

Automatisierung der Weißlichtinterferometrie zur  
Inspektion rotationssymmetrischer  
Triebwerksbauteile

Vom Promotionsausschuss der  
Technischen Universität Hamburg-Harburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von

Dipl.-Ing. Tomas Domaschke

aus

Hamburg

2017

1.Gutachter

Prof. Dr.-Ing. Thorsten Schüppstuhl  
Institut für Flugzeug-Produktionstechnik  
Technische Universität Hamburg-Harburg

2.Gutachter

Prof. Dr.-Ing. Bernd Kuhlenkötter  
Lehrstuhl für Produktionssysteme  
Ruhr-Universität Bochum

Tag der mündlichen Prüfung:

22.09.2017



# Danksagung

Diese Dissertation ist während meiner Zeit als wissenschaftlicher Mitarbeiter am Institut für Flugzeug-Produktionstechnik der Technischen Universität Hamburg-Harburg entstanden. Ich möchte mich bei dem Institutsleiter und meinem Doktorvater Professor Thorsten Schüppstuhl für die stets hervorragende Betreuungsarbeit, die anregenden Gespräche und das unerschütterliche Vertrauen in meine Arbeit bedanken. Ebenfalls möchte ich Professor Bernd Kuhlenkötter für die Betreuung meiner Arbeit und Professor Hermann Lödding für die Übernahme des Vorsitzes meiner Prüfungskommission danken.

Außerdem möchte ich mich bei meinen Arbeitskollegen und Freunden aus dem AutoInspect-Team und vom Institut bedanken, die mit ihren unzähligen Denkanstößen und konstruktiver Kritik einen großen Teil zu dieser Arbeit beigetragen haben und mir bei der Korrektur behilflich waren. Mein besonderer Dank gilt hierbei Marc Otto und Tobias Flüh, mit denen ich viele Tage und Nächte durchprogrammiert habe und die mir auch in schwersten Zeiten selbstlos zur Seite gestanden und dadurch den erfolgreichen Abschluss dieser Arbeit erst ermöglicht haben.

Abschließend möchte ich mich auch bei meinen Eltern Silke und Wolfgang und besonders bei meiner Frau Alice dafür bedanken, dass sie zu keinem Zeitpunkt auf meinem langen und nicht immer einfachen Weg zur Promotion an mir gezweifelt und mich durchgängig in meinen Plänen und Zielen bestärkt und in der Umsetzung dieser unterstützt haben.



# Zusammenfassung

Aus Gründen der Sicherheit schreiben Luftfahrtbehörden die Instandhaltung von Flugzeugen in engen Intervallen vor. Diese trägt daher einen erheblichen Teil zu den Kosten bei, die im Laufe eines Flugzeuglebens anfallen. Dies gilt insbesondere für die hochbelastete Baugruppe der Triebwerke, die bei der Instandhaltung gesondert betrachtet wird. Viele Bauteile aus dem Triebwerk stellen teure Investitionsgüter dar, weswegen sie in den meisten Fällen nicht ausgetauscht, sondern stattdessen instandgesetzt werden. Im Bereich der Brennkammer sind typische Schäden Korrosion, Verbrennungen sowie Risse. Damit letztere zuverlässig detektiert werden, sind besondere Inspektionsverfahren nötig.

Im Rahmen eines Forschungsvorhabens sollte das lang etablierte, aber gleichzeitig Schwächen aufweisende Verfahren der FPI durch die optische Technologie der WLI substituiert werden. Damit diese effektiv eingesetzt werden kann, ist ein automatisches System zur Handhabung notwendig. Die Zielstellung dieser Arbeit ist die Entwicklung eines solchen automatischen Inspektionssystems. Hierfür mussten der Inspektionsaufbau und ein Inspektionsprozess konzipiert und später implementiert werden. Bei der Auswahl der Systemkomponenten mussten Erreichbarkeit, Zugänglichkeit, Messabdeckung sowie Schwingungen berücksichtigt werden. Die Herausforderungen im Inspektionsprozess entstammen der individuellen Geometrie der beschädigten Bauteile, die einen hohen Grad an Adaptivität bei dem System voraussetzen. Hierfür wird in dieser Arbeit ein neuartiger zweistufiger Ansatz für die Pfadplanung vorgestellt. Dieser besteht aus einer modellbasierten Inspektionsplanung und einer sich anschließenden Pfadadaption. Dabei wird zur Datenreduktion die Rotationssymmetrie des Bauteils ausgenutzt. Außerdem wurden die entsprechend benötigten intelligenten Softwarebausteine in einem Framework zu einer Software für die Off-Line-Programmierung sowie On-Line-Steuerung von Industrierobotern kombiniert. Für diesen off-Line ausgeführten Pfadplanungsansatz wird die reale Bauteiloberfläche benötigt. Diese wird in einem vorgelagerten Hilfsprozess mit einem zusätzlichen LLS-Sensor akquiriert und der Planung bereitgestellt. Alle Komponenten des Systems mussten außerdem zueinander kalibriert werden, um die notwendige Genauigkeit zu garantieren. Anhand eines realen Brennkammerbauteils wurde der gesamte Inspektionsprozess im Ganzen und jeder einzelne Hilfsprozess für sich qualitativ validiert. Schon mit der Entscheidung für die Substitution der FPI durch eine automatisierte WLI wurden die formulierten Anforderungen, was die Belastung von Mensch und Umwelt sowie die Reproduzierbarkeit betrifft, berücksichtigt. Zusätzlich generiert das in dieser Arbeit implementierte Inspektionssystem in einer vergleichbar kurzen Durchlaufzeit durchschnittlich 60.000 Einzelaufnahmen und 130 GB der vollständigen Bauteiloberfläche in Mikrometerauflösung, womit auch die weiteren quantitativen globalen Anforderungen erfüllt sind. Diese hoch aufgelösten Geometrieinformationen stellen eine wertvolle Erweiterung des Inspektionsresultates dar und stärken somit die Position des instandhaltenden Unternehmens gegenüber dem Hersteller der betrachteten Bauteile.



# Inhaltsverzeichnis

Abbildungsverzeichnis	i
Tabellenverzeichnis	iii
Abkürzungsverzeichnis	v
<b>1. Einleitung</b>	<b>1</b>
1.1. Hintergrund . . . . .	1
1.1.1. Instandhaltung in der Luftfahrtindustrie . . . . .	1
1.1.2. Optimierung der Instandhaltung . . . . .	2
1.1.3. Automatisierte Rissinspektion . . . . .	4
1.2. Motivation - Das Projekt AutoInspect . . . . .	5
1.3. Randbedingungen bei der Rissinspektion . . . . .	6
1.3.1. Betrachtetes Bauteilspektrum . . . . .	7
1.3.2. Messprinzip . . . . .	8
1.3.3. Handhabung . . . . .	10
1.4. Zielsetzung der Arbeit . . . . .	11
<b>2. Stand von Wissenschaft und Technik</b>	<b>15</b>
2.1. Weißlichtinterferometrie für die Inspektion . . . . .	15
2.2. Pfadplanung für die automatische Inspektion . . . . .	17
2.2.1. On-Line-Pfadplanung von Industrierobotern . . . . .	17
2.2.2. Off-Line-Pfadplanung von Industrierobotern . . . . .	18
2.2.3. Modellbasierte Pfadplanung für die Inspektion . . . . .	20
2.2.3.1. Graphentheoretische Ansätze . . . . .	22
2.2.3.2. Mengentheoretische Ansätze . . . . .	22
2.3. Software für eine automatische robotergestützte Inspektion . . . . .	26
2.3.1. Proprietäre und generische Softwarelösungen . . . . .	27
2.3.2. Frameworks . . . . .	27
2.4. Diskussion des aktuellen Standes der Wissenschaft . . . . .	29
2.4.1. Weißlichtinterferometrie für die Inspektion . . . . .	29
2.4.2. Automatische Inspektionsplanung . . . . .	30
2.4.3. Softwarebasierte Steuerungssysteme . . . . .	30
2.5. Aufbau der Arbeit . . . . .	31
<b>3. Konzipierung des Inspektionsaufbaus</b>	<b>33</b>
3.1. Vollständige Inspektion . . . . .	34
3.1.1. Auswahl eines Industrieroboters . . . . .	34
3.1.2. Auswahl zusätzlicher Handhabungskomponenten . . . . .	35

3.1.3.	Konzept zur Bestimmung der Messabdeckung . . . . .	37
3.2.	Reduktion von Schwingungen . . . . .	39
3.2.1.	Schwingungen innerhalb des Systems . . . . .	39
3.2.1.1.	Schwingungen durch den Roboter . . . . .	40
3.2.1.2.	Schwingungen durch die Drehachse . . . . .	42
3.2.2.	Schwingungen von außerhalb des Systems . . . . .	45
3.3.	Zusätzliche Sensorik . . . . .	48
3.3.1.	Optisch messende 3D-Verfahren . . . . .	48
3.3.2.	Lichtstrahlverfahren . . . . .	49
3.3.3.	Lichtschnittverfahren . . . . .	50
3.3.4.	Steifenprojektion . . . . .	50
3.3.5.	Photogrammetrie . . . . .	52
3.3.6.	Die Verfahren im Vergleich . . . . .	52
3.3.7.	Integration des Sensors in das Inspektionssystem . . . . .	54
<b>4.</b>	<b>Konzipierung des Inspektionsprozesses</b>	<b>55</b>
4.1.	Übergeordnete Prozessstruktur . . . . .	55
4.2.	Kalibrierung des Inspektionssystems . . . . .	56
4.2.1.	Kalibrierung der Werkzeugspitze . . . . .	57
4.2.2.	Kalibrierung der Drehachsbasis . . . . .	58
4.3.	Bauteildigitalisierung mit dem Laserlichtschnittverfahren . . . . .	59
4.4.	Pfadplanung für die Inspektion . . . . .	61
4.5.	Automatische Rissinspektion mit einem Weißlichtinterferometer . . . . .	63
<b>5.</b>	<b>Software zur automatischen Inspektion</b>	<b>65</b>
5.1.	Simulation zur Bestimmung der Messabdeckung . . . . .	65
5.2.	Steuersoftware AutoXIV . . . . .	68
5.2.1.	Aufbau der Software . . . . .	69
5.2.2.	Abbildung des Inspektionsprozesses in <i>AutoXIV</i> . . . . .	70
5.2.3.	Implementierung von Arbeiterklassen . . . . .	73
5.2.4.	Netzwerkarchitektur . . . . .	73
<b>6.</b>	<b>Ausgestaltung und Implementierung des Inspektionsprozesses</b>	<b>75</b>
6.1.	Kalibrierung des Inspektionssystems . . . . .	75
6.1.1.	Kalibrierung der Werkzeugspitzen . . . . .	75
6.1.1.1.	Schritt 1: Kalibrierung des rotatorischen Anteils . . . . .	76
6.1.1.2.	Schritt 2: Kalibrierung des translatorischen Anteils . . . . .	77
6.1.1.3.	Bestimmung von Kugelzentren . . . . .	77
6.1.2.	Dokumentation des Kalibrierergebnisses . . . . .	79
6.1.3.	Kalibrierung der Drehachsbasis . . . . .	80

6.2. Bauteildigitalisierung mit dem Laserlichtschnittverfahren . . . . .	81
6.2.1. Datensynchronisation . . . . .	82
6.2.2. Homogenität der Oberflächendaten . . . . .	83
6.2.2.1. Achsgeschwindigkeit . . . . .	84
6.2.2.2. Datenzugriff . . . . .	86
6.3. Pfadplanung für die Inspektion mit einem WLI . . . . .	88
6.3.1. Stufe 1: Modellbasierter Ansatz zur lokalen Pfadplanung . . . . .	89
6.3.1.1. Gewinnung von Oberflächenkandidaten . . . . .	89
6.3.1.2. Gewinnung von Sensorpositionskandidaten . . . . .	91
6.3.1.3. Bestimmung der Messbarkeit . . . . .	93
6.3.1.4. Extraktion einer Untermenge . . . . .	98
6.3.2. Stufe 2: globale Pfadadaption auf die individuelle Bauteilgeometrie . . . .	100
6.3.2.1. Optimierung des Eingangspfades . . . . .	100
6.3.2.2. Adaption der Messpunkte an die individuelle Bauteilgeometrie .	103
6.3.2.3. Erweiterung der Adaption auf zwei Dimensionen . . . . .	105
6.4. Automatische Rissinspektion mit einem Weißlichtinterferometer . . . . .	107
<b>7. Validierung des Inspektionssystems</b>	<b>109</b>
7.1. Inspektionsaufbau . . . . .	109
7.1.1. Zugänglichkeit und Erreichbarkeit . . . . .	110
7.1.2. Messabdeckung . . . . .	112
7.2. Validierung des Prozessfunktionalität . . . . .	114
7.2.1. Kalibrierung . . . . .	114
7.2.1.1. Kalibrierung der Werkzeugspitzen . . . . .	114
7.2.1.2. Kalibrierung der Drehachsbasis . . . . .	115
7.2.2. Digitalisierung . . . . .	116
7.2.3. Pfadplanung . . . . .	118
7.2.3.1. Modellbasierte Pfadplanung . . . . .	118
7.2.3.2. Optimierung des Eingangspfades . . . . .	124
7.2.3.3. Adaption der Messpunkte . . . . .	127
7.2.4. Automatische Rissinspektion mit einem Weißlichtinterferometer . . . . .	129
<b>8. Zusammenfassung und Ausblick</b>	<b>133</b>
8.1. Zusammenfassung . . . . .	133
8.2. Ausblick . . . . .	134
<b>A. Anhang</b>	<b>137</b>
A.1. Kalibrierprotokolle WLI-Sensor . . . . .	137
A.2. Kalibrierprotokolle LLS-Sensor . . . . .	146
<b>B. Lebenslauf</b>	<b>161</b>





# Abbildungsverzeichnis

1.1. Konstellation und Abhängigkeiten in der Luftfahrtinstandhaltung . . . . .	4
1.2. 3D-Modell des betrachteten Brennkammerbauteils . . . . .	7
1.3. Messprinzip und Intensitätsverteilung bei der Weißlichtinterferometrie . . . . .	9
(a). Messprinzip . . . . .	9
(b). Intensität auf einem Pixel über Gangunterschied von Proben- und Referenzstrahl . . . . .	9
1.4. farbcodierte Höhentopologie als Ergebnis einer WLI-Messung . . . . .	9
1.5. globale Anforderungen sowie aus den Randbedingungen resultierende Herausforderungen . . . . .	12
1.6. Aufbau der übergeordneten Zielstellung . . . . .	13
2.1. typischer Ablauf einer CAD-basierten Off-Line-Programmierung . . . . .	19
2.2. Einordnung der modellbasierten Inspektionsplanung in die Roboterprogrammierung	20
2.3. Algorithmen des MBVP . . . . .	21
2.4. Diskretisierung einer „Viewsphere“ . . . . .	23
3.1. Schematischer Arbeitsraum des Industrieroboters Kuka KR30 in Seiten- und Draufsicht mit Bauteil . . . . .	35
3.2. Positionierung der zusätzlichen Drehachse in Relation zum Industrieroboter . . .	36
(a). isometrische Ansicht . . . . .	36
(b). Draufsicht . . . . .	36
3.3. prinzipielles Vorgehen bei der Inspektion mit einem Weißlichtinterferometer . . .	36
3.4. Adaption durch Verfahren der WLI-Linearachse . . . . .	37
3.5. Messbereichsscharfe Abdeckungsbestimmung . . . . .	39
(a). schematisches Vorgehen . . . . .	39
(b). nicht messbereichsscharfes „Ausfransen“ bei <i>FAMOS</i> . . . . .	39
3.6. Messaufbau mit Lasertracker in Roboterzelle . . . . .	40
3.7. Tripelspiegel-Target an Werkzeug und Werkstückhalterung . . . . .	40
3.8. Ergebnis der Lasertracker-Messung mit Roboter in Bremse . . . . .	41
3.9. Ergebnis der Lasertracker-Messung mit Roboter in Regelung . . . . .	41
3.10. Vermessung einer Drehachsbewegung . . . . .	43
3.11. Betrachtung des Ausschwingverhaltens . . . . .	43
3.12. Schwingungsmessung der Drehachse mit Hilfe des WLI-Sensors . . . . .	44
3.13. Schwingungsmessung der Drehachse mit Hilfe des WLI-Sensors und neuer Spannvorrichtung . . . . .	45
3.14. Dämpfungstöpfe unter dem Stahlbau der Anlage zur Reduktion der von außen eingebrachten Schwingungen . . . . .	45
3.15. Zeitsignal des Roboters nach Anregung durch einen mechanischen Impuls . . . .	46

3.16. Frequenzspektrum des Zeitsignals nach Anregung der Drehachse durch einen mechanischen Impuls . . . . .	47
3.17. Frequenzspektrum des Zeitsignals nach Anregung des Roboters durch einen mechanischen Impuls . . . . .	47
3.18. Übersicht der zerstörungsfreien Prüfverfahren zur Oberflächenvermessung . . . .	48
3.19. Übersicht der 3D-Triangulationsverfahren . . . . .	49
3.20. Geometrischer Aufbau beim Triangulationsverfahren . . . . .	50
3.21. Aufbau beim Laserlichtschnittverfahren . . . . .	51
3.22. Signalmodell der Streifenprojektion . . . . .	51
3.23. Gemeinsame Werkzeughalterung von WLI- und LLS-Sensor am Roboterflansch .	54
4.1. Übersicht der notwendigen Subprozesse im Inspektionsprozess und deren Anordnung	56
4.2. Verkettete Transformation im Inspektionssystem . . . . .	57
4.3. Varianten zur Kalibrierung eines Scanner TCP's nach Gan.2011: links „eye-to-hand“, rechts „eye-in-hand“ . . . . .	58
4.4. Drehachsbasis im Zentrum der Drehachse in Höhe der Werkstückaufnahme mit Kugeltarget . . . . .	59
4.5. Ringweise Aufnahmestrategie mit dem LLS-Sensor durch das Verfahren der Drehachse . . . . .	60
4.6. Mögliches Vorgehen bei unbeschädigter, idealer Bauteilgeometrie . . . . .	61
4.7. Prinzipieller Ablauf der Pfadplanung . . . . .	62
4.8. Merkmale und Störgeometrien an dem Bauteil, links: Panele und Löcher, Mitte: weitere Löcher, rechts: Zündkerzeneinlässe . . . . .	63
4.9. Messstrategie bei der Inspektion . . . . .	64
5.1. Prozessablaufplan des <i>MATLAB</i> -Skriptes zur Bestimmung der Messabdeckung .	66
5.2. Klassendiagramm der wichtigsten Klassen des <i>MATLAB</i> -Skriptes zur Bestimmung der Messabdeckung . . . . .	67
5.3. Ergebnisse bei der Bestimmung der Messabdeckung . . . . .	68
(a). Ergebnis der Messabdeckung in <i>MATLAB</i> . . . . .	68
(b). STL-Modell der Messabdeckung in <i>FAMOS</i> . . . . .	68
5.4. Benutzeroberfläche der Software <i>AutoXIV</i> . . . . .	70
5.5. Vereinfachte Übersicht der unterschiedlichen Schichten der verwendeten Zustandsautomaten . . . . .	71
5.6. Netzwerk im Inspektionsaufbau . . . . .	74
6.1. Schematische Messbereiche der unterschiedlichen Sensoren auf dem Kugeltarget: links WLI, rechts LLS . . . . .	77
6.2. Kalibrierergebnis visualisiert in <i>AutoXIV</i> . . . . .	79
6.3. Variablen bei der Kalibrierung der Werkstückbasis . . . . .	81

6.4. Überlagerung verschiedener Messpositionen zur Veranschaulichung des sukzessiven Vorgehens bei der Digitalisierung . . . . .	82
6.5. Beschleunigungs-, Geschwindigkeits- und Bewegungsverlauf einer Verfahrbewegung der Drehachse . . . . .	85
6.6. Einfluss von Auflösung und Extrapolation auf Datenhomogenität . . . . .	86
(a).    passende Auflösung und Extrapolation . . . . .	86
(b).    falsche Auflösung . . . . .	86
(c).    keine Extrapolation . . . . .	86
6.7. Ablauf bei asynchroner Datenaufnahme und Visualisierung in dem Prozessschritt der Digitalisierung . . . . .	87
6.8. Die zwei Stufen des neuen Ansatzes für die Inspektionsplanung . . . . .	88
6.9. Aus dem Gesamtmodell herausgefiltertes Segment des Umfangs . . . . .	89
6.10. Vereinfachung durch Bauteiltransformation . . . . .	91
(a).    Schema der Bauteiltransformation vom Zylinderraum in den kartesischen Raum . . . . .	91
(b).    Schema der Abbildung vom dreidimensionalen in den zweidimensionalen Raum . . . . .	91
6.11. Schritte bei der Gewinnung von Sensorpositions-kandidaten . . . . .	93
(a).    Ableitung der Sensorpositions-kandidaten aus den Objektkandidaten . . .	93
(b).    Aufstellen der Messbarkeitsmatrix . . . . .	93
6.12. Bounding volumes unterschiedlicher Komplexität . . . . .	95
6.13. Überprüfung des Kriteriums Messbereich . . . . .	96
6.14. Bestimmung der Verdeckung . . . . .	97
6.15. Positive Verdeckungsüberprüfung mit einem Quadtree . . . . .	98
6.16. Greedy-Algorithmus auf eine Beispielmatrix angewendet . . . . .	99
6.17. Prinzipieller Ablauf des Greedy-Algorithmus . . . . .	100
6.18. Filterung des Bauteilbereichs in der messbereichrelevanten Höhe . . . . .	101
6.19. 5 Schritte einer Euclidean-Cluster-Extraktion . . . . .	102
6.20. Darstellung der benötigten Schritte zur globalen Optimierung der Pfadpunkte auf dem Schema eines Bauteilsegmentes. . . . .	103
6.21. Bestimmung des Adaptionwertes und der Messtiefe . . . . .	104
6.22. Zentrierung und iterative Vervielfachung des Pfadpunktes im horizontalen Bauteilbereich . . . . .	105
6.23. Detaillierte Übersicht aller Prozessschritte der Inspektionsplanung . . . . .	106
6.24. Ablauf bei der Inspektion . . . . .	107
7.1. Generierung eines Pfades aus einem Ebenenschnitt. Graph nach Ebenenschnitt mit Bauteil und Roboterpfad auf Schnittgraph . . . . .	110
7.2. Vermessung der reflektierten Lichtintensität unter verschiedenen Winkeln zur Bauteiloberfläche . . . . .	111

7.3.	Versuch zu Intensitäten über die Messwinkel . . . . .	112
7.4.	Simulation der Zugängigkeit mit dem konzipierten Aufbau . . . . .	112
7.5.	Messabdeckung außen und innen beim betrachteten Brennkammerbauteil . . . . .	113
7.6.	Verdeckte Bereiche der Oberfläche in Hinterschnitten des Bauteils (orange) . . . . .	114
7.7.	Modell als Resultat der Digitalisierung mit dem LLS-Sensor . . . . .	117
(a).	Gesamtansicht . . . . .	117
(b).	Nahansicht . . . . .	117
7.8.	Überlappung der Daten von zwei aufeinanderfolgenden Messringen bei der Digitalisierung . . . . .	117
7.9.	Teilergebnisse bei der Generierung von Objektkandidaten . . . . .	119
(a).	Bauteilsegment . . . . .	119
(b).	Transformiertes Segment . . . . .	119
(c).	Abbildung in $y'z'$ -Ebene . . . . .	119
7.10.	Messbarkeitsmatrix mit Sensorpositionskandidaten und Objektkandidaten farbcodiert als Bild nach Anwendung der einzelnen Messbarkeitskriterien . . . . .	122
7.11.	Aus der Messbarkeitsmatrix gefilterte Sensorpositionen, verbunden mit ihren messbaren Objektpunkten . . . . .	123
7.12.	Die Transformation der Sensorpositionen zu Zylinderkoordinaten führt zu dem Roboterpfad als Ergebnis der modellbasierten Pfadplanung . . . . .	124
7.13.	Verschiedene Ansichten des aufgenommenen Modells nach der automatischen Inspektion, dargestellt in der Software <i>AutoProcess</i> . . . . .	130
(a).	Gesamtmodell . . . . .	130
(b).	Ein Segment mit Riss . . . . .	130
(c).	Rohdaten eines Segmentes . . . . .	130
(d).	Riss in Nahansicht . . . . .	130
8.1.	Demonstratoranlage zur automatischen Inspektion mit Weißlichtinterferometrie . . . . .	134

# Tabellenverzeichnis

3.1. Eigenschaften der Subkomponenten . . . . .	33
3.2. ZIELERTRÄGE der einzelnen Verfahren hinsichtlich der formulierten Zielkriterien . .	53
3.3. Gewichtete Zielwerte der einzelnen Verfahren . . . . .	53
3.4. Nutzwerte der betrachteten Verfahren . . . . .	53
7.1. Übersicht der Anforderungen an den Inspektionsprozess . . . . .	109
7.2. Statistische Auswertung der TCP Kalibrierung . . . . .	115
7.3. Statistische Auswertung bei der Kalibrierung der Werkstückbasis . . . . .	116
7.4. Unterschiedliche Kriterien bei der Überprüfung der Messbarkeit . . . . .	121
7.5. Teilschritte der Pfadoptimierung als 1.Stufe der Pfadadaption auf die globale Bauteilgeometrie . . . . .	127
7.6. Teilschritte der Pfadadaption als 2.Stufe der übergeordneten Adaption auf die globale Bauteilgeometrie . . . . .	129



# Abkürzungsverzeichnis

<b>API</b>	Application Programming Interface
<b>ASCII</b>	American Standard Code for Information Interchange
<b>BMWi</b>	Bundesministerium für Wirtschaft und Energie
<b>BUOP</b>	Bediener-unterstützte-Online Programmierung
<b>BV</b>	Bildverarbeitung
<b>CAD</b>	Computer-Aided-Design
<b>CMM</b>	Koordinatenmessmaschine
<b>CUDA</b>	Compute Unified Device Architecture
<b>DFG</b>	Deutsche Forschungsgemeinschaft
<b>DOF</b>	Degrees Of Freedom
<b>FIS</b>	Flexible Inspection System
<b>FPI</b>	Flourescent Penetrant Inspection
<b>GPU</b>	Graphical Computing Unit
<b>GUI</b>	Graphical User Interface
<b>HA</b>	High Accuracy
<b>ICAO</b>	International Civil Aviation Organization
<b>IFPT</b>	Institut für Flugzeug-Produktionstechnik
<b>IND</b>	Imaging Network Designer
<b>IR</b>	Industrieroboter
<b>KS</b>	Koordinatensystem
<b>LBA</b>	Luftfahrt Bundesamt
<b>LHT</b>	Lufthansa Technik AG
<b>LLS</b>	Laserlichtschnitt
<b>MBVP</b>	Model-based view planning
<b>MM</b>	Messbarkeitmatrix

<b>MRDS</b>	Microsoft Robotics Developer Studio
<b>MRO</b>	Maintenance, Repair and Overhaul
<b>MSC</b>	Minimum Set Cover
<b>NBVP</b>	Next best view planning
<b>NTP</b>	Network Time Protocol
<b>OBB</b>	Oriented Bounding Box
<b>PCL</b>	Pointcloud Library
<b>POD</b>	Probability of Detection
<b>RANSAC</b>	Random Sample Consensus
<b>RBS</b>	Reference Broadcast Synchronisation
<b>ROS</b>	Robot Operating System
<b>RSI</b>	Robot Sensor Interface
<b>SCP</b>	Set Covering Problem
<b>SOP</b>	sensorgestützte-Online-Programmierung
<b>SPK</b>	Sensorpositions-kandidaten
<b>SPS</b>	Speicherprogrammierbare Steuerung
<b>STL</b>	Surface Tessellation Language
<b>SVD</b>	Singular Value Decomposition
<b>TAT</b>	Turnaround Time
<b>TCP</b>	Tool Center Point
<b>TCPIP</b>	Transmission Control Protocol
<b>TOF</b>	Time-of-Flight
<b>TPSN</b>	Timing-Sync Protocol for Sensor Networks
<b>UDP</b>	User Datagram Protocol
<b>UML</b>	Unified Modeling Language
<b>UV</b>	Ultraviolettstrahlung



**VPP** View Planning Problem

**V-REP** Virtual Robot Experimentation Platform

**VTK** The Visualization Toolkit

**WLI** Weißlichtinterferometrie

**XML** Extensible Markup Language

**zfP** zerstörungsfreie Prüfverfahren



# 1. Einleitung

## 1.1. Hintergrund

### 1.1.1. Instandhaltung in der Luftfahrtindustrie

Unter Instandhaltung (engl. Maintenance, Repair and Overhaul (MRO)) versteht man nach DIN 31051 (DIN00) ganz allgemein:

Die Kombination aller technischen und administrativen Maßnahmen während des Lebenszyklus einer Einheit, die dem Erhalt oder der Wiederherstellung ihres funktionsfähigen Zustands dient, sodass sie die geforderte Funktion erfüllen kann.

Die Instandhaltung ist in die vier Grundmaßnahmen *Wartung*, *Inspektion*, *Instandsetzung* und *Verbesserung* unterteilt. Aktuelle Flugzeugmodelle sind durchschnittlich 26 Jahre im aktiven Flugbetrieb und absolvieren dabei bis zu 50 000 Flugstunden (For15). Um über so eine lange Zeit den lufttüchtigen und zuverlässigen Betrieb des Flugzeuges gewährleisten zu können, schreiben die nationalen (Luftfahrt Bundesamt (LBA)) und internationalen (International Civil Aviation Organization (ICAO)) Luftfahrtbehörden in engen Intervallen die Durchführung einer Instandhaltung vor. Gerade in C- und D-Checks in der sogenannten Base-Maintenance sind die Instandhaltungsprozesse oft sehr komplex und umfangreich und beinhalten die Demontage des gesamten Flugzeuges und aller Baugruppen, was bis zu 50 000 Mannstunden beziehungsweise 4-8 Wochen beanspruchen kann (Hin13). Aufgrund des großen Umfangs und der Häufigkeit entfallen bis zu 17 % der operativen Kosten eines Flugzeug auf die Instandhaltung (BOB15).

Triebwerksteile werden bei der Instandhaltung gesondert betrachtet, da sie im Betrieb im Vergleich zu anderen Flugzeugteilen nicht nur hohen mechanischen, sondern im Brennkammer- und Abgasbereich auch noch hohen thermischen Belastungen ausgesetzt sind. Deswegen werden sie aus besonders hitzeresistenten, teuren Superlegierungen gefertigt, was sie zu hochwertigen Investitionsgütern macht. Im Laufe der Zeit wurden für Triebwerke eigene Instandhaltungsprozesse mit engeren Intervallen und eigenen Anforderungen entwickelt. Hierdurch können größere Beschädigungen vermieden werden, da sie schon in einem frühen Stadium erkannt und repariert werden können. Dies wiederum führt zu einer Kostenreduktion, da das reparierte Bauteil nicht durch ein Neuteil ersetzt werden muss. Dieses gesonderte Vorgehen beansprucht fast 20 % der Arbeitskraft im gesamten Instandhaltungsvorgang (Aub04).

An dem Prozess der Instandhaltung sind, wie in Abbildung 1.1 dargestellt, drei verschiedene Parteien beteiligt. Neben dem von der Luftfahrtbehörde gemäß EASA Part 145 zugelassenen Instandhaltungsunternehmen sind das zum einen der Hersteller der Flugzeugbauteile oder -baugruppen und zum anderen die Betreiber der Flugzeuge wie Fluglinien und Leasinggesellschaften. Hersteller und Betreiber stellen individuelle Anforderungen an das Instandhaltungsunternehmen. Der Hersteller, der das jeweilige Bauteil ausgelegt, konstruiert und später gefertigt hat, kennt die eingesetzten Materialien und das entsprechende Ermüdungsverhalten bei mechanischen und

thermischen Belastungen und gibt deswegen einem Instandhalter in sogenannten Manuals Toleranzen vor, die festlegen, ab wann ein Schaden oder Defekt repariert werden oder das defekte Bauteil durch ein Neuteil ersetzt werden muss. Will der Instandhalter neu- beziehungsweise weiterentwickelte Reparaturprozesse einsetzen, müssen diese außerdem vorher durch den jeweiligen Hersteller zertifiziert werden.

Die Betreiber wiederum treffen ihre Entscheidung für die Beauftragung eines Instandhaltungsunternehmens vor allem über wirtschaftlich geprägte Kriterien. Neben den Kosten ist das vor allem die Zeit, die ein Flugzeug in der Instandhaltung verbringt und die damit verbundene „Downtime“. In dieser Zeit steht das Flugzeug am Boden und kann nicht wirtschaftlich genutzt werden. Um diesen verschiedenen Anforderungen gleichzeitig entsprechen zu können, muss das Instandhaltungsunternehmen versuchen, seine Wettbewerbsfähigkeit zu erhöhen, indem es seine Prozesse hinsichtlich Zeit und Qualität verbessert. Dies gilt vor allem für die aufwändige und arbeitsintensive Triebwerksinstandhaltung. Gleichzeitig muss allerdings sichergestellt sein, dass die entsprechenden strengen Vorgaben und Toleranzen bei dieser Optimierung weiterhin eingehalten werden.

### 1.1.2. Optimierung der Instandhaltung

Ein hohes Potential zur Optimierung der Instandhaltung hinsichtlich der genannten Kriterien bietet die Inspektion. Diese spielt eine zentrale Rolle in der Luftfahrt-MRO und hat einen großen Einfluss auf die Ergebnisse einer Instandhaltung. Nur Schäden und Defekte, die in einer Inspektion detektiert werden, können in einem nachgelagerten Prozessschritt behoben werden.

Der Betrieb von Flugzeugteilen führt aufgrund der hohen Belastungen zu Materialermüdung, wodurch unterschiedliche Beschädigungen und Defekten wie Risse, Korrosion und Verbrennungen entstehen können. Die Aufgabe der Inspektion im Rahmen der Instandhaltung ist diese mit Hilfe von zerstörungsfreien Prüfverfahren (zfPn) verlässlich zu detektieren. Die Wahrscheinlichkeit, bei einer Inspektion eine Beschädigung zu detektieren, wird über die Probability of Detection (POD) quantifiziert. Neben der POD ist das Ergebnis einer Inspektion noch abhängig von der Höhe seines Informationsgehaltes. Dieser beschreibt ganz allgemein, wieviel Information eine Nachricht beziehungsweise speziell in diesem Fall ein Inspektionsergebnis enthält. Liefert die Inspektion zum Beispiel neben der reinen binären Information des Vorhandenseins einer Beschädigung zusätzlich auch noch dessen Lage auf dem inspizierten Bauteil und geometrische Daten, kann der Instandhalter durch Auswertung der fusionierten Daten dessen genaues Stadium bestimmen und exakt evaluieren, ob die vorgegebenen Toleranzen vom Hersteller überschritten wurden. Das erleichtert im Anschluss an die Inspektion die qualitative Entscheidung darüber, ob ein Bauteil durch ein neues ersetzt, lediglich instandgesetzt werden muss oder aber, ob ein Schaden bis zum nächsten Instandhaltungszyklus im Material verbleiben kann. Detaillierte Informationen können folglich in vielen Fällen zu einer Verlängerung der Bauteillebenszeit und dadurch gerade bei teuren Investitionsgütern zu einer deutlichen Kostenreduktion für die Betreiber führen. Dieser Faktor stellt für das Instandhaltungsunternehmen, das eine entsprechende Technologie anbietet,

einen direkten Wettbewerbsvorteil gegenüber seinen Konkurrenten dar.

Die Erhöhung des Informationsgehaltes der Beschädigungsdaten stärkt gleichzeitig die Stellung des Instandhalters gegenüber dem Hersteller, da dieser ein Bauteil mehrmals während dessen Lebenszeit inspiziert und eine Historie aller Beschädigungen aufstellen kann. Diese für den Hersteller hochrelevanten Daten können über die Zeit zu einem besseren Verständnis der Entwicklung einer Beschädigung führen und in deren Konstruktionsabteilung zurückfließen, wo entweder das Design des aktuellen Bauteils angepasst oder ein verbessertes Design der nachfolgenden Generation entsprechend ausgelegt werden kann. Dadurch können die Lebenszeit und das Instandhaltungsintervall des Bauteils verlängert und seine Attraktivität für den Betreiber erhöht werden, da er dieses Bauteil dann länger einsetzen kann.

Zusätzlich zu dem Informationsgehalt lässt sich eine Inspektion auch noch über die Reproduzierbarkeit bewerten. Reproduzierbarkeit beziehungsweise Replizierbarkeit gibt in dem Fall der Inspektion an, wie stabil sich vergleichbare Inspektionsergebnisse unter gleichen Versuchsbedingungen erzeugen lassen. Außerdem ist es ein Maß für die Objektivität eines Prozesses. Bei einer hohen Reproduzierbarkeit lassen sich systematische Fehler und die Grenzen eines Inspektionsverfahrens leicht erkennen. So kann auch hier wieder exakt bestimmt werden, dass die vorgegebenen Toleranzen eingehalten werden, was wiederum bei der Zertifizierung eines neuen Inspektionsprozesses eine zentrale Rolle spielt. Eine weitere Kennzahl, die in die Bewertung einer Inspektion einfließt, ist die für den Prozess benötigte Durchlaufzeit. Da ein einzelnes Inspektionsverfahren gleichzeitig bei einer Vielzahl von unterschiedlichen Flugzeugteilen wie Struktur-, Triebwerks- und Fahrwerkteilen Anwendung findet, würde sich eine Zeiteinsparung um ein Vielfaches multiplizieren und sich erheblich auf die für die gesamte Instandhaltung benötigte Zeit auswirken. Die Downtime eines Flugzeuges könnte verringert werden, was wiederum einen positiven Einfluss auf die Wettbewerbsfähigkeit des Instandhalters hätte. Das bedeutet zusammengefasst, dass durch Optimierung eines Inspektionsprozesses alle drei an der Instandhaltung beteiligten Parteien profitieren. Neben der Erhöhung der Wettbewerbsfähigkeit gegenüber Konkurrenten führt diese Optimierung außerdem zu einer Verbesserung der Position des Instandhalters gegenüber dem Hersteller, da sich durch den Rückfluss der Schadensdaten eine gegenseitige Abhängigkeit ergibt. Die Abhängigkeiten der einzelnen an der Instandhaltung beteiligten Parteien sowie deren zentrale Qualitätsmerkmale sind in Abbildung 1.1 veranschaulicht.

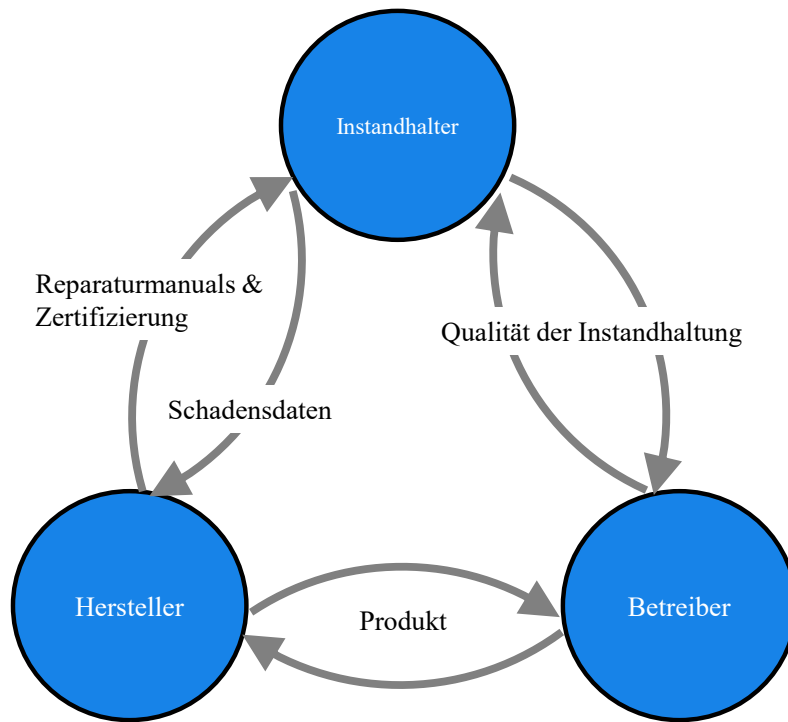


Abbildung 1.1.: Konstellation und Abhängigkeiten in der Luftfahrtinstandhaltung

### 1.1.3. Automatisierte Rissinspektion

Im Rahmen der Inspektion in der Instandhaltung wird auch eine Rissinspektion durchgeführt. Hierbei wird versucht, die für Flugzeugteile sehr gefährlichen Ermüdungsrisse zu detektieren. Diese treten vermehrt in Bauteilen auf, die über eine lange Zeit sich wiederholenden hohen Belastungen ausgesetzt sind. Start, Landung und Beschleunigung bei einem Flugzeug stellen exemplarisch solche Belastungen für zum Beispiel Fahr- und Triebwerke dar. Ein so entstehender Riss kann sich, wenn unentdeckt, im Material weiterentwickeln und schlussendlich zum Bruch des entsprechenden Bauteils führen. Wird durch solch eine Beschädigung zum Beispiel ein Investitionsgut wie ein Triebwerk irreparabel beschädigt, kann das den wirtschaftlichen Totalverlust des gesamten Flugzeugs bedeuten. Um Risse schon in einem frühen Stadium mit einer hohen POD detektieren zu können, müssen hochauflösende Verfahren eingesetzt werden. Ein gemeinsames Merkmal von hochauflösenden Inspektionsverfahren ist, aufgrund von endlichen Hardwareressourcen, die Eigenschaft eines begrenzten Messbereichs. Die Mehrzahl der Bauteile aus dem Flugzeugtriebwerk weisen ein Volumen auf, das mehrere Größenordnungen oberhalb des Volu-

mens eines Sensormessbereichs liegt. Eine Inspektion dieser Bauteile setzt sich deswegen aus sehr vielen Einzelaufnahmen zusammen. Wird die Zeit für Rüst- und Hilfsprozesse nicht mitberücksichtigt, besteht die Durchlaufzeit für eine Inspektion fast ausschließlich aus der Zeit für die eigentliche Aufnahme der Oberfläche und der Zeit für die Umpositionierung des Sensors zwischen diesen Aufnahmen. Eine Reduktion der Durchlaufzeit kann also erreicht werden, indem eine dieser beiden Komponenten reduziert wird.

Großes Potential zur Reduktion der Durchlaufzeit bietet deswegen die Automatisierung. Hierbei wird die zur Inspektion verwendete Sensorhardware und/oder das zu inspizierende Bauteil automatisch durch ein Handhabungsgerät geführt. Neben der Optimierung der Durchlaufzeit können dadurch außerdem die aus der Inspektion resultierenden Beschädigungsdaten um eine exakte Messposition ergänzt werden. Das ermöglicht wiederum, diese Daten auch bei nachfolgenden Inspektionen sicher einer bestimmten Beschädigung zuzuordnen.

Die entscheidenden Faktoren bei der Auslegung eines solchen automatischen Inspektionssystems sind die Messstrategie, die spezifizierten Kennwerte des Sensors und die geometrischen Abmaße des Bauteils. Außerdem muss bestimmt werden, wie sich die Handhabung und das Inspektionsverfahren gegenseitig beeinflussen, damit sichergestellt werden kann, dass die hohen Ansprüche an die Qualität weiterhin eingehalten werden können.

## 1.2. Motivation - Das Projekt AutoInspect

Der aktuelle Weltmarktführer in der Luftfahrt-MRO, Lufthansa Technik AG (LHT) setzt heute bei der Inspektion von unterschiedlichen Flugzeugbauteilen zur Detektion von Rissen das von den Bauteilherstellern zertifizierte und bei der Inspektion von Flugzeugteilen seit langer Zeit etablierte Verfahren der Flouruscent Penetrant Inspection (FPI) ein. Für dieses Verfahren ist ein vollständig manueller Prozess implementiert, bei dem ein zuvor von Rückständen und Beschichtungen gereinigtes Bauteil in eine unter Ultraviolettstrahlung (UV) fluoreszierende, niedrigviskose Chemikalie getaucht und anschließend wieder oberflächlich gereinigt wird. Diese Chemikalie tritt aus den Rissen hervor und wird unter dem Einsatz von Schwarzlicht sichtbar gemacht. Ein Experte entscheidet im Anschluss, welche Markierungen durch einen Riss entstanden sein könnten und kennzeichnet diese wiederum mit einem Stift, der auch unter normalem Weißlicht sichtbar ist. In einer weiteren Station werden die gekennzeichneten Markierungen dann manuell mit einer Lupe mit 10-fach Vergrößerung visuell befundet und es wird entschieden, welche Risse im Anschluss repariert werden.

