \left( \frac{\Delta \mathbf{x}_{j,x}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,x}}{\|\Delta \mathbf{x}_{j-1}\|} \right)}{\|\Delta \mathbf{x}_j\|^2} \\ &\quad - \frac{2(\Delta \mathbf{x}_{j,z})(\Delta \mathbf{x}_{j,y}) \left( \frac{\Delta \mathbf{x}_{j,y}}{\|\Delta \mathbf{x}_j\|} - \frac{\Delta \mathbf{x}_{j-1,y}}{\|\Delta \mathbf{x}_{j-1}\|} \right)}{\|\Delta \mathbf{x}_j\|^2}, \end{aligned} \quad (\text{A.20})$$



$$\begin{aligned}
\frac{\partial \|e_{c,j}(\boldsymbol{x})\|^2}{\partial \boldsymbol{x}_{j,x}} &= 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,x})^2}{\|\Delta \boldsymbol{x}_{j+1}\|^2} - \frac{1}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{1}{\|\Delta \boldsymbol{x}_j\|} + \frac{(\Delta \boldsymbol{x}_{j,x})^2}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,x}}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{\Delta \boldsymbol{x}_{j,x}}{\|\Delta \boldsymbol{x}_j\|} \right) \\
&+ 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,x})(\Delta \boldsymbol{x}_{j+1,y})}{\|\Delta \boldsymbol{x}_{j+1}\|^2} + \frac{(\Delta \boldsymbol{x}_{j,x})(\Delta \boldsymbol{x}_{j,y})}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,y}}{\|\Delta \boldsymbol{x}_{j+1}\|} + \frac{\Delta \boldsymbol{x}_{j,y}}{\|\Delta \boldsymbol{x}_j\|} \right) \\
&+ 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,x})(\Delta \boldsymbol{x}_{j+1,z})}{\|\Delta \boldsymbol{x}_{j+1}\|^2} + \frac{(\Delta \boldsymbol{x}_{j,x})(\Delta \boldsymbol{x}_{j,z})}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,z}}{\|\Delta \boldsymbol{x}_{j+1}\|} + \frac{\Delta \boldsymbol{x}_{j,z}}{\|\Delta \boldsymbol{x}_j\|} \right), \tag{A.21}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \|e_{c,j}(\boldsymbol{x})\|^2}{\partial \boldsymbol{x}_{j,y}} &= 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,y})^2}{\|\Delta \boldsymbol{x}_{j+1}\|^2} - \frac{1}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{1}{\|\Delta \boldsymbol{x}_j\|} + \frac{(\Delta \boldsymbol{x}_{j,y})^2}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,y}}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{\Delta \boldsymbol{x}_{j,y}}{\|\Delta \boldsymbol{x}_j\|} \right) \\
&+ 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,y})(\Delta \boldsymbol{x}_{j+1,x})}{\|\Delta \boldsymbol{x}_{j+1}\|^2} + \frac{(\Delta \boldsymbol{x}_{j,y})(\Delta \boldsymbol{x}_{j,x})}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,x}}{\|\Delta \boldsymbol{x}_{j+1}\|} + \frac{\Delta \boldsymbol{x}_{j,x}}{\|\Delta \boldsymbol{x}_j\|} \right) \\
&+ 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,y})(\Delta \boldsymbol{x}_{j+1,z})}{\|\Delta \boldsymbol{x}_{j+1}\|^2} + \frac{(\Delta \boldsymbol{x}_{j,y})(\Delta \boldsymbol{x}_{j,z})}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,z}}{\|\Delta \boldsymbol{x}_{j+1}\|} + \frac{\Delta \boldsymbol{x}_{j,z}}{\|\Delta \boldsymbol{x}_j\|} \right), \tag{A.22}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \|e_{c,j}(\boldsymbol{x})\|^2}{\partial \boldsymbol{x}_{j,z}} &= 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,z})^2}{\|\Delta \boldsymbol{x}_{j+1}\|^2} - \frac{1}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{1}{\|\Delta \boldsymbol{x}_j\|} + \frac{(\Delta \boldsymbol{x}_{j,z})^2}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,z}}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{\Delta \boldsymbol{x}_{j,z}}{\|\Delta \boldsymbol{x}_j\|} \right) \\
&+ 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,z})(\Delta \boldsymbol{x}_{j+1,x})}{\|\Delta \boldsymbol{x}_{j+1}\|^2} + \frac{(\Delta \boldsymbol{x}_{j,z})(\Delta \boldsymbol{x}_{j,x})}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,x}}{\|\Delta \boldsymbol{x}_{j+1}\|} + \frac{\Delta \boldsymbol{x}_{j,x}}{\|\Delta \boldsymbol{x}_j\|} \right) \\
&+ 2 \left( \frac{(\Delta \boldsymbol{x}_{j+1,z})(\Delta \boldsymbol{x}_{j+1,y})}{\|\Delta \boldsymbol{x}_{j+1}\|^2} + \frac{(\Delta \boldsymbol{x}_{j,z})(\Delta \boldsymbol{x}_{j,y})}{\|\Delta \boldsymbol{x}_j\|^2} \right) \\
&\cdot \left( \frac{\Delta \boldsymbol{x}_{j+1,y}}{\|\Delta \boldsymbol{x}_{j+1}\|} + \frac{\Delta \boldsymbol{x}_{j,y}}{\|\Delta \boldsymbol{x}_j\|} \right), \tag{A.23}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \|e_{c,j+1}(\boldsymbol{x})\|^2}{\partial \boldsymbol{x}_{j,x}} &= 2 \left( \frac{1}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{(\Delta \boldsymbol{x}_{j+1,x})^2}{\|\Delta \boldsymbol{x}_{j+1}\|^2} \right) \left( \frac{\Delta \boldsymbol{x}_{j+2,x}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,x}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right) \\
&- \frac{2(\Delta \boldsymbol{x}_{j+1,x})(\Delta \boldsymbol{x}_{j+1,y}) \left( \frac{\Delta \boldsymbol{x}_{j+2,y}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,y}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right)}{\|\Delta \boldsymbol{x}_{j+1}\|^2} \\
&- \frac{2(\Delta \boldsymbol{x}_{j+1,x})(\Delta \boldsymbol{x}_{j+1,z}) \left( \frac{\Delta \boldsymbol{x}_{j+2,z}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,z}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right)}{\|\Delta \boldsymbol{x}_{j+1}\|^2}, \tag{A.24}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \|e_{c,j+1}(\boldsymbol{x})\|^2}{\partial \boldsymbol{x}_{j,y}} &= 2 \left( \frac{1}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{(\Delta \boldsymbol{x}_{j+1,y})^2}{\|\Delta \boldsymbol{x}_{j+1}\|^2} \right) \left( \frac{\Delta \boldsymbol{x}_{j+2,y}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,y}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right) \\
&- \frac{2(\Delta \boldsymbol{x}_{j+1,y})(\Delta \boldsymbol{x}_{j+1,x}) \left( \frac{\Delta \boldsymbol{x}_{j+2,x}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,x}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right)}{\|\Delta \boldsymbol{x}_{j+1}\|^2} \\
&- \frac{2(\Delta \boldsymbol{x}_{j+1,y})(\Delta \boldsymbol{x}_{j+1,z}) \left( \frac{\Delta \boldsymbol{x}_{j+2,z}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,z}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right)}{\|\Delta \boldsymbol{x}_{j+1}\|^2}, \tag{A.25}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \|e_{c,j+1}(\boldsymbol{x})\|^2}{\partial \boldsymbol{x}_{j,z}} &= 2 \left( \frac{1}{\|\Delta \boldsymbol{x}_{j+1}\|} - \frac{(\Delta \boldsymbol{x}_{j+1,z})^2}{\|\Delta \boldsymbol{x}_{j+1}\|^2} \right) \left( \frac{\Delta \boldsymbol{x}_{j+2,z}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,z}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right) \\
&- \frac{2(\Delta \boldsymbol{x}_{j+1,z})(\Delta \boldsymbol{x}_{j+1,y}) \left( \frac{\Delta \boldsymbol{x}_{j+2,y}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,y}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right)}{\|\Delta \boldsymbol{x}_{j+1}\|^2} \\
&- \frac{2(\Delta \boldsymbol{x}_{j+1,z})(\Delta \boldsymbol{x}_{j+1,x}) \left( \frac{\Delta \boldsymbol{x}_{j+2,x}}{\|\Delta \boldsymbol{x}_{j+2}\|} - \frac{\Delta \boldsymbol{x}_{j+1,x}}{\|\Delta \boldsymbol{x}_{j+1}\|} \right)}{\|\Delta \boldsymbol{x}_{j+1}\|^2}. \tag{A.26}
\end{aligned}$$

Im Folgenden wird zusätzlich  $\Delta \mathbf{q}_j := \mathbf{q}_j - \mathbf{q}_{j-1}$ ,  $\Delta \mathbf{r}_j := \mathbf{r}_j - \mathbf{r}_{j-1}$  und  $\Delta \mathbf{s}_j := \mathbf{s}_j - \mathbf{s}_{j-1}$  genutzt. Die Ableitung von  $E_m(\boldsymbol{\mathcal{X}})$  nach  $t_j$  ist damit

$$\begin{aligned} \nabla_{t_j} E_m(\boldsymbol{\mathcal{X}}) &= \frac{\lambda_a}{n-1} \cdot \nabla_{t_j} \sum_{k=1}^{n-1} \|e_{a,k}(\boldsymbol{\mathcal{X}})\|^2 \\ &+ \frac{\lambda_{aa}}{n-1} \cdot \nabla_{t_j} \sum_{k=1}^{n-1} \left( \|e_{aa,q,k}(\boldsymbol{\mathcal{X}})\|^2 + \|e_{aa,r,k}(\boldsymbol{\mathcal{X}})\|^2 + \|e_{aa,s,k}(\boldsymbol{\mathcal{X}})\|^2 \right) \end{aligned} \quad (\text{A.27})$$

mit

$$\nabla_{t_j} \sum_{k=1}^{n-1} \|e_{a,k}(\boldsymbol{\mathcal{X}})\|^2 = 2e_{a,j}(\boldsymbol{\mathcal{X}})^\top \frac{-\Delta \mathbf{x}_{j+1}}{t_j^2} + 2e_{a,j+1}(\boldsymbol{\mathcal{X}})^\top \frac{\Delta \mathbf{x}_{j+1}}{t_j^2} \quad (\text{A.28})$$

und

$$\nabla_{t_j} \sum_{k=1}^{n-1} \|e_{aa,q,k}(\boldsymbol{\mathcal{X}})\|^2 = 2e_{aa,q,j}(\boldsymbol{\mathcal{X}})^\top \frac{-\Delta \mathbf{q}_{j+1}}{t_j^2} + e_{aa,q,j+1}(\boldsymbol{\mathcal{X}})^\top \frac{\Delta \mathbf{q}_{j+1}}{t_j^2} \quad (\text{A.29})$$

$$\nabla_{t_j} \sum_{k=1}^{n-1} \|e_{aa,r,k}(\boldsymbol{\mathcal{X}})\|^2 = 2e_{aa,r,j}(\boldsymbol{\mathcal{X}})^\top \frac{-\Delta \mathbf{r}_{j+1}}{t_j^2} + e_{aa,r,j+1}(\boldsymbol{\mathcal{X}})^\top \frac{\Delta \mathbf{r}_{j+1}}{t_j^2} \quad (\text{A.30})$$

$$\nabla_{t_j} \sum_{k=1}^{n-1} \|e_{aa,s,k}(\boldsymbol{\mathcal{X}})\|^2 = 2e_{aa,s,j}(\boldsymbol{\mathcal{X}})^\top \frac{-\Delta \mathbf{s}_{j+1}}{t_j^2} + e_{aa,s,j+1}(\boldsymbol{\mathcal{X}})^\top \frac{\Delta \mathbf{s}_{j+1}}{t_j^2}. \quad (\text{A.31})$$

Als Letztes wird die Ableitung von  $E_m(\boldsymbol{\mathcal{X}})$  nach  $\phi$  bestimmt zu:

$$\nabla_{\phi_j} E_m(\boldsymbol{\mathcal{X}}) = \frac{\lambda_{aa}}{n-1} \cdot \nabla_{\phi_j} \sum_{k=1}^{n-1} \left( \|e_{aa,q,k}(\boldsymbol{\mathcal{X}})\|^2 + \|e_{aa,r,k}(\boldsymbol{\mathcal{X}})\|^2 + \|e_{aa,s,k}(\boldsymbol{\mathcal{X}})\|^2 \right) \quad (\text{A.32})$$

mit

$$\begin{aligned} \nabla_{\phi_j} \sum_{k=1}^{n-1} \|e_{aa,q,k}(\boldsymbol{\mathcal{X}})\|^2 &= 2e_{aa,q,j-1}(\boldsymbol{\mathcal{X}}) \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_q}{t_{j-1}} \\ &+ 2e_{aa,q,j}(\boldsymbol{\mathcal{X}}) \left( \frac{-\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_q}{t_j} - \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_q}{t_{j-1}} \right) + 2e_{aa,q,j+1}(\boldsymbol{\mathcal{X}}) \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_q}{t_j}, \end{aligned} \quad (\text{A.33})$$

$$\begin{aligned} \nabla_{\phi_j} \sum_{k=1}^{n-1} \|e_{aa,r,k}(\boldsymbol{\mathcal{X}})\|^2 &= 2e_{aa,r,j-1}(\boldsymbol{\mathcal{X}}) \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_r}{t_{j-1}} \\ &+ 2e_{aa,r,j}(\boldsymbol{\mathcal{X}}) \left( \frac{-\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_r}{t_j} - \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_r}{t_{j-1}} \right) + 2e_{aa,r,j+1}(\boldsymbol{\mathcal{X}}) \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_r}{t_j}, \end{aligned} \quad (\text{A.34})$$

$$\begin{aligned} \nabla_{\phi_j} \sum_{k=1}^{n-1} \|e_{aa,s,k}(\boldsymbol{\mathcal{X}})\|^2 &= 2e_{aa,s,j-1}(\boldsymbol{\mathcal{X}}) \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_s}{t_{j-1}} \\ &+ 2e_{aa,s,j}(\boldsymbol{\mathcal{X}}) \left( \frac{-\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_s}{t_j} - \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_s}{t_{j-1}} \right) + 2e_{aa,s,j+1}(\boldsymbol{\mathcal{X}}) \frac{\nabla_{\phi_j} \mathbf{R}(\boldsymbol{\rho}_j)_s}{t_j}. \end{aligned} \quad (\text{A.35})$$

## A.2 Roboterbasierte Optimierung

Im Folgenden werden die Ableitungen für die roboterbasierte Optimierung angegeben.

### A.2.1 Ableitungen der Zielfunktion

Für die Ableitung der Zielfunktionskomponente  $E_l(\mathcal{X})$  muss die Funktion  $\tilde{z}_i$  nach  $\Phi_j$  und  $t_j$  abgeleitet werden. Die Ableitung nach  $\Phi_j$  ist

$$\begin{aligned} \nabla_{\Phi_j} \tilde{z}_i = & \left( \nabla_{\Phi_j} a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_M(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \right. \\ & + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,q}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\Phi_j} q(\mathbf{v}_i, \boldsymbol{\rho}_j) \\ & + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,r}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\Phi_j} r(\mathbf{v}_i, \boldsymbol{\rho}_j) \\ & \left. + a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) \cdot z_{M,d}(q(\mathbf{v}_i, \boldsymbol{\rho}_j), r(\mathbf{v}_i, \boldsymbol{\rho}_j), d(\mathbf{v}_i, \boldsymbol{\rho}_j)) \cdot \nabla_{\Phi_j} d(\mathbf{v}_i, \boldsymbol{\rho}_j) \right) \cdot \frac{t_{j-1} + t_j}{2}, \end{aligned} \quad (\text{A.36})$$

während die nach  $t_j$  bereits in Gleichung (A.3) bestimmt wurde.

Als Nächstes müssen die weiteren Ableitungen nach  $\Phi_j$  bestimmt werden. Dies ergibt für  $a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j)$

$$\nabla_{\Phi_j} a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j) = \nabla_{\Phi_j} \mathbf{x}_j(\boldsymbol{\rho}_j)^\top \frac{\|\mathbf{x}_j(\boldsymbol{\rho}_j) - \mathbf{v}_i\| \cdot \mathbf{n}_i - a_I(\mathbf{v}_i, \mathbf{n}_i, \boldsymbol{\rho}_j)(\mathbf{x}_j(\boldsymbol{\rho}_j) - \mathbf{v}_i)}{d(\mathbf{v}_i, \boldsymbol{\rho}_j)^2}. \quad (\text{A.37})$$

Nun folgt die Ableitung von  $q(\mathbf{v}_i, \boldsymbol{\rho}_j)$  mit

$$\begin{aligned} \nabla_{\Phi_j} q(\mathbf{v}_i, \boldsymbol{\rho}_j) = & - \left( \left( \nabla_{\Phi_j} \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) - \mathbf{R}(\boldsymbol{\rho}_j)^\top (\nabla_{\Phi_j} \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_q \cdot \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s \right. \\ & - \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_q \cdot \left( \nabla_{\Phi_j} \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) - \mathbf{R}(\boldsymbol{\rho}_j)^\top (\nabla_{\Phi_j} \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s \left. \right) \\ & / \left( \mathbf{R}(\boldsymbol{\rho}_j)^\top (\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)) \right)_s^2. \end{aligned} \quad (\text{A.38})$$

Für  $r(\mathbf{v}_i, \boldsymbol{\rho}_j)$  ist die Ableitung äquivalent. Um die Ableitungen von  $\mathbf{x}_j(\boldsymbol{\rho}_j)$  und  $\mathbf{R}(\boldsymbol{\rho}_j)$  nach jeder Komponente von  $\Phi$  zu bestimmen, müssen die Ableitungen der Matrizen  $\mathbf{T}_i(\Phi_i)$  bestimmt werden. Da in diesen Matrizen wieder ausschließlich Sinus- und Kosinusfunktionen vorkommen, werden die Ableitungen hier nicht explizit aufgeführt.

Die Ableitung von  $d(\mathbf{v}_i, \boldsymbol{\rho}_j)$  nach  $\nabla_{\Phi_j}$  bestimmt sich zu

$$\nabla_{\Phi_j} d(\mathbf{v}_i, \boldsymbol{\rho}_j) = \nabla_{\Phi_j} \mathbf{x}_j(\boldsymbol{\rho}_j)^\top \frac{\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)}{\|\mathbf{v}_i - \mathbf{x}_j(\boldsymbol{\rho}_j)\|}, \quad (\text{A.39})$$

wobei wie oben die Ableitungen von  $\mathbf{x}_j(\boldsymbol{\rho}_j)$  durch einfache Matrizen gegeben sind.

Die verbleibenden Ableitungen für  $z_{M,q}$ ,  $z_{M,r}$  und  $z_{M,d}$  sind gleich denen bei der bahnbasierten Variante aus Abschnitt A.1.1

### A.2.2 Ableitungen der Glättungsterme

Nun werden noch die Ableitungen der Funktion  $E_m(\boldsymbol{\mathcal{X}})$  aus Gleichung (5.11) für das roboterbasierte Verfahren angegeben. Die Ableitungen von  $E_m(\boldsymbol{\mathcal{X}})$  bezüglich  $\Phi_j$  je Komponente ist

$$\begin{aligned} \nabla_{\Phi_{i,j}} E_m(\boldsymbol{\mathcal{X}}) &= \frac{2e_{aa,i,j-1}(\boldsymbol{\mathcal{X}})}{6(n-1)} \frac{1}{t_{j-1}t_{j-2}} \\ &+ \frac{2e_{aa,i,j}(\boldsymbol{\mathcal{X}})}{6(n-1)} \left( \frac{-1}{t_j t_{j-1}} - \frac{1}{t_{j-1}^2} \right) \\ &+ \frac{2e_{aa,i,j+1}(\boldsymbol{\mathcal{X}})}{6(n-1)} \frac{1}{t_j^2}. \end{aligned} \quad (\text{A.40})$$

Die Ableitung von  $E_m(\boldsymbol{\mathcal{X}})$  bezüglich  $t_j$  ist

$$\begin{aligned} \nabla_{t_j} E_m(\boldsymbol{\mathcal{X}}) &= \frac{2}{6(n-1)} \sum_{i=1}^6 \left( e_{aa,i,j+1}(\boldsymbol{\mathcal{X}}) \left( \frac{-(\Phi_{i,j+2} - \Phi_{i,j+1})}{t_{j+1}t_j^2} - \frac{-2(\Phi_{i,j+1} - \Phi_{i,j})}{t_j^3} \right) \right. \\ &\quad \left. + e_{aa,i,j}(\boldsymbol{\mathcal{X}}) \frac{-(\Phi_{i,j+1} - \Phi_{i,j})}{t_j^2 t_{j-1}} \right). \end{aligned} \quad (\text{A.41})$$

## A.3 Werkstückbasierte Optimierung

Im Folgenden werden die Ableitungen für die werkstückbasierte Optimierung angegeben.