Die FPI weist sogar für Rissbreiten in Dimensionen kleiner als  $1\text{ }\mu\text{m}$  eine hohe POD auf. Das Verfahren hat jedoch abseits der Detektionswahrscheinlichkeit eine Reihe von Einschränkungen. Bei dem etablierten manuellen Prozess durchläuft das Bauteil mehrere Stationen, die unterschiedlich ausgelastet sind und sich in verschiedenen Werkstätten und Hallen befinden. Die Durchlaufzeit der gesamten Inspektion wird hierbei also durch Wartezeiten in diesen Stationen unnötig verlängert. Außerdem ist das Inspektionsergebnis abhängig von der subjektiven Interpretation und der

aktuellen Verfassung des inspizierenden Experten und dadurch nur eingeschränkt reproduzierbar. Es liefert zudem keine Daten über den Defekt, sondern ausschließlich die Information über dessen Vorhandensein, was einem minimalen Informationsgehalt entspricht.

Vor diesem Hintergrund wurde durch die LHT in Kooperation mit weiteren Projektpartnern 2011 das vom Bundesministerium für Wirtschaft und Energie (BMWi) im Rahmen der Luftfahrtforschung geförderte Forschungsvorhaben *AutoInspect* gestartet. Dieses beschäftigt sich exemplarisch mit der Substitution der FPI bei rotationssymmetrischen Brennkammerbauteilen aus dem Triebwerk als ersten Schritt in einem vollständig automatisierten Reparaturprozess. Für die angestrebte Zertifizierung des erforschten Inspektionsprozesses durch den Hersteller des Bauteils müssen Risse mit einer POD vergleichbar oder besser zu der FPI detektiert werden können. Folgend sind alle von der LHT formulierten Anforderungen aufgelistet.

#### **Anforderungen der LHT an ein automatisches Inspektionssystem**

- Zerstörungsfreies Prüfverfahren
- Vergleichbare oder bessere POD wie bei der FPI
- Reduktion der Durchlaufzeit
- Erhöhung des Informationsgehaltes durch Geometrie-Informationen der Beschädigung
- Höhere Prozesssicherheit durch bessere Reproduzierbarkeit
- Reduktion umweltschädlicher Chemikalien
- Lebenszyklus-Verlängerung der Bauteile
- Optimierung der Inspektionsintervalle
- Reduktion der körperlichen Belastung

### **1.3. Randbedingungen bei der Rissinspektion**

Zusätzlich zu den vorgestellten Anforderungen ergeben sich aus den Randbedingungen im betrachteten Anwendungsfall weitere Herausforderungen. Diese müssen bei der Entwicklung eines automatischen Inspektionssystems berücksichtigt werden. Die Randbedingungen sind das zu inspizierende Brennkammerbauteil mit seinen einzigartigen Eigenschaften, das entsprechende Handhabungsgerät und die hochauflösende Sensortechnologie. Zusätzlich muss auch der Einfluss der Randbedingungen aufeinander überprüft werden. Die Identifikation des passenden Handhabungsgerätes und der Sensortechnologie war Inhalt einer LHT-internen Vorstudie zu dem Forschungsprojekt *AutoInspect*. In den folgenden Abschnitten werden die Randbedingungen näher vorgestellt und deren Herausforderungen herausgearbeitet.



### 1.3.1. Betrachtetes Bauteilspektrum

Das betrachtete Bauteil ist die äußere Brennkammerwand eines *CFM-56* Triebwerks der Firma *CFM-International*. Dieses Bauteil wird in verschiedenen Reparaturphasen inspiziert, die jeweils mit individuellen Modifikationen der Geometrie und der Oberfläche einhergehen. Insgesamt umfasst das Bauteilspektrum deswegen vier unterschiedliche Bauteilvarianten. Das Bauteil ist aus der Nickel-Cobalt Superlegierung Hastelloy X gefertigt, da diese aufgrund der besonderen Eigenschaften der unterschiedlichen Legierungsbestandteile bei der hohen thermischen Belastung, die durch die Kerosinverbrennung in der Brennkammer entsteht, nicht expandiert oder schmilzt (Hay08). Da dieses Material sehr hochwertig ist, stellt das Bauteil für die Betreiber der Flugzeuge ein Investitionsgut dar. Deshalb lohnt sich der Aufwand, das Bauteil aus dem Triebwerk zu demontieren und wieder instandzusetzen, anstatt es durch ein Neuteil zu ersetzen. Die Geometrie des Bauteils lässt sich durch einen dünnwandigen rotationssymmetrischen Hohlkegelstumpf beschreiben, der stufenweise in verschiedene Paneele unterteilt ist. Die Abbildung 1.2 zeigt ein dreidimensionales Oberflächenmodell dieses Bauteils. Das Bauteil wiegt circa 11 kg und hat einen maximalen Durchmesser von circa 800 mm und einen minimalen Durchmesser von 600 mm.

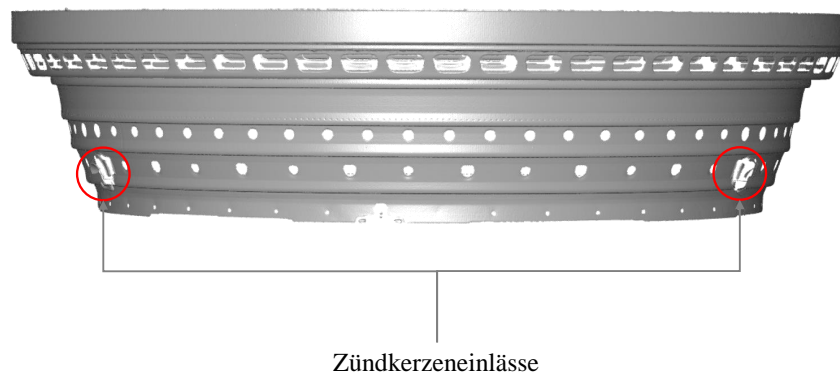


Abbildung 1.2.: 3D-Modell des betrachteten Brennkammerbauteils

Die Geometrie der Bauteile ist komplex und weist verdeckte Bereiche sowie Löcher für die Vermischung von Kerosin mit Luft und zur Kühlluft einleitung auf. Seine Rotationssymmetrie wird auf dem gesamten Umfang nur an zwei Stellen durch Zündkerzeneinlässe unterbrochen. Aufgrund der eben genannten hohen thermischen Belastungen verformen sich die Brennkammerwände im operativen Betrieb individuell lokal sowie global, es kommt zu Verbrennungen und eben auch zu den zu detektierenden Rissen. Kein Bauteil entspricht deshalb seinem idealen CAD-Modell. Während des Reparaturprozesses werden die Bauteile unterschiedlichen Bearbeitungsprozessen

wie Fräs-, Polier- sowie Plasmaaktivierungs- und Waterjet-Prozessen unterzogen, wodurch dessen Oberflächenreflektivität während des Instandhaltungsprozesses zwischen matt und hochglänzend variiert.

### Herausforderungen des Bauteilspektrums hinsichtlich automatisierter Inspektion

1. unterschiedliche Varianten
2. Gewicht von 11 kg
3. Abmessungen: Radius= 300 mm bis 400 mm, Höhe= 200 mm
4. komplexes Profil mit verdeckten Bereichen
5. unterbrochene Rotationssymmetrie
6. Verformungen sowie Beschädigungen

#### 1.3.2. Messprinzip

In der einleitend benannten Vorstudie wurden unterschiedliche zFP daraufhin untersucht, inwieweit sie den im Projekt formulierten Anforderungen entsprechen. Die Weißlichtinterferometrie (WLI) ist zerstörungs- und chemikalienfrei, bietet grundsätzlich eine sehr hohe Auflösung, den geforderten Informationsgehalt und eignet sich aufgrund ihres niedrigen Gewichtes und der geringen Abmaße für die Integration in ein automatisches System. Deswegen wurde es im Michelson-Aufbau für die Inspektionsaufgabe als Sensortechnik ausgewählt. Das Verfahren der WLI basiert auf der kurzen Kohärenzlänge von Licht mit einem breitem Frequenzspektrum. Diese berechnet sich mit der mittleren Wellenlänge  $\lambda$  und einer Spektrumbreite von  $\Delta\lambda$  nach Formel 1.1 (GG06).

$$l_C = \frac{\lambda^2}{\Delta\lambda} \quad (1.1)$$

Das benötigte kohärente Licht wird bei einem Michelson-Interferometer von einer entsprechenden Lichtquelle auf einen Strahlenteiler geworfen, wo es in einen Referenzstrahl und einen Probenstrahl aufgeteilt wird. Der Referenzstrahl trifft nach einer bekannten Weglänge auf einen Spiegel und wird von diesem zurück zum Strahlenteiler reflektiert. Der Probenstrahl trifft auf eine Probe und wird teilweise von dieser zurück auf den Strahlenteiler reflektiert, wo es dann zur Überlagerung der beiden Strahlen kommt. Aufgrund des breiten Wellenlängenspektrums von Weißlicht treten nur dann konstruktive Interferenzen auf, wenn der Gangunterschied zwischen den Strahlen kleiner ist als die halbe Kohärenzlänge des Lichtes (BWC13). An einem Detektor kann in diesem Fall ein Intensitätsanstieg wie in 1.3b gemessen werden. Durch verschiedene Algorithmen wie der Hilbert- oder Fouriertransformation wird in diesem Profil das Maximum bestimmt, dem dann wiederum eine Entfernung zugeordnet werden kann. Die Zuverlässigkeit der Detektion ist dabei abhängig davon, wieviel Intensität von dem Objektstrahl reflektiert wird. Abbildung 1.3a

veranschaulicht das beschriebene Verfahren

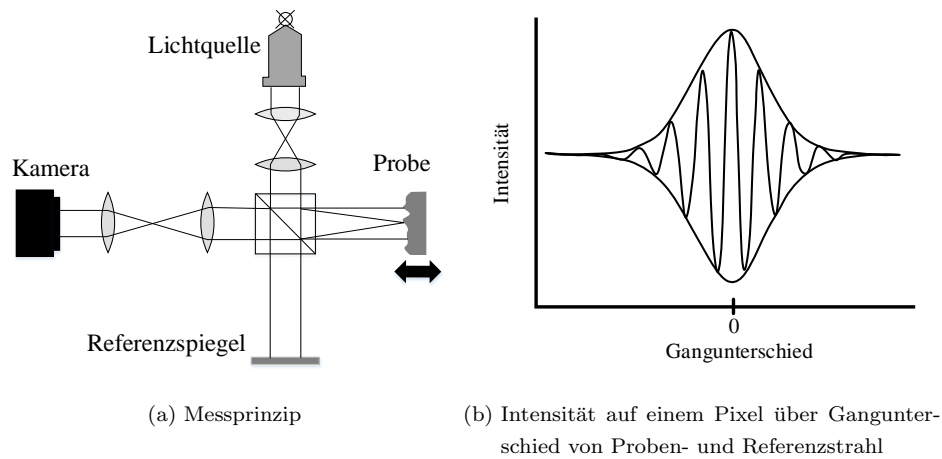


Abbildung 1.3.: Messprinzip (a) und Intensitätsverteilung (b) bei der WLI (GG06)

Wird als Detektor ein Kamerachip eingesetzt, kann bei einer Relativbewegung zwischen der Probe und dem Sensor in Strahlrichtung aus den Intensitätsmaxima der einzelnen Pixel eine dreidimensionale Topologie der Probenoberfläche im Messbereich rekonstruiert werden (siehe Abbildung 1.4). So lässt sich damit, anders als bei der FPI, neben des Vorhandenseins eines Risses auch dessen Länge, die Breite und bei klaffenden Rissen auch dessen Tiefe bestimmen.

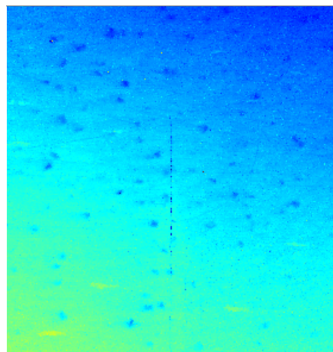


Abbildung 1.4.: farbcodierte Höhentopologie als Ergebnis einer WLI-Messung

Die Auflösung des Verfahrens in Strahlrichtung liegt theoretisch im Submikrometerbereich. Praktisch ist sie jedoch abhängig von der Menge an reflektiertem Licht und der damit verbundenen Intensität sowie von der Genauigkeit, mit der die aktuelle Position des Interferenzmaximums festgestellt werden kann. Bei optisch rauen Oberflächen kann eine Auflösung in Strahlrichtung von circa  $1\text{ }\mu\text{m}$  erreicht werden. Die Auflösung senkrecht zur Strahlrichtung (lateral) ist aufgrund der

Abbe'schen Auflösungsbedingung auf wenige Mikrometer begrenzt (GG06). Die Bedingung besagt, dass das Auflösungsvermögen  $x_{min}$  abhängig ist von der numerischen Apertur  $NA$  und der zur Betrachtung verwendeten Wellenlänge  $\lambda$  und sich über Formel 1.2 berechnen lässt (Dem06).

$$x_{min} \geq \frac{\lambda}{NA} \quad (1.2)$$

Die hohe Auflösung bewirkt allerdings aufgrund endlicher Hardwareressourcen eine Reduktion des Messbereichs auf wenige Quadratmillimeter. Der Messabstand ist bei dieser Technologie prinzipbedingt abhängig von der Baugröße des Sensors und der damit verbundenen Länge der optischen Strecke bis zum Referenzspiegel. Bei dem eingesetzten Sensor beträgt dieser aktuell circa 50 mm. Um die Oberfläche des betrachteten Bauteilspektrums vollständig inspizieren zu können, sind daher bei den geometrischen Dimensionen eines Brennkammerbauteils abhängig von der gewählten Sensorauflösung bis zu 60 000 Einzelaufnahmen nötig. Die notwendige Relativbewegung zwischen WLI und Bauteil wird bei dem betrachteten Aufbau von einer hochpräzisen direktangetriebenen Linearachse realisiert. Diese Achse liefert dem Weißlichtinterferometer die für die Zuordnung der Interferenzmaxima benötigte hochgenaue Position von einem Linearencoder mit Glasmaßstab. Wird zwischen den auflösbaren Skalenteilen keine Interpolation vorgenommen, beschränkt die Auflösung dieser Achse praktisch die laterale Auflösung des gesamten Weißlichtinterferometers. Während dieser Bewegung muss außerdem vermieden werden, dass Schwingungen und Stöße in den Sensor einkoppeln, um sicherzustellen, dass Interferenzmaxima gar nicht oder an einer falschen Stelle detektiert werden. Dies würde wiederum die Relativposition verfälschen und die aktuelle Aufnahme als Resultat unbrauchbar machen.

### **Herausforderungen der Sensortechnik hinsichtlich automatisierter Inspektion**

1. kleines Messvolumen
2. kleiner Messabstand
3. Aufnahmequalität ist abhängig von dem Betrachtungswinkel
4. Schwingungsanfälligkeit
5. diskontinuierliches Messverfahren
6. Gewicht von 2 kg

#### **1.3.3. Handhabung**

Durch den Einsatz eines automatischen Handhabungsgerätes bei der Inspektion wird die Reproduzierbarkeit erhöht und gleichzeitig die körperliche Belastung für den Menschen reduziert. Das selektierte Handhabungsgerät muss den entsprechenden WLI-Sensor in einem Arbeitsraum flexibel positionieren und orientieren können, dessen geometrische Abmessungen mindestens de-

nen des Bauteils entsprechen. Es muss außerdem das Gewicht des Sensors zusammen mit der prinzipbedingt vorhandenen Linearachse und der notwendigen Peripherie handhaben können. In Frage kommen hierbei Industrieroboter (IR) in der klassischen Knickarm-Konfiguration, Werkzeugmaschinen sowie Sonderkinematiken. Ein IR ist preiswert, weist einen großen Arbeitsraum und gleichzeitig hohe Handhabungsgewichte auf. Deswegen wurde er einer Werkzeugmaschine und einer Sonderkinematik vorgezogen. Die im Vergleich zu einer Werkzeugmaschine schlechtere Positioniergenauigkeit ist für den betrachteten Inspektionsprozess irrelevant, weil die Daten im Bereich der um einige Größenordnungen besseren Wiederholgenauigkeit aufgenommen werden sollen. Damit die Inspektion vollautomatisch durchgeführt werden kann, muss auch die Pfadplanung für den IR automatisch ablaufen. Die besonderen Anforderungen, die aus dem Bauteil, der Sensortechnologie und den Wechselwirkungen der beiden Komponenten untereinander resultieren, müssen bei dieser Pfadplanung unbedingt berücksichtigt werden. Dabei stellt vor allem der kleine Messbereich in Kombination mit der individuellen Verformung der beschädigten Bauteile eine besondere Herausforderung dar. Die WLI ist prinzipbedingt sensibel gegenüber Schwingungen und Vibrationen, weswegen der zu selektierende IR den Sensor nicht zum Schwingen anregen darf.

#### **Herausforderungen des IR hinsichtlich automatisierter Inspektion**

1. schwingungsanfällige Kinematik
2. diskontinuierliche Aufnahmestrategie

### **1.4. Zielsetzung der Arbeit**

Das übergeordnete Ziel dieser Arbeit ist die *Automatisierung der Weißlichtinterferometrie zur Inspektion rotationssymmetrischer Triebwerksbauteile*. Für die Erfüllung dieser übergeordneten Zielsetzung müssen als untergeordnete Ziele ein automatisiertes System zur Handhabung eines WLI-Sensors sowie ein vollautomatischer WLI-Inspektionsprozess konzipiert und zusätzlich in Software implementiert werden. Aus dem dieser Arbeit zugrunde liegenden Forschungsvorhaben ergeben sich dabei zusätzlich globale Anforderungen an die Inspektion, was die Vollständigkeit und die benötigte Durchlaufzeit betrifft. Neben diesen globalen Anforderungen müssen bei der Konzipierung des Inspektionssystems zusätzlich die vorher genannten und in Abbildung 1.5 in den entsprechenden Kategorien zusammengefassten Herausforderungen berücksichtigt werden. Diese haben ihren Ursprung in der speziellen Aufgabenstellung in der Flugzeuginstandhaltung, dem damit verbundenen speziellen Bauteilspektrum, der WLI als Technologie für die Inspektion sowie in der Handhabungstechnik. Hierbei ist vor allem zu berücksichtigen, dass die hochbelasteten Bauteile verformt sind, weswegen der entsprechende Inspektionsprozess im Allgemeinen sowie die Pfadplanung für die Handhabung im Speziellen adaptiv entworfen werden müssen. Neben der eigentlichen Rissinspektion und der dafür benötigten Pfadplanung setzt sich der gesamte Prozess außerdem aus Hilfsprozessen aus dem Bereich der Kalibrierung und Oberflächenerfassung

zusammen. Für die Implementierung der unterschiedlichen Teilprozesse sowie für die Ablaufsteuerung des Gesamtprozesses wird außerdem eine Steuerungssoftware benötigt, die sowohl in der Lage sein muss mit der zum Einsatz kommenden Sensorik sowie Handhabungstechnik über unterschiedliche generische Schnittstellen zu kommunizieren als auch komplexe Algorithmen auf den unterschiedlichen Sensordaten auszuführen. Die übergeordnete Zielsetzung, die dafür zu erreichenden untergeordneten Ziele und deren jeweiligen Charakter veranschaulicht die folgende Abbildung.

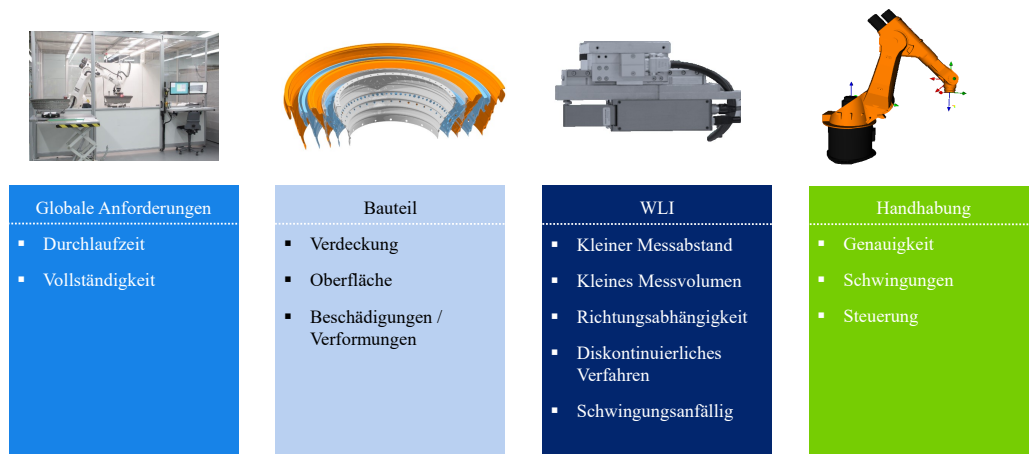


Abbildung 1.5.: globale Anforderungen sowie aus den Randbedingungen resultierende Herausforderungen

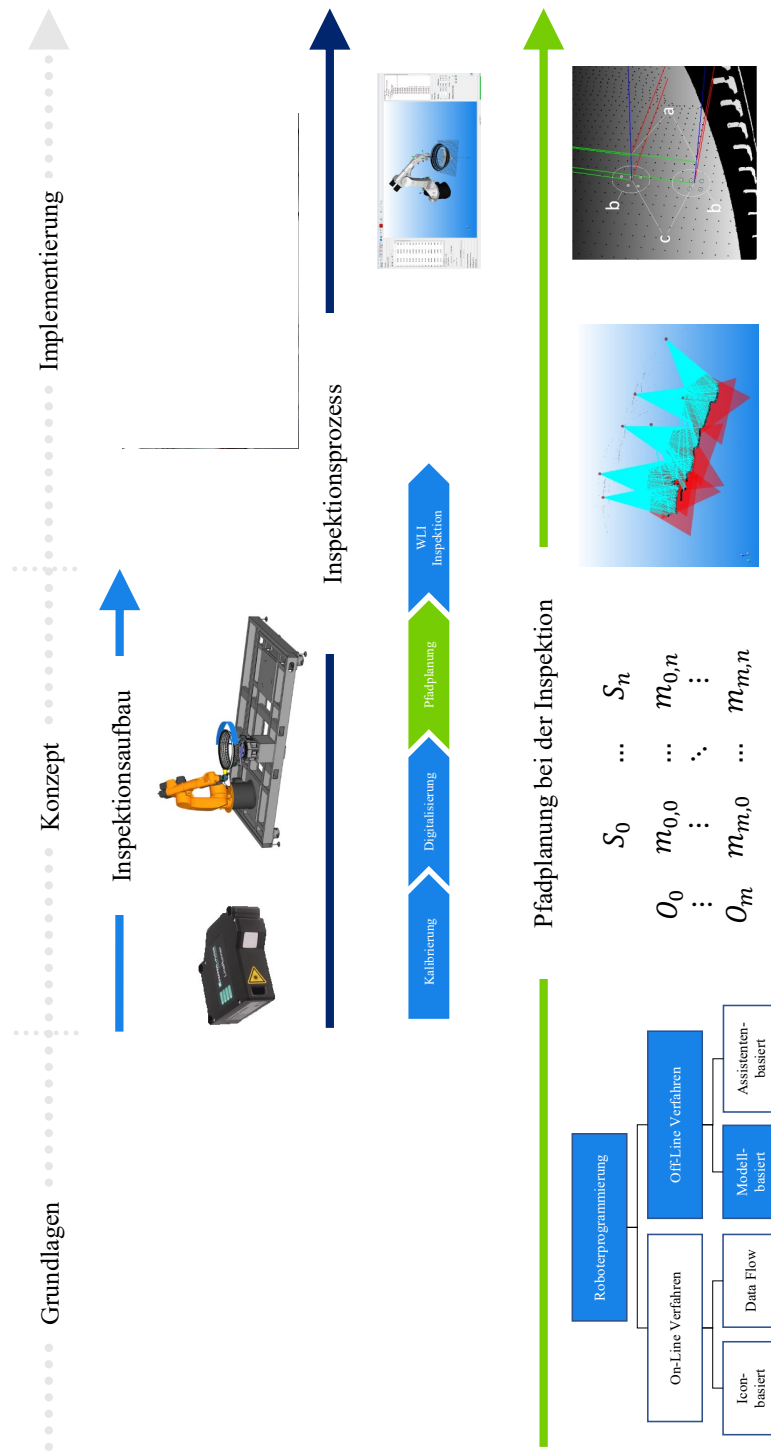


Abbildung 1.6.: Aufbau der übergeordneten Zielstellung





## 2. Stand von Wissenschaft und Technik

In diesem Kapitel wird der aktuelle Stand von Wissenschaft und Technik bezüglich der in der Zielsetzung formulierten Kernthemen dieser Arbeit vorgestellt. Begonnen wird dabei mit aktuellen Applikationen der WLI in Industrie und Wissenschaft. Da sowohl auf der Forschungsseite als auch auf der Industrieseite noch keine automatische Applikation der WLI veröffentlicht wurde, beschränkt sich die Betrachtung auf Weißlichtinterferometer, die für Inspektionsaufgaben spezialisiert sind. Des Weiteren werden aktuelle Ansätze der Pfadplanung mit einem IR im Allgemeinen und für Inspektionsaufgaben im Speziellen vorgestellt. Anschließend folgt eine Betrachtung des Standes der Technik und Wissenschaft im Bereich softwarebasierter Programmier-, und Steuerungssysteme für robotergestützte Inspektionsanlagen. Abschließend werden Defizite der vorgestellten Themen bezüglich der übergeordneten Zielsetzung präsentiert. Aus diesen Defiziten werden im Anschluss die neuartigen eigenen Beiträge dieser Arbeit abgeleitet. Zuletzt wird die Struktur der weiteren Kapitel dieser Arbeit vorgestellt.

### 2.1. Weißlichtinterferometrie für die Inspektion

Das optische Verfahren der Interferometrie wird schon seit über hundert Jahren im Laborumfeld eingesetzt und basiert auf den Arbeiten von Mireau und Michelson. Typische Anwendungsgebiete sind die Vermessung von Länge, Längenabweichung und Vibration. Übliche Applikation sind die Bestimmung von Gas- und Plasmaströmungen, von Oberflächentopologien sowie von Temperatur, Druck und elektrischen sowie magnetischen Feldern (Har07). Die WLI als spezielle Unterform der Interferometrie ist aufgrund der sehr kurzen Kohärenzlänge des eingesetzten Lichtspektrums ideal geeignet, um Oberflächentopologien zu vermessen (siehe Abbildung 1.3.2). Aufgrund des speziellen Aufnahmeverfahrens fällt für jede Einzelaufnahme eine große Datenmenge an. Auf dieser großen Datenmenge müssen in der Vorverarbeitung Algorithmen zur Verbesserung der Datenqualität angewendet und im Anschluss das Intensitätsmaximum bestimmt werden. Der hohe Rechenaufwand in Kombination mit einer Sensibilität des Verfahrens gegenüber Vibrationen und Schwingungen erschwert die Integration des Verfahrens in einen Prozess im industriellen Umfeld. In der Literatur findet sich deswegen trotz der hohen Rechenleistung heute noch immer keine industrielle Applikation für dieses Verfahren. Stattdessen arbeiten verschiedene Gruppen in Industrie und Wissenschaft noch immer an der Verbesserung und Effizienzsteigerung der eigentlichen Sensortechnik (JD11). Im Folgenden werden aus diesem Grund anstatt industrieller Applikationen der WLI diese unterschiedlichen Ansätze und aktuelle Sensoren präsentiert.

Ein Beispielsensor aus der Industrie ist das Weißlichtinterferometer der Firma 3D-Shape. Laut Datenblatt weist ihr schnellster Sensor eine vertikale Messgeschwindigkeit von  $90.8 \mu\text{m/s}$  auf (Gmb16). Dies würde schon bei einer benötigten Messtiefe von zwei Millimetern bedeuten, dass eine vollständige Aufnahme abgerundet 22 s in Anspruch nehmen würde. Multipliziert man diesen Wert mit der hohen Anzahl an Aufnahmen, die für eine Inspektion eines gesamten Bauteils

des betrachten Bauteilspektrums nötig wären, erhält man eine Durchlaufzeit, die laut den in dieser Arbeit skizzierten Anforderungen deutlich zu hoch wäre. Eine wirtschaftliche Umsetzung eines Inspektionssystems für vergleichsweise große Bauteile wäre mit diesem Sensor also nicht möglich.

Die Weißlichtinterferometrie ist gleichzeitig noch immer Thema aktueller Forschungsarbeiten. Im Rahmen des von der Deutsche Forschungsgemeinschaft (DFG) geförderten und in Hannover angesiedelten Sonderforschungsbereiches 871 „Regeneration komplexer Investitionsgüter“ wird an der WLI in einem Teilprojekt als Teil eines multiskaligen Sensorsystems zur Inspektion geforscht. Die betrachteten Bauteile stammen ebenfalls aus dem Flugzeugtriebwerk. Ziel dieser Forschung ist die Entwicklung eines eigenen neuen Sensors zur Weißlichtinterferometrie. In unterschiedlichen Veröffentlichungen berichtet das Projektteam über aktuelle Arbeiten. Diese umfassen zum einen die Beschleunigung der Messgeschwindigkeit durch den Einsatz der Graphical Computing Unit (GPU) und der Anwendung der Compute Unified Device Architecture (CUDA) und zum anderen die Integration eines hochgenauen Hexapoden zur Positionierung des Sensors (LKR15). Durch den Einsatz eines solchen Hexapoden kann ein Probenteil auf eine Weise positioniert und orientiert werden, dass auch Bereiche vermessen werden können, die andernfalls einen zu steilen Messwinkel aufweisen würden. Auch Schneider et al. beschreiben in einer Veröffentlichung einen ähnlichen Ansatz zur Beschleunigung der Bestimmung der Intensitätsmaxima (SFW14).

Aus Gründen der Geheimhaltung wird der Hersteller sowie der Typ des in dieser Arbeit verwendeten WLI-Sensors nicht explizit genannt. Stattdessen wird nur auf dessen Kennwerte und Besonderheiten eingegangen. Der spezielle WLI-Sensor unterscheidet sich vor allem bei der Datenauswertung erheblich von den vorher vorgestellten. Die Bestimmung der Interferenzmaxima wird bei diesem Ansatz direkt in jedem Pixel des Sensors durchgeführt. Dies bedeutet, dass nicht jede Aufnahmescheibe zu einem externen Rechner zur Datenauswertung gesendet werden muss, sondern stattdessen nur die auf jedem einzelnen Pixel des Chips identifizierten Maxima selbst. Dieses Vorgehen verringert die zu übertragende Datenmenge erheblich, wodurch die gesamte Aufnahme beschleunigt wird. Dieses Vorgehen ermöglicht außerdem für jeden Pixel eine individuelle Belichtungsgrenze festzulegen, sodass dieser Sensor auch mit unterschiedlichen Reflektivitäten innerhalb einer Aufnahme umgehen kann. Andere Sensoren können diese Grenze erst über die gesamte Aufnahme global bei der Datenauswertung ermitteln. An dieser Stelle kann allerdings die Belichtungszeit nicht mehr angepasst werden und es kommt zu über- oder unterbelichteten Pixeln. Als weiterer Vorteil kommt der überzeugende Formfaktor des betrachteten Sensors hinzu. Der komplette Sensor wiegt zusammen mit der prinzipbedingt notwendigen Linearachse unter 2 kg und seine längste Seite in Strahlrichtung ist unter 200 mm lang, was eine einfache Handhabung und eine gute Zugänglichkeit des Sensors in einem automatischen Prozess ermöglicht.

## 2.2. Pfadplanung für die automatische Inspektion

Pfadplanung bei Robotern beschreibt ganz allgemein das Verbinden einer Anfangskonfiguration mit einer Zielkonfiguration unter Betrachtung von individuellen Zielsetzungen (GC13). Klassische Zielsetzungen aus der Literatur sind zum Beispiel eine optimale Zeit, ein minimaler Energieeinsatz oder die Vermeidung von Kollisionen (TLJ02). Oft wird auch eine Kombination mehrerer unterschiedlicher Ziele gleichzeitig verfolgt. Die Pfadplanung bei einer IR-gestützten Rissinspektion, wie sie in dieser Arbeit entwickelt wird, stellt allerdings spezielle Anforderungen. Die zentrale Anforderung ist eine vollständige Messabdeckung. Konkret muss garantiert werden, dass die gesamte Oberfläche inspiziert wird, sodass der mitgeführte Sensor im Rahmen seiner möglichen Sensorauflösung befähigt wird, alle Risse in der Bauteiloberfläche zu finden. Außerdem muss gleichzeitig garantiert werden, dass die Einzelaufnahmen eine vorgegebene Messüberdeckung aufweisen, damit sie zu einem Gesamtmodell registriert werden können. Der Begriff „Pfadplanung“ wird vor allem bei mobilen Robotern benutzt. Bei einem IR wird für denselben Vorgang der Begriff „Roboterprogrammierung“ verwendet.

Gegenwärtig stehen einem Roboterprogrammierer eine Vielzahl von unterschiedlichen Programmiermethoden zur Verfügung. In der Literatur existieren aktuell mehrere Ansätze zur Klassifizierung dieser Methoden, so unterteilt beispielsweise Weck sie in On-Line-Methoden und Off-Line-Methoden (Wec05). On-Line bedeutet in diesem Zusammenhang, dass während der Programmierung der physische Roboter bewegt wird und aktuelle Positionen seines Endeffektors in einem Roboterpfad abgespeichert werden. Die Off-Line-Programmierung findet hingegen ausschließlich mit einer virtuellen Repräsentation auf einem separaten Rechner statt. Die resultierenden Roboterpfade werden im Anschluss an die Programmierung von diesem Rechner auf eine entsprechende Robotersteuerung übertragen und können danach durch den Roboter abgefahren werden. Innerhalb dieser beiden skizzierten Klassen können diese Verfahren wiederum nach ihrem Automatisierungsgrad sortiert werden (GB03). Je höher der Automatisierungsgrad ist, desto kleiner wird der manuelle Aufwand für die Programmierung und desto weniger Erfahrung wird von dem entsprechenden Programmierer benötigt.

In den nächsten Abschnitten wird der Stand der Wissenschaft und Technik der einzelnen Programmiermethoden vorgestellt, diskutiert und herausgearbeitet, ob sie für eine automatische IR-gestützte Rissinspektion geeignet sind und an welchen Stellen sie gegebenenfalls Defizite aufweisen.

### 2.2.1. On-Line-Pfadplanung von Industrierobotern

Die klassische On-Line-Programmiermethode für Industrieroboter ist das Programmieren mit einem Teach-Pendant auch lead-through-teaching genannt. Bei dieser Methode verfährt ein Bediener den zu programmierenden Roboter manuell auf eine Position und speichert diese dann als neuen Pfadpunkt ab. Die einzelnen Pfadpunkte werden dann mit Bewegungsbefehlen verknüpft, so dass eine Bewegungstrajektorie für den Roboter entsteht. Diese textbasierte Methode ist nur

für einfache Roboterpfade mit wenigen Pfadpunkten geeignet, weil sie bei komplexeren Programmen sehr zeitaufwendig ist und umfangreiche Erfahrung seitens des Programmierers sowohl bei der Erstellung als auch später bei der Fehlerbehebung erfordert (RMH13). Dadurch entfällt ein nicht unerheblicher Teil der Kosten für die Inbetriebnahme eines Industrieroboters auf seine Programmierung (PPL12). Zudem sind nachträgliche Modifikationen an einem Roboterpfad sehr aufwendig und der Roboter kann während der Programmierzeit nicht operativ genutzt werden. Trotzdem sind diese Verfahren bis heute in der Industrie weit verbreitet, da es noch immer ausreichend einfache Prozesse gibt, bei denen die einmal programmierten Roboterpfade über eine lange Zeit eingesetzt werden und somit der Einfluss für den hohen Aufwand für den Programmierungsvorgang auf die gesamten Betriebskosten eines Roboters gering bleibt (LR06, Dil06). On-Line Verfahren sind generell für die Planung von Roboterpfaden für die Inspektion mit optischen Sensoren schlecht geeignet, da der vollständige Messbereich für den Programmierer nicht sichtbar ist. So ist es nur unter großem Aufwand möglich, den Roboter manuell so zu positionieren, dass eine geforderte Überlappung zweier Einzelaufnahmen und damit eine vollständige Messabdeckung garantiert werden können (CW08). Außerdem werden bei den betrachteten Bauteilen mehrere zehntausend Einzelaufnahmen benötigt. Die Programmierung dieser Menge an Aufnahmen würde schon allein für sich die geforderte Durchlaufzeit für den Gesamtprozess überschreiten. Deswegen werden diese Verfahren im Weiteren nicht mehr betrachtet.

### 2.2.2. Off-Line-Pfadplanung von Industrierobotern

Bei den Off-Line-Methoden zur Programmierung von Industrierobotern stehen dem Programmierer heutzutage eine Vielzahl von unterschiedlichen Ansätzen und deren Kombinationsmöglichkeiten zur Auswahl. Untergliedert werden diese Verfahren in die vier folgenden Kategorien (RMH13).

#### Unterschiedliche Off-Line-Programmierverfahren

1. Icon-basierte Programmierung, bei der Roboterprogramme über Icons definiert werden
2. Data Flow Diagramme, bei denen der Ablauf eines Roboterprogramms über funktionale Blöcke konfiguriert wird
3. CAD-basierte Programmierung, wo CAD-Daten zur semiautomatischen oder vollautomatischen Generierung eines Roboterpfades genutzt werden
4. Assistenten-basierte Programmierung, bei der Programmierer ähnlich wie bei einer Softwareinstallation durch ein Assistenten-Programm durch die Roboterprogrammierung geführt wird

Alle vier aufgezählten Verfahren werden auf einem externen Rechner ausgeführt. Es vereint sie also der Vorteil, dass der entsprechend zu programmierende IR während des eigentlichen Programmierprozesses nicht benötigt wird und weiterhin produktiv eingesetzt werden kann. Da in

dieser Arbeit aufgrund der Aufgabenstellung modellbasierte Verfahren betrachtet werden, wird im Folgenden nur auf aktuelle Applikationen aus der CAD-basierten Programmierung eingegangen. Unter CAD-Daten zählen verschiedene Autoren entweder Volumen-Modelle, Oberflächen-Modelle oder Punktmodelle. Diese Verfahren werden deswegen den modellbasierten Verfahren zugerechnet.

Bei modellbasierten Verfahren plant der Programmierer bzw. eine Software je nach Grad der Automatisierung einen Roboterpfad auf einer virtuellen Repräsentation des betrachteten Bauteils. Abhängig von der Komplexität des Programmiersystems werden dabei weitere Objekte wie der Roboter mit seinem Kinematikmodell, Werkzeuge, zusätzliche Achsen oder andere relevante Zellkomponenten berücksichtigt. Diese virtuellen Objekte müssen mit dem realen Objekt, das sie repräsentieren, abgeglichen bzw. kalibriert sein, damit der Programmierer in der virtuellen Umgebung entsprechend Parameter einstellen kann, um Zugänglichkeit, Erreichbarkeit und Kollisionen des Roboters zu prüfen und in manchen Systemen sogar den Prozess in der Simulation testen zu können. Im Anschluss an die Planung wird der Pfad dann auf die Steuerung des eigentlichen Roboters übertragen. Die Pfadplanung kann zwar parallel zum aktiven Betrieb durchgeführt werden, der Programmierer muss jedoch trotzdem Erfahrung in der Roboterprogrammierung aufweisen, da er auch in einer Off-Line-Programmierung mit Roboterposen, Trajektorien und unterschiedlichen Koordinatensystemen umgehen können muss. Abbildung 2.1 zeigt den typischen Ablauf einer modellbasierten Off-Line-Programmierung (PPL12).

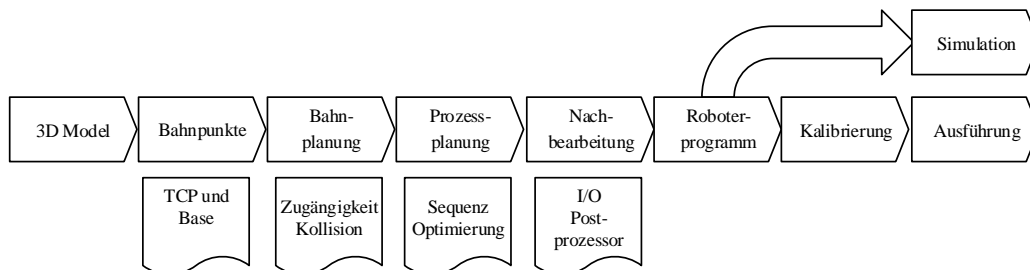


Abbildung 2.1.: typischer Ablauf einer CAD-basierten Off-Line-Programmierung (PPL12)

In der Kategorie der modellbasierten Off-Line-Programmierverfahren kann der Programmierer aus einer Vielzahl von unterschiedlichen proprietären sowie generischen Softwarelösungen auswählen, die ihn bei der Pfadplanung unterstützen. Der überwiegende Teil der proprietären Programme stammen von den verschiedenen Herstellern der Industrieroboter und werden in (PPL12) übersichtlich beschrieben. Damit jedoch möglichst flexibel unterschiedliche Roboter von verschiedenen Herstellern programmiert werden können, wurden parallel zu den proprietären generische Lösungen entwickelt. Die umfangreichsten kommerziell erwerbbaaren Softwarelösungen sind aktuell *Delmia* von der Firma *Dessault Systems* und *Robcad* der Firma *Siemens*. Das Problem dieser generischen Lösungen ist allerdings ihr großer Funktionsumfang und die daraus resultierende

Komplexität. Von den Funktionen wird ein großer Teil für die Programmierung eines Roboterpfades für eine spezifische Problemstellung nicht benötigt. Außerdem entstehen aus der hohen Spezifität des hier betrachteten Prozesses der Inspektion zusätzlich weitere anspruchsvolle Anforderungen bei der Prozessplanung, die generische Lösungen im Detail wiederum nicht leisten können. Deswegen wurden im Bereich der Wissenschaft weitere Off-Line-Lösungen entwickelt, die auf die Pfadplanung für einen bestimmten Prozess wie das Lackieren, Beschichten, Bohren oder Schweißen spezialisiert sind und die prozessspezifischen Parameter mit in die Prozessplanung einbeziehen (ZQC13, Kim04, PHS08). Ein Vorteil dieser spezialisierten Lösungen ist, dass sie aufgrund des vorhandenen Prozesswissens einen hohen Automatisierungsgrad bei der Programmierung erreichen können und dem Programmierer viele bis alle der in Abbildung 2.1 gezeigten Prozessschritte abnehmen können je nachdem, ob es sich um ein semi- oder vollautomatisches System handelt. Trotzdem ist dabei, wie schon vorher erwähnt, zu beachten, dass ein guter Kompromiss zwischen Automatisierungsgrad und Flexibilität gefunden wird. Auch für die Planung einer Inspektion mit ihren anspruchsvollen Anforderungen haben sich aus den genannten Gründen eigene Planungslösungen entwickelt, die im nächsten Abschnitt genauer betrachtet werden.

### 2.2.3. Modellbasierte Pfadplanung für die Inspektion

Aus Gründen der Übersichtlichkeit veranschaulicht Abbildung 2.2 noch einmal, wo die modellbasierten Verfahren zur Inspektionsplanung innerhalb der Roboterprogrammierung einzuordnen sind.