### A.3.1 Ableitungen der Zielfunktion

Die Gradienten von  $E_l(\boldsymbol{\mathcal{X}})$  bezüglich der Knotenverschiebung  $v_i$  können bestimmt werden, um die Verschiebung eines Knotens entlang seiner Normalenrichtung zu bestimmen. Dafür wird die alternative Schichtdicke  $\bar{z}_i$  genutzt, welche in Abschnitt 5.4 eingeführt wurde. Diese ist:

$$\bar{z}_i = z_i + v_i. \quad (\text{A.42})$$

Damit kann nun die Ableitung gebildet werden zu

$$\nabla_{v_i} \tilde{E}_l(\boldsymbol{\mathcal{X}}) = \frac{1}{A} \sum_{i=0}^m \frac{2A_i}{Z_i^2} \cdot (z_i + v_i - Z_i). \quad (\text{A.43})$$

Sowohl die Zielschichtdicke  $Z_i$  als auch die Normalenrichtung müssen nach jeder Iteration entsprechend angepasst werden, da die Verschiebung diese direkt beeinflusst.

### A.3.2 Ableitungen der Glättungsterme

Das Verschieben der Knoten verändert das Netz des Werkstücks. Eine Glättung des Netzes ist aber nicht notwendig. Allerdings sollten die Verschiebungen von nahegelegenen Knoten ähnlich sein, damit zum Beispiel ein Fräsprozess, mit dem das Werkstück hergestellt wird, nicht zu feine Strukturen erstellen muss. Daher ist das Zielkriterium das Glätten der Abweichung der Knotenverschiebung bezüglich der Nachbarn, was durch  $E_{vc}$  beschrieben wird. Die Ableitung nach  $v_i$  ergibt

$$\nabla_{v_i} E_{vc} = \frac{1}{m+1} \left( \sum_{j \in N_1(i)} 2(v_i - v_j) - \sum_{k \in N_1(i)} 2(v_k - v_i) \right). \quad (\text{A.44})$$

Die zweite Bedingung ist das möglichst gute Erhalten des Originalnetzes, damit nicht die Beschichtung vollständig durch die Änderung des Netzes ersetzt wird. Dies wird durch  $E_v$  beschrieben. Die Ableitung dieser Funktion ergibt

$$\nabla_{v_i} E_v = \frac{2v_i}{m+1}. \quad (\text{A.45})$$

# Mathematische Notation

Notation	Einheit	Bedeutung
$a_I(\mathbf{v}_i, \mathbf{n}_i, \mathbf{x}_j)$		Einschlags-Richtungs-Komponente des Auftragsmodells
$A_i$	$\text{m}^2$	Fläche des $i$ -ten Oberflächenpunkts
$b_{ijk}$		Ausgleich für Schichthöhenunterschied
$c_k$		Skalierungskonstante des $k$ -ten Gaußmodells
$d(\mathbf{v}_i, \mathbf{x}_j)$	$\text{m}$	Entfernung des $i$ -ten Oberflächenpunkts vom $j$ -ten Bahnpunkt
$\mathbf{d}_k$		Suchrichtung im konjugierten Gradienten Verfahren
$e_i$		$i$ -te Kante des Dreiecksnetzes des Werkstücks
$E_x(\mathcal{X})$		$x$ -te Zielfunktionskomponente
$f_i$		$i$ -te Facette des Dreiecksnetzes des Werkstücks
$f_k(\cdot)$		Ableitungsfunktion im konjugierten Gradienten Verfahren
$\mathbf{g}_k$		Gradientenvektor im konjugierten Gradienten Verfahren
$i$		Index eines Oberflächenpunkts
$j$		Index eines Bahnpunkts
$K$		Anzahl Gaußmodelle im Beschichtungsmodell
$m$		Anzahl der Oberflächenknoten
$m_e$		Anzahl der Kanten des Dreiecksnetzes
$m_f$		Anzahl der Facetten des Dreiecksnetzes
$M_\alpha$	$\text{W}/(\text{m}^2 \text{K})$	Wärmeübergangskoeffizient
$M_c$	$\text{J}/(\text{kg K})$	Wärmekapazität
$M_\epsilon$		Strahlungskoeffizient: Unterschied zum idealen schwarzen Strahler
$M_k$	$\text{W}/(\text{m K})$	Wärmeleitfähigkeit
$M_\rho$	$\text{kg}/\text{m}^3$	Materialdichte
$n$		Anzahl der Bahnpunkte
$\mathbf{n}_i$		Normale am $i$ -ten Oberflächenpunkt
$N_1(i)$		Knoten des 1-Ringes des $i$ -ten Oberflächenpunkts

Notation	Einheit	Bedeutung
$\mathbf{p}$		Einheitsvektor des Spritzpistolen-Bezugssystems
$P$		Population des Evolutionären Algorithmus
$q(\mathbf{v}_i, \boldsymbol{\rho}_j)$		$q$ -Komponente des Schnittpunkts von $\overline{(\mathbf{v}_i - \mathbf{x}_j)}$ mit der ( $s = -1$ )-Ebene des Spritzpistolen-Bezugssystems
$q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)$		$q_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) = q(\mathbf{v}_i, \boldsymbol{\rho}_j) - \mu_{q,k}$
$\mathbf{q}$		Einheitsvektor des Spritzpistolen-Bezugssystems
$r(\mathbf{v}_i, \boldsymbol{\rho}_j)$		$r$ -Komponente des Schnittpunkts von $\overline{(\mathbf{v}_i - \mathbf{x}_j)}$ mit der ( $s = -1$ )-Ebene des Spritzpistolen-Bezugssystems
$r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j)$		$r_{d,k}(\mathbf{v}_i, \boldsymbol{\rho}_j) = r(\mathbf{v}_i, \boldsymbol{\rho}_j) - \mu_{r,k}$
$\mathbf{r}$		Einheitsvektor des Spritzpistolen-Bezugssystems
$\mathbf{R}_w(\phi)$		Rotationsmatrix einer Rotation um Koordinatenachse $\mathbf{w}$ mit Winkel $\phi$
$\mathbf{R}(\boldsymbol{\rho})$		Transformation vom lokalen Spritzkoordinatensystem ins Weltkoordinatensystem
$\mathbf{R}(\Phi)$		Transformationsmatrix zur Orientierungsbestimmung des TCP über Denavit-Hartenberg Parameter
$\mathbf{s}_j$		Inverse Spritzrichtung am $j$ -ten Bahnpunkt
$t_j$	s	Zeitdifferenz zwischen dem $j$ -ten und $j + 1$ -ten Bahnpunkt
$\hat{t}_j$	s	Spritzdauer am $j$ -ten Bahnpunkt
$\mathbf{T}_i(\Phi)$		Denavit-Hartenberg Matrix mit Index $i$ und Rotationswinkel $\Phi$
$\mathbf{T}^r$		Initialtransformation des Roboters
$\mathbf{T}^t$		Transformation vom Roboterflansch zum TCP
$T_x^\alpha$	°	Denavit-Hartenberg Parameter
$T_x^a$	m	Denavit-Hartenberg Parameter
$T_x^d$	m	Denavit-Hartenberg Parameter
$T_x^\Phi$	°	Denavit-Hartenberg Parameter
$v_i$	m	Knotenverschiebung am Knoten mit Index $i$
$\mathbf{v}_i$		Position des $i$ -ten Oberflächenpunkts
$V_s$	m <sup>3</sup> /s	Ausgegebenes Materialvolumen der Spritzpistole pro Sekunde
$\mathbf{x}_j$		Position des $j$ -ten Bahnpunkts
$\mathbf{x}_j(\boldsymbol{\rho})$		Position des $j$ -ten Bahnpunkts bei gegebener Roboterachsstellung
$\bar{\mathbf{x}}_j$		Position des $j$ -ten Oberflächenbahnpunkts
$z(\mathbf{v}_i, \mathbf{n}_i, \mathbf{x}_j, \Phi_j, \hat{t}_j)$	m	Vom $j$ -ten Bahnpunkt generierter Schichtauftrag am $i$ -ten Oberflächenpunkt bei Spritzdauer $\hat{t}_j$
$z_M(\mathbf{v}_i, \mathbf{x}_j, \Phi_j)$	m	Materialverteilungsfunktion des Auftragsmodells
$z_i$	m	Schichtauftrag am $i$ -ten Oberflächenpunkt



Notation	Einheit	Bedeutung
$z_{i,\tau}$	m	Schichthöhe am $i$ -ten Oberflächenpunkt zum Zeitpunkt $\tau$
$\Delta z_{i,\tau}$	m	$\Delta z_{i,\tau} = z_{i,\tau+1} - z_{i,\tau}$
$Z_i$	m	Zielschichtdicke am $i$ -ten Oberflächenpunkt
$Z_{i,0}$	m	Zielschichtdicke am $i$ -ten Oberflächenpunkt vor Knotenverschiebung
$\alpha_{i,k}$	°	gegenüber der Kante $\overline{i,k}$ liegende Winkel
$\beta_{i,k}$	°	gegenüber der Kante $\overline{i,k}$ liegende Winkel
$\beta_k^X$		Suchrichtungparameter des konjugierten Gradienten Verfahrens
$\epsilon$		Vektor des Tschebyscheff Algorithmus
$\Theta_{i,\tau}$	K	Temperatur am $i$ -ten Oberflächenpunkt zum Zeitpunkt $\tau$
$\tilde{\Theta}$	K	Temperatur neuer Partikel
$\hat{\Theta}$	K	Temperatur der Umgebungsluft
$\overline{\Theta}_j$	K	Mittelwert der Temperatur auf der Oberfläche zum Zeitpunkt $j$
$\lambda, \lambda_x$		Gewichtungparameter
$\mu$		Schrittweitenparameter des konjugierten Gradienten Verfahrens
$\mu_q$		Mittelwert des Gaußmodells in $q$ -Richtung
$\mu_r$		Mittelwert des Gaußmodells in $r$ -Richtung
$\rho_j$		Spritzpistolenpose am $j$ -ten Bahnpunkt
$\sigma_B$	J/K	Stefan-Boltzmann Konstante
$\sigma_q$		Standardabweichung des Gaußmodells in $q$ -Richtung
$\sigma_r$		Standardabweichung des Gaußmodells in $r$ -Richtung
$\Sigma_k^{-1}$		Kovarianzmatrix des $k$ -ten Gaußmodells
$\tau_j$	s	Ankunftszeit am $j$ -ten Bahnpunkt
$\Delta\tau$	s	Zeitschritt
$\Phi_{j,x}$		Roboterachswinkel am $j$ -ten Bahnpunkt ( $x = 1, \dots, 6$ )
$\Phi_j$		Vektor der Roboterachswinkel $\Phi_{j,x}$ am $j$ -ten Bahnpunkt
$\phi_{j,x}$		Rotationswinkel am $j$ -ten Bahnpunkt ( $x = 1, \dots, 3$ )
$\phi_j$		Vektor der Rotationen $\phi_{j,x}$ am $j$ -ten Bahnpunkt
$\chi_j$		Bahnpunkt mit Index $j$
$\nabla$		Ableitungssymbol
$\mathcal{X}$		Spritzpistolenbahn



# Abbildungsverzeichnis

1.1	Industrieroboter beim thermischen Spritzen . . . . .	3
1.2	Gliederung thermischer Spritzmethoden . . . . .	3
1.3	Schematischer Aufbau einer Lichtbogen-Spritzpistole . . . . .	4
1.4	Systemarchitektur der Bahnplanung beim thermischen Spritzen . . . . .	7
2.1	Visualisierung verschiedener Bahntypen . . . . .	16
2.2	Fläche eines Knotens . . . . .	20
2.3	CAD-Modell und Denavit-Hartenberg Parameter des ABB IRB 4600 Roboters	22
2.4	Beispiel einer Denavit-Hartenberg Transformation . . . . .	23
3.1	Auftragsmodell der Simulation . . . . .	30
3.2	Thermisches Modell der Simulation . . . . .	36
3.3	Kalibrierung eines Gauß-Footprints . . . . .	41
3.4	Kalibrierung eines Libo-Footprints . . . . .	41
3.5	Kalibrierung eines Libo-Footprints mit erweitertem Modell . . . . .	42
3.6	Auftragsfehler der Modelle nach der Kalibrierung . . . . .	42
3.7	Simulierte Temperaturentwicklung auf einer Flachprobe . . . . .	45
4.1	Konzeptioneller Aufbau der Initialbahnplanung . . . . .	55
4.2	Vergleich von Bahnstrukturen . . . . .	58
4.3	Beispiel für ein Region of Interest . . . . .	59
4.4	Konstruktion von Rand-zu-Rand Bahnen . . . . .	60
4.5	Beispiel für mehrkriterielle Optimierung . . . . .	75
4.6	Desirability Functions . . . . .	88
4.7	Beispiele für das Latin Hypercube Sampling . . . . .	91
4.8	Beispiel zur Flächengewichtung beim LHS . . . . .	92
4.9	Grundprinzip der Parameteroptimierung mit SPOT . . . . .	104
4.10	Ergebnisse der Parameteroptimierung mit SPOT . . . . .	107
4.11	Analyse der Netz- und Bahnauflösung: Versuchsaufbau . . . . .	110
4.12	Analyse der Netz- und Bahnauflösung: Ergebnisse . . . . .	111
4.13	Werkstücke und Werkstücknetze in verschiedenen Auflösungen . . . . .	113
4.14	Region of Interest und Krümmung auf Werkstück A . . . . .	115
4.15	Zielfunktionsverlauf bei der Initialbahnplanung auf Werkstück A . . . . .	116

4.16	Thermische Varianzen verschiedener Bahnen . . . . .	117
4.17	Visualisierung der Beschichtungsfehler vor und nach der Initialbahnplanung	118
4.18	Fehler der Zwischenergebnisbahnen vor und nach der Bahnoptimierung . .	119
4.19	Visualisierung der Beschichtungsfehler vor und nach der Bahnoptimierung .	120
4.20	Streuung der Zielfunktionswerte bei der Initialbahnplanung . . . . .	121
4.21	Histogramm der optimierten Zielfunktionswerte . . . . .	122
4.22	Populationsvielfalt im Verlauf der Initialbahnplanung . . . . .	124
4.23	Einfluss der Zielschichtdicke . . . . .	125
4.24	Einfluss konstanter Bahngeschwindigkeit auf TCP- und Werkzeugbahn . . .	127
4.25	Einfluss der Bahngeschwindigkeit . . . . .	128
4.26	Zusammensetzung der Laufzeit der Initialbahnplanung . . . . .	129
4.27	Zusammensetzung der Simulationslaufzeit . . . . .	130
4.28	Zusammensetzung der Laufzeit zur Zielfunktionsbestimmung . . . . .	131
4.29	Zeitbedarf beim Generieren einer Bahn und beim Laden aus dem Cache . .	132
4.30	Thermische Varianzen von unterschiedlich generierten Bahnen . . . . .	133
4.31	Region of Interest auf Werkstück B . . . . .	138
4.32	Visualisierung quasi-konturparalleler Bahnen verschiedener Richtungen . . .	139
4.33	Thermische Varianzen verschiedener Bahnen . . . . .	140
4.34	Zielfunktionswerte von quasi-konturparallelen und Punkt-zu-Punkt Bahnen	141
4.35	Bahnen verschiedener Bahnstrukturen nach den Optimierungsschritten . . .	144
4.36	Ausschnitte von Bahnen verschiedener Bahnstrukturen nach der Optimierung	145
5.1	Konzeptioneller Aufbau der Bahnoptimierung . . . . .	150
5.2	Zielfunktionsverlauf bei der Bahnoptimierung mit verschiedenen Verfahren .	163
5.3	Zielfunktionsverlauf bei der Bahnoptimierung auf Werkstück B . . . . .	164
5.4	Bahnoptimierung mit unterschiedlichem $\lambda$ . . . . .	165
5.5	Thermische Varianzen verschiedener Bahnen . . . . .	167
5.6	Bahnoptimierung mit unterschiedlichen Modellen . . . . .	168
5.7	Bahnoptimierung mit falscher Zielschichtdicke . . . . .	169
5.8	Bahnoptimierung mit geglätteter und nicht geglätteter Bahn . . . . .	170
5.9	Winkel der Gradienten bei der Bahnoptimierung . . . . .	171
5.10	Bahnoptimierung mit Knotenverschiebung . . . . .	173
5.11	Zusammensetzung der Laufzeit der Bahnoptimierung . . . . .	174
5.12	Fallbeispiele für die Bahnoptimierung . . . . .	176
6.1	Konzeptioneller Aufbau der Dynamikkorrektur . . . . .	181
6.2	Visualisierung einer Überschleifzone . . . . .	183
6.3	Vergleich verschiedener Glättungsstufen für Bahnen mit B-Splines . . . . .	184
6.4	Bahn und Beschichtung auf Werkstück A nach Dynamikkorrektur . . . . .	186
6.5	Zielfunktionswerte im Verlauf der Dynamikoptimierung auf Werkstück A .	187
6.6	Bahn und Beschichtung auf Werkstück B nach Dynamikkorrektur . . . . .	189
6.7	Zielfunktionswerte im Verlauf der Dynamikoptimierung auf Werkstück B . .	190
6.8	Zielfunktionswerte verschiedener Bahnen nach Dynamikkorrektur . . . . .	191

# Tabellenverzeichnis

2.1	Technische Daten des ABB IRB 4600 Roboters . . . . .	22
2.2	Denavit-Hartenberg Parameter des ABB IRB 4600 Roboters . . . . .	23
3.1	Parameter der Auftragsmodelle nach der Kalibrierung . . . . .	44
3.2	Parameter des thermischen Modells nach der Kalibrierung . . . . .	45
4.1	Vergleich der Bahnstrukturen . . . . .	67
4.2	Mutationsoperatoren . . . . .	97
4.3	Werte der Parameter des EA nach deren Optimierung durch SPOT . . . . .	106
4.4	Indexzuordnung der Methoden in SPOT . . . . .	106
4.5	Auswirkung verschiedener Parametersätze auf den EA . . . . .	108



# Literaturverzeichnis

- [1] ABB Robotics. Operating manual robotstudio: Document id: 3hac032104-001. Technical report, ABB Robotics, Västerås, Sweden, 2007. [https://library.e.abb.com/public/244a8a5c10ef8875c1257b4b0052193c/3HAC032104-001\\_revD\\_en.pdf](https://library.e.abb.com/public/244a8a5c10ef8875c1257b4b0052193c/3HAC032104-001_revD_en.pdf). (Zitiert auf Seiten 180 und 183.)
- [2] ABB Robotics Products AB. Rapid reference manual: Rapid overview on-line. article number: 3hac 0966-13. Technical report, ABB Robotics, Västerås, Sweden, 2017. <http://rab.ict.pwr.wroc.pl/irb1400/overviewrev1.pdf>. (Zitiert auf Seite 183.)
- [3] E. Alba und M. Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6(5):443–462, 2002. (Zitiert auf Seite 102.)
- [4] D. L. Applegate, R. E. Bixby, V. Chvátal und W. J. Cook. *The Travelling Salesman Problem: A Computational Study*. Princeton University Press, 2011. (Zitiert auf Seite 52.)
- [5] N. A. Aspragathos und S. Foussias. Optimal location of a robot path when considering velocity performance. *Robotica*, 20(02):139–147, 2002. (Zitiert auf Seite 69.)
- [6] A. A. Ata. Optimal trajectory planning of manipulators: a review. *Journal of Engineering Science and Technology*, 2(1):32–54, 2007. (Zitiert auf Seite 69.)
- [7] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset und A. A. Rizzi. Hierarchical segmentation of surface embedded in  $\mathbb{R}^3$  for auto-body painting. *IEEE Transactions on Robotics and Automation*, Seiten 572–577, 2005. (Zitiert auf Seite 51.)
- [8] P. N. Atkar, A. Greenfield, D. C. Conner, H. Choset und A. A. Rizzi. Uniform coverage of automotive surface patches. *The International Journal of Robotics Research*, 24(11):883–898, 2005. (Zitiert auf Seite 51.)
- [9] T. Bäck. Self-adaptation in genetic algorithms. In *Proceedings of the First European Conference on Artificial Life*, Seiten 263–271. MIT Press, 1992. (Zitiert auf Seite 103.)