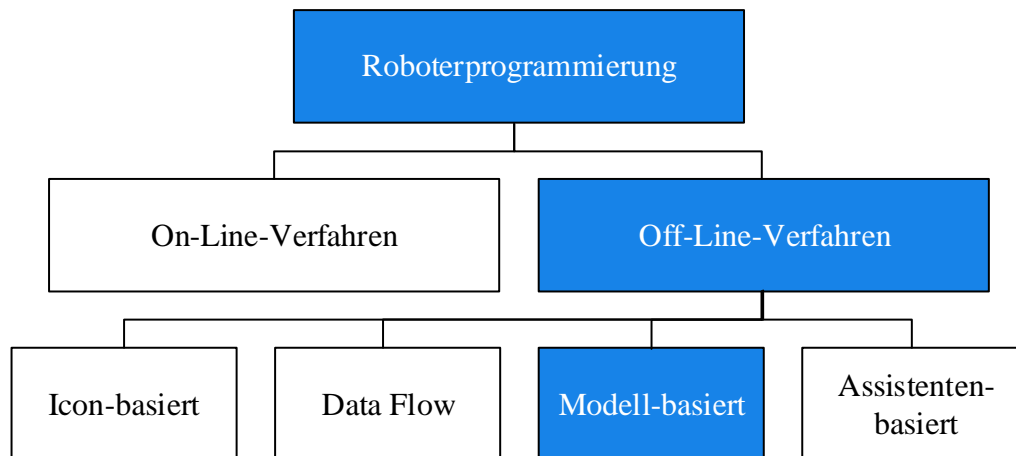


Abbildung 2.2.: Einordnung der modellbasierten Inspektionsplanung in die Roboterprogrammierung

Vollautomatische modellbasierte Pfadplanung für die Inspektion ist zurückzuführen auf das allgemeine Ansichtenplanungsproblem (View Planning Problem (VPP)). Das Problem wird nach Scott (SRR03) wie folgt definiert:

Finde innerhalb akzeptabler Berechnungszeit einen Ansichtenplan  $N$  angemessener Länge, welcher für einen gegebenen Messaufbau und ein gegebenes Messobjekt die spezifizierten Rekonstruktionsziele erfüllt.

Das Problem der Sensorplanung, auf dem das Ansichtenplanungsproblem basiert, ist seit über drei Dekaden ein Thema in der Forschung. Frühe Übersichtsarbeiten aus dem Jahr 1995 dazu stammen von Trabanis (TAT95), Newman & Jain (NJ95) sowie Tarbox & Gottschlich (TG95), die wiederum alle auf die Arbeit von Cowan aus dem Jahr 1988 zurückzuführen sind (CK88). Basierend auf den Arbeiten von Scott et al. (SRR03) und Li & Gu (LG04) lassen sich die Verfahren zur Lösung des Ansichtenplanungsproblems in On-Line- und Off-Line-Verfahren unterteilen, je nachdem, ob bei der Planung a priori Information über das zu inspizierende Bauteil vorliegt. Diese a priori Information kann ein Modell sein, was in einem mehrstufigen Sensorplanungsansatzes vor der eigentlichen Inspektion mit einer anderen Sensortechnologie aufgenommen worden ist (SRR01) oder ein CAD-Modell des Bauteils aus der Konstruktion. Da bei der aktuellen Arbeit die Digitalisierung der Bauteiloberfläche aufgrund der individuellen Beschädigungen eine Voraussetzung darstellt, wird im Folgenden nur der Stand der Technik der Off-Line-Verfahren vorgestellt. Zur Vollständigkeit wird bei den On-Line-Verfahren auf die umfassenden Arbeiten von Impoco (ICS05) Mehdi & Souzani (MSTL06) und Larsson & Kjellander (LK08) verwiesen. Die Off-Line Verfahren lassen sich laut Scott in drei verschiedene Ansätze kategorisieren (SRR03), die Abbildung 2.3 zusammen mit ihrem zentralem Element veranschaulicht:

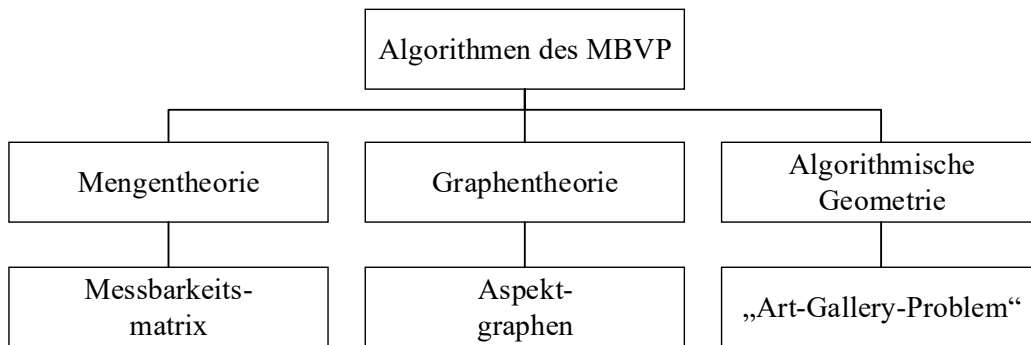


Abbildung 2.3.: Algorithmen des MBVP

Die Verfahren der algorithmischen Geometrie mit dem „Art-Gallery-Problem“ als Beispiel werden schon für sehr kleine Modelle sehr rechenintensiv. Scott beschreibt dieses Vorgehen als unpassend

für dreidimensionale Probleme, weswegen im Folgenden nur die graphen- sowie mengentheoretischen Verfahren genauer betrachtet werden. Diese Verfahren haben sich in den letzten Jahren in der Literatur als die zentralen Ansätze herausgestellt. Bei den algorithmischen Verfahren wird auf die Arbeit von Prieto verwiesen (PLB03).

### 2.2.3.1. Graphentheoretische Ansätze

Die graphentheoretischen Algorithmen bauen auf sogenannten Aspekt-Graphen auf. Bei diesen Graphen beschreiben die Knoten alle Aspekte des betrachteten Bauteils. Benachbarte Knoten sind durch Kanten miteinander verbunden (BD90, TG95). Ein Aspekt ist definiert als die Menge vom Sensorpositionsraum, von dem qualitativ der gleiche unverdeckte Bereich eines Objektes betrachtet werden kann. Dafür wird das Modell des zu inspizierenden Bauteils in einem Präprozessorschnitt zu einem Polyhedron diskretisiert. „Qualitativ gleich“ meint allerdings nur, dass ein Polygon von allen Positionen dieses Unterraums sichtbar ist, was nach (TUWR97) ein Problem dieses Ansatzes darstellt, da bei einer Inspektion parallel weitere Kriterien einbezogen werden müssen. Ein weiteres Problem der Aspektgraphen ist ihre Komplexität und der damit verbundene Rechenaufwand bei schon durchschnittlich komplexen Objekten (SRR03). In weiteren Veröffentlichungen werden die Aspektgraphen vereinfacht, um dadurch deren Komplexität zu reduzieren. Yang & Ciarallo und später Rivera-Ros et al. erreichen dies, indem sie den Aspektansatz auf einzelne Bauteilmerkmale reduzieren. Konkret geht es in deren Arbeit um Liniensegmente, bei denen die Länge genau vermessen werden soll. Der kürzeste Pfad wird bei ihnen mit genetischen Algorithmen bestimmt. Da bei der vorliegenden Aufgabenstellung nicht nur bestimmte Merkmale betrachtet werden, sondern im Gegenteil der Anspruch besteht, die vollständige Bauteiloberfläche zu inspizieren, sind die vorgeschlagenen Maßnahmen zur Reduktion der Komplexität nicht anwendbar. Deswegen sind die graphentheoretischen Ansätze nicht auf den hier skizzierten Fall mit großen Bauteilmodellen anwendbar, die aus mehreren Millionen Punkten bestehen.

### 2.2.3.2. Mengentheoretische Ansätze

Die mengentheoretischen Ansätze bei der Inspektionsplanung wurden erstmalig in einer grundlegenden Arbeit von Tarbox & Gottschlich beschrieben (TG95). Der zentrale Grundgedanke dieses Ansatzes ist die Abbildung des Ansichtenplanungsproblems (engl. VPP) auf das Mengenüberdeckungsproblem (engl. Minimum Set Cover (MSC)). Dabei werden drei unterschiedliche Räume betrachtet. Der Raum der Objektoberfläche  $O$ , der aus diskreten Elementen (Punkten, Dreiecken usw.) der Oberfläche approximiert wird, der Raum der möglichen Sensorpositionen beziehungsweise Sensorkandidaten  $S$  und der mit der Positioniereinheit des Sensors inspizierbare Raum  $I$  (SRR03). Ziel ist es, eine Untermenge  $N$  an Sensorpositionen zu finden, mit der jedes Element  $o_i \in O$  von mindestens einem Sichtkandidaten  $s_i \in S$  betrachtet werden kann. Es gilt also  $N \subseteq S$ . Diese Untermenge kann durch Operationen auf einer binären Sichtbarkeitsmatrix  $M$  bestimmt werden. Ein Eintrag  $m_{ij}$  von  $M$  ist genau dann 1, wenn ein Oberflächenelement  $o_i$  für einen Sichtkandidaten  $s_j$  sichtbar ist. Bei einem weiteren Beitrag von Scott zur „spezifikationsgetriebenen“



Ansichtenplanung wird diese Sichtbarkeitsmatrix zu einer Messbarkeitmatrix (MM) erweitert. Dabei fließen dann neben der Sichtbarkeit weitere sensor- oder verfahrensspezifische Kriterien der Inspektion in einen Matrixeintrag mit ein. Diese Kriterien können auf der Sensorseite zum Beispiel der Sensorfokus, die Abweichung der Sensorstrahlrichtung von der Elementnormalen oder eine mögliche Verdeckung eines Elementes durch ein anderes sein. Der entsprechende Matrixeintrag ist jedoch weiterhin binär, also entweder 0 oder 1. Ein Eintrag wird 0, wenn ein Grenzwert eines der Kriterien überschritten wird. Die Daten über Sensor und Bauteil stammen dabei von entsprechenden Modellen. Das Bauteilmodell kann hier das Ergebnis eines vorgelagerten Digitalisierungsschrittes in einem mehrstufigen Ansatz sein wie zum Beispiel beim 3M-Ansatz (Modified Measurebility Matrix) von Scott (Sco09). Dieser mehrstufige Ansatz kann gut auf die vorliegende Aufgabenstellung abgebildet werden, da aufgrund der Bauteileigenschaften ein mehrstufiges Vorgehen vorgesehen ist. Die Auswahl der besten Ansichtenkandidaten erfolgt dabei durch Ansätze des MSC. Da in dieser Arbeit der gesamte Ablauf der Inspektionsplanung durchgeführt werden muss, werden im Folgenden alle sechs Einzelschritte des 3M-Ansatzes von Scott betrachtet und mit alternativen Varianten aus der Literatur abgeglichen.

**1. Akquirierung des Grobmodells:** In der Literatur wird die Pfadplanung für diesen Prozessschritt noch immer manuell durchgeführt (Sco09). Die Bauteiloberfläche wird dabei mit einem Sensor mit großem Messbereich vollständig aufgenommen und anschließend zu einem Netz verbunden. Bei Modellen, die vollständig in den Messbereich des Sensors passen, schlägt Scott den „Viewsphere-Ansatz“ von Tarbox vor, bei dem eine Menge von Ansichtenkandidaten in einem gleichmäßigen Raster mit festem Radius auf einer Kugel diskretisiert werden, die das Bauteil vollständig einhüllt (Abbildung 2.4) (TG95). Bei größeren Objekten rät Scott in seiner Übersichtsarbeit zu Ansätzen der Raumteilung, bei der immer nur ein Teil des Bauteils in die Pfadplanung eingeht (SRR03).

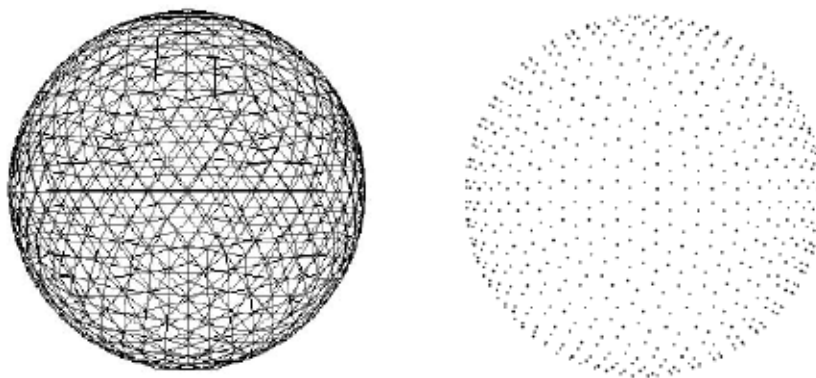


Abbildung 2.4.: Diskretisierung einer „Viewsphere“ (MF98)

**2. Dezimierung des Grobmodells:** In diesem Schritt soll das Grobmodell weiter vereinfacht werden. Für den Grad der Dezimierung verweist Scott auf durchgeführte Experimente, bei denen ein Faktor von der Elementdichte des Feinmodells zu der des Grobmodells von 32 experimentell ermittelt worden ist.

**3. Segmentierung des Grobmodells:** Die Segmentierung des Objektes in Teilbereiche bietet eine weitere Möglichkeit zur Reduktion der Rechenkomplexität. Scott schlägt vor, das Bauteil an natürlichen Kanten zu trennen. Er merkt allerdings selber an, dass verschiedene Parameter bei der Aufstellung der MM, wie zum Beispiel die Verdeckung bei dem vorgeschlagenen Vorgehen nur lokal auf einem Segment anstatt global für das ganze Bauteil betrachtet werden können.

**4. Generierung der Ansichtenkandidaten für die Inspektion:** Bei der Generierung der Ansichten für die Inspektion weist die Literatur eine Vielzahl an unterschiedlichen Ansätzen auf. Moutchiho hat sich in seiner Dissertation auf genau diesen Prozessschritt der Inspektionsplanung konzentriert. Hierbei unterscheidet er zwischen ein- und mehrstufigen Verfahren. Die mehrstufigen Verfahren werden von ihm in fünf verschiedene Kategorien eingeordnet (MB12). Alle betrachteten Ansätze basieren darauf, dass die Bauteiloberfläche in einem Präprozessorschritt für die Generierung der Ansichtenkandidaten segmentiert wird, um dadurch im nachfolgenden Schritt den Rechenaufwand zu reduzieren. Die resultierenden Segmente können entweder durch Rasterung des Bauteils oder durch Zusammenfassung von Oberflächenelementen mit gleichen optischen Charakteristika entstehen. Moutchiho weist auf zwei notwendige Voraussetzungen hin, die bei den von ihm betrachteten mehrstufigen Ansätzen erfüllt sein müssen: die betrachteten Bauteile müssen eine einfache Geometrie aufweisen und es muss eine zusätzliche Linearachse oder ein Drehtisches eingesetzt werden. Wichtige Arbeiten zu dieser Übersicht stammen von Martins et al. (MGBC05), Son & Lee (SKL03) und Prieto et al. (PLB03).

**5. Aufstellen der Messbarkeitsmatrix:** Die Datenstruktur einer Sichtbarkeitsmatrix, bzw. später Messbarkeitsmatrix wurde erstmals durch Tarbox et al. eingeführt (TG95). In seiner Arbeit kommen bistatische Matrizen für Kamera sowie Beleuchtung zum Einsatz. Die Rechenkomplexität bei der Optimierung der Matrix ist in dieser Arbeit noch vergleichbar hoch, da Objekt- und Ansichtenkandidaten keiner Filterung unterzogen werden. Scott greift diesen Ansatz in mehreren Veröffentlichungen auf und erweitert diesen in seinem einleitend beschriebenen 3M-Ansatz. Hierbei reduziert er die Anzahl der Ansichtenkandidaten durch Abtastung des Ansichtenraums hinsichtlich der durch Sensor- und Objektmodell definierten optimalen Zone. Diese Zone ist durch unterschiedliche Anforderungen definiert (Sco09).

**6. Auswahl der Ansichten:** Bei der Auswahl der Ansichtskandidaten werden die MM und verschiedene Algorithmen zum Lösen des Mengenüberdeckungsproblems (engl. Set Covering Problem (SCP)) herangezogen. Beispiialgorithmen sind das simulative Ausglühen (Sen93), genetische Algorithmen (BC96) und die Lagrange Relaxation (CTF00). Die besten Resultate sind aller-

dings laut Literatur mit den sehr einfachen, aber schnellen Greedy-Suchalgorithmen zu erzielen. Diese sind allerdings NP-schwer und führen nicht immer zu der kleinsten und damit optimalen Untermenge der entsprechenden Grundgesamtheit der Ansichtenkandidaten (Sco09).

Chen nennt die Inspektion im industriellen Umfeld in einer wichtigen Übersichtsarbeit den relevantesten Anwendungsbereich für die modellbasierte Ansichtenplanung (CLK11). In den zentralen Veröffentlichungen zur modellbasierten Inspektionsplanung nach Scott kommen ausschließlich Tiefenkameras mit aktiver Beleuchtung wie Laserlichtschnitt- oder Streifenprojektionssensoren zum Einsatz. Bei diesen Sensoren ist der Messbereich so groß, dass das zu inspizierende Bauteil schon durch wenige Aufnahmen vollständig inspiziert werden kann. Die speziellen Herausforderungen, die durch die einzigartigen Eigenschaften der WLI wie dem kleinen Messbereich und dem beschränkten Messwinkel in Kombination mit den verformten Bauteilen stammen, treten in diesen Arbeiten gar nicht auf und werden deswegen nicht behandelt. Der Fokus liegt deswegen auf anderen Aspekten der Pfadplanung.

Munkelt stellt zum Beispiel in seiner Dissertation ein Inspektionssystem für Karosserieteile von Automobilen vor, bei dem er Scotts Ansichtenplanung anwendet, um die Positionen eines Streifenlichtsensors zu planen (Mun11). Bei ihm kann das gesamte Bauteil in circa 10 Aufnahmen inspiziert werden. Seine Hauptbeiträge sind die direkte Einbeziehung der Messunsicherheit in den Planungsprozess, die Kombination von Verfahren ohne a priori Wissen mit Verfahren mit a priori Wissen und die Einführung einer Bewertungsmetrik für Ansichtenplanungsergebnisse.

Die Zielsetzung in der Dissertation von Moutchiho ist die Erhöhung des Automatisierungsgrades eines roboterbasierten Inspektionssystems (Flexible Inspection System (FIS)) durch Einsatz der Ansichtenplanung. Auch in diesem System kommt ein Laserlichtschnittsensor zum Einsatz. Das Ziel dieses Systems ist nicht die vollständige Inspektion, sondern stattdessen die Kontrolle einzelner Bauteilmerkmale. Über seine übergeordnete Zielsetzung zu erreichen, benennt er als zentrale Schritte eine Konzipierung und Entwicklung einer Methode zur optimalen Positionierung eines Bauteils in einer Inspektionszelle, eine CAD-basierte Beschreibungsmethode für Bauteile und ein automatisches System zur Trajektorienplanung unter Einbezug verschiedener Limitierungen (MB12). Für diese Arbeit nützlich ist die Vorstellung vieler Ansätze zur Vereinfachung des Ausgangsmodells. Hosseininaveh et al. stellen in einer aktuellen Veröffentlichung das System Imaging Network Designer (IND) vor, das die Planungskomponente eines roboterbasierten Systems zur Rekonstruktion von Kulturgütern mit einer Stereokamera darstellt. Er wendet ähnlich wie von Scott vorgeschlagen und anders als die beiden anderen zentralen Arbeiten einen mehrstufigen Ansatz an, bei dem ein Grobmodell eines Objektes mit einer *Kinect*-Kamera der Firma *Microsoft* erfasst wird und dann auf der errechneten Hüllkörperfläche dieses Modells in einem Raster Ansichtenkandidaten erstellt werden. Aus diesen werden dann unter Berücksichtigung verschiedener Randbedingungen die besten ausgewählt (HSE14). Der gerasterte Hüllkörperansatz kann nur gewählt werden, weil die Kamera einen großen Messbereich aufweist. Außerdem dürfen die zu inspizierenden Bauteile keine Hinterschnitte oder Innenseiten aufweisen, weil beim Hüllkörperan-

satz immer von konvexen Objekten ausgegangen wird. Interessant ist allerdings die Ausnutzung von Bauteilsymmetrien, die auch für diese Arbeit hilfreich ist.

### 2.3. Software für eine automatische robotergestützte Inspektion

Wie in der Zielsetzung herausgearbeitet, wird für die automatische Inspektion mit einem WLI-Sensor ein flexibles Steuerungssystem benötigt. Dieses System muss aufgrund von heterogenen Anforderungen eine Vielzahl an unterschiedlich gearteten Funktionalitäten ermöglichen. Neben klassischen Steuerungsfunktionen werden sowohl Anteile der On-Line- als auch Anteile der Off-Line-Programmierung benötigt. Zusätzlich muss die Kommunikation mit unterschiedlichen Sensoren über verschiedene Schnittstellen und eine anschließende Datenverarbeitung möglich sein. Dieser breitgefächerte Funktionsumfang übersteigt die Möglichkeiten einer Speicherprogrammierbare Steuerung (SPS), die üblicherweise in Industrieroboteranlagen für Steuerungsaufgaben eingesetzt werden, sodass stattdessen der Einsatz einer hochsprachenbasierten PC-Steuerung mit entsprechender zusätzlicher Software nötig wird. Außerdem ist unbedingt zu beachten, dass diese Software in einer Applikation im industriellen Umfeld eingesetzt werden soll und deswegen zusätzliche Sicherheitsfunktionen beinhalten muss. Die folgende Aufzählung veranschaulicht den formulierten minimalen Funktionsumfang des benötigten Softwaresystems:

#### Geforderter Minimalumfang der Systemsteuerung

- Ablauf- und Prozesssteuerung bei der Inspektion
- Bewegungssteuerung eines Industrieroboters
- Robotersimulation für Kollision und Erreichbarkeit
- Datenschnittstellen zu unterschiedlichen optischen Sensoren
- grundlegende mathematische Algorithmen
- Datenvisualisierung und -manipulation
- Benutzerschnittstelle (engl. Graphical User Interface (GUI))

In den folgenden Abschnitten werden sowohl proprietäre als auch generische Softwarelösungen und verschiedene Frameworks vorgestellt, diskutiert und entsprechend überprüft, in welchem Umfang sie die gestellten Anforderungen erfüllen können. Danach kann eingeschätzt werden, ob sie geeignet sind, in dem betrachteten Inspektionssystem im industriellen Umfeld eingesetzt zu werden.

### 2.3.1. Proprietäre und generische Softwarelösungen

Aktuell ist eine Vielzahl von unterschiedlichen proprietären Softwarepaketen verschiedener Hersteller kommerziell erhältlich, die eine oder mehrere der geforderten Funktionalitäten abdecken können. So bieten zum Beispiel alle großen Roboterhersteller Off-Line-Simulationssoftware für ihre eigenen Robotermodelle an. Hierbei können neben mehreren Robotern auch CAD-Modelle importiert werden, um eine entsprechende Roboterzelle detailgetreu abbilden und simulieren zu können. Zusätzlich sind diese Softwarepakete in der Lage, die Robotersteuerung der entsprechenden Hersteller in der Simulation widerzuspiegeln, sodass schon offline Fehler bei der Ausführung des Programms erkannt werden können. Auch Hersteller von optischen Sensoren bieten unterschiedlich komplexe Softwarepakete an, die eine Akquise und Visualisierung von optischen Daten ihrer entsprechenden Sensoren sowie die einfache Manipulation dieser Daten ermöglichen. Die größte Überdeckung bezüglich des Funktionsumfangs bietet die Software *GOM-Inspect* des Unternehmens *gom*. Diese Software bietet Features zur automatischen Planung für eine vollständige Inspektion und eine Vielzahl von Funktionen zur Datenmanipulation, zum Ist-Soll-Abgleich sowie zur Objektrekonstruktion. Allerdings ist die Software auf den Betrieb ihrer eigenen Sensoren beschränkt, weswegen sie bei der Betrachtung ausgeschlossen wurde. Zusätzlich zu den proprietären Systemen ist auf dem Markt auch generische Software erhältlich. Diese bietet neben der herstellerübergreifenden Off-Line-Simulation von Industrierobotern in manchen Fällen auch noch eine Simulation ganzer Prozessabläufe an. Doch genau wie bei den proprietären Softwaresystemen deckt auch kein generisches Softwaresystem das geforderte heterogene Funktionsspektrum ab. Die unbedingt benötigten Schnittstellen zu Industrierobotern und optischer Sensortechnik in Kombination mit dem hohen Anteil an komplexer Algorithmik und mathematischen Funktionen ist in dem vorliegenden Fall so speziell, dass hier auf kein kommerzielles Softwarepaket zurückgegriffen werden kann und stattdessen eine neue Applikation programmiert werden muss. Im Folgenden werden diese Pakete deswegen nicht weiter betrachtet.

### 2.3.2. Frameworks

Eigenständige und inhaltlich abgegrenzte Softwarebibliotheken bieten ganz allgemein die Möglichkeit, flexibel ein auf spezielle Anforderungen angepasstes Framework zusammenzustellen. Dieses kann neben vorgefertigten Methoden und Funktionen auch Komponenten zur Software-Infrastruktur bieten, die eine effektivere Programmierung ermöglichen. Ältere Programmiersprachen können auf diese Weise durch weitere Konzepte ergänzt werden. Aktuell sind unterschiedliche Bibliotheken für alle Bereiche der Robotik wie zum Beispiel zur Kinematikberechnung und -simulation, zur Datenvisualisierung und -manipulation, zur Kommunikation über unterschiedliche Bussysteme und Protokolle, für alle grundlegenden mathematischen Aufgabenstellungen sowie für Benutzerschnittstellen verfügbar. In manchen Fällen sind diese Bibliotheken quelloffen (engl. open source) und dadurch sogar im kommerziellen Umfeld frei nutz- und beliebig erweiterbar. Zusätzlich bieten viele Roboter- wie Sensorhersteller Programmierschnittstellen (engl.

Application Programming Interface (API)) zur Kommunikation mit ihren Hardwarekomponenten für unterschiedliche Programmiersprachen an, sodass leicht angepasste Treiber für die Hardware entwickelt werden können. Mit Hilfe eines solchen Frameworks kann eine individuelle Software programmiert werden, die den geforderten Funktionsumfang vollständig bieten kann. Prozessabhängige Programmanteile müssen jedoch auch in diesem Fall zusätzlich implementiert und dem zusammengestellten Framework hinzugefügt werden.

Im Bereich der Robotik-Forschung sind in den letzten Jahren eine Vielzahl Frameworks entstanden. Einige populäre Beispiele hierfür sind USARSim, ROS, UberSIM, EyeSim, SubSim, MRDS und robotSim pro (HC11), von denen exemplarisch das Robot Operating System (ROS) das Microsoft Robotics Developer Studio (MRDS) und die Virtual Robot Experimentation Platform (V-REP) genauer betrachtet werden.

Das aktuell populärste und mit weit über 1500 Bibliotheken umfangreichste und relevanteste Framework für die Robotik ist das ROS. Die Motivation zur Entwicklung dieses Frameworks ist das immer schneller wachsende Feld der Robotik und die damit verbundenen heterogenen Bibliotheken für Roboter, Sensoren und verschiedenen Algorithmen zur Pfadplanung und Navigation. Das ROS ist ein open source Framework (unter der BSD Lizenz) und implementiert nicht einzelne Softwarebausteine, sondern setzt auf einem tiefen und abstrakten Programmierlevel auf und bietet dem Entwickler eine graphenbasierte Kommunikationsinfrastruktur. Die zentralen Elemente dieser Struktur sind *Knoten*, die Softwarebausteine beliebiger Programmiersprachen kapseln, *Nachrichten* (engl. messages), die zwischen diesen Knoten gesendet werden können, *Themen* (engl. Topics) die Nachrichten erhalten und von anderen Knoten abonniert werden können und *Dienste* (engl. services), die zwei Knoten exklusiv mit einander verbinden (QCG09, Kou16). Der Code dieser Plattform ist quelloffen und viele Forschungsinstitute und Unternehmen weltweit haben auf Basis dieses Frameworks Module für unterschiedliche Aufgaben im Bereich der Robotik entwickelt. Der Fokus liegt dabei nicht auf typischen Industrierobotern, sondern aktuell eher auf mobilen Robotern mit einer oft sehr hohen Anzahl an Achsen und der schnell wachsenden Servicerobotik.

Trotz der genannten Features des ROS ist die Akzeptanz in der Industrie aufgrund seines Forschungscharakters, der mangelnden Langzeiterfahrung im industriellen Umfeld sowie fehlender Verantwortlicher im Fall eines Systemausfalls noch immer beschränkt. Hinzu kommt, dass das Framework ROS Linux basiert ist, was den Einsatz in der von Microsoft beherrschten Industrie zusätzlich erschwert (PH13). Bei der speziell für die Industrie entwickelten Variante ROS-industrial wird diesen Einschränkungen aktuell bewusst entgegengearbeitet (Bau16).

Einen ähnlichen Ansatz wie beim ROS haben Jackson et al. als Mitarbeiter des Unternehmens Microsoft verfolgt. Sie haben zusammen das Framework MRDS entwickelt. Dieses Framework ist nicht quelloffen, stattdessen muss zur Entwicklung eine Lizenz erworben werden. Ähnlich wie das ROS baut dieses Framework auf voneinander unabhängigen Knoten auf, die allerdings in eine Zustandsautomatenstruktur eingebettet sind. Auch bei diesem Framework kommunizieren die einzelnen Knoten über Peer-to-Peer Verbindungen in einem verteilten Nachrichtensystem. Ein

großes Problem des MRDS ist die Abhängigkeit von .NET-Programmiersprachen (Jac07). Gerade auf Hardwareebene kommen Rechnerstrukturen vor, die .NET Architekturen nicht unterstützen. Für diese Fälle werden zusätzliche Rechner als Schnittstellen zwischen dem .NET-System und der entsprechenden Hardware benötigt. Anders als das ROS wird das MRDS nicht durch die Forschung getrieben und nur partiell von der Industrie angenommen, sodass die Nutzbarkeit aufgrund des noch immer geringen Funktionsumfanges eingeschränkt ist.

Auch das V-REP Framework stellt eine umfangreiche Simulationsumgebung für unterschiedliche Roboterapplikationen dar. Seine Alleinstellungsmerkmale sind die plattformunabhängige Lua-Skriptinfrastruktur (IFF96), die eine sehr gute Portabilität von Programmen ermöglicht und die Vielzahl von Schnittstellen (RSF13). Hierdurch kann das Framework auch fehlende Funktionalitäten abdecken, die aktuell nur von anderen Frameworks wie dem ROS abgedeckt werden. Durch diese beiden Merkmale erhält das Framework eine sehr hohe Flexibilität. Die Steuerung von Robotern ist allerdings keine Kernkompetenz dieses Frameworks, wird allerdings trotzdem zur Verfügung gestellt. Diese sind aber für industrielle Applikationen ungeeignet, da sie die eigentliche Steuerung ersetzen, was in der Industrie aufgrund der hohen Sicherheitsanforderungen nur schwer umzusetzen ist.

## **2.4. Diskussion des aktuellen Standes der Wissenschaft**

Die Betrachtung des aktuellen Standes der Wissenschaft führte zur Ableitung des notwendigen eigenen Beitrags. Er gliedert sich entsprechend der vorangegangenen Darstellung in die Bereiche: robotergeführte WLI, Pfadplanung für die automatische Inspektion und softwarebasierte Steuerungssysteme für den übergeordneten Inspektionsprozess.

### **2.4.1. Weißlichtinterferometrie für die Inspektion**

Im Bereich der automatisierten WLI ist eine Literaturrecherche in wissenschaftlichen Veröffentlichungen und die zusätzliche Betrachtung der Industrie ohne Ergebnis geblieben. Aktuell wird die WLI in keiner anderen robotergestützten Anwendung zur Inspektion eingesetzt. Im Rahmen dieser Arbeit wurden deswegen Weißlichtinterferometer betrachtet, die in Tischaufbauten zur Anwendung kommen. Keines dieser betrachteten Systeme kommt aufgrund des mangelnden Reifegrades für die hier skizzierte Aufgabenstellung in Frage. Das bedeutet, dass in dieser Arbeit ein automatisches System für die Inspektion mit einem WLI-Sensor grundlegend neu konzipiert werden muss. Dieses automatische System setzt sich aus den benötigten Hardwarekomponenten und einem entsprechenden Inspektionsprozess zusammen. Bei dem Prozess geht es sowohl um den übergeordneten Ablauf bei der Inspektion als auch um die einzelnen Teil- und Hilfsprozesse. Hierbei reicht der eigene Beitrag über die Konzipierung hinaus auch in eine Implementierung. Bei der Hardware wird die Implementierung durch ein Unternehmen im Rahmen des Forschungsvorhabens übernommen. Bei jedem der genannten Arbeitspakete müssen die in der Einleitung

vorgestellten globalen Zielformulierungen erfüllt sowie die speziellen Randbedingungen berücksichtigt werden.

### 2.4.2. Automatische Inspektionsplanung

Im Bereich der automatischen Pfadplanung für die Inspektion ist die Literatur bei den modellbasierten Off-Line-Verfahren dominiert von dem mengentheoretischen Ansatz von Scott beziehungsweise Ableitungen davon. Speziell bei dem von ihm vorgeschlagenen 3M-Ansatz lassen sich viele Aspekte auf den hier betrachteten Anwendungsfall übertragen. Aufgrund der sehr speziellen Kennwerte und Eigenschaften des WLI-Sensors in Kombination mit den zu inspizierenden beschädigten Bauteilen muss dieser Ansatz in einem eigenen Beitrag allerdings an mehreren Stellen über den aktuellen Stand der Wissenschaft hinaus weiterentwickelt werden. In der Literatur sind ausschließlich Applikationen dieses Ansatzes zu finden, bei denen das Bauteil von nur einer oder wenigen Positionen durch einen Sensor mit einem großen Messbereich vollständig inspiziert werden kann. Das besondere Größenverhältnis von Messbereich des Weißlichtinterferometers zu der Bauteiloberfläche der Triebwerkskomponente in dieser Arbeit führt allerdings zu neuartigen Problemen, die in der Literatur noch nicht betrachtet worden sind und nachfolgend erläutert werden.

Die aufgrund des kleinen Messbereichs hohe Anzahl an notwendigen Einzelmessungen bei der WLI würde die zentrale Struktur der MM in dem betrachteten Fall so groß werden lassen, dass die Bestimmung eines Inspektionspfades allein, die geforderte Taktzeit überschreiten würde. Speziell bei dem Prozessschritt der Generierung von Ansichtenkandidaten bei Scotts 3M-Ansatz muss für die vorliegenden Anforderungen aus der Aufgabenstellung ein neues Verfahren entwickelt werden, da dessen Komplexität von dem digitalisierten Modell abhängt, das als Eingangsgröße in diesen Prozessschritt miteinfließt. Dieses besteht in dem betrachteten Fall aus mehreren Millionen Punkten. Moutchino beschäftigt sich zwar mit der Vereinfachung der Daten durch Ausnutzung unterschiedlicher Bauteileigenschaften wie zum Beispiel Symmetrien. Sein Ansatz ist allerdings nicht übertragbar auf Bauteile, bei denen wie in diesem Fall die lokale beziehungsweise globale Verformung im Größenbereich des ausgenutzten Bauteilmerkmals liegt. Keine der zentralen Veröffentlichungen liefert einen Ansatz, wie bei den speziellen Anforderungen dieser Arbeit vorgegangen werden könnte, weswegen speziell bei dem Prozessschritt der Generierung von Ansichtenkandidaten an dieser Stelle ein eigener Beitrag zu dem Stand der Wissenschaft notwendig ist. Dieser kann aus den genannten Gründen nicht allein auf dem mengentheoretischen Ansatz von Scott aufbauen, sondern muss stattdessen mehrstufig ausgelegt und an die speziellen Eigenschaften der Bauteile angepasst sein.

### 2.4.3. Softwarebasierte Steuerungssysteme

Bei softwarebasierten Steuerungssystemen für die Automatisierung wurde herausgearbeitet, dass die abgeleiteten breitgefächerten Anforderungen der Aufgabenstellung weder mit einer proprietä-



ren Software eines Roboter- oder Sensorherstellers noch mit einem aktuellen generischen Softwareprodukt vollständig erfüllt werden. Besser geeignet sind stattdessen einfach individualisierbare Frameworks, die sich aus einer beliebigen Anzahl von eigenständigen Softwarebibliotheken derselben Programmiersprache zusammengesetzt werden können. Bei der Auswahl sind quelloffene Frameworks kommerziellen vorzuziehen, weil erstere flexibel erweitert und beliebig auf die vorliegende Aufgabenstellung angepasst werden können und parallel von einer großen Gemeinde gepflegt und weiterentwickelt werden. Bestehende Frameworks wie das ROS oder das MRDS erfüllen zwar viele der formulierten Anforderungen, sind aber entweder in ihrem aktuellen Zustand noch ungeeignet für den Einsatz zusammen mit industriellen Komponenten oder sie beschränken sich auf Programmiersprachen, für die nur wenige Bibliotheken zur Verfügung stehen oder für die ein zusätzlicher Aufwand für das Einhüllen (engl. wrapping) erbracht werden müsste. Außerdem wird bei aktuellen Frameworks vor allem auf eine leicht erweiterbare Infrastruktur fokussiert, anstatt auf die eigentlichen Inhalte. Die Infrastruktur ist allerdings bei einer abgegrenzten Aufgabenstellung mit einem vorformulierten Anforderungsrahmen niedriger priorisiert und ist stattdessen wichtiger bei Systemen, bei denen es darum geht, effektiv einzelne Module in weitere Applikationen zu integrieren. Im Rahmen dieser Arbeit wird deswegen aus verschiedenen Bibliotheken ein eigenes Framework zusammengestellt und daraus eine individuelle Steuersoftware konzipiert und implementiert.

## 2.5. Aufbau der Arbeit

Nachdem in diesem Kapitel der Stand von Wissenschaft und Technik zu den Kernthemen dieser Arbeit dargestellt und bestehende Defizite herausgearbeitet worden sind, folgen die inhaltlichen Kapitel, die der Zielsetzung entsprechend strukturiert sind.

Im **Kapitel 3** wird die Konzipierung des Aufbaus für die automatische Inspektion mit einem Industrieroboter vorgestellt. Hierzu gehört unter anderem die Auswahl eines passenden Roboters und zusätzlicher Handhabungskomponenten. Außerdem muss neben dem Weißlichtinterferometer ein weiterer Sensor in den Aufbau integriert werden, mit dem die individuelle Bauteiloberfläche möglichst schnell akquiriert werden kann.

**Kapitel 4** umfasst die Konzeption des gesamten Inspektionsprozesses. Nachdem die notwendigen Hilfsprozesse identifiziert und in einer sinnvollen Struktur zu einem Gesamtprozess miteinander verknüpft worden sind, wird für die Umsetzung jedes Teilprozesses ein grundsätzliches Konzept aufgestellt.

**Kapitel 5** zeigt die Implementation der Softwarekomponenten auf. Diese werden für die Ausgestaltung der Konzepte für die einzelnen Teilprozesse benötigt. Hier wird vor allem der Kern des Handhabungsanteils des Inspektionssystems, die Software *AutoXIV*, mit seiner Architektur eingeführt und im Netzwerk des Gesamtsystems eingeordnet.

**Kapitel 6** beschreibt die Ausgestaltung und Implementierung des vorher konzipierten Inspekti-

onsprozesses. Hierbei liegt der Fokus auf den verwendeten grundsätzlichen Prinzipien, Algorithmen und Funktionen.

**Kapitel 7** dient der qualitativen sowie quantitativen Validierung der Anforderungen an den konzipierten Inspektionsaufbau und Inspektionsprozess. Hierzu wird ein Testbauteil sukzessive durch den gesamten Prozess geführt und an ihm jeder Teilschritt validiert. Zusätzlich werden die entsprechenden Ergebnisse im Detail visualisiert.

**Kapitel 8** enthält eine Zusammenfassung von Inhalt und Ergebnissen. Abschließend wird ein Ausblick sowohl auf mögliche Erweiterungen des vorgestellten Ansatzes als auch potentielle Anschlussarbeiten gegeben.

### 3. Konzipierung des Inspektionsaufbaus

Die globalen Anforderungen an das automatische Inspektionssystem mit einem Weißlichtinterferometer, wie sie in der Einleitung identifiziert und beschrieben wurden, umfassen eine *vollständige Messabdeckung* und eine *Durchlaufzeit* von unter acht Stunden. Die Durchlaufzeit spielt bei der Konzipierung des Inspektionsaufbaus eine eher untergeordnete Rolle und wird detaillierter in der Konzipierung des Inspektionsprozesses betrachtet. Allerdings gilt es schon an dieser Stelle die Anforderung der Vollständigkeit zu berücksichtigen, da nur bei einer vollständigen Inspektion der vom Hersteller spezifizierten Bauteiloberfläche garantiert werden kann, dass jede Beschädigung zumindest mit dem WLI-Sensor erfasst wird und einer späteren Auswertung zugeführt werden kann.

Zu Beginn wurden neben den globalen Anforderungen auch Randbedingungen für das Inspektionssystem formuliert, die sich ebenfalls auf die vollständige Abdeckung bei der Inspektion auswirken. Sie haben ihren Ursprung in dem betrachteten Brennkammerbauteil, der Handhabung sowie der verwendeten Sensortechnik. Die Eigenschaften der verschiedenen Komponenten sind in Tabelle 3.1 zusammengefasst und, wenn möglich, mit einem Kennwert quantifiziert.

Komponente	Eigenschaft	Kennwert
Bauteil	Abmessungen	800 mm × 800 mm × 200 mm
	Gewicht	ca. 10 kg
	Geometrie	Hohlkegelstumpf
	Geometrie	rotationssymmetrisch
	Geometrie	komplexes Profil
	Geometrie	globale und lokale Verformungen
Sensor	Abmessungen	120 mm × 80 mm × 50 mm
	Gewicht	1.2 kg
	Messabstand	ca. 50 mm
	Messvolumen	5.7 mm × 5.7 mm × 2.7 mm
	Aufnahmequalität	abhängig vom Messwinkel
	Aufnahmequalität	abhängig von Schwingungen
Handhabung	Kinematik	anfällig gegenüber Schwingungen
	Genauigkeit	schlechte Positioniergenauigkeit, gute Wiederholgenauigkeit

Tabelle 3.1.: Eigenschaften der Subkomponenten

Es wird deutlich, dass aus der Kombination von IR und WLI, aufgrund der Sensibilität der beiden Komponenten gegenüber Schwingungen eine weitere Anforderung erwächst. Diese ist die Minimierung der Einbringung von Schwingungen in das Inspektionssystem. Zusätzlich muss schon früh in der Konzeptionsphase des Systems beachtet werden, dass die Bauteile von ihrer idea-

len Geometrie abweichen. Damit diese Abweichung vor der eigentlichen Inspektion quantifiziert und später einer Off-Line-Pfadplanung zugeführt werden kann, muss neben dem Weißlichtinterferometer ein weiterer Sensor für das Inspektionssystem ausgewählt und mit in den Messaufbau integriert werden. Anhand dieser drei unterschiedlichen Anforderungen *Vollständigkeit*, *Schwingungsreduktion* und *zusätzliche Sensorik* ist das vorliegende Konzipierungskapitel strukturiert.

### 3.1. Vollständige Inspektion

Bei der globalen Anforderung der vollständigen Inspizierbarkeit spielen drei unterschiedliche Faktoren eine Rolle, die bei der Konzipierung des Aufbaus berücksichtigt werden müssen. Der erste Faktor ist die *Erreichbarkeit*, die angibt, ob eine geplante Inspektionspose mit einer ausgewählten Roboterkinematik erreicht werden kann. Der zweite Faktor ist die *Zugänglichkeit*, die zusätzlich zur Erreichbarkeit beschreibt, ob der Sensor durch den Roboter auf eine erreichbare Pose positioniert werden kann, ohne bei der Umsetzungsbewegung mit dem Bauteil zu kollidieren. Der letzte Faktor ist die *Messabdeckung*, die zusätzlich angibt, welcher Teil der Oberfläche mit dem Messbereich des kollisionsfrei positionierten Sensors auf erreichbaren Posen inspiziert werden kann. Dieser Faktor gibt also schlussendlich an, zu welchem Maß die Anforderung der vollständigen Inspektion erfüllt werden kann.