- 
- [10] T. Bäck. Optimal mutation rates in genetic search. In *Proceedings of the 5th International Conference on Genetic Algorithms*, Seiten 2–8, 1993. (Zitiert auf Seite 90.)
- [11] T. Bäck, U. Hammel und H.-P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997. (Zitiert auf Seite 74.)
- [12] T. Bäck und H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):1–23, 1993. (Zitiert auf Seite 74.)
- [13] T. Bartz-Beielstein, M. Friese, M. Zaefferer, B. Naujoks, O. Flasch, W. Konen und P. Koch. Noisy optimization with sequential parameter optimization and optimal computational budget allocation. In *Proceedings of the 2011 Annual Conference on Genetic and Evolutionary Computation*, Seiten 119–120, 2011. (Zitiert auf Seite 104.)
- [14] T. Bartz-Beielstein, C. Jung und M. Zaefferer. Uncertainty management using sequential parameter optimization. In *Uncertainty Management in Simulation-Optimization of Complex Systems*, Seiten 79–99. Springer, 2015. (Zitiert auf Seite 104.)
- [15] T. Bartz-Beielstein und M. Preuss. Considerations of budget allocation for sequential parameter optimization (spo). In *Proceedings of the Workshop on Empirical Methods for the Analysis of Algorithms*, Seiten 35–40, 2006. (Zitiert auf Seite 104.)
- [16] T. Bartz-Beielstein und M. Zaefferer. A gentle introduction to sequential parameter optimization. *CIplus*, 1(2), 2012. (Zitiert auf Seiten 104 und 105.)
- [17] T. Bartz-Beielstein, M. Zaefferer, J. Ziegenhirt, W. Konen, O. Flasch, P. Koch und M. Friese. Package 'spot', 2016. (Zitiert auf Seiten 104 und 106.)
- [18] W. Benenson und H. Stöcker. *Handbook of Physics*. Springer, New York, 2002. (Zitiert auf Seite 36.)
- [19] T. L. Bergman, F. P. Incropera, A. S. Lavine und D. P. Dewitt. *Fundamentals of Heat and Mass Transfer*. Wiley, Hoboken, NJ, 7. Edition, 2011. (Zitiert auf Seite 37.)
- [20] R. Berthelsen, T. Wiederkehr, R. Denzer, A. Menzel und H. Müller. Efficient simulation of nonlinear heat transfer during thermal spraying of complex workpieces. *World Journal of Mechanics*, 04(09):289–301, 2014. (Zitiert auf Seite 35.)
- [21] D. Biermann, A. Sacharow, T. Surmann und T. Wagner. Direct free-form deformation of nc programs for surface reconstruction and form-error compensation. *Production Engineering*, 4(5):501, 2010. (Zitiert auf Seite 149.)



- [22] R. Bolot, S. Deng, Z. H. Cai, H. Liao und G. Montavon. A coupled model between robot trajectories and thermal history of the workpiece during thermal spray operation. *Journal of Thermal Spray Technology*, 23(3):296–303, 2014. (Zitiert auf Seite 29.)
- [23] M. R. Bonyadi, Z. Michalewicz und L. Barone. The travelling thief problem: the first step in the transition from theoretical problems to realistic problems. In *2013 IEEE Congress on Evolutionary Computation*, Seiten 1037–1044, 2013. (Zitiert auf Seite 83.)
- [24] M. R. Bonyadi, Z. Michalewicz, M. R. Przybyłok und A. Wierzbicki. Socially inspired algorithms for the travelling thief problem. In D. V. Arnold, Editor, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Seiten 421–428, New York, NY, USA, 2014. ACM. (Zitiert auf Seite 84.)
- [25] G. Boschetti, R. Rosa und A. Trevisani. Optimal robot positioning using task-dependent and direction-selective performance indexes: General definitions and application to a parallel robot. *Robotics and Computer-Integrated Manufacturing*, 29(2):431–443, 2013. (Zitiert auf Seite 69.)
- [26] M. Botsch, M. Pauly, C. Rössl, S. Bischoff und L. Kobbelt. Geometric modeling based on triangle meshes. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM. (Zitiert auf Seite 19.)
- [27] Z. H. Cai, S. Deng, H. Liao, C. Zeng und G. Montavon. The effect of spray distance and scanning step on the coating thickness uniformity in cold spray process. *Journal of Thermal Spray Technology*, 23(3):354–362, 2014. (Zitiert auf Seiten 9, 17 und 52.)
- [28] A. Candel und R. Gadow. Optimized multiaxis robot kinematic for hvof spray coatings on complex shaped substrates. *Surface and Coatings Technology*, 201(5):2065–2071, 2006. (Zitiert auf Seite 17.)
- [29] A. Candel und R. Gadow. Trajectory generation and coupled numerical simulation for thermal spraying applications on complex geometries. *Journal of Thermal Spray Technology*, 18(5-6):981–987, 2009. (Zitiert auf Seiten 17, 54 und 149.)
- [30] E. Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs Paralleles, Reseaux et Systems Repartis*, 10(2):141–171, 1998. (Zitiert auf Seite 102.)
- [31] U. K. Chakraborty und C. Z. Janikow. An analysis of gray versus binary encoding in genetic search. *Information Sciences*, 156(3):253–269, 2003. (Zitiert auf Seite 90.)
- [32] S. Chand und M. Wagner. Fast heuristics for the multiple traveling thieves problem. In *Proceedings of the 2016 Annual Conference on Genetic and Evolutionary Computation*, Seiten 293–300, 2016. (Zitiert auf Seite 84.)

- [33] S. Chandra und P. Fauchais. Formation of solid splats during thermal spray deposition. *Journal of Thermal Spray Technology*, 18(2):148–180, 2009. (Zitiert auf Seite 29.)
- [34] P.-C. Chang, M.-H. Chen, M. K. Tiwari und A. S. Iquebal. A block-based evolutionary algorithm for flow-shop scheduling problem. *Applied Soft Computing*, 13(12):4536–4547, 2013. (Zitiert auf Seite 85.)
- [35] H. Chen, T. Fuhlbrigge und L. Xiongzi. A review of CAD-based robot path planning for spray painting. *Industrial robot: An International Journal*, 36(1):45–50, 2009. (Zitiert auf Seite 51.)
- [36] H. Chen und N. Xi. Automated robot tool trajectory connection for spray forming process. *Journal of Manufacturing Science and Engineering*, 134(2):21017, 2012. (Zitiert auf Seite 53.)
- [37] Y. Chen, X. Liang, Y. Liu, J. Bai und B. Xu. Finite element modeling of coating formation and transient heat transfer in the electric arc spray process. *International Journal of Heat and Mass Transfer*, 53(9-10):2012–2021, 2010. (Zitiert auf Seite 55.)
- [38] T.-C. Chiang und H.-J. Lin. A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling. *International Journal of Production Economics*, 141(1):87–98, 2013. (Zitiert auf Seite 85.)
- [39] D. C. Conner, A. Greenfield, P. N. Atkar, A. A. Rizzi und H. Choset. Paint deposition modeling for trajectory planning on automotive surfaces. *IEEE Transactions on Automation Science and Engineering*, 2(4):381–392, 2005. (Zitiert auf Seite 51.)
- [40] C. Contreras-Bolton und V. Parada. Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. *PloS one*, 10(9):e0137724, 2015. (Zitiert auf Seite 81.)
- [41] J. R. Current und D. A. Schilling. The covering salesman problem. *Transportation Science*, 23(3):208–213, 1989. (Zitiert auf Seiten 52 und 80.)
- [42] Y.-H. Dai. Nonlinear conjugate gradient methods. Technical report, State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China, 2010. (Zitiert auf Seite 164.)
- [43] M. Dakulovic, S. Horvatic und I. Petrovic. Complete coverage d\* algorithm for path planning of a floor-cleaning mobile robot. In *Preprints of the 18th IFAC World Congress*, Band 18, Seiten 5950–5955. IFAC, 2011. (Zitiert auf Seite 52.)
- [44] C. de Boor. B(asic)-spline basics. Technical report, Madison Mathematics Research Center, Wisconsin University, 1986. (Zitiert auf Seite 184.)

- [45] K. Deb und H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014. (Zitiert auf Seiten 79 und 86.)
- [46] K. Deb, A. Pratap, S. Agarwal und T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. (Zitiert auf Seiten 79 und 86.)
- [47] S. Deng, Z. H. Cai, D. Fang, H. Liao und G. Montavon. Application of robot offline programming in thermal spraying. *Surface and Coatings Technology*, 206(19-20):3875–3882, 2012. (Zitiert auf Seiten 9, 15, 21 und 53.)
- [48] S. Deng, H. Liang, Z. H. Cai, H. Liao und G. Montavon. Kinematic optimization of robot trajectories for thermal spray coating application. *Journal of Thermal Spray Technology*, 23(8):1382–1389, 2014. (Zitiert auf Seiten 9 und 53.)
- [49] R. Denysiuk, L. Costa und I. Espírito Santo. Moea/van: Multiobjective evolutionary algorithm based on vector angle neighborhood. In S. Silva, Editor, *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, Seiten 663–670, 2015. (Zitiert auf Seiten 99 und 101.)
- [50] G. Derringer und R. Suich. Simultaneous optimization of several response variables. *Journal of Quality Technology*, 12(4):214–219, 1980. (Zitiert auf Seite 88.)
- [51] DIN Deutsches Institut für Normierung e.V. Thermisches spritzen - begriffe, einteilung, 06.2005. DIN EN 657:2005, ICS 01.040.25; 25.220.20, Beuth Verlag, Berlin. (Zitiert auf Seiten 3 und 4.)
- [52] F. Dobsław. Recent development in automatic parameter tuning for metaheuristics. In *Proceedings of the 19th Annual Conference of Doctoral Students-WDS*, Band 2010, 2010. (Zitiert auf Seite 103.)
- [53] R. R. dos Santos, V. Steffen, Jr. und S. d. F. P. Saramago. Optimal task placement of a serial robot manipulator for manipulability and mechanical power optimization. *Intelligent Information Management*, 02(09):512–525, 2010. (Zitiert auf Seite 69.)
- [54] S. R. Duncan, P. D. A. Jones und P. Wellstead. A frequency-domain approach to determining the path separation for spray coating. *IEEE Transactions on Automation Science and Engineering*, 2(3):233–239, 2005. (Zitiert auf Seiten 2, 29, 30, 54 und 60.)
- [55] A. E. Eiben, R. Hinterding und Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999. (Zitiert auf Seite 103.)

- [56] K. Endl und W. Luh. *Analysis I.: Eine integrierte Darstellung: Studienbuch für Studierende der Mathematik, Physik und anderer Naturwissenschaften ab 1. Semester*. Studien-Texte: Mathematik. Aula-Verlag GmbH, 1989. (Zitiert auf Seite 64.)
- [57] K. Endl und W. Luh. *Analysis II.: Eine integrierte Darstellung: Studienbuch für Studierende der Mathematik, Physik und anderer Naturwissenschaften ab 1. Semester*. Studien-Texte: Mathematik. Aula-Verlag GmbH, 1989. (Zitiert auf Seite 64.)
- [58] B. Fardanesh und J. Rastegar. Minimum cycle time location of a task in the workspace of a robot arm. In *Proceedings of the 27th IEEE Conference on Decision and Control*, Seiten 2280–2283, 1988. (Zitiert auf Seite 69.)
- [59] M. M. Fasching, F. B. Prinz und L. E. Weiss. Planning robotic trajectories for thermal spray shape deposition. *Journal of Thermal Spray Technology*, 2(1):47–57, 1993. (Zitiert auf Seiten 17, 51 und 149.)
- [60] P. Fauchais, M. Fukumoto, A. Vardelle und M. Vardelle. Knowledge concerning splat formation: An invited review. *Journal of Thermal Spray Technology*, 13(3):337–360, 2004. (Zitiert auf Seite 29.)
- [61] H. Faulkner, S. Polyakovskiy, T. Schultz und M. Wagner. Approximate approaches to the traveling thief problem. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, Seiten 385–392, 2015. (Zitiert auf Seite 84.)
- [62] J. T. Feddema. Kinematically optimal robot placement for minimum time coordinated motion. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Band 4, Seiten 3395–3400, 1996. (Zitiert auf Seite 69.)
- [63] H. Y. Feng und H. Li. Constant scallop-height tool path generation for three-axis sculptured surface machining. *Computer-Aided Design*, 34(9):647–654, 2002. (Zitiert auf Seite 53.)
- [64] E. Foroozmehr und R. Kovacevic. Effect of path planning on the laser powder deposition process: thermal and structural evaluation. *International Journal of Advanced Manufacturing Technology*, 51(5-8):659–669, 2010. (Zitiert auf Seite 54.)
- [65] Y. Fu und W. Luo. Optimal task assignment for serial-parallel hybrid robots cooperation via ant colony optimization. In *9th International Conference on Electronic Measurement & Instruments*, Band 9, Seiten 795–800. IEEE, 2009. (Zitiert auf Seite 51.)
- [66] R. Gadow, A. Candel und M. Floristán. Optimized robot trajectory generation for thermal spraying operations and high quality coatings on free-form surfaces. *Surface and Coatings Technology*, 205(4):1074–1079, 2010. (Zitiert auf Seiten 17, 54 und 149.)

- 
- [67] E. Galceran und M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, 2013. (Zitiert auf Seite 52.)
- [68] M. Gen und L. Lin. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. *Journal of Intelligent Manufacturing*, 25(5):849–866, 2014. (Zitiert auf Seite 85.)
- [69] J. C. Gilbert und J. Nocedal. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, 2(1):21–42, 1992. (Zitiert auf Seite 157.)
- [70] A. S. Glassner, Editor. *An introduction to ray tracing*. Elsevier, 1989. (Zitiert auf Seite 70.)
- [71] M. Glickman und K. Sycara. Evolutionary algorithms: Exploring the dynamics of self-adaptation. In *Proceedings of the Third Annual Conference on Genetic Programming*, Seiten 762–769, 1998. (Zitiert auf Seite 103.)
- [72] C. Glynn, T. O’Donovan, D. B. Murray und M. Feidt. Jet impingement cooling. In *Proceedings of the 9th UK National Heat Transfer Conference*, Seiten 5–6, 2005. (Zitiert auf Seite 37.)
- [73] B. Golden, Z. Naji-Azimi, S. Raghavan, M. Salari und P. Toth. The generalized covering salesman problem. *INFORMS Journal on Computing*, 2011. (Zitiert auf Seiten 52 und 80.)
- [74] P. S. Grant, S. R. Duncan, A. Roche und C. F. Johnson. Scientific, technological, and economic aspects of rapid tooling by electric arc spray forming. *Journal of Thermal Spray Technology*, 15(4):796–801, 2006. (Zitiert auf Seite 54.)
- [75] D. J. Gulczynski, J. W. Heath und C. C. Price. The close enough traveling salesman problem: A discussion of several heuristics. *Perspectives in Operations Research*, 36:271–283, 2006. (Zitiert auf Seiten 52 und 80.)
- [76] W. W. Hager und H. Zhang. A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization*, 16(1):170–192, 2005. (Zitiert auf Seiten 157 und 158.)
- [77] W. W. Hager und H. Zhang. A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*, 2(1):35–58, 2006. (Zitiert auf Seiten 157 und 158.)
- [78] A. Hansbo und P. Nylén. Models for the simulation of spray deposition and robot motion optimization in thermal spraying of rotating objects. *Surface and Coatings Technology*, 122(2-3):191–201, 1999. (Zitiert auf Seite 53.)
- [79] J. Harrington. The desirability function. *Industrial Quality Control*, 21(10):494–498, 1965. (Zitiert auf Seite 88.)

- [80] J. He, Y. Wang und Y. Zhou. Analysis of solution quality of a multiobjective optimization-based evolutionary algorithm for knapsack problem. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, Seiten 74–85, 2015. (Zitiert auf Seite 82.)
- [81] H. C. Hege und K. Polthier. *Visualization and Mathematics III*. Springer, 2003. (Zitiert auf Seiten 20 und 36.)
- [82] D. Hegels. *Heuristische Wegfindung zur Reduktion thermischer Varianz beim roboterbasierten thermischen Spritzen*. Masterarbeit, TU Dortmund, Dortmund, 12.09.2012. (Zitiert auf Seiten 10 und 13.)
- [83] D. Hegels und H. Müller. Heuristische Wegfindung zur Reduktion thermischer Varianz beim roboterbasierten thermischen Spritzen. In W. Tillmann, Editor, *5. Öffentliches Kolloquium - SFB 708*, Seiten 111–120. Verlag Praxiswissen, 2012. (Zitiert auf Seite 13.)
- [84] D. Hegels und H. Müller. Evolutionary path generation for reduction of thermal variations in thermal spray coating. In C. Blum, Editor, *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation (GECCO2013)*, Seiten 1277–1284, New York, NY, USA, 2013. ACM. (Zitiert auf Seiten 10, 13, 52, 54 und 60.)
- [85] D. Hegels und H. Müller. A simulation based path optimization system for robot guided thermal spray. In W. Tillmann, Editor, *6. Öffentliches Kolloquium - SFB 708*, Seiten 127–136. Verlag Praxiswissen, 2013. (Zitiert auf Seiten 13 und 156.)
- [86] D. Hegels, T. Wiederkehr und H. Müller. Simulation based iterative post-optimization of paths of robot guided thermal spraying. *Robotics and Computer-Integrated Manufacturing*, 35:1–15, 2015. (Zitiert auf Seiten 13 und 17.)
- [87] S.-C. Horng, S.-S. Lin und F.-Y. Yang. Evolutionary algorithm for stochastic job shop scheduling with random processing time. *Expert Systems with Applications*, 39(3):3603–3610, 2012. (Zitiert auf Seite 85.)
- [88] F. Hutter, H. H. Hoos, K. Leyton-Brown und K. Murphy. Time-bounded sequential parameter optimization. In *International Conference on Learning and Intelligent Optimization*, Seiten 281–298, 2010. (Zitiert auf Seite 104.)
- [89] Y. K. Hwang und P. A. Watterberg. Optimizing robot placement for visit-point tasks. In *Proceedings of the AAAI Workshop on Artificial Intelligence for Manufacturing*, Seiten 81–86, 1996. (Zitiert auf Seite 69.)
- [90] H. Ishibuchi, N. Akedo und Y. Nojima. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 19(2):264–283, 2015. (Zitiert auf Seite 82.)