#### 3.1.1. Auswahl eines Industrieroboters

In einer dieser Arbeit vorausgegangenen, nicht öffentlichen Vorstudie wurde ein IR in der Vertikal-Knickarm-Konfiguration als am besten geeignetes Handhabungsgerät für das zu entwickelnde Inspektionssystem identifiziert. Dieser bietet einen großen Arbeitsraum, in dem Werkzeuge frei positioniert und orientiert werden können und weist gleichzeitig ein hohes Handhabungsgewicht auf. Bei der Auswahl eines zu der Aufgabe passenden Modells muss zusätzlich ein weiterer Parameter berücksichtigt werden. Aufgrund der Komplexität des betrachteten Prozesses und der dadurch benötigten Flexibilität bei der Programmierung des Roboters spielen auch die zur Verfügung stehenden Programmierschnittstellen eine wichtige Rolle. Die beschränkte Genauigkeit eines IR wird an dieser Stelle nicht weitergehend betrachtet, weil diese später im Prozess kompensiert werden soll. Alle am Manipulator des Roboters montierten Werkzeuge wiegen zusammen mit ihren Peripheriekomponenten wie Kabel und Sicherheitseinrichtungen deutlich unter zehn Kilo, weswegen die Mindestanforderung beim Handhabungsgewicht 10 kg beträgt. Der Radius der verschiedenen Varianten des betrachteten hohlkegelstumpfförmigen Brennkammerbauteils beträgt ca. 350 mm und die Höhe ca. 200 mm. Über diese Abmessungen ist der minimal notwendige Arbeitsraum des Roboters definiert. Wie eingangs erwähnt spielen neben den geometrischen Anforderungen auch die angebotenen Schnittstellen der Robotersteuerung eine wichtige Rolle. Für eine erste Versuchsanlage wurde ein Roboter der Marke FANUC verwendet, der nicht über die notwendigen offenen Programmierschnittstellen verfügte und deswegen nicht für die finale Demonstratoranlage eingesetzt werden konnte. Stattdessen wurde den Anforderungen folgend das

Modell KR30HA der Firma KUKA mit mittlerer Traglast ausgewählt. Die Robotersteuerungen der Firma KUKA bieten eine Vielzahl unterschiedlicher Schnittstellen an. Eine davon ist das Robot Sensor Interface (RSI), über das eine Echtzeitkommunikation mit der Robotersteuerung im 4 ms-Takt möglich ist. Das vorgeschlagene Modell weist für das betrachtete Bauteil einen ausreichend großen Arbeitsraum (siehe Abbildung 3.1) und ein Handhabungsgewicht von 30 kg auf. Außerdem zeichnet ihn in der High Accuracy (HA)-Variante eine im Vergleich zu dem Standardmodell überdurchschnittlich hohe Wiederholgenauigkeit von  $50\text{ }\mu\text{m}$  aus. Das Modell erfüllt also alle beschriebenen Anforderungen und ist deshalb besonders gut für das zu konzipierende Inspektionssystem geeignet.

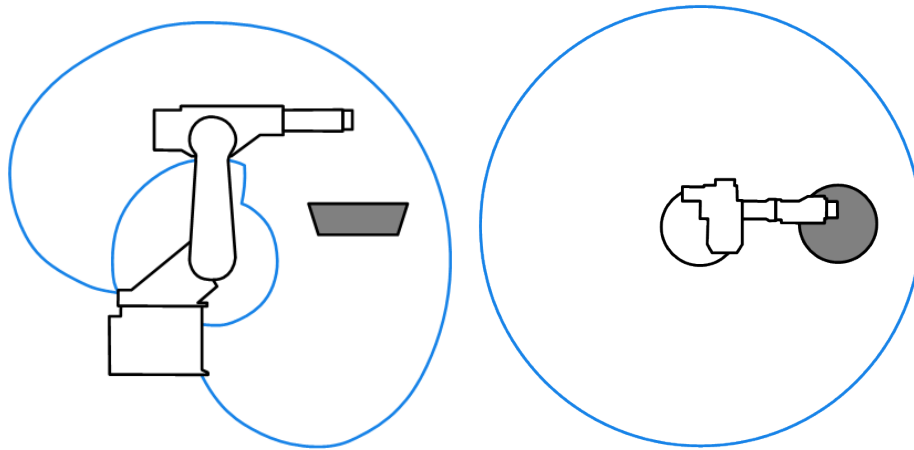


Abbildung 3.1.: Schematischer Arbeitsraum (blau) des Industrieroboters Kuka KR30 in Seiten- und Draufsicht mit Bauteil (grau)

### 3.1.2. Auswahl zusätzlicher Handhabungskomponenten

Um die Rotationssymmetrie des betrachteten Bauteils für die Vereinfachung des Inspektionsprozesses ausnutzen zu können und zur weiteren Vergrößerung des Arbeitsraumes, wurde zusätzlich zu dem ausgewählten IR eine Drehachse in das Handhabungssystem integriert. Die Position der Drehachse in der Anlage seitlich vorne innerhalb des Arbeitsraumes des IR wurde gewählt, damit sowohl Randsingularitäten und innere Singularitäten aufgrund von kollinearen Roboterachsen bei der Inspektion vermieden werden (Abbildung 3.2b).

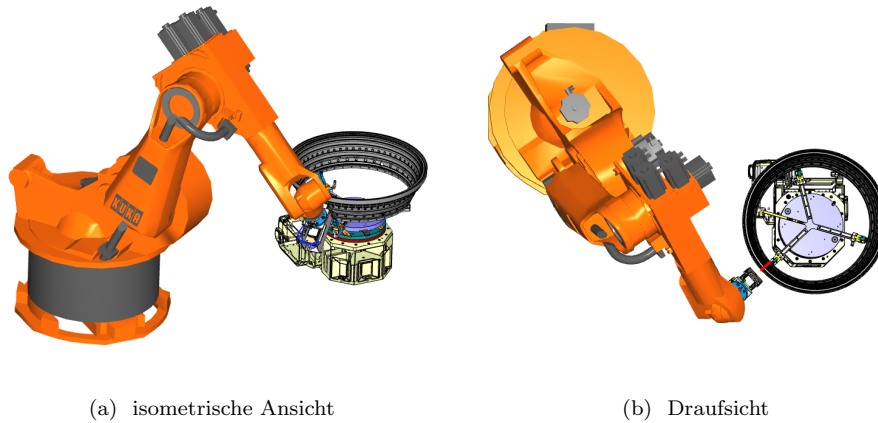


Abbildung 3.2.: Positionierung der zusätzlichen Drehachse in Relation zum Industrieroboter

Dieser Aufbau erlaubt eine ringweise Strategie bei der Bauteilinspektion mit einem WLI (Abbildung 3.3). Hierbei verfährt der Roboter den Sensor auf eine Messpose (2.) und steht dann im weiteren Verlauf still. Durch anschließendes inkrementelles Drehen (3.) der zusätzlichen Drehachse kann auf diese Weise ein vollständiger horizontaler Bauteilring inspiziert werden. Erst wenn ein Ring vollständig inspiziert ist, wird der Sensor von dem IR auf die nächste Pose verfahren. Durch diesen Ansatz führt die Drehachse den Großteil der Relativbewegungen zwischen Sensor und Bauteil durch. Das bedeutet allerdings, dass die Gesamttaktzeit der Inspektion erheblich von den dynamischen Werten dieser Drehachse abhängt. Deswegen stellen diese Werte das Hauptkriterium bei der Auswahl eines passenden Drehachsmodells dar.

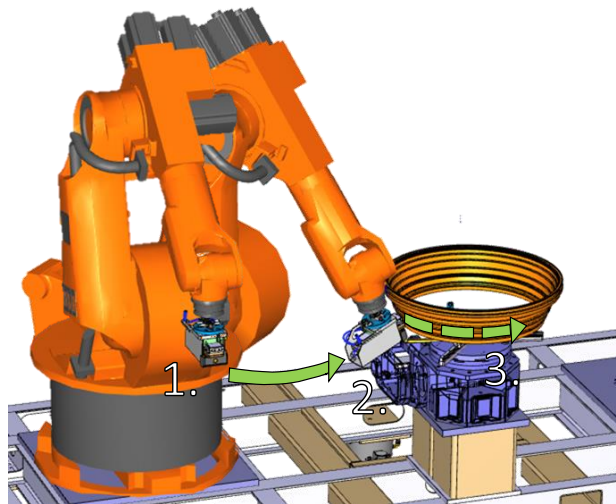


Abbildung 3.3.: prinzipielles Vorgehen bei der Inspektion mit einem Weißlichtinterferometer

Auf einem ideal runden Teil könnte mit dieser Strategie das gesamte Bauteil inspiziert werden. Das Bauteil weist jedoch Verformungen auf, die größer sind als der Messbereich des Sensors. Damit sich die Bauteiloberfläche während der Inspektion trotzdem garantiert im Messbereich des Sensors befindet muss dieser also zugestellt werden. Hierzu kann, wie in Abbildung 3.4 veranschaulicht, die für die Weißlichtinterferometrie prinzipbedingt benötigte Linearachse des WLI-Sensors genutzt werden. Es muss also für diesen Zweck keine weitere Aktorik in den Aufbau integriert werden (siehe Kapitel 2.1).

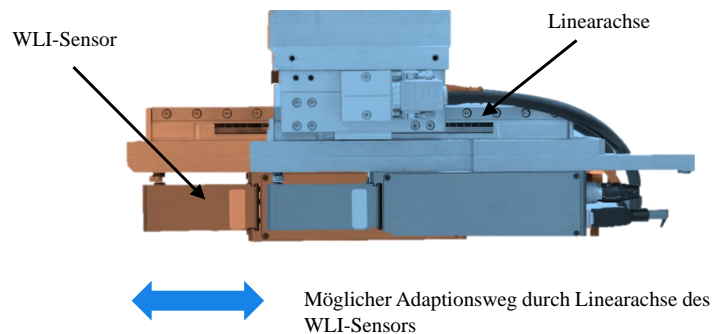


Abbildung 3.4.: Adaption durch Verfahren der WLI-Linearachse

### 3.1.3. Konzept zur Bestimmung der Messabdeckung

Um objektiv nachzuweisen, dass die Bauteiloberfläche mit der ausgewählten Handhabungskonfiguration vollständig inspizierbar ist, müssen die Parameter der Bauteilzugänglichkeit und die Erreichbarkeit aller Messposen zusammen mit einer exakten Messabdeckung bestimmt werden. Ein Konzept, die genannten Parameter mit geringem Aufwand zu validieren, ist eine Off-Line-Simulation.

Die für diese Arbeit ausgewählte Simulationsumgebung ist *FAMOS* der Firma *carat robotic innovation GmbH*. Dem Anwender steht bei *FAMOS* eine umfangreiche Bibliothek mit den gängigen IR-Modellen zur Verfügung. Da es sich bei *FAMOS* um eine generische Software handelt, besteht die Bibliothek, anders als bei den proprietären Vergleichssystemen, aus IR-Modellen aller relevanten Hersteller. Hierdurch ist es universell einsetzbar. Für jeden IR ist außerdem das entsprechende Kinematikmodell hinterlegt, sodass auch dessen Bewegungen realitätsnah simuliert werden können. Zusätzlich können Computer-Aided-Design (CAD)-Modelle in allen gängigen Formaten importiert und in die Simulation integriert werden, wodurch eine vollständige Roboterzelle virtuell nachgebildet werden kann. Der Hauptfokus der Simulationsumgebung *FAMOS* liegt auf der effektiven Off-Line-Planung und einfachen Manipulation von Roboterpfaden. Einzelne Pfadpunkte können dabei unter anderem mit Koordinaten im Raum sowie auf Modellflächen oder Graphen generiert und anschließend zu einem vollständigen Pfad verbunden werden. Diese

Pfade können anschließend in einem vorgegebenen Intervall segmentiert oder dreidimensional an Vektoren ausgerichtet und dann mit der ausgewählten Roboterkinematik simuliert werden. Dabei wird auch die Interpolation der Roboterposen zwischen zwei Pfadpunkten berücksichtigt. Kommt es auf einem geplanten Roboterpfad zu Kollisionen, werden diese dem Anwender entsprechend signalisiert. Bei *FAMOS* stehen also alle zur Validierung der *Erreichbarkeit* und *Zugänglichkeit* benötigten Werkzeuge zu Verfügung.

Die Bestimmung einer Abdeckung muss auch bei anderen Prozessen im industriellen Umfeld durchgeführt werden. Ein Beispiel hierfür ist die Sprühabdeckung bei einem automatischen Lackierprozess. Auch bei diesem Prozess kann in einer Off-Line-Simulation im Vorfeld überprüft werden, ob sich die bei einer Lackierpose zu lackierende Oberfläche innerhalb einer Geometrie befindet. In diesem Fall geht es allerdings um die abstrahierte Geometrie des Sprühstrahls eines Lackierwerkzeugs und nicht um den Messbereich eines Sensors. Diese Funktionalität ist ebenfalls im Umfang der Simulationsumgebung *FAMOS* enthalten. Allerdings wird diese Überprüfung hier nicht „messbereichsscharf“ durchgeführt. Der Begriff „messbereichsscharf“ bedeutet in diesem Zusammenhang, dass ausschließlich derjenige Teil der Bauteiloberfläche bestimmt und visualisiert werden sollen, der sich innerhalb des Messbereichs befindet (siehe Schema in Abbildung 3.5 (a)). Messbereichsschärfe ist zwingend notwendig, um eine belastbare Aussage über eine vollständige Messbereichsabdeckung treffen zu können. Besteht die Oberfläche eines Modells aus diskreten Elementen wie zum Beispiel Dreiecken, sollen ebenfalls nur die Teile dieser Elemente ermittelt und später visualisiert werden, die innerhalb der Messbereichsgeometrie des entsprechenden Sensors liegen. Dieser Anspruch erhöht den Aufwand bei der Überprüfung erheblich und ist in der *FAMOS*-Methode nicht umgesetzt. Stattdessen werden nur die Elemente bestimmt, die Anteile innerhalb des Messbereichs aufweisen. Zwischen den innen- und außenliegenden Punkten dieser Elemente wird dann im Anschluss interpoliert und es kommt zu einem „Ausfransen“ (siehe Abbildung 3.5 (b)). Dies ist vor allem bei Flächen mit kleiner Krümmung, die im Surface Tessellation Language (STL)-Format durch große Dreiecke repräsentiert werden, ein Problem.

Eine weitere Recherche ergab, dass auch keine andere Simulationsumgebung diese spezielle Funktionalität anbietet. Das Konzept zur Bestimmung der Messbereichsabdeckung in dieser Arbeit sieht deswegen die Implementation einer eigenen Simulation vor. Die abstrahierte Aufgabe dieser Simulation ist die messbereichsscharfe Bestimmung der Oberfläche innerhalb des Messbereichs des WLI-Sensors.



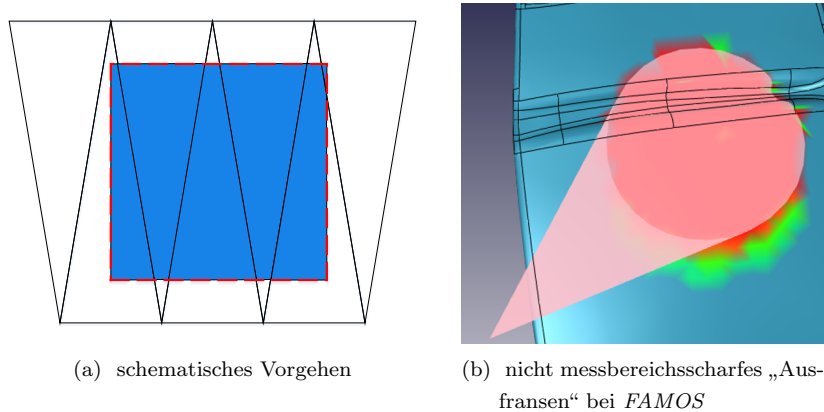


Abbildung 3.5.: Messbereichsscharfe Abdeckungsbestimmung

## 3.2. Reduktion von Schwingungen

Wie schon eingangs mit Bezug auf die Randbedingungen erwähnt, ist das Verfahren der WLI prinzipbedingt sensibel gegenüber Schwingungen. Findet während einer Messung eine Relativbewegung zwischen Messobjekt und Sensor statt, wird das Bestimmen der Interferenzmaxima in der Auswertung erheblich erschwert. Bei der Konzipierung des Aufbaus mit einem ebenfalls für Schwingungen anfälligen IR als Handhabungsgerät müssen also unbedingt sowohl Schwingungen innerhalb des Inspektionssystems als auch von außerhalb des Systems vermieden oder wenigstens auf ein Minimum reduziert werden. Das betrachtete System besteht dabei aus dem ausgewählten IR mit dem WLI-Sensor an dem Roboterflansch und der zusätzlich in den Messaufbau integrierten Drehachse.

### 3.2.1. Schwingungen innerhalb des Systems

Die Integration einer zusätzlichen Drehachse in den Aufbau wirkt sich auf zwei unterschiedliche Weisen positiv auf das Schwingungsverhalten innerhalb des Inspektionssystems aus. Zum einen kann durch die Integration wie vorher herausgearbeitet die grundsätzliche Strategie bei der Inspektion angepasst werden, sodass das Bauteil durch eben diese Achse geführt wird und der IR den Sensor nur zwischen zwei Messringen positionieren und orientieren muss. Der Roboter mit seiner schwingungsanfälligen Kinematik in Form von sechs seriell verketteten Achsen befindet sich also während der Messung in einem ausgeschwungenen, stationären Zustand. Es kann angenommen werden, dass deswegen nur minimal Schwingungen in den eigentlichen Messvorgang induziert werden. Zum anderen konnte bei der Auswahl einer geeigneten Drehachse spezifisch darauf geachtet werden, dass diese entsprechend hohe Beschleunigungswerte aufweist. Aufgrund eines im Forschungsvorhaben geforderten Massenträgheitsmoments konnte nicht auf einen bezüglich Schwingungen günstigen Direktantrieb zurückgegriffen werden. Stattdessen musste ein

gewöhnliche Drehachse mit Übersetzung für das höhere Drehmoment eingesetzt werden. Im Folgenden wird eine Reihe an Versuchen beschrieben, mit denen überprüft werden soll, ob die getroffenen Annahmen bei den ausgewählten Handhabungskomponenten zutreffen und sie für den Einsatz in diesem Aufbau geeignet sind. Die aufgenommenen Daten wurden in *MATLAB* aufbereitet und als Graphen visualisiert.

#### 3.2.1.1. Schwingungen durch den Roboter

In einem ersten Versuch wird mit Hilfe eines Lasertrackers überprüft, ob der ausgewählte IR in unterschiedlichen Betriebszuständen Schwingungen erzeugt, die während einer Messung in den an der Roboterhand montierten WLI-Sensor induziert werden. Dazu wurden entsprechende Tripelspiegel-Targets am Roboter befestigt und deren Position hochfrequent mit 1000 Hz über einen Zeitraum von 10 s aufgenommen (siehe Abbildungen 3.6 und 3.7).

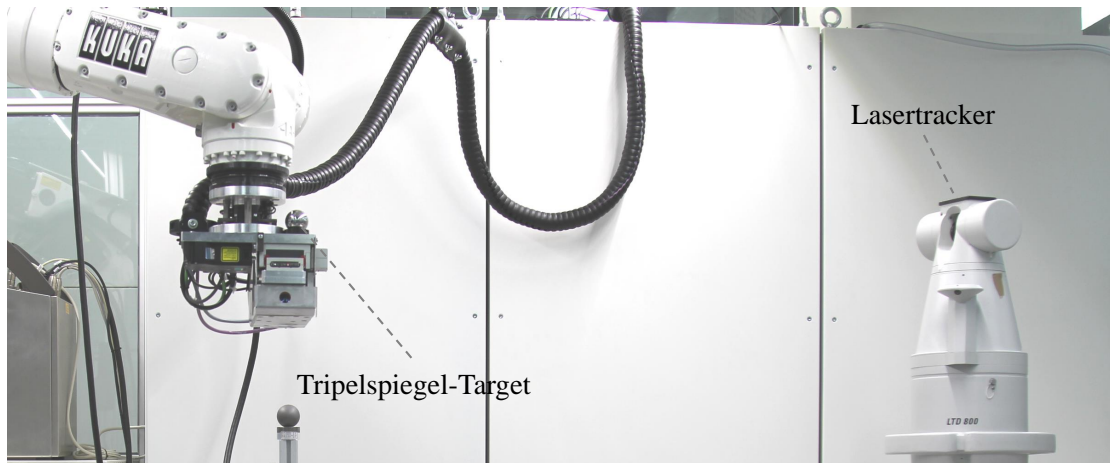


Abbildung 3.6.: Messaufbau mit Lasertracker in Roboterzelle

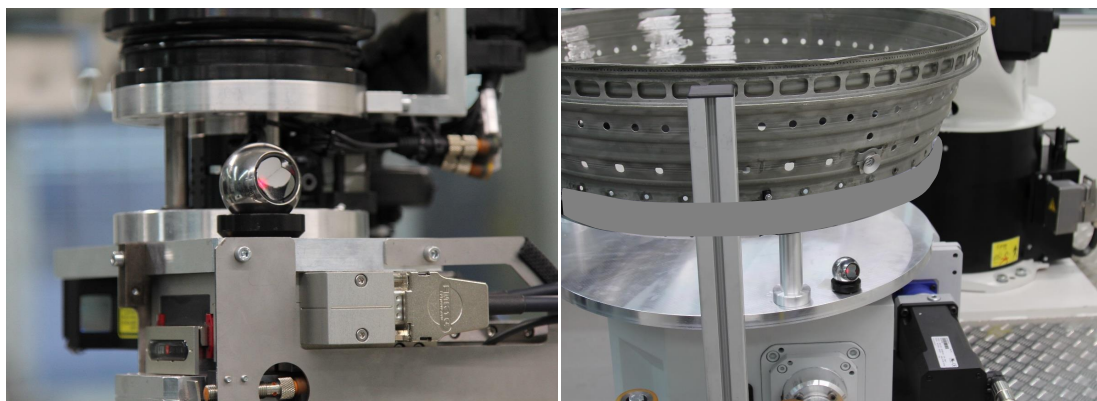


Abbildung 3.7.: Tripelspiegel-Target an Werkzeug (links) und Werkstückhalterung (rechts)

Die verschiedenen Betriebszustände des IR waren zum einen „aktive Bremsen“, bei dem alle sechs Achsen der Roboterkinematik durch Bremsen in Position gehalten werden und zum anderen „in Positionsregelung“, bei der die Position der Achsen über eine Positionsregelschleife gehalten wird. Die folgenden Graphen veranschaulichen die Ergebnisse dieser Messungen.

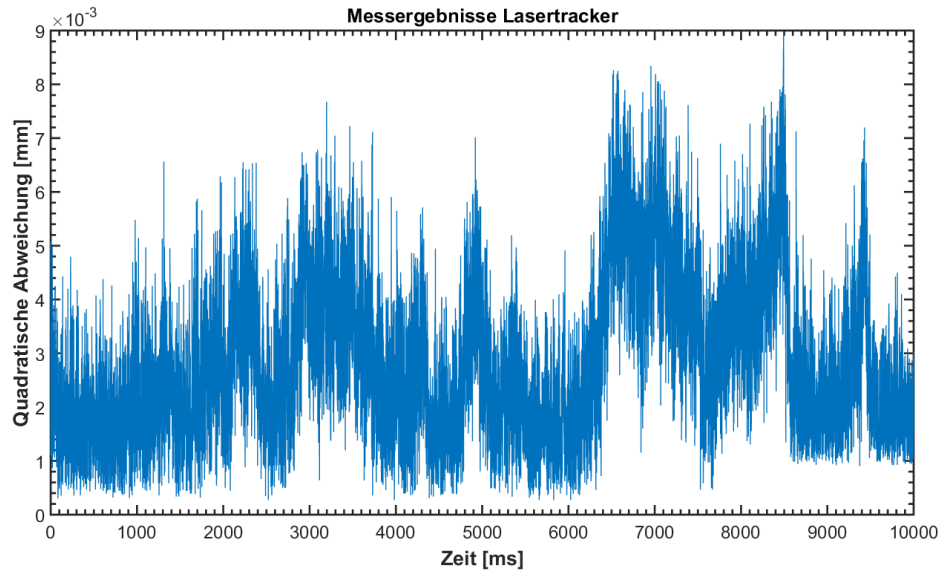


Abbildung 3.8.: Ergebnis der Lasertracker-Messung mit Roboter in Bremse

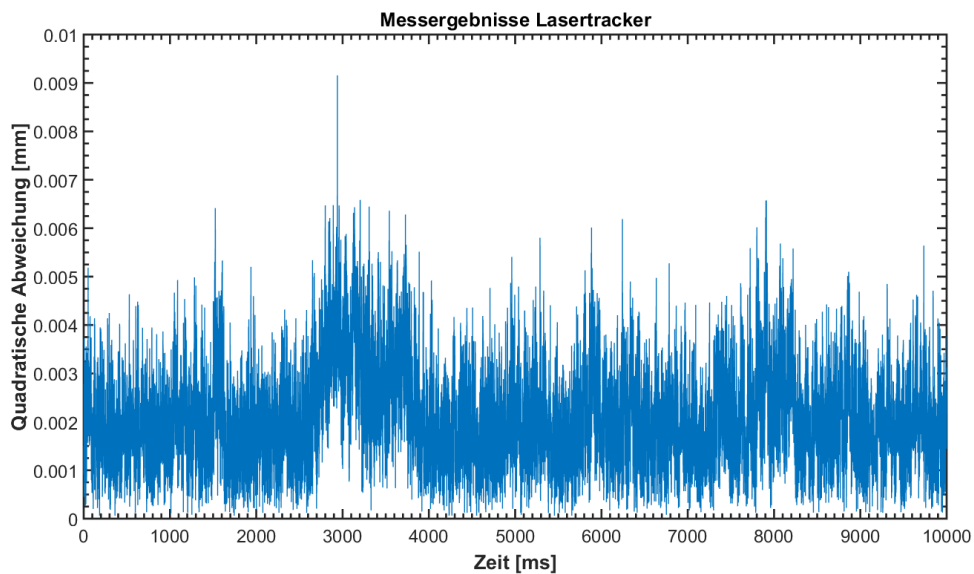


Abbildung 3.9.: Ergebnis der Lasertracker-Messung mit Roboter in Regelung

In beiden betrachteten Betriebszuständen des Roboters variierte die gemessene maximale Schwingungsamplitude in einer Größenordnung von 0.01 mm. Diese Amplitude ist so klein, dass sie im Messrauschen des Lasertrackers untergeht. Der Roboter kann also in beiden betrachteten Betriebszuständen als stationär angenommen und daher als Schwingungsquelle vernachlässigt werden.

#### 3.2.1.2. Schwingungen durch die Drehachse

Auch die zusätzliche Drehachse musste im Rahmen der Anlagenkonzipierung auf Schwingungen untersucht werden. Die Messstrategie in diesem Prozess sieht vor, dass die Relativbewegung zwischen Bauteil und WLI durch die Drehachse realisiert wird. Für eine möglichst geringe Durchlaufzeit ist ihr Geschwindigkeits- und Beschleunigungswert so hoch und der Ruckwert so gering wie möglich gewählt. Schwingungen aus der Verfahrbewegung oder aus dem zugehörigen Antrieb könnten potentiell in das Bauteil einkoppeln und dadurch wiederum den Messvorgang negativ beeinflussen. Zur Bewertung dieser Einflüsse wurden deswegen zwei unterschiedliche Versuche durchgeführt. Zum einen wurde eine Verfahrbewegung der Achse mit Hilfe des Lasertrackers aufgezeichnet und im Anschluss bestimmt, welchen Zeitraum die Achse benötigt, um wieder still zu stehen (siehe Abbildungen 3.10 und 3.11). Zum anderen wurde der WLI-Sensor selbst herangezogen, um potentielle Schwingungen im Bauteil zu identifizieren, die nicht durch den Lasertracker erfasst werden können (siehe Abbildung 3.12). Dazu wurde der Sensor wie im eigentlichen Messvorgang vor das Bauteil platziert und anschließend die Drehachse im Schrittbetrieb um eine vollständige Umdrehung verfahren. Nach jedem Schritt wurde bei dem WLI-Sensor eine Aufnahme ausgelöst, in welcher dann im Anschluss die durchschnittliche Intensität bestimmt wurde. Diese nimmt ab, sobald während seiner Vermessung Relativbewegungen zwischen Messobjekt und Sensor auftreten. Dieser Versuch hat einen eher qualitativen Charakter, da die Intensität in einer Aufnahme nicht nur von Schwingungen abhängt. Die relative Veränderung dieses Wertes auf einem Bauteil ist allerdings ein guter Indikator für auftretende Schwingungen. Die in den Abbildungen angegebenen Achsparameter sind in der folgenden Aufzählung beschrieben:

#### Versuchsparameter

- $v$  = Drehgeschwindigkeit in % der Maximalgeschwindigkeit
- $t$  = Totzeit nach Erreichen des Zielwinkels in ms
- $r$  = Achsenruck in s

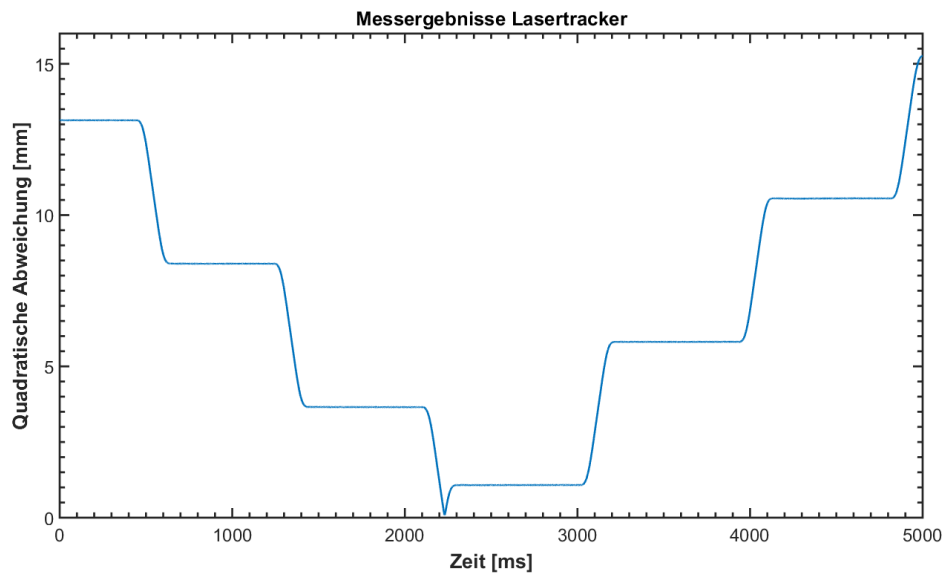


Abbildung 3.10.: Vermessung einer Drehachsbewegung,  $v=100$ ,  $r=0.05$

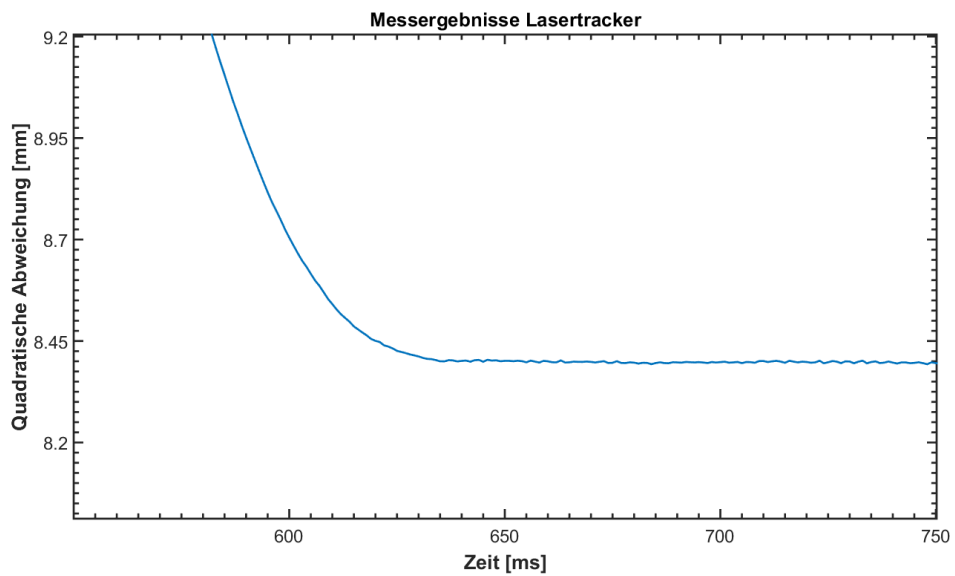


Abbildung 3.11.: Betrachtung des Ausschwingverhaltens

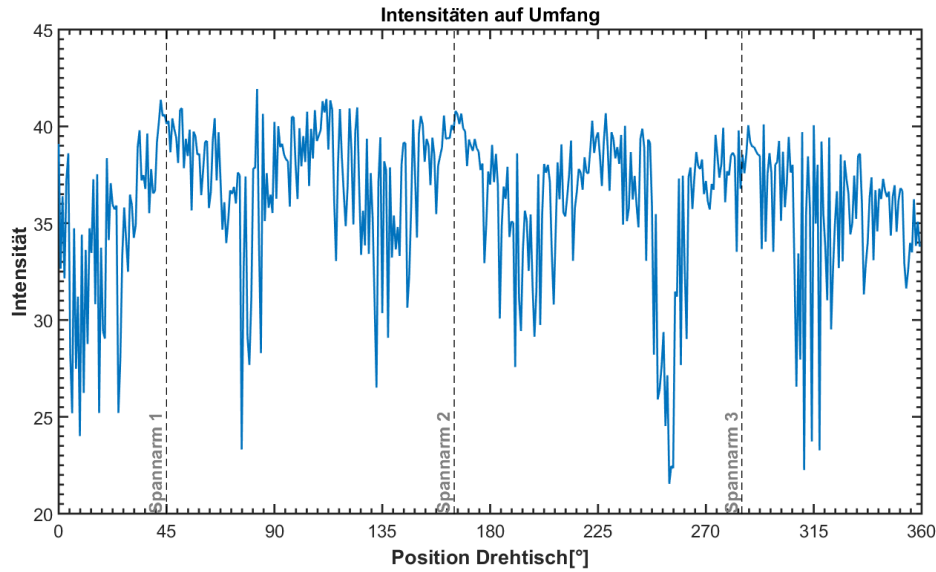


Abbildung 3.12.: Schwingungsmessung der Drehachse mit Hilfe des WLI-Sensors,  $v=100, r=0.05, t=175$

Der Graph in Abbildung 3.10 zeigt sechs Schrittbewegungen der Drehachse. Um prozessähnliche Werte zu erhalten, ist eine Schrittweite von  $0.8^\circ$  gewählt worden, die ungefähr denjenigen im realen Inspektionsprozess entspricht. Nachdem die Achse vor einem Zielwinkel abgebremst wird, zeigt diese Abbildung kein Ausschwingverhalten der Drehachse. Diese Beobachtung bestätigt der Graph in Abbildung 3.11, die einen Ausschnitt des ersten Graphen bei Erreichen eines Zielwinkels vergrößert zeigt. Die Amplitude der verbleibenden Bewegung der Achse liegt innerhalb des Rauschens des Messgerätes und kann daher vernachlässigt werden.

Die Ergebnisse des Messversuchs mit dem WLI-Sensor in Abbildung 3.12 zeigen jedoch, dass das Bauteil durch die Bewegung der Drehachse angeregt wird. Das Bauteil ist in einer Werkstückvorrichtung auf drei Spannarmen aufgenommen und geklemmt. Diese Spannarme liegen jeweils um  $120^\circ$  verdreht zueinander. In dem Ergebnisgraph sind sie als vertikale Linien eingezeichnet. Zwischen diesen Aufnahmepunkten scheint sich eine stehende Welle mit jeweils 2 Maxima auszubilden. An den Stellen der Maxima fällt die Intensität in den Aufnahmen erheblich ab und es kommt zu Fehlmessungen. Als Reaktion auf diese Beobachtung wurde die Werkzeugvorrichtung nochmal konstruktiv überarbeitet. In der neukonstruierten Vorrichtung wird das Bauteil am vollen Umfang geklemmt, wodurch das beobachtete Verhalten sogar bei deutlich reduzierter Totzeit erheblich reduziert werden konnte (siehe Abbildung 3.13). Die höhere Durchschnittsintensität bei diesem Versuch resultiert aus angepassten WLI-Parametern.

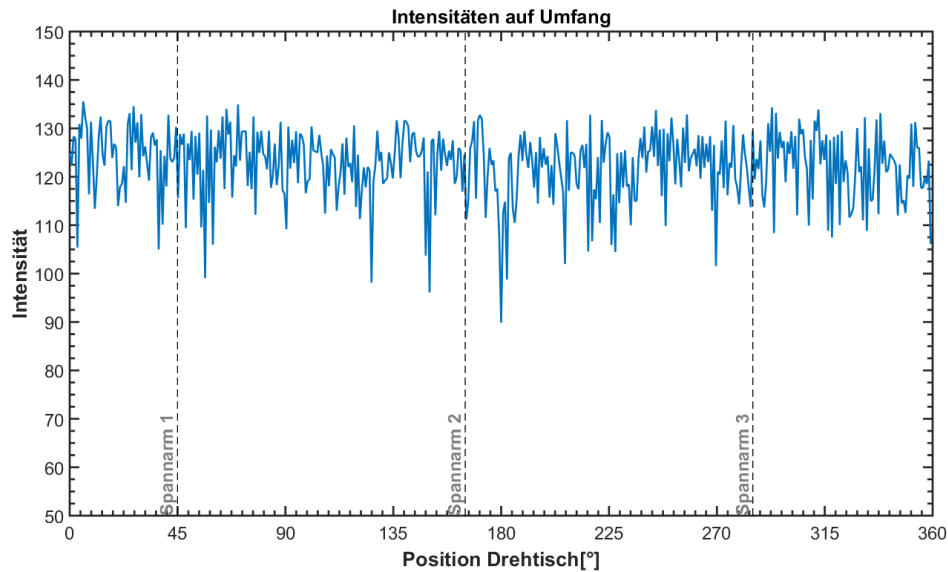


Abbildung 3.13.: Schwingungsmessung der Drehachse mit Hilfe des WLI-Sensors und neuer Spannvorrichtung,  $v=100, r=0.05, t=20$

### 3.2.2. Schwingungen von außerhalb des Systems

Das Inspektionssystem soll nach seiner Fertigstellung in einem Maschinenpark innerhalb einer Werkstattumgebung zum Einsatz kommen. Es muss also unbedingt gewährleistet sein, dass keine Schwingungen von Quellen in der näheren Umgebung den sensiblen Inspektionsaufbau zum Schwingen anregen können. Beispiele für potentielle Quellen in der betrachteten Applikation sind andere Bearbeitungszentren oder der Betrieb von Gabelstaplern oder Hubwagen in unmittelbarer Nähe zum Aufstellort der Anlage. Deswegen werden alle Handhabungskomponenten zusammen auf eine massive Stahlkonstruktion montiert. Diese Konstruktion ist wiederum auf sechs passive Pneumatikdämpfungstöpfe gestellt (siehe Abbildung 3.14).

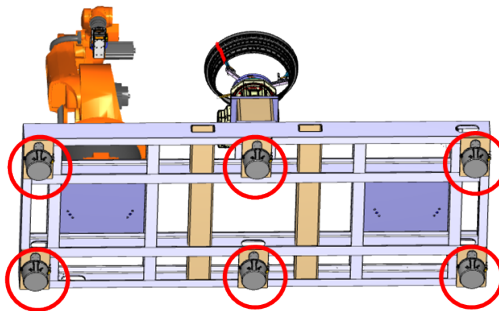


Abbildung 3.14.: Dämpfungstöpfe unter dem Stahlbau der Anlage zur Reduktion der von außen eingebrachten Schwingungen

Die Dämpfungstöpfe weisen laut Herstellerangaben eine vertikale Eigenfrequenz von 2.5 Hz und eine horizontale Eigenfrequenz von 2.8 Hz auf. Um sicherzustellen, dass diese Werte ausreichen, um eine Anregung des Systems durch Schwingungen von außerhalb zu verhindern, wurden in einem praktischen Versuch die jeweiligen Eigenfrequenzen von Roboter und Bauteil auf der Vorrichtung bestimmt. Dazu wurden sie mit einem mechanischen Impuls angeregt. An der jeweiligen Komponente wurde ein Reflektortarget befestigt. Die absolute Position dieses Targets wurde während des Versuches mit einer Messfrequenz von 1000 Hz mit Hilfe eines Lasertrackers aufgezeichnet. Die Auswertung der Daten wurde wiederum mit MATLAB durchgeführt. Dabei wurde das aufgenommene Zeitsignal durch eine Fourier-Transformation in den Frequenzbereich transformiert. In dem Frequenzbereich können die in dem Signal enthaltenen Frequenzen direkt auf der Abszissenachse abgelesen werden. Abbildung 3.15 zeigt das aufgenommene Zeitsignal für die Drehachse. Abbildungen 3.16 und 3.17 zeigen die entsprechenden Frequenzspektren, aus denen die Eigenfrequenzen  $f_R = 9.5$  Hz des Roboters und  $f_R = 26.73$  Hz von dem in der entsprechenden Werkstückhalterung eingespannten Bauteil abgelesen werden können. Beide Eigenfrequenzen liegen deutlich oberhalb der Eigenfrequenz der Dämpfungselemente. Das bedeutet, dass diese Komponenten durch die Pneumatikdämpfer vor Anregungen von Schwingungsquellen außerhalb des Systems geschützt werden und sie eine geeignete Entkopplung darstellen.

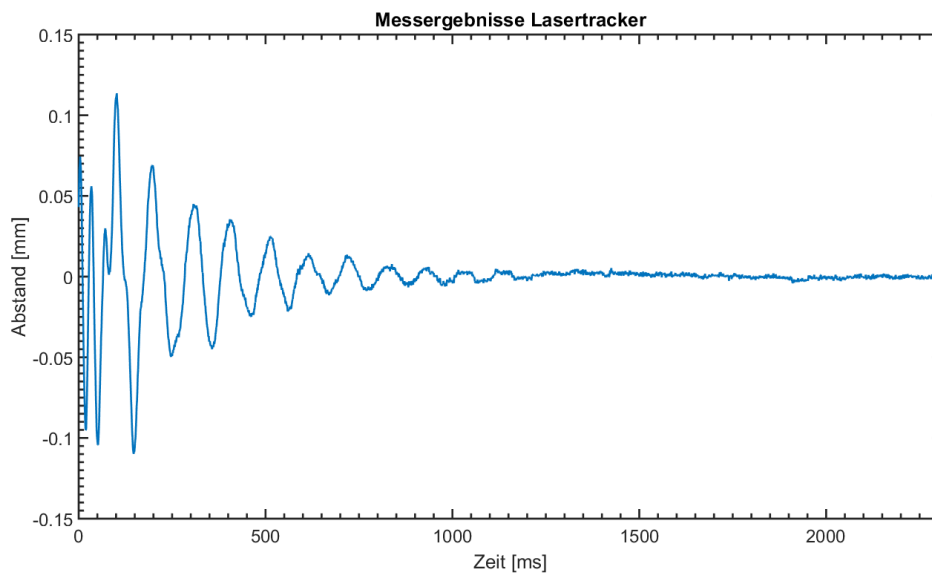


Abbildung 3.15.: Zeitsignal des Roboters nach Anregung durch einen mechanischen Impuls



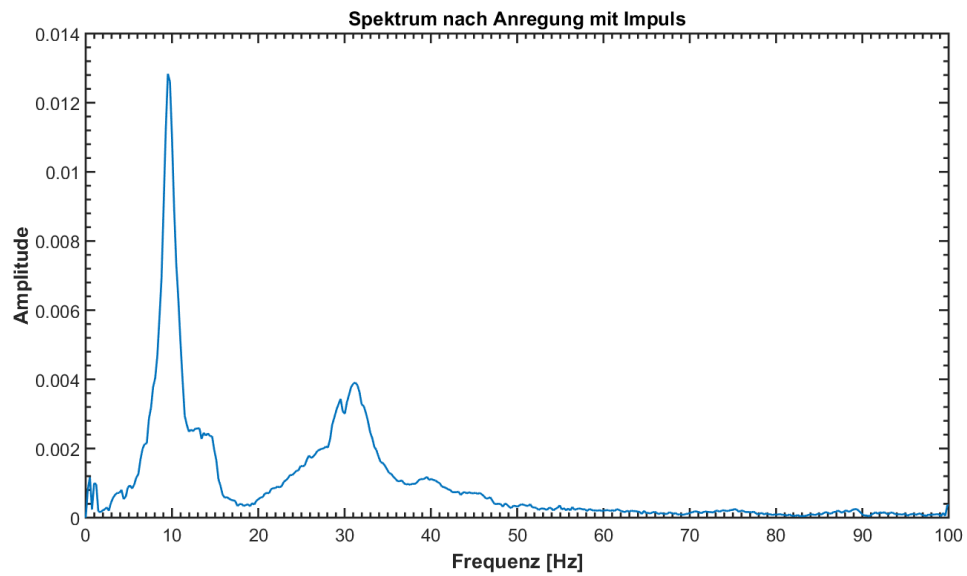


Abbildung 3.16.: Frequenzspektrum des Zeitsignals nach Anregung der Drehachse durch einen mechanischen Impuls

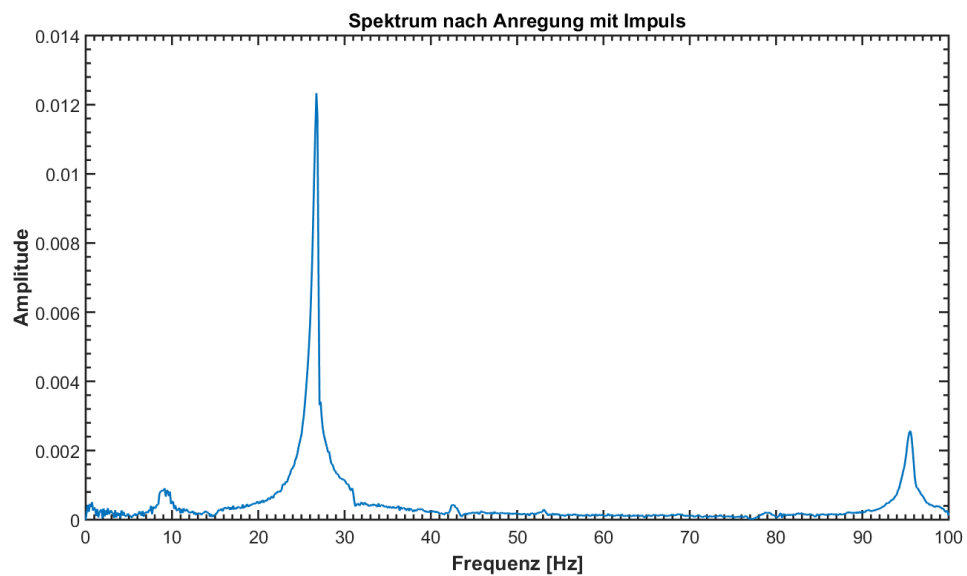


Abbildung 3.17.: Frequenzspektrum des Zeitsignals nach Anregung des Roboters durch einen mechanischen Impuls

### 3.3. Zusätzliche Sensorik

Eine weitere Anforderung ist die Möglichkeit zur schnellen Akquirierung der Bauteilgeometrie für die anschließende individuelle Adaption des jeweiligen Inspektionsprogramms des Roboters. Aktuell ist eine ganze Reihe von optisch messenden Verfahren zur Akquirierung von dreidimensionalen Oberflächendaten erhältlich, die noch immer Bestandteil aktueller Forschung ist. Zur Übersicht werden die wichtigsten Verfahren kurz vorgestellt, miteinander verglichen und danach das für diesen Anwendungsfall optimal passende ausgewählt.

#### 3.3.1. Optisch messende 3D-Verfahren

Schwarte zeigt in der Abbildung 3.18 (SHM00) eine Übersicht der relevanten zerstörungsfreien Prüfverfahren zur Oberflächeninspektion. Die Vor- und Nachteile dieser unterschiedlichen Verfahren wurden schon in anderen Arbeiten vorgestellt (BPF12, STD09), weswegen darauf an dieser Stelle verzichtet wird. Stattdessen werden für diesen vorgelagerten Prozessschritt die Verfahren mit sichtbarem Licht in den Fokus gestellt, das durch eine Wellenlänge von  $\lambda = 0.5 - 1 \mu\text{m}$  charakterisiert ist.

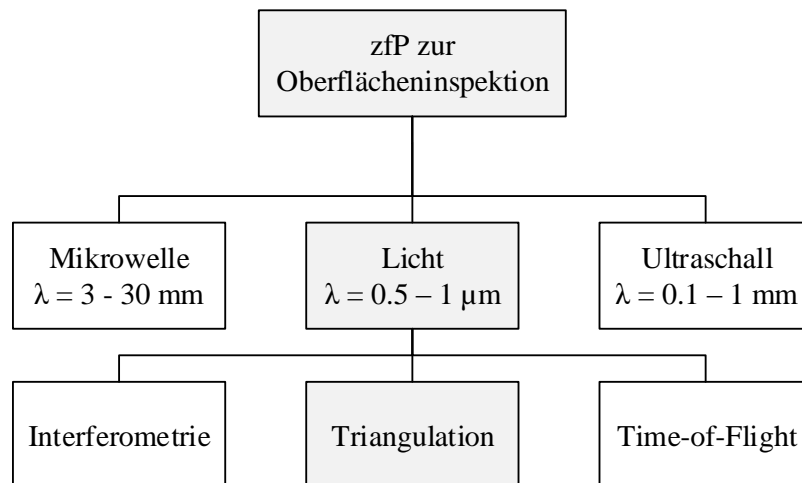


Abbildung 3.18.: Übersicht der zerstörungsfreien Prüfverfahren zur Oberflächenvermessung

Diese spezielle Gruppe ist in drei Verfahren aufgegliedert: die interferometrischen Verfahren, die Interferenzen von kohärenten Licht zur Abstandsmessung verwenden, die Time-of-Flight (TOF)-Verfahren, bei denen über die Laufzeit von Impulsen ein Abstand gemessen wird und die Triangulationsverfahren, bei denen über geometrische Zusammenhänge ein Abstand zu einem Objekt bestimmt wird. Im nächsten Abschnitt werden ausschließlich die unterschiedlichen Triangulationsverfahren vorgestellt und hinsichtlich der Anforderungen aus dem Prozess des neuartigen

Inspektionssysteme bewertet. Die TOF-Verfahren werden ausgeschlossen, weil sie eine unzureichende Genauigkeit aufweisen und deswegen eher für Abstandsmessungen bei mittleren und langen Distanzen geeignet sind (BH88) und die interferometrischen Verfahren werden ausgeschlossen, da sie schon für die anschließende Inspektion eingesetzt werden und ein schnelleres Verfahren zur Akquirierung von Oberflächeninformationen gesucht wird. Schwarte zeigt in einer Veröffentlichung, welche die relevanten Verfahren bei der Triangulation sind und untergliedert diese wie in Abbildung 3.19 in passive und aktive Verfahren (SHM00).

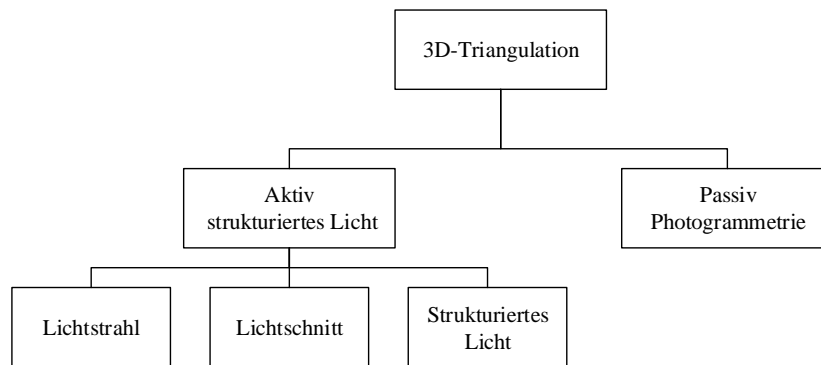


Abbildung 3.19.: Übersicht der 3D-Triangulationsverfahren

### 3.3.2. Lichtstrahlverfahren

Die Triangulation ist ein Verfahren zur Entfernungsmessung. Es beruht auf der Geometrie von ebenen Dreiecken<sup>1</sup>. In der Stereokonfiguration, bei der Laserquelle und Detektor räumlich voneinander getrennt sind, wird ein Laserstrahl von einer Quelle auf die zu vermessende Oberfläche geworfen und von dieser teilweise zu einem Detektor gestreut. Damit ein Teil des Lichtes den Detektor erreicht, ist es wichtig, dass die beleuchtete Oberfläche weder das gesamte Licht reflektiert noch vollständig absorbiert. Für ein scharfes Bild muss bei dem Aufbau außerdem die Scheimpflug-Bedingung eingehalten werden, die besagt, dass sich Laserstrahl und Bildebene im Detektor in der Objektebene schneiden müssen. In der Konfiguration wie in Abbildung 3.20 sind durch Vermessen oder Kalibrieren die Brennweite des eingesetzten Objektivs  $f$ , der Schnittpunkt von Laserebene und Bildsensorebene  $l$  und der Verkippwinkel von Bildebene zu Objektebene  $\alpha$  bekannt. Die Triangulationsbasis  $b$  lässt sich deswegen durch einsetzen dieser Werte in die Formel 3.1 berechnen (BPF12).

$$b = x \cos \phi. \quad (3.1)$$

<sup>1</sup>Die Innenwinkel eines Dreiecks in einer Ebene summieren sich zu  $180^\circ$

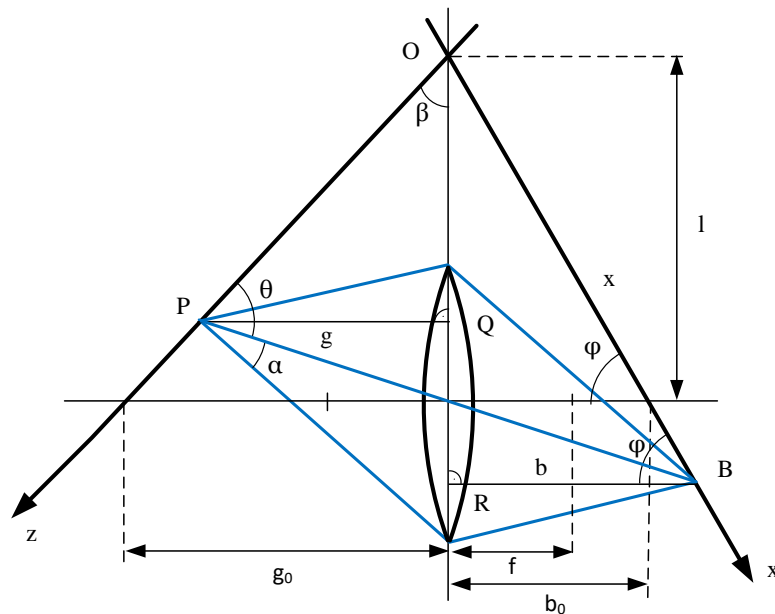


Abbildung 3.20.: Geometrischer Aufbau beim Triangulationsverfahren (BPF12)

### 3.3.3. Lichtschnittverfahren

Das Lichtschnittverfahren basiert ebenfalls auf dem Verfahren der Triangulation. Dabei wird jedoch nicht nur die Entfernung eines Punktes auf der Oberfläche des Objektes berechnet, sondern die Entfernung einer ganzen Linie. Diese wird im klassischen Aufbau durch Auffächerung eines Lasers durch zum Beispiel eine Zylinderlinse erzeugt und auf die Oberfläche geworfen und von ihr wieder gestreut (siehe Abbildung 3.21).

Ein Teil des gestreuten Lichtes trifft wiederum auf den Detektor, der in diesem Fall aus einer bidimensionalen Flächenkamera besteht. Auflösung und Messbereich der Linie sind dabei abhängig von dem Aperturwinkel zwischen Laser und Detektor und der Auflösung des Detektors. Aus der Position eines Lichtpunktes in der jeweiligen Zeile kann dann ein zweidimensionales Höhenprofil der Oberfläche berechnet werden.

### 3.3.4. Streifenprojektion

Während sich mit dem Lichtstrahlverfahren nur ein Punkt im Raum und bei dem Laserlichtschnittverfahren eine 2D-Kontur auf der Bauteiloberfläche bestimmen lässt, kann bei der Streifenprojektion die ganze Bauteiloberfläche innerhalb eines Volumens vermessen werden. Dazu wird ein Streifenmuster auf die Oberfläche projiziert, dessen Streuung dann wiederum von einem bidimensionalen Detektor in einer Kamera aufgenommen wird. Die Aufgabe ist hierbei die Zuord-

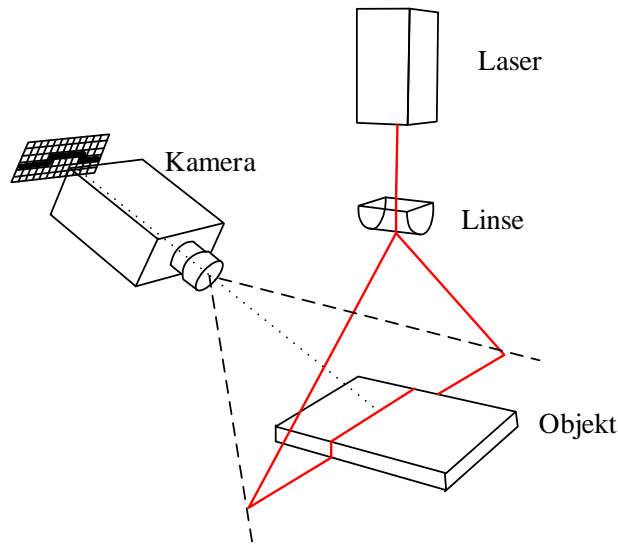


Abbildung 3.21.: Aufbau beim Laserlichtschnittverfahren (BRB88)

nung einer bestimmten Linie in unterschiedlichen Aufnahmen. Deswegen werden in diesem Fall inhomogene Streifenmuster verwendet, die in allen Kamerabildern eindeutig bestimmt werden können. Die Scheimpflug-Bedingung ist bei diesem Verfahren nur für eine bestimmte Ebene im Bild erfüllt, sodass nicht das ganze Bild scharf dargestellt werden kann, was wiederum zu einer Einschränkung in der Genauigkeit entsprechend der Schärfentiefe führt.

(BPF12) stellt die Projektion des Muster  $b(x)$  auf das Höhenrelief  $z(x)$  durch ein Signalmodell, wie in Abbildung 3.22 zu sehen, dar. Auf der Kamera erhält man das Signal  $b_{\text{zerr}}(x)$  aus dem sich dann das reale Höhenrelief  $z'(x)$  in einem Bildauswertungsschritt zurückrechnen läßt. Die

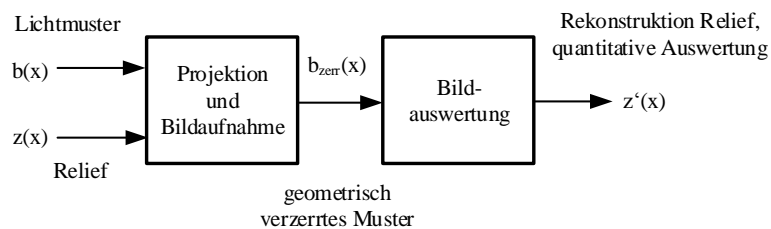


Abbildung 3.22.: Signalmodell der Streifenprojektion (BPF12)

eindeutige Zuordnung eines Bildpunktes zu einem Projektionspunkt kann zum Beispiel erreicht werden, indem das Objekt in mehreren Belichtungen entsprechend dem Graycode codiert wird.

„Beleuchtet“ bedeutet binär „1“ und „nicht beleuchtet“ binär „0“. Das  $m - te$  Bild entspricht dann dem  $m - ten$  Bit der entsprechenden Codierung. Jedes Pixel kann dann im Anschluss eindeutig einer projizierten Linie zugeordnet werden. Dadurch ist das zur Triangulation benötigte Dreieck und die Triangulationsbasis definiert und es kann eine Entfernung errechnet werden.

### 3.3.5. Photogrammetrie

Die Photogrammetrie ist ein passives Triangulationsverfahren und arbeitet mit einer beliebigen Anzahl unstrukturierter Lichtquellen. Das von dem Objekt reflektierte Licht wird dabei mit mindestens einer Zeilenkamera detektiert. Wird tatsächlich nur eine Kamera verwendet, muss diese auf verschiedenen Messposen positioniert werden. Ein Punkt kann nämlich nur dann im 3D-Raum bestimmt werden, wenn er eindeutig in zwei Aufnahmen identifiziert werden kann. Hierbei können entweder aufgebrachte oder eindeutige, objekteneigene Merkmale verwendet werden. Aufgrund der Rotationssymmetrie und dem sich immer wiederholenden Muster auf dem Bauteilumfang in dem betrachteten Anwendungsfall ist es schwer, zwei Aufnahmen miteinander zu registrieren. Da die Photogrammetrie in diesem Anwendungsfall deshalb nur schwer zu realisieren ist, scheidet sie in der weiteren Betrachtung aus.

### 3.3.6. Die Verfahren im Vergleich

Nach Ausschluss aller Verfahren, die nicht im sichtbaren Bereich von Licht operieren sowie des photogrammetrischen Triangulationsverfahrens, verbleiben für den entscheidenden Vergleich nur noch die Triangulationsverfahren mit aktiver Beleuchtung. Diese sind die weiter oben detailliert beschriebenen Lichtstrahl-, Lichtschnitt- und Streifenprojektionsverfahren. Bei der Entscheidung für eines der Verfahren wird die Nutzwertanalyse (engl. benefit-, utility- oder worthanalysis) angewendet, die bei der Analyse von mehrdimensionalen Entscheidungsprozessen schon lange etabliert ist (Zan14). Die Zielgröße in dem vorliegenden Fall ist ein Verfahren zur Digitalisierung der Bauteiloberfläche. Das Zielsystem besteht aus Zielkriterien, die aus den Anforderungen an den zusätzlichen Sensor abgeleitet und im Folgenden zusammen mit ihrem subjektiven Gewicht  $g$  aufgeführt sind. Bei der Auswahl der Kriterien wurde darauf geachtet, dass sie möglichst wenig voneinander abhängen. Die Bewertung der Verfahren hinsichtlich der einzelnen Kriterien orientiert sich an dem Rangordnungsverfahren. Dieses ist für die betrachtete Zielgröße passend, da nur eine kleine Anzahl von Verfahren betrachtet werden. Tabelle 3.3 zeigt die aus der Bewertung resultierende Zielertragsmatrix.

**Gewicht des Sensors ( $k_1, g = 0.1$ ):** Das Gewicht des Sensors muss aufgrund des begrenzten Handhabungsgewichtes des selektierten IR betrachtet werden. In das Gewicht gehen auch benötigte Peripheriegeräte wie Leitungen und Beleuchtung mit ein. Weniger Gewicht an der Roboterhand führt zu einem besseren dynamischen Verfahrenverhalten.

**Kosten** ( $k_2, g = 0.2$ ): Die Kosten der Sensorik müssen ebenfalls als Zielkriterium berücksichtigt werden. Ein preiswerter Sensor führt bei der Markteinführung im Anschluss an die Forschungs- und Entwicklungsphase zu einem günstigeren Produkt.

**Durchlaufzeit** ( $k_3, g = 0.2$ ): Eine möglichst geringe Durchlaufzeit ist ein globales Ziel bei der Konzipierung dieses Systems. Deswegen muss es auch in der Analyse des Nutzwertes eines Verfahrens zur Digitalisierung betrachtet werden.

**Zugänglichkeit** ( $k_4, g = 0.3$ ): Die Relevanz der Zugänglichkeit als Kriterium bei der Digitalisierung leitet sich aus der Vollständigkeit bei der Inspektion ab. Diese findet auch als globales Ziel an dieser Stelle Berücksichtigung.

**Auflösung** ( $k_5, g = 0.1$ ): Eine hohe Auflösung ist auch bei dem Schritt der Digitalisierung ein zu beachtendes Kriterium. Wie früher beschrieben, führt eine hohe Auflösung auch immer zu einem kleinen Messbereich, weswegen dieses Kriterium nicht vollständig unabhängig vom Kriterium Durchlaufzeit ist.

	Zielkriterien				
Verfahren	$k_1$	$k_2$	$k_3$	$k_4$	$k_5$
Lichtschnitt	2	3	2	2	2
Laserstrahl	3	1	1	3	1
Streifenprojektion	1	2	3	1	3

Tabelle 3.2.: Zielerträge der einzelnen Verfahren hinsichtlich der formulierten Zielkriterien

	Gewichte $k_1 - k_5$				
Verfahren	0.1	0.2	0.2	0.3	0.1
Lichtschnitt	0.2	0.6	0.4	0.6	0.2
Laserstrahl	0.3	0.2	0.2	0.9	0.1
Streifenprojektion	0.1	0.4	0.6	0.3	0.3

Tabelle 3.3.: Gewichtete Zielwerte der einzelnen Verfahren

Verfahren	Nutzwert
Lichtschnitt	<b>2.0</b>
Laserstrahl	1.7
Streifenprojektion	1.7

Tabelle 3.4.: Nutzwerte der betrachteten Verfahren

Aus den aufsummierten Zielwerten ist ersichtlich, dass das Lichtschnittverfahren als Verfahren zur Digitalisierung der Bauteiloberfläche am besten geeignet ist. Sensoren beim Lichtstrahlverfahren sind zwar günstiger und leichter, weisen aber aufgrund der eindimensionalen Messung eine schlechte Auflösung und die höchste Durchlaufzeit auf. Die Streifenprojektion bietet zwar den größten Messbereich der Sensoren mit aktiver Beleuchtung, kostet allerdings deutlich mehr als Sensoren aus den anderen beiden Kategorien. Außerdem wäre die Zugänglichkeit bei den betrachteten Bauteilen nur mit einer zusätzlichen Spiegeloptik möglich, deren Entwicklung zu einem erheblichen Mehraufwand führen würde. Auch die Durchlaufzeit ist bei diesem Verfahren vergleichbar hoch, da vor dem Prozess durch einen Werkstattmitarbeiter Marker auf das zu digitalisierende Bauteil aufgebracht und später wieder entfernt werden müssen. Die Integration des Laserlichtschnitt (LLS)-Sensors in das Zielsystem wird im Folgenden vorgestellt.

### 3.3.7. Integration des Sensors in das Inspektionssystem

Die Adaption des Roboterpfades für die anschließende Inspektion mit dem WLI-Sensor wird auf dem vorher vom LLS-Sensor akquirierten Punktmodell durchgeführt. Aufgrund der Aufteilung von Digitalisierung und Inspektion auf zwei unterschiedliche Sensoren muss der Einfluss der Positioniergenauigkeit des IR betrachtet werden. Damit dieser Einfluss so gering wie möglich gehalten werden kann, sollte sichergestellt sein, dass diese beiden Prozesse an den selben Positionen im Arbeitsraum und dadurch in ähnlichen Achskonfigurationen des IR ausgeführt werden und sich die Positioniergenauigkeit des Roboters dadurch seiner deutlich besseren Wiederholgenauigkeit annähert. Bei einem Standard-IR liegt die Positioniergenauigkeit bei etwa 1 mm. Im Vergleich dazu wird die Wiederholgenauigkeit der genauen HA-Variante des eingesetzten Robotermodells vom Hersteller mit 0.05 mm angegeben. Eine ähnliche Achskonfiguration bei beiden Prozessen wird wiederum erreicht, indem die beiden Werkzeugspitzenkoordinatensysteme der Sensoren so nah wie möglich beieinanderliegen. In dem betrachteten Aufbau wurden die Werkzeuge aus diesem Grund wie in Abbildung 3.23 veranschaulicht an eine gemeinsame Werkzeughalterung montiert.

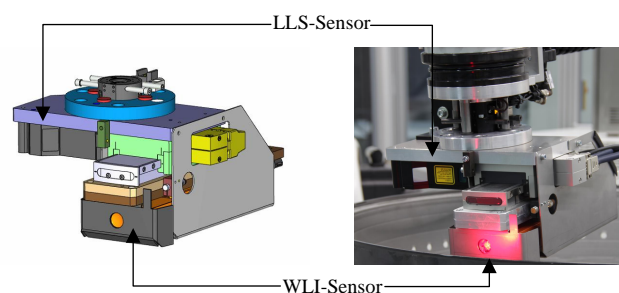


Abbildung 3.23.: Gemeinsame Werkzeughalterung von WLI- und LLS-Sensor am Roboterflansch



## 4. Konzipierung des Inspektionsprozesses

Im folgenden Kapitel wird das grundsätzliche Vorgehen bei der Rissinspektion vorgestellt. Bei den betrachteten Bauteilen werden neben dem eigentlichen Inspektionsschritt mit dem hoch auflösenden WLI-Sensor zusätzliche Prozessschritte benötigt, um deren individuelle Geometrie zu berücksichtigen. Die zusätzlichen Schritte werden aus der übergeordneten Problemstellung abgeleitet und zu einem Gesamtprozess miteinander verknüpft. Zur besseren Übersicht beschränkt sich dieses Kapitel vorerst auf die Konzipierung grundlegender Lösungsansätze für jeden Prozessschritt. Die detaillierte Ausgestaltung der verwendeten Ansätze in mathematische Algorithmen folgt dann anschließend in Kapitel 6.

### 4.1. Übergeordnete Prozessstruktur

Bei der Identifizierung notwendiger Prozessschritte für die Inspektion wird Top-down vorgegangen. Es wird mit dem letzten Teilschritt begonnen und von diesem im Folgenden dann der jeweilige zur Ausführung benötigte Vorgänger bestimmt.

Der letzte Schritt der automatischen Rissinspektion ist die Rissdetektion in einer Bildverarbeitung, die nicht zur Aufgabenstellung dieser Arbeit gehört. Damit in diesem Schritt Risse detektiert werden können, müssen zuvor in dem eigentlichen Inspektionsschritt mit einem IR-geführten Weißlichtinterferometer die entsprechenden hochauflösten Oberflächendaten der Bauteile aufgenommen werden. Da diese Bauteile verformt sind und Beschädigungen aufweisen, muss vor der Ausführung dieser automatischen Inspektion in einer Pfadplanung für jedes einzelne Bauteil ein individueller Inspektionspfad geplant werden. Für diesen offline ausgeführten Prozessschritt wird wiederum als Eingangsinformation die entsprechende IST-Oberfläche benötigt. Diese muss in einem weiteren vorgelagerten Digitalisierungsschritt mit dem zusätzlich an dem Roboter manipulator montierten LLS-Sensor aufgenommen werden. Damit die Daten von dieser Digitalisierung und der nachfolgenden Inspektion in ein gemeinsames Referenzkoordinatensystem transformiert und zu einem Modell registriert werden können, muss zu Beginn jeder Inspektion eine Kalibrierung der zusätzlichen Drehachse sowie der virtuellen Tool Center Points (Tool Center Point (TCP)) der verwendeten Werkzeuge durchgeführt werden. Rüstschrte für die Bauteile zu Anfang des Prozesses werden in dieser Betrachtung vernachlässigt. Wie in Abbildung 4.1 veranschaulicht, besteht der gesamte Inspektionsprozess aus den genannten fünf Teilprozessen, von denen allerdings nur die ersten vier Gegenstand dieser Arbeit sind.

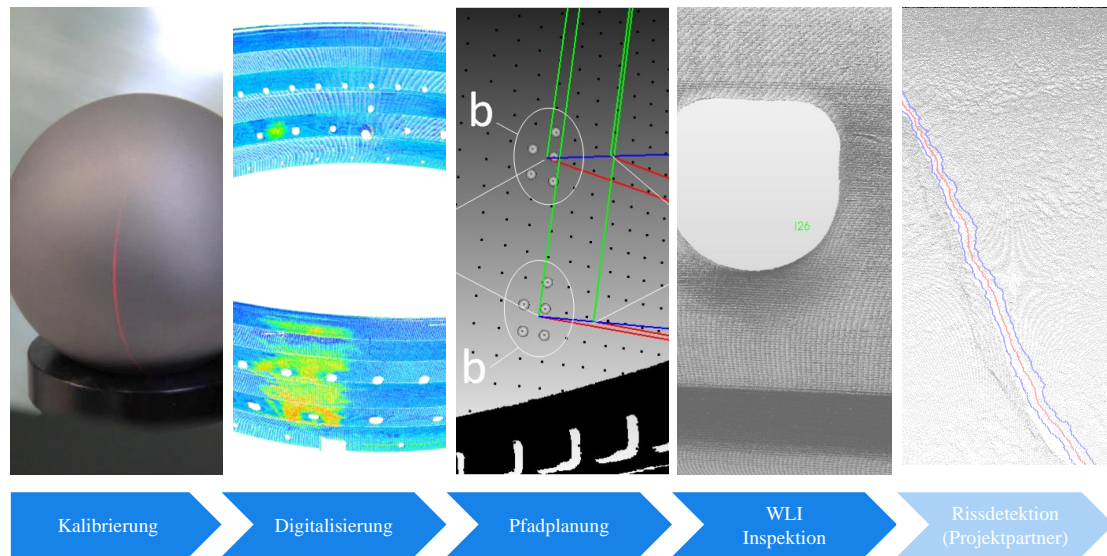


Abbildung 4.1.: Übersicht der notwendigen Subprozesse im Inspektionsprozess und deren Anordnung

## 4.2. Kalibrierung des Inspektionssystems

Damit die mit den zwei unterschiedlichen Sensoren akquirierten Daten der Bauteiloberfläche zusammen in einem gemeinsamen Werkstück Koordinatensystem (KS)  $K_W$  zu einem Modell vereinigt werden können, müssen alle Teiltransformationen der in Abbildung 4.2 dargestellten verketteten Transformation bekannt sein. Die Transformation  ${}^F_R T$  von dem Roboter-KS  $K_R$  zu seinem Handgelenksflansch-KS  $K_F$  wird über die aktuellen Parameter der Kinematik von der Robotersteuerung berechnet und bereitgestellt. In einer Kalibrierung bestimmt werden müssen also noch die Transformationen  ${}^{TCP}_F T$  von  $K_F$  zu den virtuellen KS der Werkzeugspitzen (TCP) der beiden eingesetzten Sensoren und die Transformation  ${}^W_R T$  von  $K_R$  zu  $K_W$ . Die einzelnen Roboterachsen sowie die geometrischen und dynamischen Parameter des eingesetzten IR werden in dieser Arbeit als kalibriert angenommen.

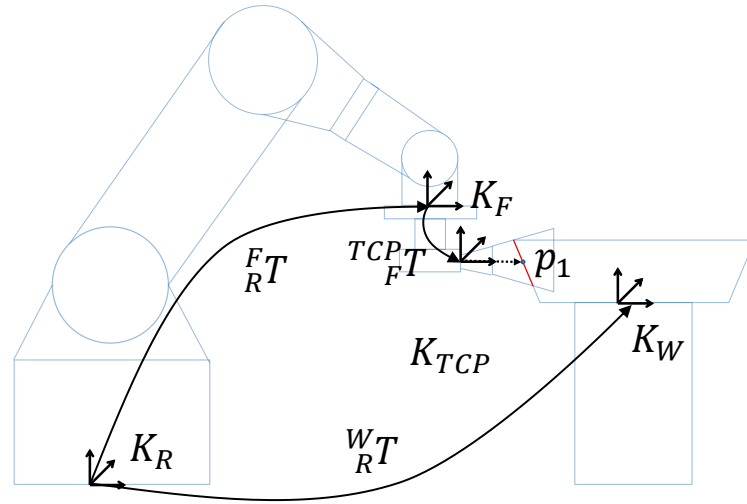


Abbildung 4.2.: Verkettete Transformation im Inspektionssystem

#### 4.2.1. Kalibrierung der Werkzeugspitze ( ${}^{TCP}_R T$ )

Beyer stellt in seiner Arbeit zur Genauigkeitssteigerung von einem IR, Kalibrierverfahren zur Bestimmung von Werkzeugspitzen vor. Als das genaueste Verfahren sieht er eine Vermessung mit einer hoch genauen Koordinatenmessmaschine (CMM) oder einem vergleichbaren oder sogar genaueren Messgerät (Bey05). Für einen automatischen Prozess mit vielen Werkzeugwechseln ist der Aufwand zur Vermessung des TCP auf einer externen CMM zu hoch, was eine unnötige Verlängerung der Durchlaufzeit bedeuten würde. Aus diesem Grund muss für das System in dieser Arbeit ein anderes Verfahren herangezogen werden. Eine schon lang etablierte Methode ist die sogenannte Spitze-auf-Spitze Methode. Hierbei wird ein Fixpunkt im Raum in Form einer Mess-Spitze aus mindestens drei Richtungen mit dem zu kalibrierenden TCP angefahren. Aus den unterschiedlichen Positions- und Orientierungsdaten an diesen Messpunkten können durch Gleichsetzen mehrerer aus der Kinematik abgeleiteter Gleichungen drei der notwendigen sechs Freiheitsgrade bestimmt werden. Hierbei handelt es sich um die translatorischen Freiheitsgrade. Beyer weist darauf hin, dass zur Feinpositionierung des Werkzeugs durch den Roboter anstatt des menschlichen Auges auch eine Kamera eingesetzt werden kann, weil diese eine höhere Auflösung bietet. Die verbleibenden drei rotatorischen Freiheitsgrade der Werkzeugspitze können dann beispielsweise anhand einer bekannten Pose im Raum ausgerichtet werden.

Auch bei optischen Sensoren kann das beschriebene Verfahren zur Bestimmung eines TCP über die Betrachtung der kinematischen Kette angewendet werden. Bei den optischen Sensoren wird allerdings mit dem TCP keine Spitze angefahren, da es sich um eine virtuelle Werkzeugspitze an der Nullposition des entsprechenden Sensor-Messbereichs handelt. Stattdessen wird ein Fixpunkt in den Sensordaten gesucht. Zieht man als Kalibriertarget eine Kugel heran, kann in den Sensor-

aufnahmen dieser Kugel zum Beispiel das zugehörige Kugelzentrum als entsprechender Fixpunkt bestimmt werden. Gan schlägt hierbei eine zweistufige Methode vor, bei der zuerst die rotatorischen Anteile des TCP und im Anschluss die translatorischen ermittelt werden (GT11). Von dieser Methode existieren zwei unterschiedliche Varianten. Bei der ersten von Gan vorgestellten Variante führt der Roboter das Target und der Sensor ist in der Roboterzelle fest installiert. Diese Methode wird „eye-to-hand“-Methode genannt. Bei der zweiten Methode führt der Roboter den Sensor und das Target ist entsprechend in der Zelle fest installiert (siehe Abbildung 4.3). Die zweite Variante wird als „eye-in-hand“-Methode bezeichnet. Da sie dem Aufbau in dieser Aufgabenstellung entspricht, wird sie als passendes Konzept herangezogen.

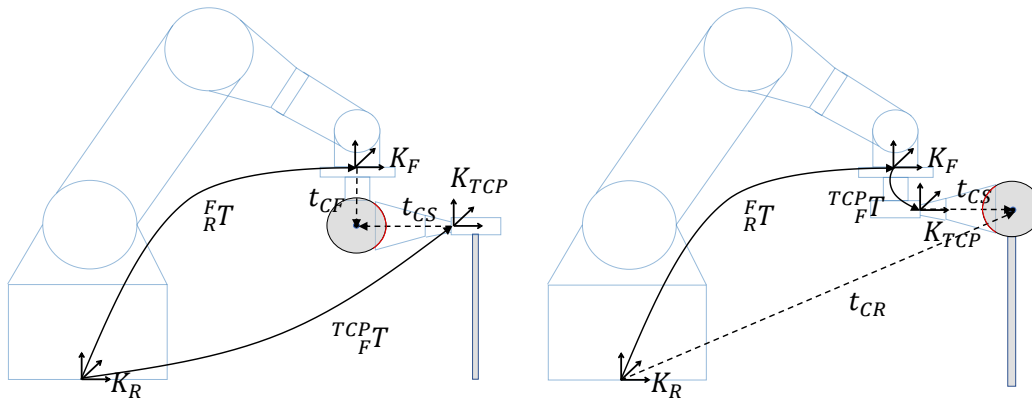


Abbildung 4.3.: Varianten zur Kalibrierung eines Scanner TCP's nach Gan.2011:  
links „eye-to-hand“, rechts „eye-in-hand“

#### 4.2.2. Kalibrierung der Drehachsbasis ( ${}^W_R T$ )

Die Drehachsbasis ist das Koordinatensystem der zusätzlichen Drehachse im Inspektionsaufbau. Alle aufgenommenen Daten des Bauteils, das von dieser zusätzliche Drehachse positioniert wird, werden in dieses KS transformiert. Erst durch diesen Transformationsschritt werden die Daten der Oberfläche auch in einem gemeinsamen KS abgebildet. Die Position des KS kann frei auf dieser Achse definiert werden. In dem betrachteten Fall wurde die Oberfläche der Werkstückaufnahme als Position für das KS ausgewählt (siehe Abbildung 4.4).

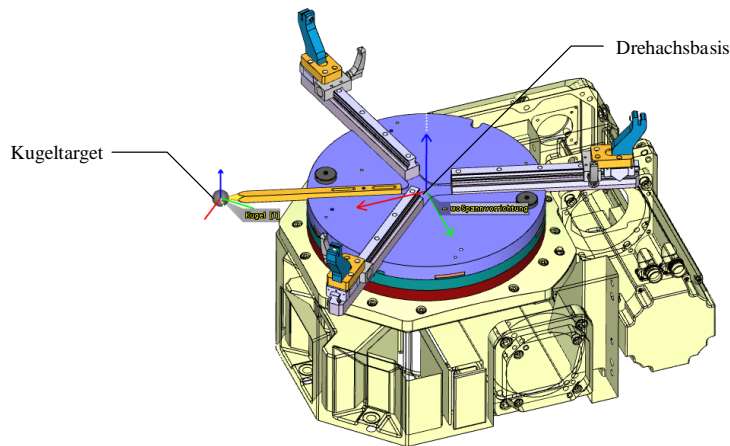


Abbildung 4.4.: Drehachsbasis im Zentrum der Drehachse in Höhe der Werkstückaufnahme mit Kugeltarget

Für die Bestimmung dieses Koordinatensystems sieht das Konzept dieser Arbeit analog zu der Bestimmung der Werkzeugspitzen die Vermessung eines Kugeltargets vor. Dieses ist in diesem Fall allerdings nicht stationär in der Zelle montiert, sondern an der Werkstückaufnahme, um sich mit dieser zusammen zu drehen. Über die Bestimmung des Zentrums dieses Kugeltargets an unterschiedlichen Winkeln kann das KS der Drehachsbasis ermittelt werden (siehe Abbildung 4.4). Damit bei anschließenden Prozessen auch Anlagen ohne Drehachse verwendet werden können, kann anstatt der Drehachsbasis auch die Basis der Werkstückaufnahme kalibriert werden. Hierfür wären allerdings drei an der Werkstückaufnahme montierte Kugeltargets notwendig.

### 4.3. Bauteildigitalisierung mit dem Laserlichtschnittverfahren

In der Konzipierung des Inspektionsaufbaus wurde in einer Nutzwertanalyse ein LLS-Sensor als passender Sensor zur schnellen Digitalisierung der verformten Bauteiloberfläche identifiziert. In diesem Abschnitt wird das Grobkonzept für den Digitalisierungsprozess mit diesem Sensor vorgestellt und später im Implementationsteil auskonzipiert. Die Anforderungen an die Digitalisierung sind eine *geringe Durchlaufzeit* zur Minimierung der Gesamttaktzeit und eine *homogene Punktedichte* der resultierenden Punktwolke. Die Homogenität spielt nicht nur eine entscheidende Rolle für die anschließende Pfadplanung bei der dieses Modell als Repräsentation der Realgeometrie herangezogen wird, sondern auch später im Gesamtprozess bei der Rissdetektion. Hier wird das selbe Modell herangezogen, um Bauteilmerkmale zu bestimmen, die wiederum für andere Al-

gorithmen benötigt werden. Der LLS-Sensor ist in dem Inspektionsaufbau zusammen mit dem WLI-Sensor an einer gemeinsamen Werkzeughalterung am Manipulator des IR montiert. Der Sensor weist einen zweidimensionalen, trapezförmigen Messbereich auf. Die kurze Seite dieses Trapezes hat eine Länge von 30 mm und die lange Seite 43 mm. Die Laserlinie wird an 476 Punkten abgetastet. Daraus resultiert eine Auflösung, die abstandsabhängig zwischen 0.063 mm und 0.09 mm variiert. Zu der Auflösung in Strahlrichtung macht der Hersteller in seiner Spezifikation keine Aussagen. Die Messfrequenz des Sensors beträgt in Abhängigkeit von der ausgewählten Betriebsart bis zu 90 Hz.

Das Konzept für die Digitalisierung sieht eine ringweise Messstrategie analog zu der bei der Rissinspektion vor. Der Sensor wird dabei durch den Roboter auf eine Pose vor das Bauteil verfahren, sodass seine Laserlinie vertikal auf dem Bauteil ausgerichtet ist. Durch das anschließende Drehen der Drehachse um  $360^\circ$  wird auf diese Weise ein vollständiger Höhenabschnitt des Bauteils aufgenommen (siehe Abbildung 4.5). Anders als beim Inspektionsprozess mit dem WLI-Sensor können mit dem LLS-Sensor dynamisch und kontinuierlich Daten akquiriert werden. Damit die einzelnen Linien nach der Aufnahme in ein gemeinsames Koordinatensystem transformiert werden können, müssen sie mit einer Roboterpose sowie einem Drehachswinkel der zusätzlichen Achse synchronisiert werden. Aus diesen fusionierten Daten kann dann in dem vollständig kalibrierten Aufbau die entsprechende Transformation in das Werkstückkoordinatensystem errechnet werden. Nachdem alle Punkte eines inneren und eines äußeren Pfades für die Digitalisierung abgearbeitet worden sind, liegt eine vollständige Punktwolke der Bauteiloberfläche vor. Dabei muss beachtet werden, dass die Verfahrensgeschwindigkeit der Drehachse zu der eingestellten Messfrequenz passt, damit das resultierende Modell eine ausreichend hohe Auflösung aufweist.



Abbildung 4.5.: Ringweise Aufnahmestrategie mit dem LLS-Sensor durch das Verfahren der Drehachse

#### 4.4. Pfadplanung für die Inspektion

Nach der Digitalisierung wird auf dem aufgenommenen Modell der realen Bauteiloberfläche eine Pfadplanung für die anschließende Inspektion mit dem WLI-Sensor durchgeführt. Wie im Kapitel 2 herausgearbeitet, besteht im Bereich der modellbasierten Ansätze für die Inspektionsplanung ein Defizit bei Anwendungsfällen, bei denen die geometrischen Abmessungen des Sensormessbereichs mehrere Größenordnungen kleiner sind als die des zu inspizierenden Bauteils. Genau solch ein Verhältnis liegt bei der Kombination vom betrachteten Brennkammerbauteil und dem WLI-Messbereich vor. Die aufgezeigten modellbasierten Methoden basieren auf der Optimierung einer MM, deren Anzahl von Zeilen und Spalten der Anzahl an Objektpunkten und potentiellen Sensorpositionskandidaten (SPK) entsprechen. Mit SPK werden des Weiteren die Ansichtenkandidaten aus Scotts Ansatz bezeichnet. Eine einfache Hochskalierung dieses Ansatzes auf die große Datenmenge in dieser Anwendung ist aufgrund des quadratisch steigenden Rechenaufwandes nicht möglich. Eine Strategie zur Bewältigung dieses Problems ist, wie durch Moutchino vorgeschlagen, das Ausnutzen von Bauteilsymmetrien (MB12).