- [91] H. Jain und K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014. (Zitiert auf Seite 79.)
- [92] R. N. Jazar. Inverse kinematics. In R. N. Jazar, Editor, *Theory of applied robotics*, Seiten 263–296. Springer, New York, 2007. (Zitiert auf Seite 22.)
- [93] D. R. Jones, M. Schonlau und W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. (Zitiert auf Seite 104.)
- [94] P. D. A. Jones, S. R. Duncan, T. Rayment und P. S. Grant. Control of temperature profile for a spray deposition process. *IEEE Transactions on Control Systems Technology*, 11(5):656–667, 2003. (Zitiert auf Seite 2.)
- [95] P. D. A. Jones, S. R. Duncan, T. Rayment und P. S. Grant. Optimal robot path for minimizing thermal variations in a spray deposition process. *IEEE Transactions on Control Systems Technology*, 15(1):1–11, 2007. (Zitiert auf Seiten 2, 10, 35, 36, 54 und 60.)
- [96] B. Kamrani, V. Berbyuk, D. Wäppling, U. Stickelmann und X. Feng. Optimal robot placement using response surface method. *The International Journal of Advanced Manufacturing Technology*, 44(1-2):201–210, 2009. (Zitiert auf Seite 69.)
- [97] S. Kang, S.-S. Kim, J.-H. Won und Y.-M. Kang. Bidirectional constructive crossover for evolutionary approach to travelling salesman problem. In *5th International Conference on IT Convergence and Security (ICITCS)*, Seiten 1–4, 2015. (Zitiert auf Seiten 77 und 81.)
- [98] Q. Y. Kenny, W. Li und A. Sudjianto. Algorithmic construction of optimal symmetric latin hypercube designs. *Journal of Statistical Planning and Inference*, 90(1):145–159, 2000. (Zitiert auf Seite 91.)
- [99] S. Khuri, T. Bäck und J. Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In *Proceedings of the 1994 ACM symposium on Applied computing*, Seiten 188–193, 1994. (Zitiert auf Seite 82.)
- [100] A. Kout. *Geometric and algorithmic aspects of automatic path planning with relation to spray deposition processes*. Dissertation, TU Dortmund, Dortmund, 14.07.2015. (Zitiert auf Seiten 17, 51 und 52.)
- [101] A. Kout und H. Müller. A framework for the generation of distance field based curves on triangular mesh surfaces for nc manufacturing. Technical report, Chair VII - Computer Science, Technische Universität Dortmund, 2010. (Zitiert auf Seite 64.)

- [102] A. Kout und H. Müller. Quantitative improvement of tool impact paths defined by isolines of scalar functions on triangular mesh workpiece surfaces. *International Journal of Advanced Manufacturing Technology*, Seiten 1–19, 2012. (Zitiert auf Seiten 51 und 52.)
- [103] A. Kout und H. Müller. Tool-adaptive offset paths on triangular mesh workpiece surfaces. *Computer-Aided Design*, 50:61–73, 2014. (Zitiert auf Seiten 15, 17, 51 und 52.)
- [104] P. Larranaga, C. M. H. Kuijpers, R. H. Murga, I. Inza und S. Dizdarevic. Genetic algorithms for the travelling salesman problem: a review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999. (Zitiert auf Seiten 52, 76, 77, 81, 92 und 137.)
- [105] A. Lasemi, D. Xue und P. Gu. Recent development in cnc machining of freeform surfaces: A state-of-the-art review. *Computer-Aided Design*, 42(7):641–654, 2010. (Zitiert auf Seite 53.)
- [106] J. Lee. *Introduction to topological manifolds*, Band 940 von *Graduate Texts in Mathematics*. Springer Verlag, New York, 2. Edition, 2011. (Zitiert auf Seite 18.)
- [107] R. S. Lin und Y. Koren. Efficient tool-path planning for machining free-form surfaces. *ASME Transaction Journal of Engineering for Industry*, Seiten 20–28, 1996. (Zitiert auf Seite 53.)
- [108] S. Lin und B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973. (Zitiert auf Seite 96.)
- [109] A. M. Lopes und E. J. Solteiro Pires. Optimization of the workpiece location in a machining robotic cell. *International Journal of Advanced Robotic Systems*, 8(6):37–46, 2011. (Zitiert auf Seite 69.)
- [110] N. Lourenço, F. B. Pereira und E. Costa. An evolutionary approach to the full optimization of the traveling thief problem. In F. Chicano, B. Hu und P. García-Sánchez, Editoren, *Evolutionary Computation in Combinatorial Optimization*, LNCS sublibrary. SL 1, Theoretical computer science and general issues, Seiten 34–45. Springer, Cham, 2015. (Zitiert auf Seite 84.)
- [111] T.-C. Lu und G.-R. Yu. An adaptive population multi-objective quantum-inspired evolutionary algorithm for multi-objective 0/1 knapsack problems. *Information Sciences*, 243:39–56, 2013. (Zitiert auf Seite 82.)
- [112] S. Luke. *Essentials of Metaheuristics*. Lulu, 2. Edition, 2013. (Zitiert auf Seiten 75, 77 und 102.)
- [113] M. L. Mauldin. Maintaining diversity in genetic search. In *Proceedings of the National Conference on Artificial Intelligence*, Seiten 247–250, 1984. (Zitiert auf Seite 78.)



- 
- [114] M. D. McKay, R. J. Beckman und W. J. Conover. Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979. (Zitiert auf Seiten 77 und 90.)
- [115] J. Mehnen und H. Trautmann. Integration of expert’s preferences in pareto optimization by desirability function techniques. *Parameters*, 500:1, 2006. (Zitiert auf Seite 88.)
- [116] J. Mehnen, H. Trautmann und A. Tiwari. Introducing user preference using desirability functions in multi-objective evolutionary optimisation of noisy processes. In *2007 IEEE Congress on Evolutionary Computation*, Seiten 2687–2694, 2007. (Zitiert auf Seite 88.)
- [117] Y. Mei, X. Li, F. Salim und X. Yao. Heuristic evolution with genetic programming for traveling thief problem. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, Seiten 2753–2760, 2015. (Zitiert auf Seite 84.)
- [118] Y. Mei, X. Li und X. Yao. Improving efficiency of heuristics for the large scale traveling thief problem. In *Asia-Pacific Conference on Simulated Evolution and Learning*, Seiten 631–643, 2014. (Zitiert auf Seite 84.)
- [119] Y. Mei, X. Li und X. Yao. On investigation of interdependence between sub-problems of the travelling thief problem. *Soft Computing*, 20(1):157–172, 2016. (Zitiert auf Seite 84.)
- [120] M. Meredith und S. Maddock. Real-time inverse kinematics: the return of the jacobian: Technical report no. cs-04-06. Technical report, Department of Computer Science, University of Sheffield, 2004. (Zitiert auf Seite 24.)
- [121] D. Meschede und C. Gerthsen. *Gerthsen Physik*. Springer-Lehrbuch. Springer Spektrum, Berlin, 25. Edition, 2015. (Zitiert auf Seite 37.)
- [122] M. Meyer, M. Desbrun, P. Schröder und A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics*, 3(2):52–58, 2002. (Zitiert auf Seiten 20, 36 und 115.)
- [123] S. Meyer-Nieberg und H. G. Beyer. Self-adaptation in evolutionary algorithms. In *Parameter Setting in Evolutionary Algorithms*, Seiten 47–75. Springer, 2007. (Zitiert auf Seite 103.)
- [124] L. L. Minku, D. Sudholt und X. Yao. Improved evolutionary algorithm design for the project scheduling problem based on runtime analysis. *IEEE Transactions on Software Engineering*, 40(1):83–102, 2014. (Zitiert auf Seite 85.)
- [125] J. S. B. Mitchell, D. M. Mount und C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987. (Zitiert auf Seiten 60, 64, 65, 66 und 134.)

- [126] S. Mitsi, K.-D. Bouzakis, D. Sagris und G. Mansour. Determination of optimum robot base location considering discrete end-effector positions by means of hybrid genetic algorithm. *Robotics and Computer-Integrated Manufacturing*, 24(1):50–59, 2008. (Zitiert auf Seite 69.)
- [127] V. Nannen, S. K. Smit und A. E. Eiben. Costs and benefits of tuning parameters of evolutionary algorithms. In G. Rudolph, Editor, *Parallel Problem Solving from Nature*, Lecture notes in computer science, Seiten 528–538. Springer, Berlin, 2008. (Zitiert auf Seite 103.)
- [128] I. M. Navon und D. M. Legler. Conjugate-gradient methods for large-scale minimization in meteorology. *Monthly Weather Review*, 115(8):1479–1502, 1987. (Zitiert auf Seite 157.)
- [129] J. A. Nelder und R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965. (Zitiert auf Seite 156.)
- [130] J. Nocedal und S. J. Wright. *Numerical optimization*. Springer, New York, 2. Edition, 2006. (Zitiert auf Seite 157.)
- [131] Z. Pan, J. Polden, N. Larkin, S. van Duin und J. Norrish. Recent progress on programming methods for industrial robots. *Robotics and Computer-Integrated Manufacturing*, 28(2):87–94, 2012. (Zitiert auf Seiten 1 und 25.)
- [132] D. C. Paraskevopoulos, C. D. Tarantilis und G. Ioannou. Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm. *Expert Systems with Applications*, 39(4):3983–3994, 2012. (Zitiert auf Seite 85.)
- [133] S. Polyakovskiy, M. R. Bonyadi, M. Wagner, Z. Michalewicz und F. Neumann. A comprehensive benchmark set and heuristics for the traveling thief problem. In D. V. Arnold, Editor, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Seiten 477–484, New York, NY, USA, 2014. ACM. (Zitiert auf Seite 83.)
- [134] S. Polyakovskiy und F. Neumann. Packing while traveling: Mixed integer programming for a class of nonlinear knapsack problems. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Seiten 332–346, 2015. (Zitiert auf Seite 84.)
- [135] K. Puljic und R. Manger. Comparison of eight evolutionary crossover operators for the vehicle routing problem. In B. Bhushan, Editor, *Introduction to Tribology*, Tribology series, Seiten 1–8. Wiley, Chichester, West Sussex, United Kingdom, 2013. (Zitiert auf Seiten 77 und 81.)
- [136] C. R. Raquel und P. C. Naval Jr. An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, Seiten 257–264. ACM, 2005. (Zitiert auf Seite 78.)