Die Brennkammerbauteile des betrachteten Spektrums weisen eine nur durch wenige Merkmale unterbrochene Rotationssymmetrie auf. Bei unbeschädigten Bauteilen, die ihrem idealen CAD-Modell gleichen, könnte diese Symmetrie ausgenutzt werden, um die Pfadplanung stark zu vereinfachen. Diese müsste in diesem Fall nur auf einem Profil des Bauteilmodells durchgeführt werden. Im Anschluss würde der resultierende Pfad um das Drehachszentrum rotiert, wodurch aufgrund der Rotationssymmetrie eine Inspektion des gesamten Bauteils ermöglicht würde (siehe Abbildung 4.6).

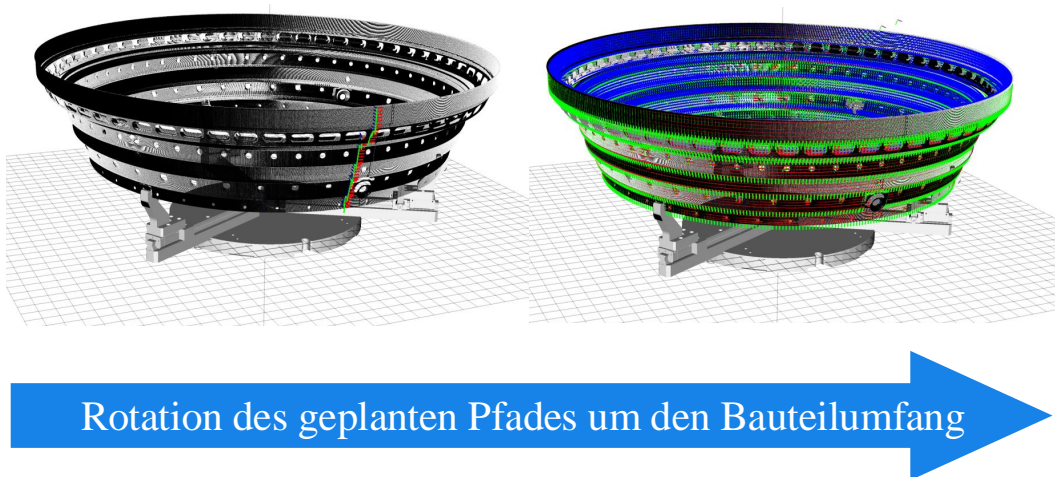


Abbildung 4.6.: Mögliches Vorgehen bei unbeschädigter, idealer Bauteilgeometrie



Die aus dem aktiven Flugbetrieb stammenden Bauteile weichen jedoch teilweise im Millimeterbereich von ihrer idealen Geometrie ab, sodass ohne weitere Maßnahmen die zentrale Anforderung der Vollständigkeit bei der Inspektion mit dem skizzierten Vorgehen nicht eingehalten werden kann.

In dieser Arbeit wird deswegen ein zweistufiges Konzept zur Inspektionsplanung eingeführt. Dieses erlaubt trotz der Beschädigungen den lokalen Einsatz einer effektiven modellbasierten Pfadplanung auf einem reduzierten Bauteilstück. Hierbei wird der mengentheoretische Ansatz nach Scott und sein entsprechend für diesen Prozess angepasster 3M-Ansatz herangezogen. Dieser setzt sich, wie in Abbildung 4.7 veranschaulicht, aus vier unterschiedlichen Schritten zusammen. Nachdem aus dem digitalisierten Modell Oberflächenkandidaten abgeleitet worden und im Anschluss entsprechende SPK ermittelt sind, werden die Kriterien für eine individuelle Messbarkeit aufgestellt und für jeden Matrixeintrag der daraus abgeleiteten MM überprüft. Den Abschluss dieses Prozesses bildet die Bestimmung eines minimalen Roboterpfades zur Inspektion aus der reduzierten MM.

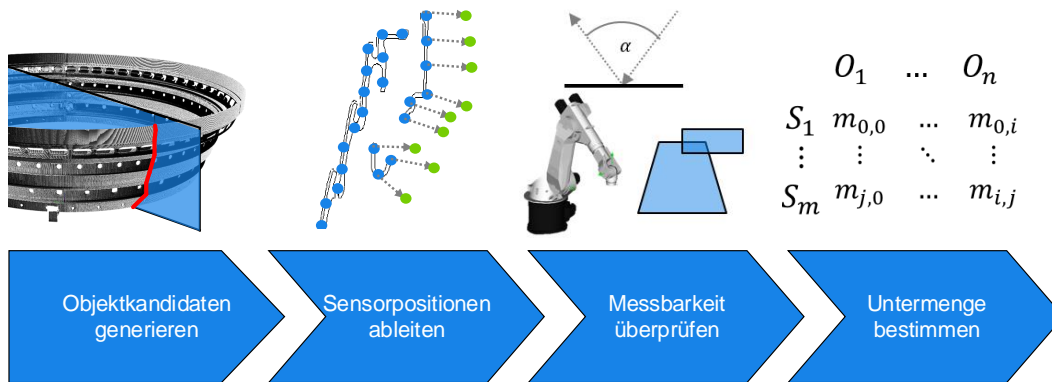


Abbildung 4.7.: Prinzipieller Ablauf der Pfadplanung

Damit der Inspektionspfad als Ergebnis der lokalen Pfadplanung auch global auf dem Rest des Bauteils eingesetzt werden kann, wird in der zweiten Stufe dieses Ansatzes eine Adaption der ermittelten Inspektionspunkte vorgenommen. Hierbei sollen die vor allem radial auftretenden Verformungen des Bauteils ausgeglichen und sichergestellt werden, dass gleichzeitig die Orientierung der Pfadpunkte global auf dem gesamten Bauteil passt und nicht nur auf einem lokalen Ausschnitt des Umfangs. Außerdem werden in dem zweiten Schritt gleichzeitig die bauteilspezifischen Merkmale und Störgeometrien berücksichtigt. Hierzu zählen zum Beispiel gewollte Fehlstellen im Bauteil wie Löcher zur Luftvermischung und Kühlung der Wandung oder andere Anbauteile wie Zündkerzeneinlässe (siehe Abbildung 4.8). Für die radiale Adaption der Inspektionspunkte kann, wie schon im Konzeptionskapitel für den Aufbau vorgestellt, die prinzipbedingte hochgenaue



Linearachse des WLI eingesetzt werden, die eine sehr schnelle und hochgenaue eindimensionale Adaption erlaubt, ohne den schwingungsanfälligen IR verfahren zu müssen.



Abbildung 4.8.: Merkmale und Störgeometrien an dem Bauteil,  
links: Panele und Löcher, Mitte: weitere Löcher, rechts: Zündkerzeneinlässe

Die Eingangsgröße des zweistufigen Pfadplanungsprozesses ist ein Modell der Bauteiloberfläche. Als Ergebnis wird dem eigentlichen Inspektionsschritt mit dem Weißlichtinterferometer sukzessive der entsprechende, aus mehreren zehntausend Pfadpunkten bestehende, Roboterpfad zugeführt. Der erste Schritt des skizzierten Vorgehens bei der Inspektionsplanung kann analog bei der Bauteildigitalisierung mit dem LLS-Sensor verwendet werden. Hierbei kann allerdings aufgrund des vergleichbar großen Messbereichs dieser Sensortechnologie auf den zweiten Adaptionsschritt verzichtet werden. Anstatt eines Punktwolkenmodells der Oberfläche ist die Eingangsgröße bei diesem Schritt ein CAD-Modell in einem der gängigen Oberflächenformate. Dieses Modell muss allerdings für die Pfadplanung aufgearbeitet werden, damit eine homogene Dichte an Oberflächenelementen erreicht wird.

## 4.5. Automatische Rissinspektion mit einem Weißlichtinterferometer

Alle bis zu diesem Punkt vorgestellten Subprozesse im Gesamtablauf stellen Hilfsprozesse dar. Sie dienen dazu, das Konzept für die eigentliche Inspektion mit dem Weißlichtinterferometer so simpel wie möglich zu halten, um dadurch einen schnellen Ablauf zu ermöglichen. Das Konzept sieht deswegen während des eigentlichen Prozesses keine weiteren Anpassungen oder Berechnungen vor. Alle notwendigen Adaptionswerte für die Inspektionsdaten an die reale Bauteilgeometrie wurden schon offline in den zwei Stufen der Pfadplanung vor der Inspektion ermittelt und können anschließend in der Inspektion einfach aus einer Speicherstruktur ausgelesen werden.

An dieser Stelle soll die ringweise Strategie bei der Inspektion noch einmal ausführlich vorgestellt werden, da sie bis jetzt nur im Zusammenhang mit der Schwingungsreduktion und zur

besseren Zugänglichkeit eingeführt wurde. Aufgrund der zusätzlichen Drehachse ist das Bauteil bei dieser Strategie in vertikale Ringe aufgeteilt, die wiederum horizontal in einzelne Messfelder aufgeteilt sind (siehe Abbildung 4.9). Die Höhe der Ringe und die Breite der Messfelder sind dabei durch den Messbereich des Sensors festgelegt und werden der Inspektion aus der Pfadplanung vorgegeben. Die Inspektion besteht aus diesem Grund ausschließlich daraus die Achse in einem vorgegebenen Intervall zu drehen und der WLI-Steuerung sukzessive die Adaptionswerte  $a_{11} \dots a_{xy}$  aus einem zweidimensionalen Array zukommen zu lassen und nach der Fertigstellung eines Ringes, den Roboter auf den nächsten Pfadpunkt zu verfahren. Die beiden Dimensionen des Arrays sind daher zum einen die Anzahl an Ringen und zum anderen die Anzahl an Messfeldern in dem korrespondierenden Ring. Das Array wird deswegen zeilenweise ausgelesen. Innen- und Außenseite des hohlkegelstumpfförmigen Bauteils werden dabei nacheinander abgearbeitet. Während des Inspektionsprozesses findet eine rege Kommunikation zwischen den Steuerungen der Handhabung und des WLI-Sensors statt. Über diese Kommunikation wird der Prozessablauf gesteuert. Wie diese Kommunikation im Detail aussieht, beschreibt Abschnitt 6.4.

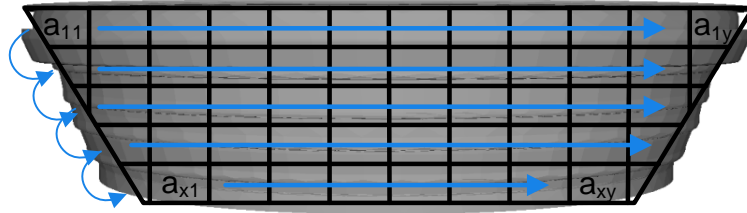


Abbildung 4.9.: Messstrategie bei der Inspektion

## 5. Software zur automatischen Inspektion

In diesem Kapitel werden die für die Planung und Steuerung des Inspektionssystems benötigten Software- beziehungsweise Simulationswerkzeuge vorgestellt. Diese sind zum einen das skizzierte *Matlab*-Skript, das zur Validierung der messbereichsscharfen Bestimmung der Abdeckung bei der Inspektion benötigt wird und zum anderen die im Rahmen dieser Arbeit entstandene Steuerungssoftware *AutoXIV*, die den Ablauf steuert und in der alle Subprozesse des gesamten Inspektionsprozesses implementiert sind. Gleichzeitig stellt *AutoXIV* auch noch eine Benutzerschnittstelle für das manuelle Durchführen einzelner Prozessschritte zur Verfügung.

### 5.1. Simulation zur Bestimmung der Messabdeckung

Die Simulation zur Bestimmung der Messabdeckung ist nicht Teil des eigentlichen Messablaufs und deswegen auch nicht in *AutoXIV* integriert. Ihre Funktionalität wird schon bei der Konzipierung des Inspektionssystems benötigt. Deswegen ist sie in der proprietären Programmiersprache von *MATLAB* implementiert. Diese bietet den Vorteil, dass der Programmierer sich nicht um das Aufsetzen der Programmierungsumgebung und das Beschaffen von Bibliotheken kümmern muss, sondern direkt mit der Implementierung beginnen kann. Der Nachteil dieser Sprache ist allerdings die Ausführungszeit. Diese ist vergleichbar niedrig, da es sich hierbei nicht um eine kompilierende, sondern um eine interpretierende Sprache handelt. Als Ressourcen stehen für die Simulation der Roboterpfad als Ergebnis der Zugänglichkeit- und Erreichbarkeitsanalyse sowie ein Oberflächenmodell des Bauteils im STL-Format zur Verfügung. In diesem Format ist eine Oberfläche durch ein Netz aus Dreiecken beschrieben. Je höher die Auflösung dieses Netzes gewählt wird, desto weniger weicht dieses Modell von seinem realen Gegenstück ab. Aufgrund der benötigten Genauigkeit bestehen diese Modelle deswegen aus bis zu 2 Millionen Dreiecken. Ein Dreieck wird dabei über seine drei Vertices und ihrer Ebenennormale beschrieben. Das Konzept sieht vor, dass die inspizierbare Oberfläche messbereichsscharf bestimmt werden soll. Dazu müssen die Dreiecke des Modells ermittelt werden, die sich zumindest teilweise innerhalb der quaderförmigen Messbereichsgeometrie des WLI-Sensors befinden. Hierbei ist zu beachten, dass rückseitige sowie vollständig verdeckte Dreiecke erkannt und von der Betrachtung ausgeschlossen werden müssen. Dreiecke, die nur teilweise durch andere Dreiecke verdeckt sind, müssen aufgetrennt und neu verknüpft werden. Hierdurch kann eine scharfe Messbereichsgrenze sichergestellt werden.

Das für diese Arbeit implementierte *MATLAB*-Skript baut auf dem in der 3D-Grafik lang etablierten Verfahren des „Raytracings“, einem binären kd-Suchbaum als Raumteilungsverfahren und der Delaunay-Triangulation für die korrekte Tessellation von Punkten zu Dreiecken auf (Hug14). Bevor die eigentliche Bestimmung der entsprechenden Dreiecke startet, wird in einem

ersten Schritt das Bauteilmodell importiert und die Vertices der Dreiecke in einen kd-Baum eingefügt. Ein kd-Suchbaum beschleunigt ganz allgemein eine Suche nach nächsten Nachbarpunkten in Punktwolken. Die Kosten für eine Nachbarsuche wachsen bei einem Suchbaum nur logarithmisch mit der Anzahl der Modellpunkte an anstatt linear ohne Suchbaum. Diese Suchbaumstruktur wird in diesem Skript an mehreren Stellen genutzt. Abbildung 5.1 zeigt den gesamten Prozess in einem stark abstrahierten Prozessablaufplan. Die zentralen Klassen sind in Abbildung 5.2 in einem Klassendiagramm veranschaulicht. Beide Diagramme sind Standarddiagramme der Unified Modeling Language (UML).

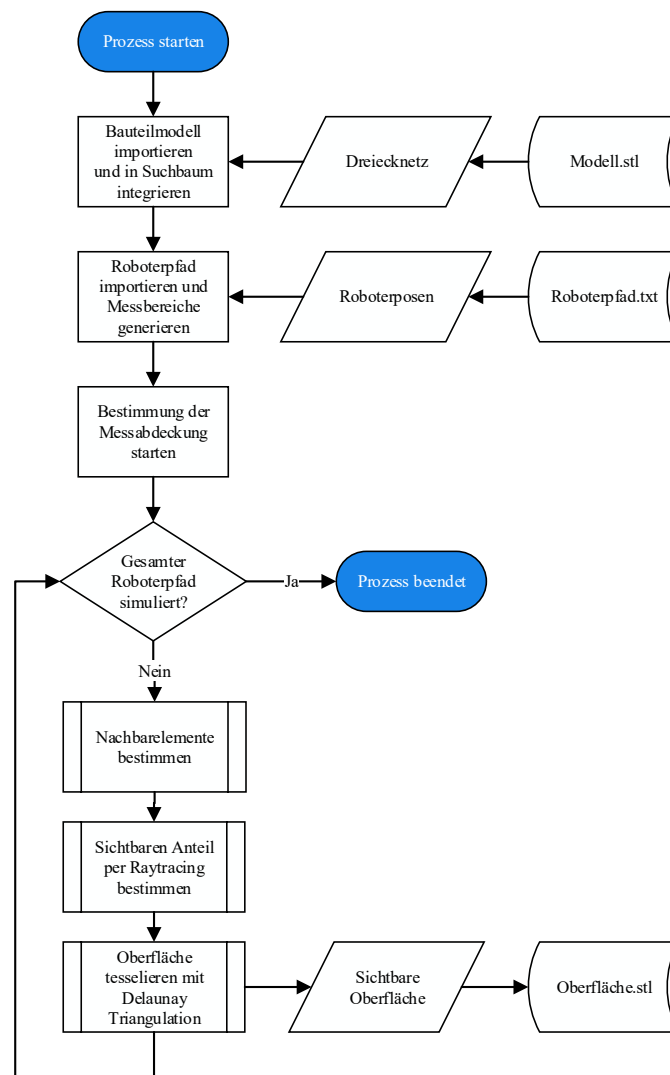


Abbildung 5.1.: Prozessablaufplan des *MATLAB*-Skriptes zur Bestimmung der Messabdeckung

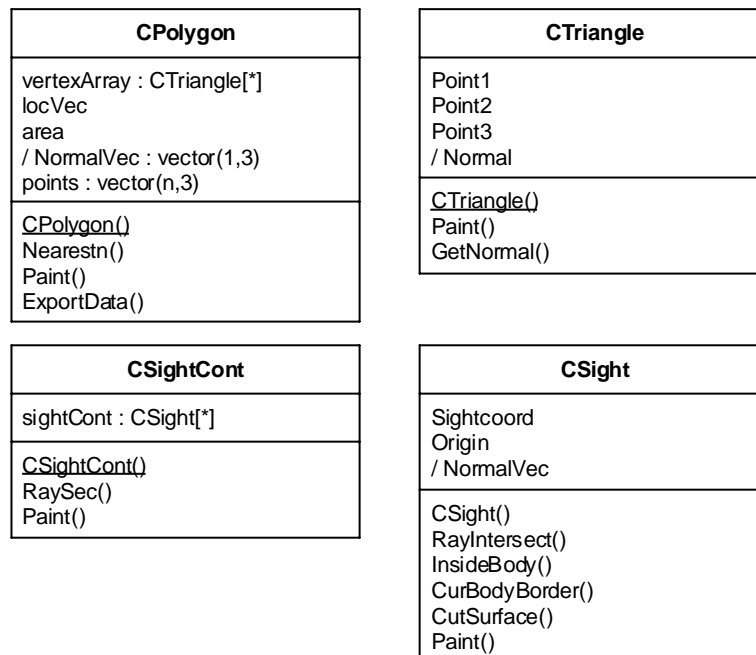


Abbildung 5.2.: Klassendiagramm der wichtigsten Klassen des *MATLAB*-Skriptes zur Bestimmung der Messabdeckung

Zu Beginn der Ausführung des Skriptes wird jeder Punkt des importierten Roboterpfades entsprechend seiner aktuellen geometrischen Parameter in einen Messbereich übersetzt. In einer Schleife werden anschließend alle Messbereiche einzeln auf Abdeckung überprüft. Es beginnt damit, dass für einen Messbereich mit Hilfe eines kd-Suchbaums die nächsten  $X$  Nachbardreiecke bestimmt werden. Die Position eines Dreiecks ist dabei über seinen Schwerpunkt beschrieben. Die Anzahl der Nachbarn ist variabel und abhängig von der Auflösung des Bauteilmodells. Dieser Wert wird dem Skript zu Anfang als Parameter vom Benutzer übergeben. Für diese reduzierte Dreiecksanzahl wird dann in einer weiteren Schleife ermittelt, ob deren Kanten eine der Seiten des Messbereichs schneiden. Ist dies der Fall, liegt ein Teil des Dreiecks innerhalb des Messbereichs. Dieser innenliegende Bereich muss des Weiteren messbereichsscharf bestimmt werden. Je nachdem wieviele Vertices eines Dreiecks sich innerhalb eines Messbereichs befinden, muss unterschiedlich fortgefahren werden. Liegen alle drei Vertices innerhalb des Messbereichs, so kann das Dreieck als „vollständig abgedeckt“ gekennzeichnet werden. Liegen allerdings kein Vertex, ein Vertex oder zwei Vertices innerhalb, muss das Dreieck in die Fläche innerhalb und außerhalb des Messbereichs aufgetrennt und wieder mit Dreiecken vernetzt werden.

Ist die sichtbare Oberfläche bestimmt, werden in einem weiteren Schritt die Dreiecke gefiltert, die nicht von anderen Dreiecken verdeckt werden. Hierfür kommt das Raytracing-Verfahren zum

Einsatz. Von jedem Vertex des betrachteten Dreiecks wird dabei ein Strahl in Richtung der Strahlrichtung des Sensors gebildet. Dieser Strahl wird dann auf Schnittpunkte mit den restlichen Nachbardreiecken überprüft. Wird bei keinem der Vertices ein Schnittpunkt ermittelt, ist das Dreieck vollständig unverdeckt. Existieren Schnittpunkte, muss wiederum ermittelt werden, welcher Teil des Dreiecks verdeckt ist. Hierzu werden die Dreiecksseiten, die einen verdeckten mit einem nicht verdeckten Vertex verbinden, iterativ halbiert und diese neuen Punkte wiederum unter Einsatz des Raytracing-Verfahrens auf Verdeckung überprüft. Sobald die neuen sichtbaren Punkte bestimmt sind, werden aus den resultierenden Flächen des verdeckten und nicht verdeckten Bereiches neue Dreiecke generiert und mit Hilfe der Delaunay-Triangulation miteinander vernetzt (Abbildung 5.3a). Am Ende des Skriptes wird aus allen sichtbaren Dreiecken dann ein neues STL-Modell in eine entsprechende Datei exportiert. Werden die Daten aller Messbereiche zu einem Datensatz vereint, kann die mögliche Messabdeckung auf einem Inspektionspfad evaluiert, in *FAMOS* importiert und visualisiert (siehe Abbildung 5.3b) und im Anschluss entsprechend optimiert werden.

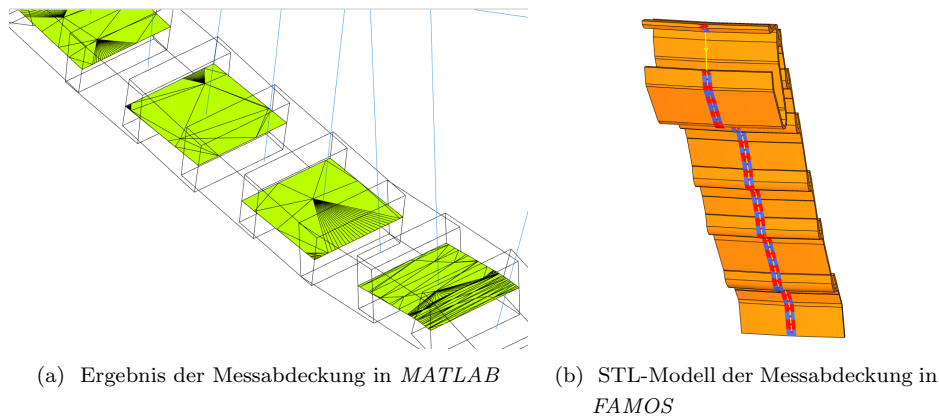


Abbildung 5.3.: Ergebnisse bei der Bestimmung der Messabdeckung

## 5.2. Steuersoftware AutoXIV

Im Kapitel 2 wurde herausgearbeitet, dass keine aktuell erhältliche Software aus dem Bereich von Industrie oder Forschung alle benannten Anforderungen des konzipierten Inspektionsprozesses gänzlich abdecken kann. Außerdem wurde festgestellt, dass für die Lösung der vorliegenden breitgefächerten Aufgabenstellung ein Framework besser geeignet ist als eine auf wenige Funktionalitäten spezialisierte Software. Deswegen wird im Rahmen dieser Arbeit das Framework *AutoXIV* entwickelt. In diesem Abschnitt wird der Aufbau der Software sowie die Einbettung in das Netzwerk des Inspektionssystems vorgestellt.

### 5.2.1. Aufbau der Software

Das Framework *AutoXIV* setzt sich aus mehreren quelloffenen (engl. open source) Bibliotheken zusammen, die für verschiedene Aufgabengebiete spezialisiert sind. Durch diesen flexiblen Aufbau ist das Framework beliebig erweiterbar und kann an die sich ändernden Anforderungen in einem Forschungsvorhaben angepasst werden. Als Programmiersprache für dieses Framework wurde das bei modernen Bibliotheken noch immer weit verbreitete C++ in der Version 03 ausgewählt. Da diese Version der Programmiersprache aufgrund ihres Alters viele moderne Programmieretechniken nicht unterstützt, wird sie durch die frei nutzbaren C++-Dialekte, *Boost* und *Qt* erweitert. Hierdurch wird es dem Programmierer ermöglicht, ohne größeren Aufwand sowohl parallel multithreadbasiert als auch ereignisorientiert zu programmieren. Auch stehen ihm weitere aktuelle Standardfeatures wie zum Beispiel Ein- und Ausgabeelemente einer Benutzerschnittstelle (engl. GUI) und verschiedene Funktionen zur Netzwerkkommunikationen für die Schnittstellenprogrammierung zur Verfügung.

Für die 3D-Visualisierung und 3D-Manipulation von Punktwolken und Oberflächenmodellen sowie für die Anteile der Offline-Programmierung sind die plattformunabhängigen Bibliotheken „The Visualization Toolkit (VTK)“ und „Pointcloud Library (PCL)“ in das Projekt eingebunden. Das VTK realisiert eine eigene Render-Pipeline, die auf der Grafikschnittstelle OpenGL aufsetzt und dem Programmierer abstrakte Grafikobjekte und verschiedene Operationen zur Verfügung stellt, die wiederum OpenGL-Elemente einhüllen. Der Programmierer muss sich dadurch nicht mehr um die tiefen Schichten des 3D-Rendering kümmern, sondern erstellt und arbeitet nur noch mit abstrakten VTK-Objekten.

Die PCL ist eine Bibliothek, die eine Vielzahl unterschiedlicher Algorithmen und Filter für 3D-Punktwolken bereitstellt. Ihre Hauptkompetenzen sind neben der Filterung, templatebasierte Datenstrukturen für Punktwolken, die Punktwolkenregistrierung und -segmentierung sowie die Erkennung und Extraktion von Objektmerkmalen. Dabei werden komplexe mathematische Algorithmen benötigt, die über die von der PCL direkt eingebundene Mathematik-Bibliothek *Eigen* abgedeckt werden. Eigen stellt Vektor und Matrixstrukturen und -operationen sowie Optimierungsverfahren bereit, die für die 3D-Programmierung von zentraler Bedeutung sind und auch unabhängig von der PCL in *AutoXIV* Anwendung finden. Abbildung 5.4 zeigt die aktuelle Oberfläche der Software. Auf der rechten Seite ist eine Baumstruktur zu erkennen, die den View Anteil des Model/View strukturierten Datenhandling des Frameworks darstellt. Neben den Punktwolken können außerdem Extensible Markup Language (XML)-basierte Roboterpfade im- und exportiert, angezeigt sowie manipuliert werden. In der unteren rechten Ecke befinden sich Werkzeugkästen (engl. Toolboxes), die Informationen über die in der Visualisierung markierten Prozessobjekte enthalten und deren Manipulation ermöglichen. In den linken Werkzeugkästen sind Eingabe- und Ausgabeelemente der verschiedenen, von *AutoXIV* implementierten, Prozesse untergebracht.

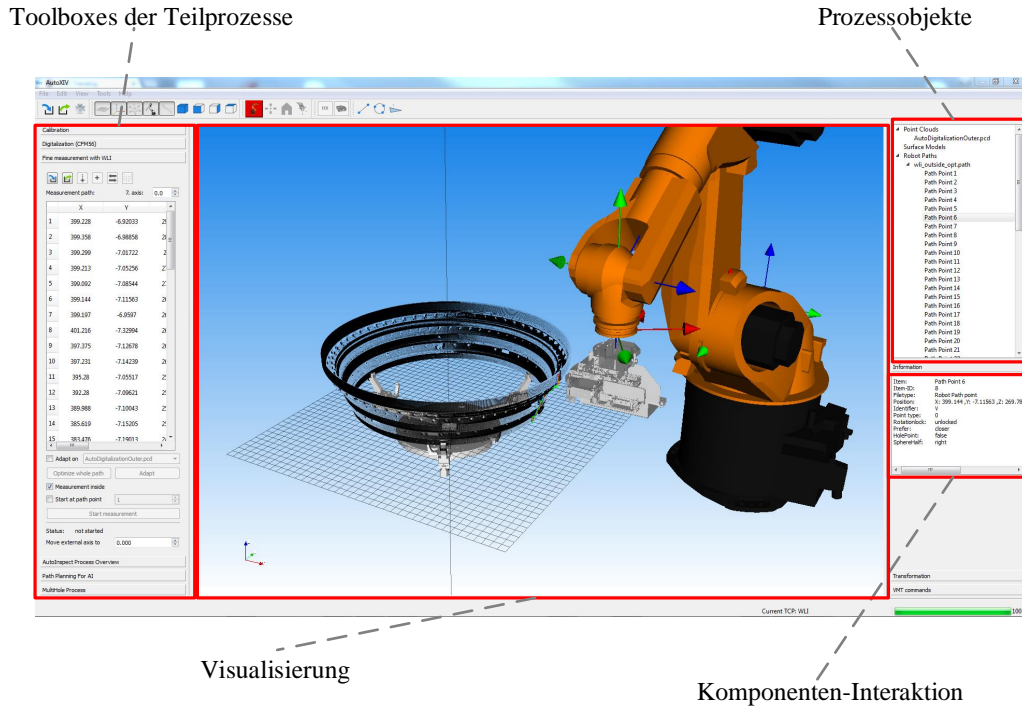


Abbildung 5.4.: Benutzeroberfläche der Software *AutoXIV*

Während des Inspektionsprozesses liegt die Benutzeroberfläche von *AutoXIV* passwortgeschützt im Hintergrund und ist dadurch nicht für den Standard-Bediener zugänglich. Sie steuert den Prozess also automatisch als Blackbox. Der Bediener hat nur Zugang zu einer stark vereinfachten Oberfläche, die aber nicht in der Arbeit betrachtet wird.

### 5.2.2. Abbildung des Inspektionsprozesses in *AutoXIV*

Eine zentrale Anforderung an die Implementierung des Inspektionsprozesses ist die Sicherstellung des ordnungsgemäßen Prozessablaufs. Denn nur ein kalibrierter Sensor darf eine Digitalisierung durchführen und nur, wenn ein Modell der aktuellen Oberfläche vorliegt, darf eine Inspektion durchgeführt werden. Hierfür wird in *AutoXIV* die Struktur des verschachtelten endlichen Zustandsautomaten (engl. finite state machine) verwendet. Zustandsautomaten setzen sich aus *Zuständen*, *Zustandsübergängen* und *Aktionen* zusammen. Zustände sind mit einer begrenzten Anzahl an Folgezuständen über logische Zustandsübergänge miteinander verbunden. Die Übergänge werden durch Aktionen in Zuständen ausgelöst. Durch diese Architektur, wie in Abbildung 5.5 veranschaulicht, ist sichergestellt, dass ein Prozess in der vorgegebenen Reihenfolge ausgeführt wird.



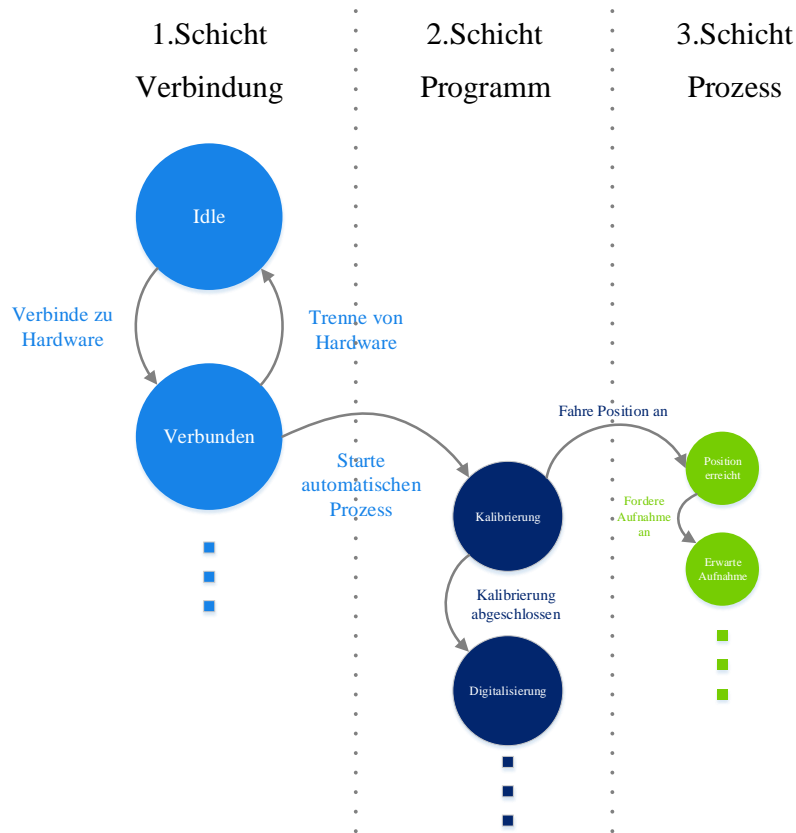


Abbildung 5.5.: Vereinfachte Übersicht der unterschiedlichen Schichten der verwendeten Zustandsautomaten

In der untersten dritten Schicht dieser verschachtelten Architektur befinden sich die Zustandsautomaten der einzelnen Teilprozesse, von denen jeder eine spezifische Aufgabe erfüllt und die grundsätzlich keine Überdeckungen in der Funktionalität aufweisen. Diese Teilprozesse sind die im Konzeptionsteil dieser Arbeit vorgestellten Prozesse wie zum Beispiel die Kalibrierung der Werkzeugspitzen, die Digitalisierung der Oberfläche oder die Rissinspektion mit einem Weißlichtinterferometer. Was diese Prozesse miteinander verbindet, sind die benötigten Hardware-Ressourcen, wie die Handhabungskomponenten oder die unterschiedlichen Sensoren. Diese Ressourcen bieten individuelle atomare Funktionen wie zum Beispiel ein „Bewegung nach XY“ bei einem Roboter oder „Starte Akquisition“ bei einem der Sensoren. Damit diese atomaren Aktionen zu einem sinnvollen Prozess verbunden werden können, benötigen sie ebenfalls einen Zustandsautomaten als steuernde Struktur. Die Zustandsautomaten sind in der einfachen switch-case Architektur implementiert und in sogenannte Handler-Klassen gehüllt. Eine switch-case Fallunterscheidung liefert alle weiter oben genannten Funktionalitäten eines Zustandsautomaten.

Auf der zweiten Schicht der Automatenstruktur werden die Prozesse zu einem sinnvollen Pro-

gramm verbunden. Dabei werden die untergeordneten Prozesse so hintereinander angeordnet, dass sie eine gegebene Problemstellung lösen können. In dem vorliegenden Fall ist dies die Rissinspektion. In den einzelnen Zuständen dieser Schicht befinden sich dann entweder, wie schon in Abbildung 5.5 veranschaulicht, die eben angesprochenen Prozesse oder auch wiederum atomare Funktionen der angeschlossenen Hardware, falls zwischen einzelnen Prozessen zum Beispiel eine weitere Roboterbewegung durchgeführt oder ein zusätzliches Modell importiert oder exportiert werden soll. Das übergeordnete Programm wird wiederum durch eine switch-case Struktur in einer Handler-Klasse gesteuert.

Die oberste Schicht stellt unter anderem die Schnittstelle der Software *AutoXIV* zu einem Benutzer dar. Zustandsübergänge können in dieser Schicht manuell von einem Nutzer über Eingabelemente wie physische Taster oder Knöpfe in der GUI ausgelöst werden. Beispiele wären der Start oder Abbruch eines Programms. Außerdem wird in dieser Schicht die Netzwerkverbindung zu den einzelnen Ressourcen auf- und später wieder abgebaut. Sind alle Verbindungen, die für ein spezielles Programm benötigt werden, hergestellt, kann durch den Benutzer eine Aktion ausgelöst werden, die einen Zustandsübergang provoziert. Ein Zustand repräsentiert hier den Beginn eines Programms in einer tieferen Schicht. Durch diese Modularität können in dieser Architektur ohne großen Aufwand andere Programme durch eine neue Zusammensetzung der Zustände erzeugt werden. Ein Beispiel für ein anderes Programm könnte ein Reparaturprogramm sein. Bezüglich der benötigten Ressourcen würde dieses eine hohe Überdeckung mit dem Inspektionsprogramm aufweisen. Auch bei diesem Programm müsste eine Kalibrierung und eine Digitalisierung durchgeführt werden. Lediglich der Inspektionsschritt müsste durch einen Reparaturschritt ausgetauscht werden.

In allen Schichten ist zu jeder Zeit der aktuelle Zustand bekannt und abgespeichert. Daher kann auch ein einfaches Wiederaufsetzen nach einem Absturz oder einer Notaus-Situation implementiert werden. Sollte *AutoXIV* abstürzen, setzt der Benutzer den Ablauf auf den letzten, erfolgreich abgeschlossenen Prozess und startet den Ablauf ab diesem Punkt neu. Die bis dahin abgespeicherten Daten werden wiederum geladen und der Prozess wird fortgesetzt. Da die Möglichkeit einer ereignisorientierten Programmierung besteht, können die einzelnen Schichten transparent in beiden Richtungen kommunizieren und sich ihre Zustände mitteilen.

Unabhängig von dem automatischen, durch die Handler organisierten Ablauf können sowohl die atomaren Funktionen als auch einzelne Prozesse manuell in dem Expertenmodus von *AutoXIV* ausgelöst werden. Hierzu stehen dem Benutzer ebenfalls die oben genannten Werkzeugkästen zur Verfügung, mit denen der Nutzer die entsprechend benötigten Ressourcen verbinden beziehungsweise importieren kann und dann den Prozess manuell auslöst. Die Steuerung des Roboters erfolgt in einem eigenen Dialog. Hier kann der Roboter manipulator wahlweise in seinem Werkzeug- oder dem Weltkoordinatensystem verfahren oder jede seiner Achsen einzeln bewegt werden.

### 5.2.3. Implementierung von Arbeiterklassen

Wie im Abschnitt zuvor beschrieben, sind die früher für die Inspektion konzipierten Teilprozesse aus Sicht des Ablaufs in eigene Handler-Klassen gekapselt. Hier ist nicht nur der Ablauf, sondern auch noch die eigentliche jeweilige Funktion der einzelnen Prozesse realisiert. Hierfür wird den Handlern für den jeweiligen Prozess eine eigene Arbeiterklasse (engl. *workerclass*) untergeordnet, in der die eigentliche Intelligenz über die konzipierten Algorithmen aus den einzelnen externen Paketen beinhaltet ist. Der Klasse *CalibrationHandler* ist zum Beispiel die Arbeiterklasse *Calibrator* untergeordnet. Sie implementiert wiederum die im Konzeptionsteil vorgestellten Methoden und Algorithmen zur Kugel beziehungsweise Kreisschätzung und für die dann benötigte Berechnung der Werkzeugschärfe.

Die meist rechen- und zeitintensiven Algorithmen in den Arbeiterklassen laufen, wenn möglich, in eigenen Threads von *AutoXIV*, sodass die restliche Applikation nicht einfriert und der Nutzer während des Prozessablaufs andere Prozesse durchführen kann. Wenn eine entsprechende Methode abgeschlossen ist, gibt die Arbeiterklasse der übergeordneten Handler-Klasse ereignisorientiert ein vor dem Start vereinbartes Signal, woraufhin die Handler-Klasse zum nächsten Schritt übergeht. Diese Architektur folgt dem Gedanken der Modularisierung und ermöglicht den unkomplizierten Austausch von Methoden und Strukturen. Diese werden nämlich nur in ihren entsprechenden Arbeiterklassen aufgerufen und weisen keine Schnittstelle zu weiteren Programmteilen auf.

### 5.2.4. Netzwerkarchitektur

Das Framework *AutoXIV* ist mit verschiedenen anderen Systemkomponenten des Inspektionssystems über unterschiedliche Schnittstellen vernetzt. Zum einen besteht eine Netzwerkverbindung mit dem LLS-Sensor, um die bei der Digitalisierung anfallenden Daten zu akquirieren und zum anderen werden sowohl IR als auch Drehachse ebenfalls durch *AutoXIV* angesteuert. Um bei dem Inspektionsprozess den Status von *AutoXIV* und WLI miteinander zu synchronisieren, besteht zwischen ihnen eine weitere Verbindung. Zusätzlich zu diesen Komponenten sind der Sicherheitskreis mit allen Sicherheitskomponenten sowie Zellentüren, Not-Aus-Knöpfe und die zusätzliche Benutzerschnittstelle der Anlage über ein *PROFIBUS*-Netzwerk in einer Soft-SPS miteinander verknüpft. Hierdurch wird sichergestellt, dass ein Not-Aus-Signal garantiert alle Systemkomponenten erreicht und nicht durch einen weiteren Prozess auf dem Rechner von *AutoXIV* unterbrochen wird. In diesen Sicherheitskreis sind ebenfalls IR und Drehachse mit eingebunden. Zwischen dieser Soft-SPS und *AutoXIV* besteht zum einen eine Kommunikationsverbindung um den Status des Prozesses für den Bediener zu visualisieren und zum anderen eine weitere „Keep-Alive“-Verbindung, über die sichergestellt wird, dass im Falle eines Absturzes von *AutoXIV* garantiert gleichzeitig die komplette Hardware gestoppt wird.

Das beschriebene Netzwerk ist über ein Transmission Control Protocol (TCP/IP)-basiertes Ethernet realisiert. Im Fall der Drehachse und der Verbindung zum *KUKA-RSI* handelt es sich um

einfache verbindungslose User Datagram Protocol (UDP)-Verknüpfungen, alle anderen Verbindungen sind im TCPIP durch Funktionsbausteine von *Qt* implementiert. Verbindungsaufbau und -abbruch werden sowohl von Aktionen in einzelnen Zuständen der Zustandsautomaten als auch von Ereignissen bei Eintritt eines Fehlers durchgeführt. Abbildung 5.6 visualisiert die grundsätzliche Struktur des Netzwerkes im Inspektionsaufbau mit *AutoXIV* im Zentrum.

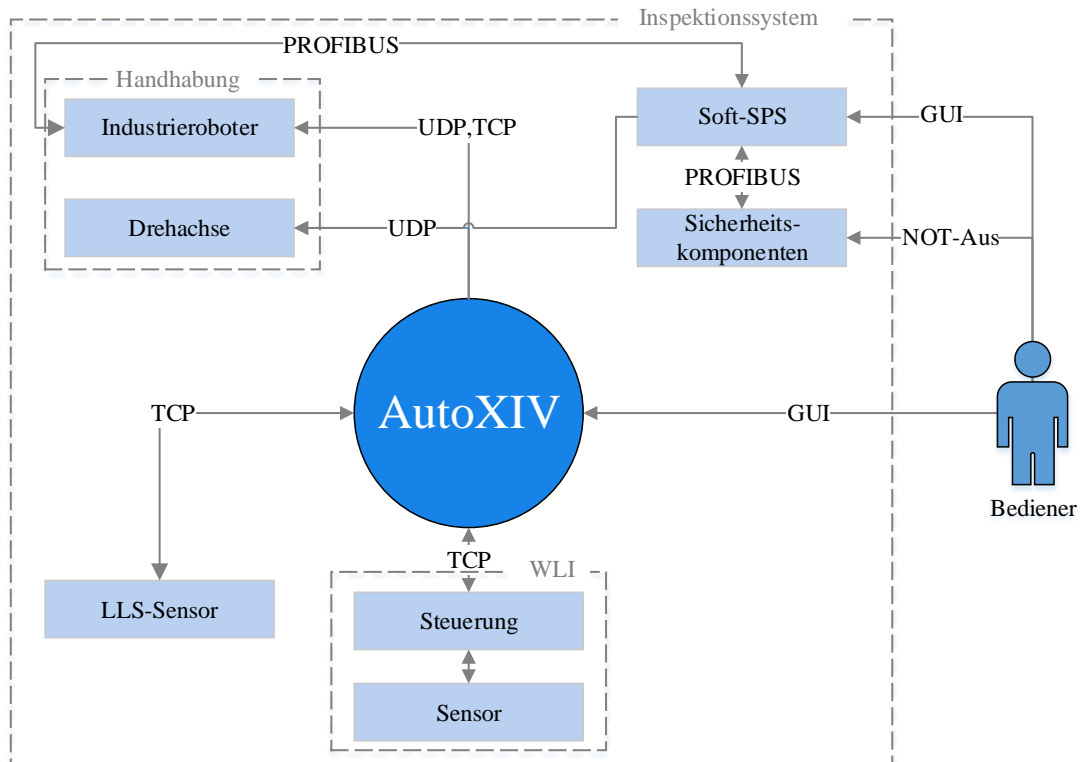


Abbildung 5.6.: Netzwerk im Inspektionsaufbau

## 6. Ausgestaltung und Implementierung des Inspektionsprozesses

In diesem Kapitel wird die Ausgestaltung und Umsetzung der in Kapitel 4 vorgestellten Konzepte für die Kalibrierung, die Digitalisierung, die Pfadplanung und die WLI-Inspektion vorgestellt. Die ausgewählten Lösungsansätze und verwendete mathematische Algorithmen werden ebenso wie die Integration der einzelnen Prozessschritte in die Softwareumgebung *AutoXIV* detailliert beschrieben.

### 6.1. Kalibrierung des Inspektionssystems

Die Kalibrierung stellt den ersten Prozessschritt jeder Bauteilinspektion dar. Sie ist unbedingt notwendig, um die mit den unterschiedlichen Sensoren aufgenommenen Daten der Oberfläche in einem Gesamtmodell zu registrieren. Kalibriert werden die Werkzeugspitzen von WLI- und LLS-Sensor sowie das Koordinatensystem der Drehachse. In allen drei Fällen stellen Kugeltargetzentren Fixpunkte innerhalb der Anlage dar, die in den Daten der verschiedenen Sensoren bestimmt werden können.

#### 6.1.1. Kalibrierung der Werkzeugspitzen

Das Konzept zur Kalibrierung der Sensor-Werkzeugspitzen sieht vor, dass die Sensoren von dem IR in unterschiedlichen Positionen vor einem Kugeltarget positioniert werden, so dass entsprechend WLI-Aufnahmen beziehungsweise Linien von der Target-Oberfläche akquiriert werden können. Diese Daten werden bei *AutoXIV* in einer Containerstruktur zwischengespeichert, solange noch weitere Messungen ausstehen. Sind alle Daten aufgenommen, werden in ihnen die entsprechenden Kugelzentren bestimmt und daraus wiederum eine Werkzeugspitze optimiert. Diese Auswertung läuft in *AutoXIV* in einem eigenen Arbeiterthread, so dass die Applikation während dieses Prozesses weitere Befehle verarbeiten kann.

Nachdem ein Sensor-TCP als Kalibrierergebnis bestimmt ist, wird dieses automatisch mit Ergebnissen vorheriger erfolgreicher Durchläufe verglichen. Übersteigt die Abweichung einen vorher festgelegten Schwellwert, ohne dass etwas an der eigentlichen Sensorhardware verändert wurde, wird die entsprechende Kalibrierung wiederholt. Wird abermals eine zu große Abweichung festgestellt, wird der gesamte Prozess abgebrochen. Liegt die Abweichung unterhalb dieses Schwellwertes entscheidet die Größe der Abweichung: Entweder werden die neuen Werte übernommen oder bei einer sehr kleinen Abweichung keine Änderungen vorgenommen. Im Folgenden wird die Umsetzung der von Gan beschriebenen, aber nicht umgesetzten Variante 2 (siehe Abbildung 4.3) mathematisch hergeleitet.

#### 6.1.1.1. Schritt 1: Kalibrierung des rotatorischen Anteils

Die Formel für die kinematische Kette ergibt sich für eine Aufnahme mit dem Index  $i$  zu:

$${}^F_R T_i * {}^{TCP}_F T * t_{CSi} = t_{CR} \quad (6.1)$$

Werden die homogenen Transformationen in ihren rotatorischen und translatorischen Teil aufgeteilt, erweitert sich diese Formel zu:

$${}^F_R R_i * {}^{TCP}_F R * t_{CSi} + {}^F_R R_i * {}^{TCP}_F t + {}^F_R t_i = t_{CR} \quad (6.2)$$

Werden nun die Kugelzentren mit der gleichen Orientierung ( ${}^F_R R_i = {}^F_R R_j$ ) an zwei unterschiedlichen Roboterpositionen bestimmt, kürzt sich der mittlere Summand heraus und die daraus resultierenden Formeln können wie folgt gleichgesetzt und später aufgelöst werden:

$${}^F_R R_i * {}^{TCP}_F R * t_{CSi} + {}^F_R t_i = {}^F_R R_j * {}^{TCP}_F R * t_{CSj} + {}^F_R t_j \quad (6.3)$$

$${}^F_R R_i * ({}^{TCP}_F R * (t_{CSi} - t_{CSj})) = {}^F_R t_j - {}^F_R t_i \quad (6.4)$$

durch linksseitige Multiplikation beider Seiten mit  ${}^F_R R_i^{-1}$  wird daraus:

$${}^{TCP}_F R * (t_{CSi} - t_{CSj}) - {}^F_R R_i^{-1} * ({}^F_R t_j - {}^F_R t_i) = 0 \quad (6.5)$$

Für die Transponierte des Produktes einer Matrix  $A \in K^{m \times n}$  mit einer Matrix  $A \in K^{n \times l}$  gilt bekannterweise

$$(A \times B)^T = B^T \times A^T \quad (6.6)$$

Auf Gleichung 6.5 angewendet ergibt sich das Gleichungssystem der Form  $A * x = b$

$$(t_{CSi} - t_{CSj})^T * {}^{TCP}_F R^T = ({}^F_R R_i^{-1} * ({}^F_R t_j - {}^F_R t_i))^T \quad (6.7)$$

Die Orientierung  ${}^{TCP}_F R$  ergibt sich dann als das Minimum des überbestimmten Optimierungsproblems der Form:

$$\min_x \|A \times x - b\| \quad (6.8)$$

Zur Minimierung bietet sich die Singular Value Decomposition (SVD) an, die als Resultat die orthogonale Matrix mit dem minimalen Abstand zu  $A$  liefert. Dabei werden Scher- und Skalieranteile entfernt.

#### 6.1.1.2. Schritt 2: Kalibrierung des translatorischen Anteils

Beim translatorischen Anteil des Sensor TCP's wird analog zu seinem rotatorischen Anteil vorgegangen. Aufgesetzt wird dabei wieder bei der Formel 6.1 für die kinematische Kette des an dem Roboter montierten LLS-Sensors. Hierbei ist nun allerdings die rotatorische Komponente  ${}^{TCP}_F R$  schon bekannt. Das Gleichsetzen und anschließende Umformen dieser Formeln an zwei Messpunkten führt in diesem Fall direkt zu der Gleichung mit der Form  $A * x = b$ :

$$({}^F_R R_i - ({}^F_R R_j) * {}^{TCP}_F t) = {}^F_R t_j + {}^F_R R_j * {}^{TCP}_F R * t_{CSj} - {}^F_R t_i - {}^F_R R_i * {}^{TCP}_F R * t_{CSi} \quad (6.9)$$

Die Variable  $x$  ist anders als bei der Bestimmung des rotatorischen Anteils der Werkzeugspitze ein  $3 \times 1$  Vektor anstatt eine  $3 \times 3$  Matrix. Um Fehlereinflüsse zu minimieren, bietet es sich auch in diesem Fall an, möglichst viele Roboterposen heranzuziehen und dann ein überbestimmtes Gleichungssystem zum Beispiel mit dem Verfahren der kleinsten Fehlerquadrate zu minimieren.

#### 6.1.1.3. Bestimmung von Kugelzentren

In den vorangegangenen Abschnitten wurde davon ausgegangen, dass der Abstand vom Sensor-TCP zum Kugelzentrum  $t_{CS}$  bekannt ist. Tatsächlich muss dieser jedoch ebenfalls in den jeweiligen Sensordaten bestimmt werden. Das Vorgehen beim Weißlichtinterferometer unterscheidet sich dabei von dem bei dem LLS-Sensor, da der WLI-Sensor in seinem Messbereich dreidimensionale Daten der Kugeloberfläche liefert und der LLS-Sensor nur zweidimensionale Daten in seiner Laserebene aufnimmt (Abbildung 6.1). Im Folgenden wird das jeweilige Vorgehen bei der Bestimmung des Kugelzentrums für beide Sensoren genauer vorgestellt.

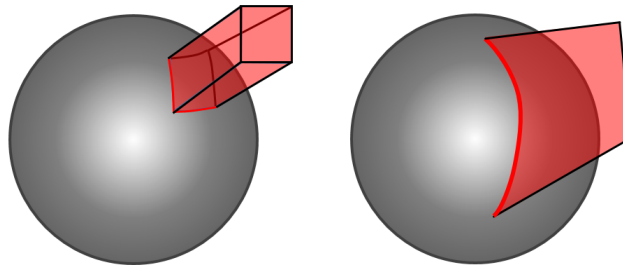


Abbildung 6.1.: Schematische Messbereiche der unterschiedlichen Sensoren auf dem Kugeltarget: links WLI, rechts LLS

**LLS-Sensor:** Bei der Bestimmung eines Kugelzentrums in den zweidimensionalen Daten eines LLS-Sensors muss beachtet werden, dass eine eindeutige Bestimmung des Kugeltarget-Zentrums nur dann möglich ist, wenn bekannt ist, von welcher Kugelhalbseite die entsprechend akquirierten Daten der Oberfläche stammen. Liegt diese Information vor, kann das Zentrum über einfache geometrische Zusammenhänge bestimmt werden. Jede Linie, die auf der Oberfläche einer Kugel liegt, beschreibt einen Kreisbogen. Aus der Kreisgleichung 6.12 ist ersichtlich, dass für die Bestimmung der drei Unbekannten  $m_1, m_2, r$  eigentlich nur drei Punkte auf dem Kreisbogen für drei unabhängige Gleichungen benötigt werden.

$$(x - m_1)^2 + (y - m_2)^2 = r^2 \quad (6.10)$$

Für eine robuste Bestimmung des Kugelzentrums ist es jedoch zu empfehlen, wiederum ein Ausgleichsproblem der Fehlerquadraten über alle  $n$  Punkte der Punktwolke  $p \in \mathbb{R}_n$  der Linie mit einem geeigneten Startwert anzuwenden. Die Form des überbestimmten Problems sieht dabei wie folgt aus:

$$\frac{1}{2} \sum_{i=1}^m ((p_i(x, y, r))^2 = \frac{1}{2} \|p(x, y, r)\|_2^2 \quad (6.11)$$

Werden hier die Parameter der Kreisgleichung eingesetzt, resultiert das folgende Problem:

$$\min \rightarrow \sum_{i=1}^n (\sqrt{(x_i - m_1)^2 + (y_i - m_2)^2} - r)^2 \quad (6.12)$$

Zur Lösung dieses Problems können unterschiedliche nichtlineare Optimierungsalgorithmen angewendet werden. Ein Beispiel ist der Levenberg-Marquardt-Algorithmus. Dieser bestimmt iterativ ein Minimum und kombiniert dabei die Vorteile der steepest-descent-Methode, die global konvergiert, mit der schnellkonvergierenden Gauß-Newton-Methode. Je nachdem wie weit das aktuelle Ergebnis eines Iterationsschrittes von einem Optimum entfernt liegt, verhält sich dieser Algorithmus wie jeweils einer der beiden Verfahren.

**WLI-Sensor:** Wenn jeder Pixel einer WLI-Aufnahme valide Abstandsinformation enthält, besteht diese aus circa 81.000 dreidimensionalen Einzelpunkten. In dieser Punktwolke muss das Kugeltargetzentrum bestimmt werden. Hierbei kann analog zu der 2D-Kreisvariante vorgegangen werden. Es kann also ebenfalls die Methode der Fehlerquadraten in einem überbestimmten, nichtlinearen Optimierungsproblem eingesetzt werden. Das Problem ist dann auch über den Levenberg-Marquardt-Algorithmus zu lösen. Anstelle der Kreisgleichung wird hierbei dann die Kugelgleichung in die Formel der Fehlerquadrateminimierung eingesetzt.



Dies führt zu der folgenden Problemformulierung, die dann zusammen mit einem geeigneten Startwert und den  $n$  Pixeln als Eingangsgröße in den Levenberg-Marquardt Algorithmus gegeben wird:

$$\min \rightarrow \sum_{i=1}^n (\sqrt{(x_i - m_1)^2 + (y_i - m_2)^2 + (z_i - m_3)^2} - r)^2 \quad (6.13)$$

### 6.1.2. Dokumentation des Kalibrierergebnisses

Bei jeder Kalibrierung einer Werkzeugspitze wird zu Dokumentationszwecken automatisch ein Protokoll der Ergebnisse angelegt und zusammen mit anderen Dateien in einem zugehörigen Prozessordner auf der Festplatte eines Steuerrechners abgelegt. Hierdurch können auch zu einem späteren Zeitpunkt alte Kalibrierungen betrachtet und Ergebnisse ausgewertet werden. Zur leichteren Bewertung der Qualität eines Ergebnisses durch einen Anwender können diese Protokoll-dateien zusätzlich in einem eigenen Dialog von *AutoXIV* importiert und dort wie in Abbildung 6.2 visualisiert werden.

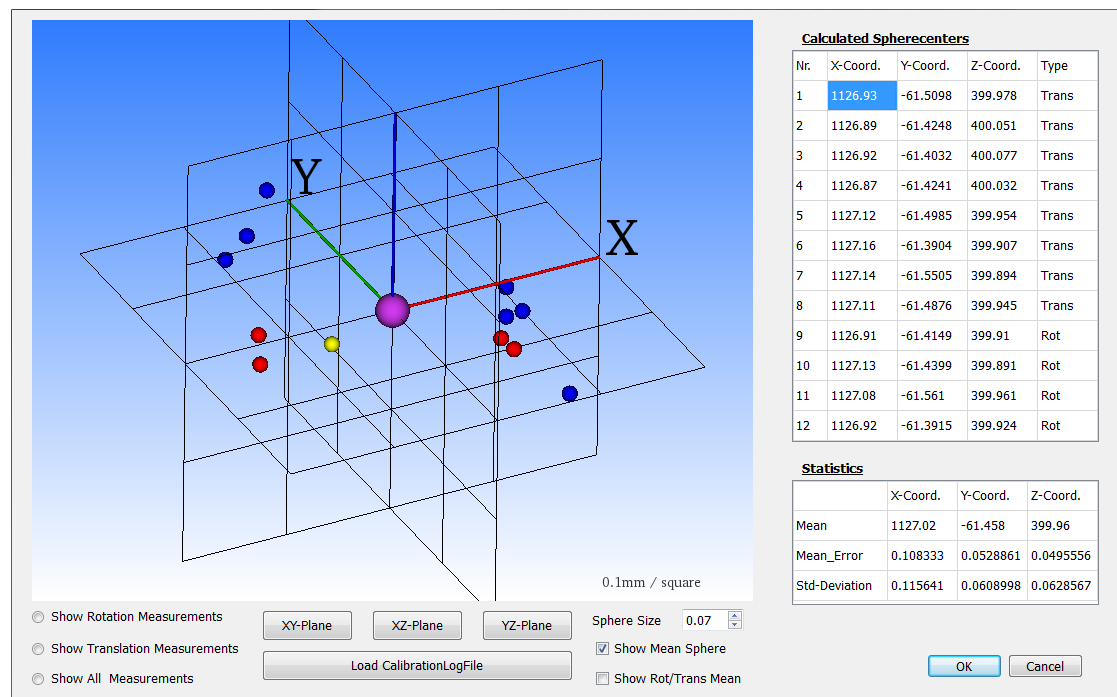


Abbildung 6.2.: Kalibrierergebnis visualisiert in AutoXIV

In diesem Dialog kann das Kalibrierergebnis an allen entsprechenden Messpositionen einzeln visualisiert werden. Hierzu wird die Roboterpose und das geschätzte Kugelzentrum an der jeweiligen Messposition herangezogen und mit dem in der Protokolldatei hinterlegten TCP transformiert. Wenn eine Kalibrierung erfolgreich war und ein TCP mit einem hohen Fitnesswert geschätzt worden ist, sollten die Werte aller Messpositionen nahe beieinanderliegen. Mittels der

vorhandenen Skala kann deren Abstand zueinander direkt abgelesen werden. Tritt bei einem der Punkte während der Kalibrierung ein Fehler auf, würde dieser Punkt durch einen großen Abstand zu den anderen Werten auffallen.

### 6.1.3. Kalibrierung der Drehachsbasis

Die Kalibrierung der Drehachsbasis ist aktuell noch nicht mit in den automatischen Inspektionsablauf integriert, sondern muss stattdessen manuell angestoßen werden. Dabei ist zu beachten, dass das für die Kalibrierung der Drehachsbasis verwendete Werkzeug selber vorher kalibriert werden muss. Die Qualität der Kalibrierung der Drehachsbasis ist dabei natürlich abhängig von der Qualität der vorherigen Kalibrierung des verwendeten Werkzeugs.

Für die Kalibrierung der Drehachsbasis bietet *AutoXIV* ebenfalls einen eigenen Dialog, der vom Nutzer aufgerufen werden kann. Hier wird das zu nutzende Werkzeug, der entsprechende Kugeltargetradius und eine Kalibrierstrategie ausgewählt und im Anschluss der Kalibrierprozess gestartet. Vorher muss das zur Kalibrierstrategie passende Kugeltarget allerdings an die Werkstückaufnahme an der Drehachse montiert werden. Im Zentrum dieser Drehachse ist die Drehachsbasis definiert. Der Ablauf des Kalibrierprozesses sieht vor, dass die Drehachse sukzessive in drei unterschiedliche Positionen gedreht wird, während der IR gleichzeitig den Sensor vor das Kugeltarget positioniert. Haben Achse und IR ihre entsprechenden Positionen erreicht, werden die Aufnahmen des Kugeltargets ausgelöst und abgespeichert. Sind alle benötigten Daten akquiriert, startet der entsprechende Kalibrier-Handler die Auswertung in der zugehörigen Arbeiterklasse. Das Ergebnis dieser Auswertung ist dann eben die Position und Orientierung des Drehachszenentrums relativ zum Roboter-KS. Wie die Bestimmung im Detail durchgeführt wird, zeigen die nachfolgenden Abschnitte sowie Abbildung 6.3.

Zwischen den drei bestimmten Kugelzentren  $p_1, p_2$ , und  $p_3$ , die auf der Rotationsebene  $E_3$  liegen, werden zwei Vektoren  $\vec{v}_1$  und  $\vec{v}_2$  gebildet. Von diesen Vektoren werden im Anschluss über die Formel  $m_i = p_i + 0.5 \cdot \vec{v}_i$  deren Mittelpunkte bestimmt. Mit Hilfe der Hesseschen-Normalenform  $\vec{x} \cdot \vec{n}_0 - d = 0$  werden in diesen Mittelpunkten wiederum zwei Ebenen  $E_1$  und  $E_2$  definiert. Hierfür wird zusätzlich zu  $m_1$  und  $m_2$  die jeweilige Ebenennormale benötigt. Für die Ebenen  $E_1$  und  $E_2$  sind das entsprechend  $\vec{v}_1$  und  $\vec{v}_2$ , für  $E_3$  das Kreuzprodukt der beiden  $\vec{n} = \vec{v}_1 \times \vec{v}_2$ . Der Schnittpunkt dieser drei Ebenen stellt dann genau das Achsenzentrum  $z$  der zu bestimmenden Drehachse dar. Die Richtung der Drehachsebene ist  $\vec{n}$ .

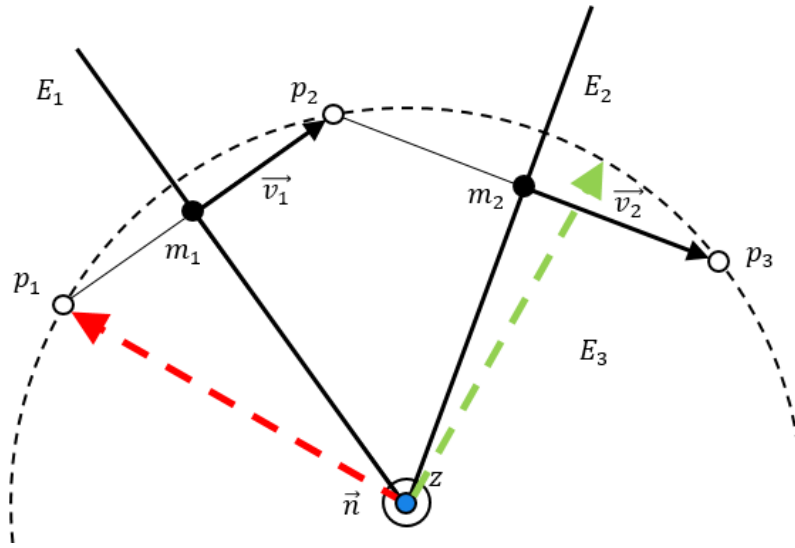


Abbildung 6.3.: Variablen bei der Kalibrierung der Werkstückbasis

Zur Berechnung des Schnittpunktes werden die drei Gleichungen der Hesseschen Normalenform in ein lineares Gleichungssystem der Form  $A \cdot x = b$  eingefügt, das sich dann wie folgt darstellt:

$$\begin{pmatrix} v_1(x) & v_1(y) & v_1(z) \\ v_2(x) & v_2(y) & v_2(z) \\ n(x) & n(y) & n(z) \end{pmatrix} \cdot \begin{pmatrix} z(x) \\ z(y) \\ z(z) \end{pmatrix} = \begin{pmatrix} \vec{v}_1^T \circ m_1 \\ \vec{v}_2^T \circ m_2 \\ \vec{n}^T \circ p_1 \end{pmatrix} \quad (6.14)$$

Damit die Drehachsbasis vollständig bestimmt ist, müssen zu den drei kartesischen Koordinaten des Achsenzentrums ebenfalls dessen drei rotatorische Komponenten beziehungsweise dessen drei Raumachsen bestimmt werden. In dieser Betrachtung kann eine beliebige Ausrichtung der X- und Y-Achse gewählt werden. Für die X-Achse wird in diesem Fall der Ortsvektor  $\vec{p}_1$  gewählt. Die Z-Achse der Werkstückbasis ist durch den Normalenvektor  $\vec{n}$  der Ebene  $E_3$  vorgegeben. Durch die Bildung des Kreuzproduktes zwischen der X-Achse mit der Z-Achse wird die Y-Achse bestimmt. Das daraus folgende vollbestimmte Koordinatensystem ist im Zentrum von Abbildung 6.3 dargestellt.

## 6.2. Bauteildigitalisierung mit dem Laserlichtschnittverfahren

Die konzipierte Strategie zur Digitalisierung der Bauteiloberfläche ist ebenfalls in *AutoXIV* integriert. Sie kann wahlweise als alleinstehender Prozess manuell angefordert werden oder als Teil des automatischen Ablaufs vom zuständigen Zustandsautomaten gestartet werden. In beiden Fällen

wird entweder manuell oder automatisch ein Roboterpfad geladen und im Anschluss sukzessive punktweise abgefahren. Das Konzept sieht vor, dass der Roboter den Sensor wie in Abbildung 6.4 positioniert und dann im Anschluss die Achse verfahren wird.

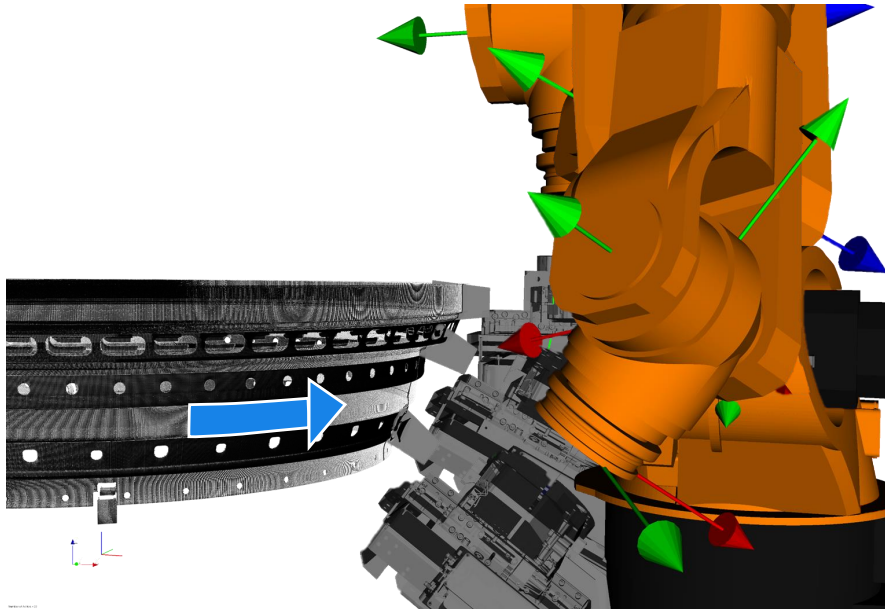


Abbildung 6.4.: Überlagerung verschiedener Messpositionen zur Veranschaulichung des sukzessiven Vorgehens bei der Digitalisierung

Beim Verfahren der Drehachse sind bei der Implementierung zwei zentrale Anforderungen zu beachten. Dies ist zum einen eine *Datensynchronisation*, bei der die Liniendaten den korrekten Positionsdaten von Roboter und Achse zugeordnet werden, und zum anderen eine möglichst hohe *Homogenität* der resultierenden Punktwolke. Die Punktwolke bildet die Grundlage für Algorithmen in den anschließenden Subprozessen der Inspektion und der Rissdetektion, weswegen die Qualität der Daten eine entscheidende Rolle spielt. Im Folgenden werden die Umsetzung von Lösungen zu diesen beiden Problemstellungen vorgestellt.

### 6.2.1. Datensynchronisation

Über die Echtzeitschnittstelle RSI der *KUKA*-Steuerung kann ein Zielrechner über einen beim Verbindungsaufbau festgelegten Netzwerkport und das Netzwerkprotokoll UDP im 4 ms-Takt verschiedene Informationen im XML-Format anfordern. Zu diesen Informationen gehören neben dem Zustand der Ein- und Ausgänge auch der Status von internen Registern sowie die aktuelle Pose des Roboters im aktuellen Werkzeug- und Base-KS. Die Drehachse liefert ebenfalls über UDP aktuelle Drehwinkel, Drehgeschwindigkeit und den Status ihrer internen Flags und Register. Bei der Achse beträgt der Sendetakt allerdings 20 ms. Kommandos in die eine sowie Drehwinkelinformationen in die andere Richtung werden dabei in einfachen American Standard

Code for Information Interchange (ASCII)-Wörtern übertragen. Der LLS-Sensor wiederum sendet seine Daten nach Anfrage im XML-Format über eine eigene TCPIP-Verbindung. Für den Prozess der Datenregistrierung muss in dieser Arbeit also ein Konzept entworfen werden, um drei unterschiedliche Datenquellen mit drei verschiedenen Sendetakten miteinander zu synchronisieren. Für die Synchronisation gibt es unterschiedliche Ansätze. Zum einen können zeitbasierte Synchronisationsmethoden eingesetzt werden. Beispiele hierfür sind das Network Time Protocol (NTP), der Reference Broadcast Synchronisation (RBS) oder das Timing-Sync Protocol for Sensor Networks (TPSN) (SY04). Beim zeitbasierten Ansatz muss allerdings schon bei der Auswahl der Komponenten darauf geachtet werden, dass sie über eine eigene interne Uhr verfügen. Diese Uhren müssen dann kontinuierlich miteinander aufsynchronisiert werden, um Offset- sowie Driftfehler zu vermeiden. Der eingesetzte LLS-Sensor verfügt nicht über eine eigene interne Uhr, weswegen diese Synchronisationsmethoden nicht weiter berücksichtigt werden konnten. Zum anderen können nicht-zeitbasierte Ansätze verfolgt werden. Bei diesen Ansätzen sind vor allem die triggerbasierten Verfahren für die vorliegende Applikation interessant, da mit diesen Verfahren vergleichsweise einfach unterschiedliche Sendetakte miteinander synchronisiert werden können und alle Komponenten über die notwendigen Schnittstellen verfügen. Mögliche Strukturen bei diesem Ansatz sind die Primary-/Secondary-Triggerung oder die Independent-Signal-Triggerung, wobei erstere dem vorliegenden Datennetzwerk aufgrund seiner Inhomogenität bei den Sendefrequenzen besser entspricht. Bei einem Trigger-Ansatz bietet der Sensor allerdings nur noch eine Sendefrequenz von ca. 30 Hz, was einem Sendetakt von ca. 30 ms entspricht. Deswegen übernimmt dieser in der ausgewählten Struktur die Funktion des Trigger-Masters. Eine Messung wird bei dem Sensor über einen Trigger aus der Steuersoftware *AutoXIV* ausgelöst. Liegen neue Liniendaten vor, werden anschließend Positionsdaten von Roboter und Drehachse angefordert und mit dieser Linie zu einem lokalisierten Datenpaket zusammengefügt. Wenn das Datenpaket abgespeichert ist, wird der nächste Trigger-Zyklus begonnen. Der maximal mögliche Winkelfehler  $\Delta\phi$  eines synchronisierten Datenpaketes ist bei diesem Vorgehen abhängig von dem langsamsten Sendetakt der Handlingkomponenten  $t_{max}$ . Das ist in dem betrachteten Fall die Drehachse. Jitter bei der Kommunikation wird in dieser Betrachtung vernachlässigt. Der Winkelfehler ergibt sich dann wie in Formel 6.2.1 aus dem Winkel, der die Achse mit ihrer Winkelgeschwindigkeit  $v_\phi$  in der Zeit  $t_{max}$  überstrichen haben kann. Auf die Winkelgeschwindigkeit der Achse wird in den folgenden Abschnitten genauer eingegangen.

$$\Delta\phi = \frac{v_\phi}{t_{max}} \quad (6.15)$$

## 6.2.2. Homogenität der Oberflächendaten

Bei der homogenen Verteilung der Punkte auf der Modelloberfläche spielen vor allem die Parameter Winkelgeschwindigkeit der Drehachse und Zuverlässigkeit der Datenakquirierung eine zentrale Rolle. Die Winkelgeschwindigkeit muss mindestens so hoch gewählt werden, dass die Minimalanforderungen der nachfolgenden Prozessschritte an die Auflösung entlang des Umfangs eingehalten werden können. Gleichzeitig muss programmatisch sichergestellt sein, dass der Prozess der Datenakquirierung zuverlässig abläuft, ohne dass er durch andere Prozesse gestört oder gar unterbrochen wird. Für beide Parameter wird im Folgenden eine Lösung vorgestellt.

### 6.2.2.1. Achsgeschwindigkeit

Aufgrund des selektierten Triggerverfahrens bei der Datensynchronisation ist die Messfrequenz  $f_M$  des LLS-Sensors bei der Digitalisierung auf ca. 30 Hz reduziert. Die minimale horizontale Auflösung, also der maximale Abstand zwischen zwei benachbarten Punkten  $s_{max}$ , wird durch den kleineren der beiden benötigten Werte bei der Pfadplanung beziehungsweise der Merkmalsextraktion in der Rissdetektion vorgegeben. Bei dem vorliegenden Bauteilradius  $r_B = 350$  mm ergibt das einen Achswinkel  $\varphi_{max}$  in  $[\circ]$  von:

$$\varphi_{max} = \frac{s_{max}}{2 \cdot \pi \cdot r} \cdot 360^\circ \quad (6.16)$$

Die maximale Winkelgeschwindigkeit  $v_\phi$  in  $\left[\frac{\circ}{s}\right]$  ergibt sich dann zu:

$$v_\phi = \varphi_{min} \cdot f_M \quad (6.17)$$

Die Rissdetektion stellt in dem Inspektionssystem die höhere Anforderung an die Auflösung bei der Digitalisierung. Sie fordert einen Maximalabstand zweier Punkte von 0.5 mm. Das entspricht einer maximalen Achsgeschwindigkeit von  $5.9^\circ \text{ s}^{-1}$ . Hierzu wird die Achse mit 60% ihrer Maximalgeschwindigkeit verfahren. Auf der Linie wird für die vertikale Auflösung entsprechend nur jeder 6. Punkt der insgesamt 476 Punkte benötigt, um eine homogene Punkteverteilung zu erreichen. Das Ergebnis bei falschen Parametersätzen zeigt die Abbildung 6.6. Bei dieser relativ hohen Verfahrensgeschwindigkeit kann es allerdings passieren, dass bei dem begrenzten internen Sendetakt der Achse zwischen zwei Triggern des LLS-Sensors noch kein neuer Winkelwert vorliegt, zwei Linien somit der gleiche Drehwinkel zugeordnet wird und es wiederum zu Inhomogenitäten bei der Punktedichte kommen würde (siehe Abbildung 6.6c). Deswegen wird bei dem Konzept für die Digitalisierung ausgenutzt, dass sich die Achse außerhalb ihrer Beschleunigungsphasen am Anfang und Ende einer Verfahrbewegung mit einer konstanten und bekannten Geschwindigkeit dreht und die Bewegung in diesem Bereich linear erfolgt (siehe Graphen in Abbildung 6.5). Dies ermöglicht eine einfache Extrapolation ihrer Drehwinkel.



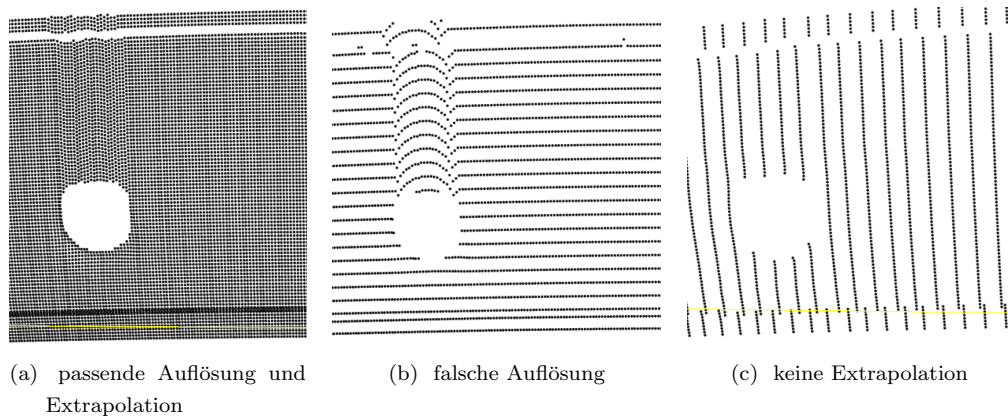


Abbildung 6.6.: Einfluss von Auflösung und Extrapolation auf Datenhomogenität

#### 6.2.2.2. Datenzugriff

Damit der Prozess der Digitalisierung in *AutoXIV* visualisiert werden kann, müssen die lokalisierten Linien während der Aufnahme in eine Visualisierungsstruktur der Bibliothek VTK abgelegt werden. Aufgrund der hohen Messfrequenz bei der Digitalisierung muss sichergestellt werden, dass die Datenaufnahme nicht durch andere Prozesse gestört wird. Solche Störungen würden zu Lücken in dem resultierenden Modell führen, die sich wiederum negativ auf die anschließenden Algorithmen auswirken würden. Deswegen wird das Schreiben der Daten in eine Datenstruktur und das anschließende Prozessieren in einer separaten Klasse ausgeführt sowie durch „Mutexe“ geschützt. Mutexe sperren in einer objektbasierten Multithread-Applikation Objekte, auf die aus unterschiedlichen Threads zugegriffen wird, so dass kein Mehrfachzugriff und ein nichtdeterminierter Zustand dieses Objektes entstehen.

Die Datenaufnahme läuft asynchron in einem eigenen Thread. Nach jeder aufgenommenen Linie fragt diese Aufnahmeklasse über den Mutex der Pufferklasse ab, ob auf seine entsprechende Speicherstruktur gerade zugegriffen wird. Ist dies der Fall, wird die aktuelle Linie direkt im Aufnahmethread zwischengespeichert und die nächste Linie angefordert. Ist die Struktur für den Zugriff frei, werden alle im Aufnahmethread angesammelten Linien in die Speicherstruktur der Pufferklasse abgelegt. Liegt in dieser Pufferklasse eine vorher festgelegte Anzahl von Linien vor, werden diese in die VTK-Struktur der Visualisierungsklasse verschoben und können danach in die entsprechende VTK-Render-Pipeline übergeben und angezeigt werden. Abbildung 6.7 zeigt das beschriebene Vorgehen mit der Pufferklasse in einem Flussdiagramm. Die verschiedenen Threads sind in der Abbildung durch unterschiedliche Farben gekennzeichnet.



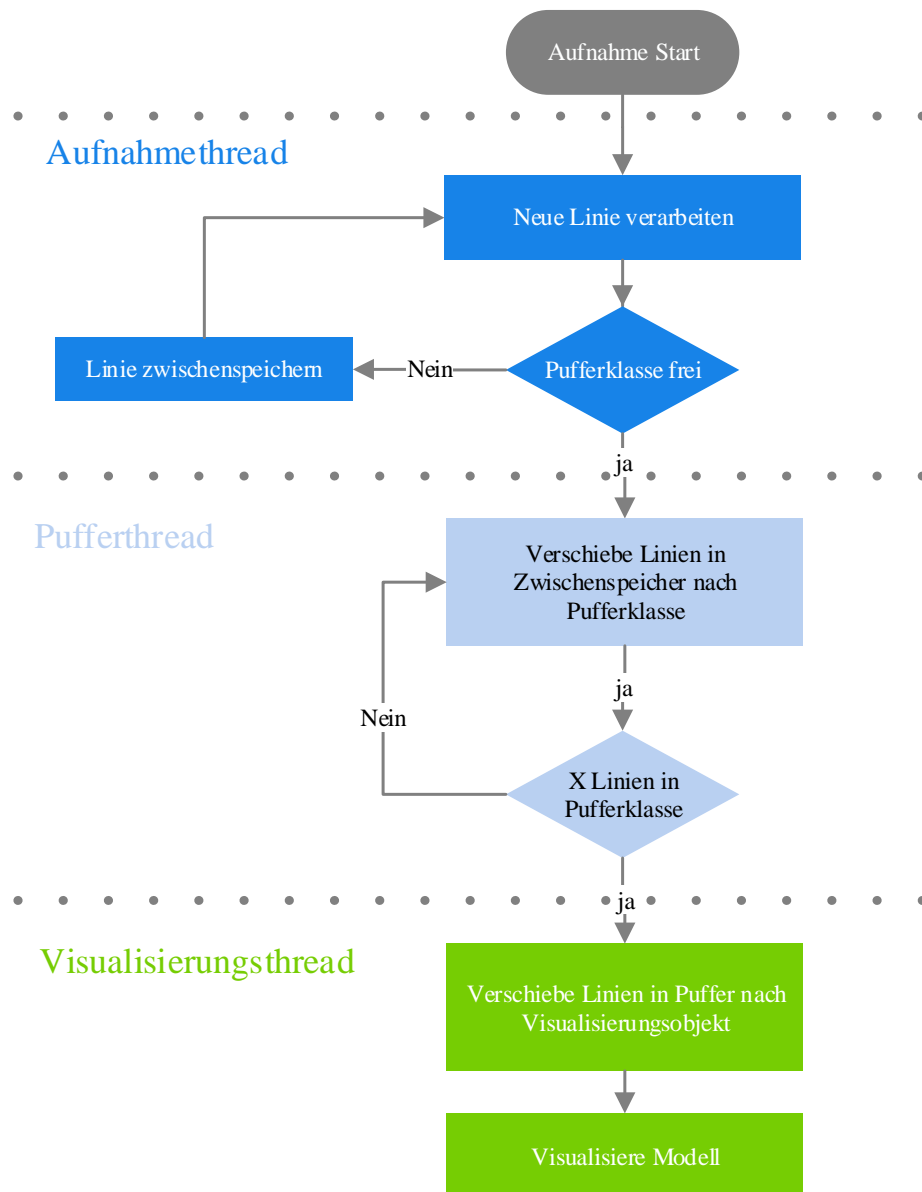


Abbildung 6.7.: Ablauf bei asynchroner Datenaufnahme und Visualisierung  
in dem Prozessschritt der Digitalisierung

### 6.3. Pfadplanung für die Inspektion mit einem WLI

Der in dieser Arbeit eingeführte Ansatz zur Pfadplanung bei der Inspektion mit einem WLI sieht, wie in Abbildung 6.8 vereinfacht dargestellt, ein zweistufiges Vorgehen vor. Im ersten Schritt sollen Symmetrien des Bauteils ausgenutzt werden, um die zu große Punktemenge des digitalisierten Modells zu reduzieren. Auf diesem gefilterten Modell soll dann im Anschluss eine modellbasierte Pfadplanung durchgeführt werden. Das Ergebnis ist ein lokal geplanter Inspektionspfad. Im zweiten Schritt wird dieser Pfad wiederum herangezogen, auf die globale Geometrie des Bauteils optimiert, um den gesamten Umfang vervielfacht und dann sukzessive einer anschließenden Pfadadaption auf die Realoberfläche zugeführt. Eine detaillierte Übersicht des gesamten Ablaufs mit allen Prozessschritten zeigt Abbildung 6.23 am Ende dieses Abschnitts.

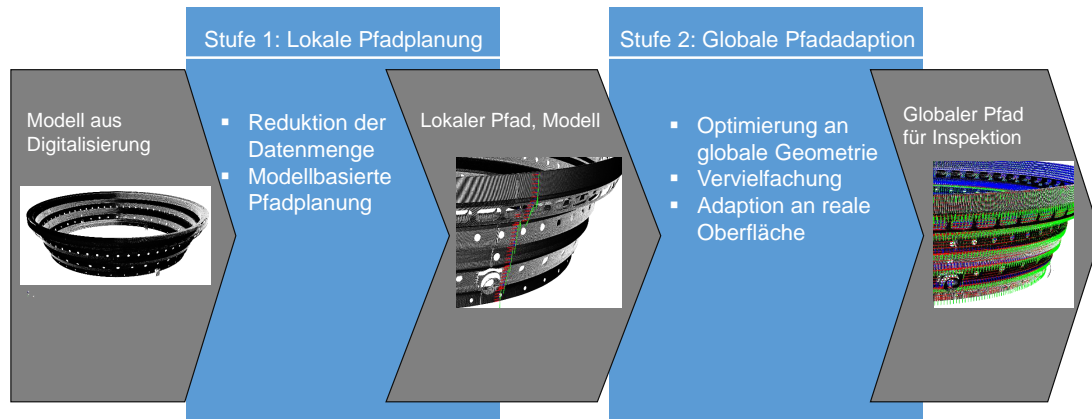


Abbildung 6.8.: Die zwei Stufen des neuen Ansatzes für die Inspektionsplanung

Der erste Schritt der Inspektionsplanung ist noch nicht in den automatischen Inspektionsprozess integriert. Ein voll funktionsfähiger Prozess ist allerdings in *AutoXIV* implementiert. Dieser kann manuell vom Systembediener gestartet werden. Das modellbasierte Konzept benötigt für die Inspektionsplanung als einzige Ressource ein Modell des Bauteils. Dieses muss manuell in *AutoXIV* vom Nutzer importiert und dann explizit ausgewählt werden. Zusätzlich zu dem Modell müssen in der zugehörigen Toolbox die entsprechende Messbereichsgeometrie sowie der zugehörige Messabstand angegeben werden. Danach kann die Pfadplanung manuell vom Benutzer gestartet werden.

Der zweite Schritt der Pfadadaption ist schon heute Bestandteil der automatischen Inspektion, kann aber ebenfalls manuell und losgelöst vom automatischen Prozess genutzt werden. Am Eingang der Adaption wird ein Roboterpfad und das Modell des Bauteils aus dem ersten Schritt erwartet.

Im Folgenden wird die Ausgestaltung und Implementierung der vorher entwickelten Konzepte für beide Schritte der Pfadplanung vorgestellt. Der Fokus liegt dabei neben dem genauen Ablauf auf den Funktionen und Algorithmen, auf denen diese Schritte basieren.