- [137] T. Rayment und P. S. Grant. Modeling the heat flow in spray formed steel shells for tooling applications. *Metallurgical and Materials Transactions B*, 37(6):1037–1047, 2006. (Zitiert auf Seiten 35 und 54.)
- [138] N. M. Razali und J. Geraghty. Genetic algorithms performance with different selection strategies in solving tsp. In *The 2011 International Conference of Computational Intelligence and Intelligent Systems*, Band 2, 2011. (Zitiert auf Seite 81.)
- [139] R. H. Refsnæs. A brief introduction to the conjugate gradient method. Technical report, NTNU, Trondheim, Norwegen, 2009. (Zitiert auf Seite 164.)
- [140] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè und M. Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009. (Zitiert auf Seite 36.)
- [141] A. Sacharow, S. Odendahl, A. Peuker, D. Biermann, T. Surmann und A. Zabel. Iterative, simulation-based shape modification by free-form deformation of the nc programs. *Advances in Engineering Software*, 56:63–71, 2013. (Zitiert auf Seite 149.)
- [142] A. Sadovoy, R. Subramanian, S. Kycheck und A. Borchardt. Optimization of thermal spray process development by off-line programming with coating thickness simulation. In K.-D. Bouzakis, K. Bobzin, B. Denkena und M. Merklein, Editoren, *Proceedings of the 10th International Conference The "A" Coatings*, Schriftenreihe Oberflächentechnik, Seiten 171–182, Aachen, 2013. Shaker. (Zitiert auf Seite 29.)
- [143] R. Sarker, M. Omar, S. M. Kamrul Hasan und D. Essam. Hybrid evolutionary algorithm for job scheduling under machine maintenance. *Applied Soft Computing*, 13(3):1440–1447, 2013. (Zitiert auf Seite 85.)
- [144] J. Schneider. The time-dependent traveling salesman problem. *Physica A: Statistical Mechanics and its Applications*, 314(1):151–155, 2002. (Zitiert auf Seite 80.)
- [145] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain: Technical report cmu-cs-94-125. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1994. (Zitiert auf Seite 157.)
- [146] P. Smid. *CNC programming handbook*. Industrial Press, 2007. (Zitiert auf Seite 15.)
- [147] J. E. Smith. Self-adaptation in evolutionary algorithms for combinatorial optimisation. In *Adaptive and Multilevel Metaheuristics*, Seiten 31–57. Springer, 2008. (Zitiert auf Seite 103.)
- [148] O. P. Solonenko, V. V. Kudinov, A. V. Smirnov, A. N. Cherepanov, V. N. Popov, A. A. Mikhalchnko und E. V. Kartaev. Micro-metallurgy of splats: Theory,

- computer simulation and experiment. *JSME International Journal Series B*, 48(3):366–380, 2005. (Zitiert auf Seite 29.)
- [149] M. W. Spong, S. Hutchinson und M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Hoboken, NJ, 2006. (Zitiert auf Seite 24.)
- [150] N. Srinivas und K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 2(3):221–248, 1994. (Zitiert auf Seite 79.)
- [151] S. G. Tanyer, A. E. Yilmaz und F. Yaman. Adaptive desirability function for multiobjective design of thinned array antennas. *Journal of Electromagnetic Waves and Applications*, 26(17-18):2410–2417, 2012. (Zitiert auf Seite 89.)
- [152] G. Tao und Z. Michalewicz. Inver-over operator for the tsp. In A. E. Eiben, Editor, *Parallel Problem Solving from Nature-PPSN V*, Lecture notes in computer science, Seiten 803–812. Springer, Berlin and New York, 1998. (Zitiert auf Seiten 81 und 94.)
- [153] M. F. Tasgetiren, Q.-K. Pan, D. Kizilay und G. Suer. A differential evolution algorithm with variable neighborhood search for multidimensional knapsack problem. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, Seiten 2797–2804, 2015. (Zitiert auf Seite 82.)
- [154] M. Thirumaleshwar. *Fundamentals of Heat and Mass Transfer*. Pearson Education India, India, 2006. (Zitiert auf Seite 37.)
- [155] W. Tillmann, P. Hollingsworth, I. Baumann, H. Müller, T. Wiederkehr, D. Hegels, B. Kuhlenkötter und B. Johnen. Adaption of the path strategy to produce near net shape coatings manufactured by hvof spraying. In *Proceedings of International Thermal Spray Conference & Exposition 2013*. DVS, 2014. (Zitiert auf Seiten 13, 14 und 156.)
- [156] R. Tinós, D. Whitley und G. Ochoa. Generalized asymmetric partition crossover (gapx) for the asymmetric tsp. In D. V. Arnold, Editor, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Seiten 501–508, New York, NY, USA, 2014. ACM. (Zitiert auf Seiten 77 und 81.)
- [157] H. Trautmann und C. Weihs. On the distribution of the desirability index using harrington’s desirability function. *Metrika*, 63(2):207–213, 2006. (Zitiert auf Seiten 88 und 98.)
- [158] H.-K. Tsai, J.-M. Yang, Y.-F. Tsai und C.-Y. Kao. An evolutionary algorithm for large traveling salesman problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 34(4):1718–1729, 2004. (Zitiert auf Seite 81.)
- [159] R. K. Ursem. Diversity-guided evolutionary algorithms. In *Parallel Problem Solving from Nature*, Seiten 462–471. Springer, 2002. (Zitiert auf Seite 78.)

- [160] N. Vahrenkamp, T. Asfour und R. Dillmann. Robot placement based on reachability inversion. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Seiten 1970–1975, 2013. (Zitiert auf Seite 69.)
- [161] K. van Moffaert, M. M. Drugan und A. Nowe. Scalarized multi-objective reinforcement learning: Novel design techniques. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, Seiten 191–199, 2013. (Zitiert auf Seiten 76 und 78.)
- [162] D. A. G. Vieira und A. C. Lisboa. Line search methods with guaranteed asymptotical convergence to an improving local optimum of multimodal functions. *European Journal of Operational Research*, 235(1):38–46, 2014. (Zitiert auf Seiten 158 und 159.)
- [163] M. Vince, A. Pichler, G. Biegelbauer, K. Häusler, H. Andersen, O. Madsen und M. Kristiansen. Automatic robotic spray painting of low volume high variant parts. *Proceedings of the 33rd ISR (International Symposium on Robotics)*, 2002. (Zitiert auf Seite 51.)
- [164] T. Vivekanandhan, A. R. Kashani und R. Echempati. Computer-aided torch trajectory generation for automated coating of parts with complex surfaces. *Journal of Thermal Spray Technology*, 3(2):208–215, 1994. (Zitiert auf Seite 17.)
- [165] T. Voss, N. Beume, G. Rudolph und C. Igel. Scalarization versus indicator-based selection in multi-objective cma evolution strategies. In *2008 IEEE Congress on Evolutionary Computation (CEC)*, Seiten 3036–3043, 2008. (Zitiert auf Seiten 76, 78 und 98.)
- [166] T. Wagner, D. Hegels, T. Wiederkehr, A. Peuker, S. Odendahl, S. Rausch, D. Biermann, C. Buchheim und H. Müller. Optimization of the process chain for the efficient manufacturing of forming tools with complex 3d surfaces. In W. Tillmann, Editor, *6. Öffentliches Kolloquium - SFB 708*, Seiten 213–221. Verlag Praxiswissen, 2013. (Zitiert auf Seite 13.)
- [167] Z. Wang, C. Liu, L. Cheng und X. Fan. Optimization of spraying path overlap rate based on matlab. In *2nd International Conference on Consumer Electronics, Communications and Networks*, Band 2, Seiten 2731–2734. IEEE, 2012. (Zitiert auf Seite 51.)
- [168] J. Watson, C. Ross, V. Eisele, J. Denton, J. Bins, C. Guerra, D. Whitley und A. Howe. The traveling salesrep problem, edge assembly crossover, and 2-opt. In A. E. Eiben, T. Bäck, M. Schoenauer und H.-P. Schwefel, Editoren, *Parallel Problem Solving from Nature PPSN V*, Lecture notes in computer science, Seiten 823–832. Springer-Verlag, Berlin and Heidelberg, 1998. (Zitiert auf Seite 94.)
- [169] T. Wiederkehr. *Effiziente, GPU-basierte Simulation thermischer Spritzprozesse*. Dissertation, TU Dortmund, Dortmund, 1.10.2015. (Zitiert auf Seiten 4 und 11.)

- [170] T. Wiederkehr und H. Müller. Acquisition and optimization of three-dimensional spray footprint profiles for coating simulations. *Journal of Thermal Spray Technology*, 22(6):1044–1052, 2013. (Zitiert auf Seite 29.)
- [171] T. Wiederkehr und H. Müller. Efficient large-scale coating microstructure formation using realistic cfd models. *Journal of Thermal Spray Technology*, 24(3):283–295, 2015. (Zitiert auf Seite 29.)
- [172] T. Wiederkehr, H. Müller, D. Hegels, W. Tillmann und L. Hagen. Fast coating deposition simulation for path planning and iterative net-shape optimization on complex workpieces. In *International Thermal Spray Conference and Exposition (ITSC)*, Seiten 390–397, Materials Park, Ohio, 2015. Asm International. (Zitiert auf Seiten 13 und 151.)
- [173] T. Wiederkehr, H. Müller, B. Krebs und M. Abdulgader. A deposition model for wire arc spraying and its computationally efficient simulation. In *Proceedings of the International Thermal Spray Conference*, Seiten 492–498. Asm International, 2009. (Zitiert auf Seiten 29 und 151.)
- [174] K. Yang, M. Liu, K. Zhou und C. Deng. Recent developments in the research of splat formation process in thermal spraying. *Journal of Materials*, 2013(2):1–14, 2013. (Zitiert auf Seite 29.)
- [175] M. Zaefferer, B. Breiderhoff, B. Naujoks, M. Friese, J. Stork, A. Fischbach, O. Flasch und T. Bartz-Beielstein. Tuning multi-objective optimization algorithms for cyclone dust separators. In D. V. Arnold, Editor, *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, Seiten 1223–1230, New York, NY, USA, 2014. ACM. (Zitiert auf Seite 104.)
- [176] H. Zhang, W. Xia, G. Wang, Y. Yang und Y. Zou. An integrated investigation approach for coating temperature measurement and control during plasma spraying. *Journal of Thermal Spray Technology*, 17(2):263–274, 2008. (Zitiert auf Seite 55.)
- [177] L. Zhang. New versions of the hestenes-stiefel nonlinear conjugate gradient method based on the secant condition for optimization. *Computational & Applied Mathematics*, 28:111–133, 2009. (Zitiert auf Seite 157.)
- [178] L. Zhang, W. Zhou und L. Dong-Hui. A descent modified polak-ribière-polyak conjugate gradient method and its global convergence. *IMA Journal of Numerical Analysis*, 26(4):629–640, 2006. (Zitiert auf Seite 158.)
- [179] E. Zitzler, M. Laumanns und L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich, 2001. (Zitiert auf Seite 79.)

- 
- [180] E. Zitzler und L. Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International Conference on Parallel Problem Solving from Nature*, Seiten 292–301, 1998. (Zitiert auf Seite 82.)
- [181] E. Zitzler und L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999. (Zitiert auf Seite 79.)