### 6.3.1. Stufe 1: Modellbasierter Ansatz zur lokalen Pfadplanung

Der modellbasierte Schritt der Pfadplanung setzt sich analog zu Scotts Vorgehen beim 3M-Ansatz aus einer Reihe von Subprozessen zusammen. Die Subprozesse sowie ihre Reihenfolge veranschaulicht Abbildung 4.7 im entsprechenden Konzeptionskapitel. Im Folgenden werden die vier Teilschritte mit den für diesen Prozess benötigten Anpassungen jeweils einzeln im Detail vorgestellt.

#### 6.3.1.1. Gewinnung von Oberflächenkandidaten

Eine Voraussetzung für die Anwendung des mengentheoretischen Ansatzes nach Scott in dem vorliegenden Kontext ist, wie einleitend vorgestellt, die Vereinfachung der zu betrachtenden Bauteiloberfläche des großen Oberflächenmodells und die damit verbundene Reduktion des Rechenaufwandes. Durch die Integration einer zusätzlichen Drehachse in das Anlagenkonzept zur Inspektion kann dessen Rotationssymmetrie ausgenutzt werden. Anstatt die Pfadplanung auf dem gesamten Umfang des abstrahierten Hohlkegelstumpfes durchzuführen, wird sie nur auf einem kleinen Segment oder wahlweise nur auf einem Profil des Bauteils durchgeführt. Dieses Segment kann aus dem Bauteil abgeleitet werden, indem man es entweder mit einer Ebene schneidet oder alle Punkte extrahiert, die sich innerhalb eines begrenzten Winkelbereichs des Bauteilumfangs befinden.

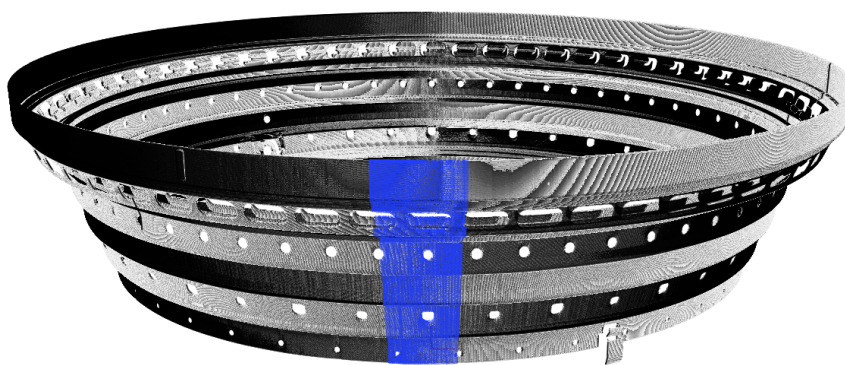


Abbildung 6.9.: Aus dem Gesamtmodell herausgefiltertes Segment des Umfangs

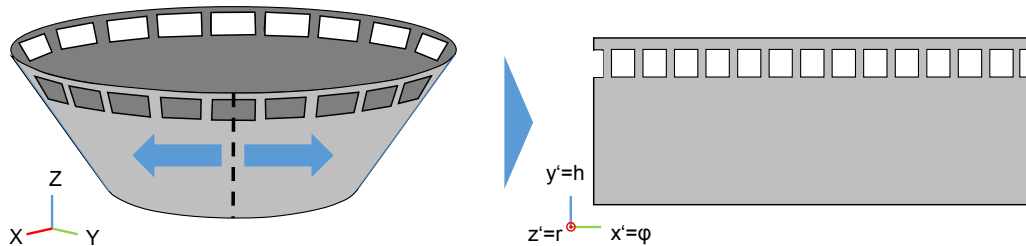
Damit keine Punkte in die Pfadplanung miteinbezogen werden, die durch Messrauschen oder andere zufällige Prozesse in das Modell gelangt sind, bietet es sich an, zu Beginn der Planung

einen einfachen statistischen Rauschfilter auf das reduzierte Bauteilsegment anzuwenden. Statistische Rauschfilter basieren auf dem arithmetischen Mittel und der Standardabweichung normalverteilter Punktwolken bezüglich ihrer Punktedichte. Für jeden Punkt wird dafür eine Anzahl von Punkten in dessen unmittelbarer Nachbarschaft herangezogen und ihr arithmetisches Mittel bezüglich ihrer Distanz bestimmt. Liegt die Entfernung zu einzelnen Punkten oberhalb eines festgelegten Grenzwertes, der einen Faktor der Standardabweichung darstellt, werden diese Punkte entsprechend aus dem Modell entfernt. Bevor die einzelnen Punkte des, von Ausreißern befreiten, Hohlkegelstumpfsegmentes in die Zeilen der zentralen Struktur der MM eingefügt werden, wird eine weitere Vereinfachung vorgenommen. Für jeden Punkt des extrahierten Segmentes werden die entsprechenden Zylinderkoordinaten in kartesische Koordinaten transformiert. Dadurch wird, wie in Abbildung 6.10a, eine Art „Abrollung“ des Hohlkegelstumpfes zu einer Ebene erreicht. Das Modell einer Ebene ist weniger komplex als das eines Kegelstumpfes und Algorithmen lassen sich einfacher und anschaulicher anwenden.

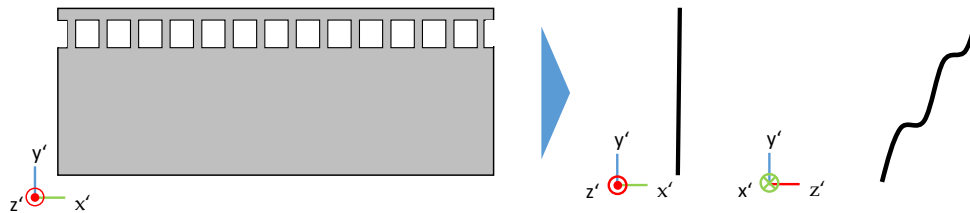
Bei der Transformation werden für jeden Punkt  $p_i$  dessen entsprechende Zylinderkoordinaten bestimmt, die dann wiederum direkt auf  $p'_i$  abgebildet werden. Es gilt dabei folgende Abbildungsregel:

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} \mapsto \begin{pmatrix} \phi_i \\ r_i \\ h_i \end{pmatrix} = \begin{pmatrix} \arccos \left\{ \frac{\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \circ \begin{pmatrix} x_i \\ y_i \\ 0 \end{pmatrix}}{\left\| \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right\| \left\| \begin{pmatrix} x_i \\ y_i \\ 0 \end{pmatrix} \right\|} \right\} \\ \sqrt{x_i^2 + y_i^2} \\ z_i \end{pmatrix} \quad (6.19)$$

Da es sich bei dem betrachteten Bauteil nicht um einen Zylinder, sondern um einen Kegelstumpf handelt und die Mantelfläche durch diese Abbildung in eine rechteckförmige Ebene transformiert wird, kommt es zu einer geometrischen Verzerrung des Segmentes. Nach der Transformation in den kartesischen Raum wird das abgerollte Segment noch weiter vereinfacht und dadurch der Rechenaufwand noch weiter reduziert. Alle Punkte werden auf die  $y'z'$ -Ebene bei  $x' = 0$  abgebildet. Hierdurch wird aus dem dreidimensionalen Segment ein zweidimensionales Profil des Bauteils und gleichzeitig aus einem dreidimensionalen Problem ein zweidimensionales. Dadurch werden allerdings Löcher des Bauteils geschlossen und somit bei diesem Schritt der Pfadplanung vernachlässigt. Dieses Procedere wird nur durch den in dieser Arbeit entwickelten, zweistufigen Ansatz ermöglicht, weil genau diese Merkmale in der zweiten Stufe Berücksichtigung finden.



(a) Schema der Bauteiltransformation vom Zylinderraum in den kartesischen Raum



(b) Schema der Abbildung vom dreidimensionalen in den zweidimensionalen Raum

Abbildung 6.10.: Vereinfachung durch Bauteiltransformation

### 6.3.1.2. Gewinnung von Sensorpositionskandidaten

Für die Gewinnung von Sensorpositionskandidaten gibt es bei der modellbasierten Pfadplanung in der Literatur grundsätzlich zwei unterschiedliche Vorgehensweisen. Bei der ersten wird das gesamte Bauteil mit einer Hüllgeometrie umschlossen. Diese Geometrie wird anschließend zu einzelnen Punkten diskretisiert. Ein Beispiel hierfür ist die im Stand der Technik erwähnte „View-sphere“ aus Abbildung 2.4. Aus den diskretisierten Punkten werden dann einzelne SPK abgeleitet und zusätzlich zu den Objektkandidaten in die MM eingefügt. Hierbei muss die Hüllgeometrie so gewählt werden, dass sie die grobe Geometrie des zu inspizierenden Bauteils abbildet, damit eine gleichmäßige Verteilung von SPK gegeben ist und so eine vollständige Inspektion durchgeführt werden kann. Dies ist weniger wichtig in Anwendungsfällen, bei denen ein vollständiger Inspektionspfad aus nur wenigen Messpunkten besteht, aber umso mehr bei Anwendungsfällen wie dem vorliegenden mit dem speziellen Größenverhältnis von Messbereich und Bauteil.

Die zweite generelle Vorgehensweise sieht vor, dass die SPK direkt aus den Objektkandidaten abgeleitet werden. Dieses Vorgehen hat gegenüber der ersten Variante den Vorteil, dass die SPK natürlicherweise der Geometrie des Bauteils folgen und dadurch eine gleichmäßige Verteilung

erzielt werden kann. Die Winkelgeschwindigkeit der Drehachse in der Digitalisierung ist so gewählt, dass das resultierende Modell eine möglichst homogene Verteilung der Punkte auf der Modelloberfläche aufweist, was das direkte Ableiten von SPK aus diesen Daten unterstützt. Eine mögliche Umsetzung dieses Ansatzes ist die Ausnutzung der Tatsache, dass die betrachteten Brennkammern annähernd eine stetige Bauteiloberfläche aufweisen und deswegen ein gutes Ergebnis bei einer Schätzung von Normalen für die Oberflächenpunkte zu erwarten ist. Hierfür wird eine Anzahl  $m$  an Nachbarpunkten herangezogen (Rus13). Der Rechenaufwand steigt mit der Anzahl der betrachteten Punkte. Eine effektive Weise, Nachbarpunkte in einer Punktwolke zu bestimmen, ist der Einsatz einer Suchbaumstruktur wie zum Beispiel dem kd-Baum. Aus diesen Nachbarpunkten wird dann über eine Hauptkomponentenanalyse eine implizite Oberfläche geschätzt. Dafür wird zuerst die Kovarianzmatrix  $K$  der betrachteten Punkte mit dem Schwerpunkt  $\bar{p}$  wie folgt aufgestellt:

$$K = \frac{1}{k} \sum_{i=1}^m (p_i - \bar{p}) (p_i - \bar{p})^T \in \mathbf{R}_{3 \times 3} \quad (6.20)$$

An den Eigenwerten dieser Kovarianzmatrix kann abgelesen werden, in welcher Raumrichtung die betrachtete Punktemenge welche Ausprägung hat. Die zwei Hauptachsen dieser Menge sind durch die zwei Eigenvektoren mit den größten Eigenwerten gegeben und liegen auf der zu bestimmenden Ebene. Die Richtung der Ebenennormalen ist also der Eigenvektor mit dem kleinsten Eigenwert. Bei symmetrischen Matrizen wie der Kovarianzmatrix sind die Eigenwerte gleich ihren Singulärwerten. Diese kann man über eine Singulärwertzerlegung bestimmen, bei der die Matrix  $M$  in die drei Matrizen  $U, V, \Sigma$  aufgeteilt wird:

$$M = U \cdot \Sigma \cdot V^* \quad (6.21)$$

Die Matrix  $U$  besteht aus den orthonormalen Eigenvektoren von  $M \cdot M^T$  und die Matrix  $V$  aus den Eigenvektoren von  $M^T \cdot M$ . Da die Matrix immer symmetrisch und quadratisch ist, sind diese identisch. Auf der Diagonalen von  $\Sigma$  kann der kleinste Singulärwert von  $M$  abgelesen und der entsprechende Eigenvektor zugeordnet werden (GR70).

Sind die Normalen bestimmt, können diese herangezogen werden, um im Anschluss die SPK  $S_{1...i}$  abzuleiten. Dafür werden die Objektkandidaten  $O_{1...i}$ , nachdem sie in den zweidimensionalen Raum transformiert worden sind, entlang der ihnen zugeordneten Normalen verschoben, sodass dadurch entlang des eigentlichen Bauteilprofils eine Hülle gebildet wird, die ihrer eigentlichen Geometrie entspricht (Abbildung 6.11a). Die SPK könnten dann zu den Objektkandidaten in die Spalten der MM eingetragen werden.

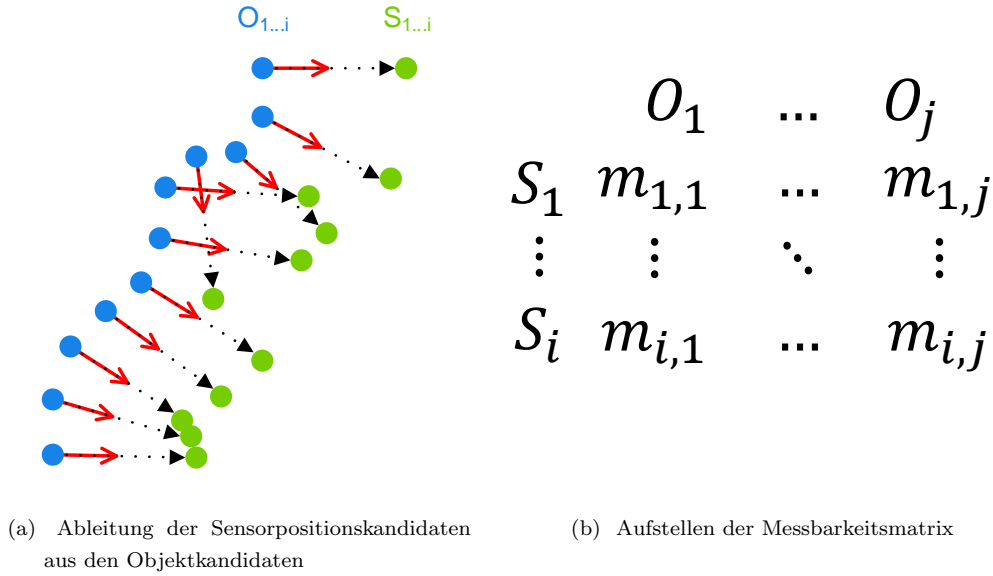


Abbildung 6.11.: Schritte bei der Gewinnung von Sensorpositionskandidaten

### 6.3.1.3. Bestimmung der Messbarkeit

In diesem Schritt werden aus Objekt- und Sensorpositionskandidaten relevante Objektpunkte und Sensorpositionen herausgefiltert und dadurch die MM für die anschließende Extraktion einer Untermenge vereinfacht. Damit ein Objektpunkt generell von einer Sensorposition gemessen werden kann, müssen in einem Inspektionsszenario verschiedene Kriterien erfüllt sein. Sie sind in dem betrachteten Fall abhängig von der eingesetzten Handhabungs- und Sensortechnik, können aber auch von anderen Umgebungsbedingungen abgeleitet werden. Ihr Erfüllungsgrad wird über einen Faktor  $c_x$  quantifiziert. Bei der Handhabungstechnik muss in der hier betrachteten Applikation zum einen beachtet werden, dass eine Sensorposition im Raum mit der eingesetzten Roboterkinematik erreichbar ist ( $c_E$ ), und zum anderen an einer Position liegt, die kollisionsfrei mit dem Sensor zugänglich ist ( $c_Z$ ). Bei dem WLI-Sensor spielt außerdem vor allem der begrenzte Messbereich eine wichtige Rolle ( $c_M$ ). Zusätzlich müssen durch andere Oberflächen verdeckte Bereiche aus der Betrachtung ausgeschlossen ( $c_V$ ) und der Grenzwert für den Winkel zwischen Strahlrichtung und Oberflächennormale eingehalten werden ( $c_S$ ). Sind alle Kriterien erfüllt und somit alle korrespondierenden Faktoren ungleich null, ist auch der entsprechende Matrixeintrag  $m_{ij}$  ungleich null und ein Oberflächenpunkt  $O_i$  ist von der Sensorposition  $S_j$  messbar.

$$m_{ij} = c_E \cdot c_K \cdot c_M \cdot c_V \cdot c_W \quad (6.22)$$

Jeder Objektkandidat, der in der Nähe eines SPK liegt und deswegen für die Betrachtung in

Frage kommt, muss hinsichtlich der aufgezählten Kriterien überprüft werden. Dabei spielt die Reihenfolge bei der Überprüfung eine entscheidende Rolle für die benötigte Durchlaufzeit der Pfadplanung. Weniger komplexe Überprüfungen werden zuerst durchgeführt, damit die Menge der zu betrachtenden Sensorpositions- bzw. Objektkandidaten und dadurch der Aufwand für nachfolgende, komplexere Überprüfungen reduziert wird. Im Folgenden wird das Vorgehen bei den für diesen Prozess selektierten Kriterien vorgestellt. Dabei werden die ersten zwei Kriterien *Erreichbarkeit* und *Kollision* auf dem vollständigen, untransformierten Segment sowie Pfad überprüft. Die Kriterien *Messbereich*, *Verdeckung* und *Messwinkel* werden dahingegen auf dem vereinfachten zweidimensionalen Profil überprüft.

**1. Erreichbarkeit ( $c_E$ ):** Bei dem Messbarkeitskriterium Erreichbarkeit wird überprüft, ob die Pose eines SPK im Raum mit der Kinematik des eingesetzten Roboters erreicht werden kann. Hierzu wird seine entsprechende Rückwärtskinematik und das zugehörige TCP sowie das Werkstück-KS herangezogen. Die dafür benötigten Parameter wie die Achsanordnungen und -längen des jeweiligen Industrieroboters werden dazu aus einer Datei importiert. Anhand dieser Informationen kann im Anschluss für jede Pose im Raum eine entsprechende Rückwärtskinematik berechnet werden. Ist eine Pose erreichbar, ist das Resultat der Rückwärtskinematik eine Achsstellung für jede der sechs Achsen des IR. Eine Pose ist also entweder erreichbar und der entsprechende Faktor  $c_E = 1$  oder nicht erreichbar, was einem Faktor  $c_E = 0$  entspricht.

**2. Kollisionserkennung ( $c_K$ ):** Bei diesem Kriterium wird ein vereinfachtes Modell des Werkzeugs mit einer beliebigen Anzahl von Punktwolkenmodellen auf Kollisionen überprüft. Diese Punktwolken können das Bauteil, eine Werkstückhalterung oder andere Zellkomponenten repräsentieren. Eine Kollision liegt vor, wenn ein Punkt einer der betrachteten Wolken innerhalb des Werkzeugmodells liegt. Das einfache Modell des Werkzeugs beschränkt sich in dieser Umsetzung auf eine Bounding-Box, die das komplette Werkzeug einschließt. Diese Bounding-Box orientiert sich mit dem Werkzeug, sodass sie in ungünstigen Winkellagen des Werkzeuges nicht unnötig groß wird. Diese Art von Bounding-Volume wird in der Literatur Oriented Bounding Box (OBB) genannt und ist in der Abbildung 6.12 aus (Eri08) veranschaulicht.



Die Abbildung zeigt, dass es noch eine Vielzahl komplexerer Bounding-Volumen-Strukturen gibt, die den realen Sensor besser annähern würden. Mit der Komplexität der Hüllgeometrie steigt allerdings gleichzeitig die Rechenkomplexität bei der Bestimmung einer Kollision mit der Struktur und damit zugleich die Rechendauer. Da eine OBB die boxartige Werkzeuggeometrie gut annähert, stellt sie einen guten Kompromiss zwischen Genauigkeit und Komplexität dar und wird in dieser Umsetzung herangezogen.

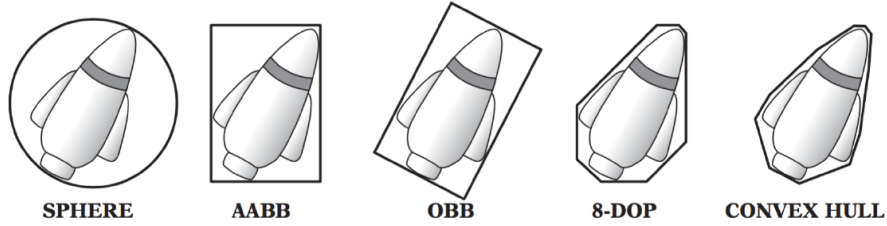


Abbildung 6.12.: Bounding volumes unterschiedlicher Komplexität (Eri08)

Auch bei der Überprüfung auf Kollisionen kommt eine Suchbaumstruktur zur Bestimmung der nächsten Nachbarn zum Einsatz. Im Fall eines 3D-Modells bietet sich ein sogenannter Octree an. Bei einem Octree wird der dreidimensionale Raum in acht Oktanten unterteilt. Die Aufteilung wird rekursiv solange wiederholt, bis wahlweise eine festgelegte Minimalgröße für einen Unterraum oder eine minimale Anzahl an Punkten pro Unterraum erreicht sind. Durch den Einsatz dieser Struktur wird sichergestellt, dass nur diejenigen Punkte auf Kollision untersucht werden, die sich in direkter Nachbarschaft des Werkzeugmodells befinden. Würde die Hüllgeometrie aus einer einfachen Kugel bestehen, würde eine einzelne Operation genügen, um eine Kollision zu überprüfen. Im Fall eines OBB-Test müssen erst das Bounding Volume ausgerichtet und zusätzlich die vier Abfragen 6.23 und 6.24 durchgeführt werden. Der Test läuft so ab, dass zuerst eine Gerade  $\vec{D}_{PC}$  zwischen dem zu testenden Punkt  $P_i$  und dem OBB-Zentrum  $C_{OBB}$  gebildet wird. Diese werden im Anschluss auf die drei Achsen  $\vec{e}_1, \vec{e}_2, \vec{e}_3$  des orthonormalen Koordinatensystem des OBB abgebildet. Sind die Beträge  $u_1, u_2, u_3$  aller drei Abbildungen kürzer als die jeweiligen halben Seitenlängen  $d_1, d_2, d_3$  des OBB, liegt der Punkt innerhalb der Struktur. Ist eine der Überprüfungen allerdings negativ, kann auf die restlichen Tests verzichtet und der aktuelle Punkt als außerhalb markiert werden. Der Faktor  $c_K$  wird dann entsprechend auf 0 gesetzt.

$$\vec{D}_{PC} = \vec{P}_i - \vec{C}_{OBB} \quad (6.23)$$

$$\begin{aligned} \vec{D}_{PC} \circ \vec{u}_1 &\leq d_1 \\ \vec{D}_{PC} \circ \vec{u}_2 &\leq d_2 \\ \vec{D}_{PC} \circ \vec{u}_3 &\leq d_3 \end{aligned} \quad (6.24)$$

**3. Messbereich ( $c_M$ ):** Der Messbereich des WLI-Sensors ist im dreidimensionalen Raum quaderförmig. Bei der Reduktion des Problems auf zwei Dimensionen wird aus diesem Quader ein Rechteck. Anstatt der eigentlichen Messtiefe des WLI-Sensor Messbereichs von nur wenigen Millimetern wird für die Überprüfung des Messbereichs der größtmögliche theoretische Messbereich betrachtet. Dieser entsteht, wenn zusätzlich der gesamte Verfahrbereich der Linearachse des WLI-Sensors miteinbezogen wird. Dadurch vergrößert sich die Messtiefe auf über 50 mm.

Für alle konvexen Vielecke wird bei der Überprüfung des Messbereichskriteriums das gleiche Konzept angewendet. Das Problem reduziert sich hier auf die Bestimmung, ob ein Objektkandidat innerhalb des auf zwei Dimensionen vereinfachten Messbereichs liegt. Dazu wird in einer entsprechenden Schleife für jede Kante des Vierecks überprüft, auf welcher Seite die Gerade liegt, die den Startpunkt der Kante mit dem überprüften Punkt verbindet (siehe Abbildung 6.13). Hierzu wird das Vorzeichen des Kreuzproduktes der beiden Geraden betrachtet. Dabei muss allerdings zusätzlich beachtet werden, dass alle Kanten des Dreiecks gegen den Uhrzeigersinn ausgerichtet sein müssen. Befindet sich ein Punkt auf der linken Seite bei einer der überprüften Kanten, wird die Überprüfung abgebrochen und der Punkt als außerhalb des Messbereichs markiert. Hierzu wird der zugehörige Koeffizient  $c_M$  und damit der zugehörige Matrixeintrag auf 0 gesetzt. Die auf 0 gesetzten Punkte werden bei den weiteren Kriterien nicht mehr berücksichtigt.

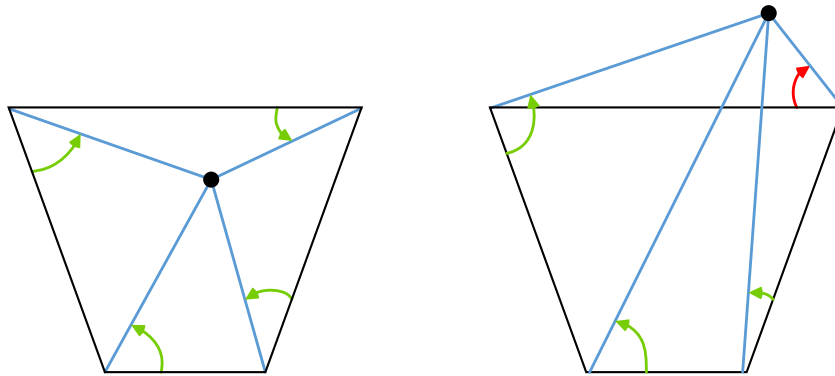


Abbildung 6.13.: Überprüfung des Kriteriums Messbereich. links: Punkt liegt links von allen Kanten und damit im Messbereich, rechts: der Punkt liegt rechts von einer Kante und damit außerhalb des Messbereichs

**4. Verdeckung ( $c_V$ ):** Wird ein Objektkandidat als innerhalb der Messbereichsgeometrie markiert, kann es trotzdem vorkommen, dass er von einem anderen Kandidaten verdeckt wird und deswegen aus der weiteren Betrachtung ausgeschlossen werden muss. Das ist zum Beispiel der Fall bei Punkten, die in Bauteilhinterschnitten liegen und dadurch von anderen Bauteiloberflächen verdeckt sind. Bei Punktmodellen liegen allerdings keine Informationen über Oberflächen vor, die direkt zur Verdeckungsbestimmung herangezogen werden könnten. Stattdessen wird vor dem Start der Pfadplanung ein Wert  $d_{th}$  festgelegt, der beschreibt, wie nah zwei Punkte einander sein müssen, damit zwischen ihnen eine Oberfläche angenommen wird. Wenn von dem aktuellen Pfadpunkt entlang der negativen Strahlrichtung der Sensorposition ein weiterer Punkt gefunden wird, dessen Normalenabstand kleiner ist als der festgelegte Abstandswert  $d_p < d_{th}$ , wird der Objektkandidat als verdeckt markiert und auch entsprechend für die weiteren Überprüfungen ausgeschlossen (Abbildung 6.14).

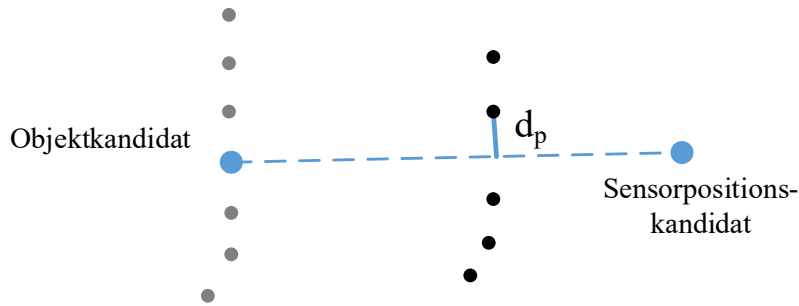


Abbildung 6.14.: Bestimmung der Verdeckung. Gilt  $d_p < d_{th}$ , liegt eine Verdeckung vor

Zur Reduktion der betrachteten Punkte und damit des Rechenaufwandes bei diesem Schritt kommt wiederum ein Suchbaum zum Einsatz. Im zweidimensionalen Fall ist die für die vorliegende Aufgabe am besten geeignete Suchbaumstruktur ein Quadtree. Dieser ist in Abbildung 6.15 veranschaulicht. Aufgrund des Profils des Bauteils besteht der verwendete Quadtree nicht aus quadratischen Knoten, sondern aus rechteckigen. Sein Wurzelknoten wird von dem maximalen sowie minimalen Punkt bezüglich der zwei Koordinatenachsen aufgespannt und dann entsprechend in beiden Dimensionen in der Hälfte geteilt, sofern der Knoten einen Oberflächenkandidaten enthält. Diese Teilung wird rekursiv so oft wiederholt, bis eine vorher festgelegte minimale Knotengröße erreicht wird. Nur die Punkte in der untersten Knotenebene dieses Quadtrees, die von der betrachteten Normalen geschnitten werden, stellen Eingangsdaten bei der endgültigen Bestimmung der Verdeckung dar. Dadurch reduziert sich die Menge der zu überprüfenden Punkte erheblich.

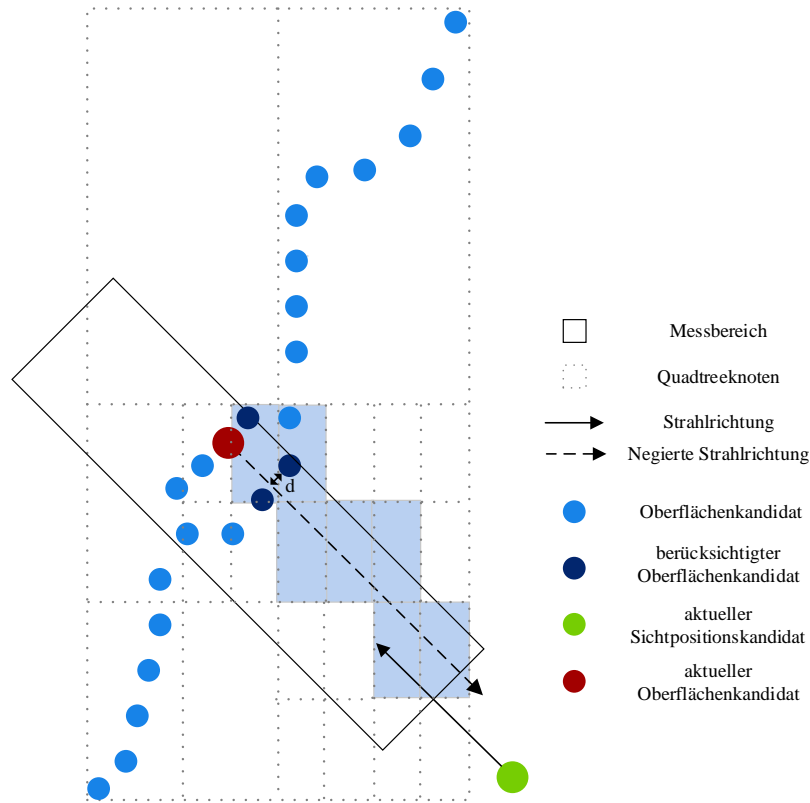


Abbildung 6.15.: Positive Verdeckungsüberprüfung mit einem Quadtree

**5. Messwinkel ( $c_W$ ):** Der Winkel zwischen der Strahlrichtung und der Normalen eines Objektkandidaten fällt als Zwischenresultat bei der Bestimmung der Verdeckung mit ab. Bei beiden Sensoren ist die Qualität und der Positionsfehler einzelner Punkte direkt abhängig von diesem Winkel. Außerdem haben beide Sensoren einen Grenzwinkel, ab dem mit einer hohen Wahrscheinlichkeit Rauschen anstatt einer Abstandsinformation akquiriert wird. Bei den anderen Messkriterien ist das Resultat immer digital, das heißt, dass ein Matriceintrag der MM entweder mit einer 1 erhalten bleibt oder durch eine 0 bei den restlichen Kriterien nicht weiter betrachtet wird. Bei dem Messwinkel wird ein Matriceintrag allerdings bis zum Grenzwinkel linear abfallend zwischen 1 und 0 gewichtet. Überschreitet der Messwinkel den vorher festgelegten Grenzwinkel, wird ebenfalls eine 0 in die Matrix eingetragen.

#### 6.3.1.4. Extraktion einer Untermenge

Nachdem alle Einträge der MM in der Bestimmung der Messbarkeit einen Wert zugewiesen bekommen haben, soll eine möglichst kleine Anzahl an Sensorpositionen als Untermenge aller möglichen SPK extrahiert werden. Hierfür gibt es, wie in Kapitel 2 beschrieben, unterschiedliche

Strategien. In dieser Arbeit wird für die Umsetzung ein sogenannter „Greedy-Ansatz“ aus dem Bereich des SCP für das Konzept der mengentheoretischen Pfadplanung herangezogen. Bei diesem Ansatz soll die kleinste Untermenge  $P_F$  aus der Grundgesamtheit aller Sensorpositionen  $P_V$  gefunden werden, mit der alle Objektkandidaten gemessen werden können. Es gilt also: Finde  $P_{Fmin} \subseteq P_V$ . Hierbei wird in einem Schleifendurchlauf immer diejenige Sensorposition in der MM bestimmt und zu dem Roboterpfad hinzugefügt, deren korrespondierender Zeilenvektor den größten Betrag aufweist. Die Zeilen der Objektkandidaten, die von dieser Sensorposition aus gemessen werden können, werden aus der MM entfernt und der Prozess durchläuft die nächste Iteration. Abbildung 6.16 zeigt zur Veranschaulichung drei Schleifendurchläufe für eine Beispiel-MM.

1. Durchlauf						2. Durchlauf						3. Durchlauf					
	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$		$S_1$	$S_2$	$S_3$	$S_4$	$S_5$		$S_1$	$S_2$	$S_3$	$S_4$	$S_5$
$O_1$	0.2	0.1	0	0.3	0	$O_1$	0.2	0.1	0	0.3	0	$O_1$	0	0	0	0	0
$O_2$	0	0.15	0	0	0	$O_2$	0	0.15	0	0	0	$O_2$	0	0.15	0	0	0
$O_3$	0.5	0	0.6	0	0	$O_3$	0	0	0	0	0	$O_3$	0	0	0	0	0
$O_4$	0	0	0.5	0	0.5	$O_4$	0	0	0	0	0	$O_4$	0	0	0	0	0
$O_5$	0	0	0	0.1	0.1	$O_5$	0	0	0	0.1	0.1	$O_5$	0	0	0	0	0
	0.7	0.25	1.1	0.4	0.6		0.2	0.25	0	0.4	0.1		0	0.15	0	0	0
$P_F = [S_3]$						$P_F = [S_3, S_4]$						$P_F = [S_3, S_4, S_2]$					

Abbildung 6.16.: Greedy-Algorithmus auf eine Beispielmatrix angewendet

Der „Greedy-Ansatz“ wird häufig zu Lösung von NP-schweren Optimierungsproblemen herangezogen. Das hier betrachtete MSC-Problem verallgemeinert das NP-vollständige Knotenüberdeckungsproblem und ist deswegen ebenso NP-schwer (CLR07). Um zu vermeiden, dass der betrachtete Algorithmus in eine Endlosschleife läuft, muss mindestens eine Terminierungsbedingung formuliert werden. In dem betrachteten Fall findet der Algorithmus zum Beispiel kein Ergebnis, wenn ein Objektkandidat von keiner Sensorposition aus gesehen werden kann. Die hier vorgestellte Umsetzung umfasst zwei entsprechende Terminierungsbedingungen. Zum einen wird der Algorithmus abgebrochen, wenn die Anzahl der restlichen Objektkandidaten unter eine vorher festgelegte Schwelle fällt und zum anderen, wenn die Differenz der Anzahl an Objektkandidaten von einer zur nächsten Iteration unter eine weitere Schwelle fällt. Dies verhindert, dass dem Pfad für einzelne Objektkandidaten weitere Sensorkandidaten hinzugefügt werden. Abbildung 6.17 veranschaulicht nochmal den prinzipiellen Ablauf beim Greedy-Algorithmus.

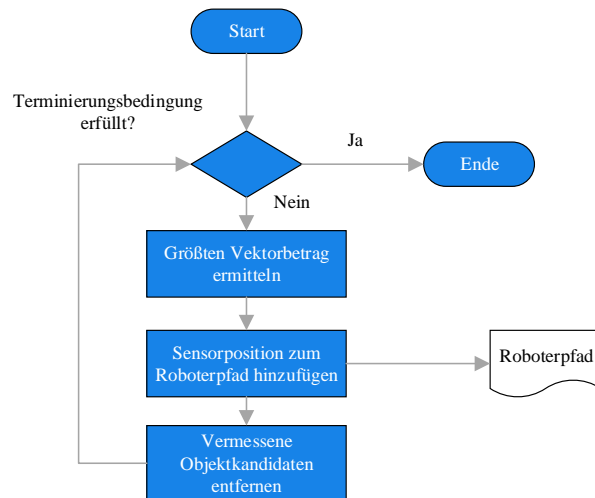


Abbildung 6.17.: Prinzipieller Ablauf des Greedy-Algorithmus

Das Ergebnis dieser ersten Stufe der Pfadplanung ist ein möglichst kurzer Inspektionspfad auf dem reduzierten Bauteilschnitt. Mit Hilfe dieses Pfades kann noch keine vollständige Inspektion durchgeführt werden, weshalb dieses Konzept zur Pfadplanung einen anschließenden Adaptionsschritt vorsieht.

### 6.3.2. Stufe 2: globale Pfadadaption auf die individuelle Bauteilgeometrie

In der zweiten Stufe der Pfadplanung soll ein lokal geplanter Pfad zu einem globalen Inspektionspfad auf dem gesamten Umfang des Bauteils erweitert werden. Dieser lokale Pfad kann entweder aus der vorherigen automatischen Pfadplanung stammen oder manuell offline geplant sein. Bei dieser Adaption müssen sowohl lokale Fehlstellen als auch globale Verformungen und eine mögliche Exzentrizität im Bauteil berücksichtigt werden. Diese werden bei dem vorangegangenen Schritt nicht berücksichtigt, da dieser nur auf einem kleinen Segment des Bauteilumfangs durchgeführt wird. Weil sehr unterschiedliche Anforderungen an diesen Prozess gestellt werden, ist auch die Umsetzung bei der Pfadadaption wiederum in zwei unterschiedliche Schritte aufgeteilt: Eine Pfadoptimierung und die eigentliche Pfadadaption.

#### 6.3.2.1. Optimierung des Eingangspfades

Ziel dieses Schrittes ist die Optimierung der einzelnen Pfadpunkte für die nachfolgende Adaption, die nur in einer Dimension entlang der Prinzip bedingten Linearachse des WLI stattfindet (siehe Kapitel 2.1). Deswegen müssen die einzelnen Pfadpunkte des Eingangspfades möglichst

ideal auf der Oberfläche des Bauteils ausgerichtet sein. Da während der Inspektion eines Ringes das WLI aufgrund von Schwingungsvermeidung und einer möglichst geringen Taktzeit nicht umpositioniert oder -orientiert werden kann, muss der entsprechende Pfadpunkt hinsichtlich der Oberfläche seines gesamten korrespondierenden Ringes optimal positioniert und orientiert werden. Das Bauteil ist außerdem in Paneele unterteilt, die sich in manchen Bereichen überlappen (siehe Abbildung 4.8). Um auszuschließen, dass ein Punkt auf ein falsches Paneel adaptiert wird, werden in der Optimierung außerdem für alle Pfadpunkte die entsprechenden Paneele auf dem Bauteil bestimmt und aus dem Modell extrahiert.

Die Optimierung wird iterativ für jeden Punkt des Eingangspfades durchgeführt. Zur Reduktion des Rechenaufwandes wird für den aktuellen Punkt, wie in Abbildung 6.18, zuerst ein Höhenabschnitt aus dem Gesamtmodell herausgefiltert.

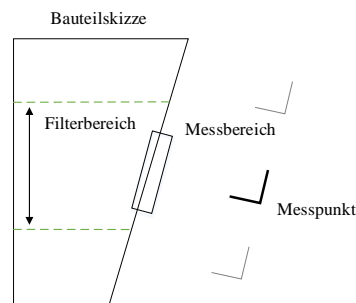


Abbildung 6.18.: Filterung des Bauteilbereichs in der messbereichrelevanten Höhe

Da Rechnungen in Zylinderkoordinaten aufwendig und wenig anschaulich sind, wird auch an dieser Stelle des Prozesses das gefilterte Höhensegment für die Optimierung auf kartesische Koordinaten abgerollt. Hierbei wird dieselbe Abbildungsregel wie in Abschnitt 6.3.1.2 angewendet. Damit die Exzentrizität des Bauteils auf der Werkstückaufnahme keinen Einfluss auf diese Transformation und allgemein auf die Optimierung hat, wird sowohl der Messpunkt als auch das Modell vorher um diesen Faktor korrigiert. Der Wert der Exzentrizität wird bestimmt, indem ein Kreis in das gefilterte Segment geschätzt wird. Die Exzentrizität ist dann der Abstand des Mittelpunktes dieses geschätzten Kreises zu dem Mittelpunkt der kalibrierten Werkzeugbasis, der wiederum im Drehzentrum der externen Achse liegt.

Zur Schätzung des Kreises kommt der Algorithmus der „Übereinstimmung mit einer zufälligen Stichprobe“ (engl. *Random Sample Consensus (RANSAC)*) zum Einsatz. Dieser Algorithmus liefert anders als „Least-Square-Verfahren“ auch bei rauschbehafteten Eingangsdaten ein robustes Ergebnis. Dazu werden aus einer Punktemenge die Anzahl an Punkten entnommen, die zu der jeweiligen Modellbestimmung benötigt werden. Im Fall eines 2D-Kreises sind dies mindestens drei. Alle anderen Punkte werden im Anschluss mit den bestimmten Modellparametern abgeglichen. Wenn eine vorher definierte Menge an Punkten innerhalb eines ebenfalls vorher festgelegten Toleranzbereichs dieses Modells liegt, wird die Suche abgebrochen. Andernfalls wird eine weitere

Iteration mit einer neuen zufälligen Stichprobe aus der Punktwolke durchgeführt. Die Modellparameter mit der größten Übereinstimmung stellen nach Abbruch die Ausgabe des Algorithmus dar. Bei einem 2D-Kreis sind das die x,y-Werte des Kreiszentrums sowie sein Radius.

In den im Anschluss abgerollten Punkten wird dann in mehreren aufeinanderfolgenden Schritten für den betrachteten Pfadpunkt seine korrespondierende Adaptionsebene auf der Bauteiloberfläche herausgefiltert.

Damit nur die Teile der Paneele zur Ermittlung der Optimierung des Pfadpunktes herangezogen werden, dessen Oberflächenvektor in Adaptionsrichtung zeigt, wird auf die gefilterten Punkte wiederum ein Normalenfilter angewendet. Punkte, deren Normale zu weit von der Adaptionrichtung abweicht, werden für die folgenden Schritte aus dem Segment entfernt. Die übrigen Punkte werden anschließend einzelnen Bauteilpaneelen zugeordnet, indem auf ihnen eine „*Euclidean-Cluster-Extraction*“ angewendet wird. Bei einer Cluster-Extraction wird wiederum eine Suchbaumstruktur der Punktwolke  $P$  zur „nächsten Nachbarn Suche“ eingesetzt. Ziel der Extraktion ist, dass alle Punkte  $p_i \in P$  über ihrem euklidischen Abstand einem zusammenhängenden Oberflächencluster zugeordnet werden. Dazu werden für einen zufälligen Punkt aus der Wolke dessen nächsten Nachbarn in einem Abstand kleiner oder gleich  $d_{th}$  bestimmt. Der selektierte Punkt und seine Nachbarn werden in einen Puffer  $Q$  gespeichert und als prozessiert markiert. Dieser Schritt wird für jeden Punkt in dem stetig wachsenden Puffer wiederholt, bis alle Punkte, die in der korrespondierenden Nachbarsuche bestimmt wurden, als prozessiert markiert sind. Danach werden alle Punkte in diesem Puffer dem Cluster  $C_k$  zugeordnet. Dieses Vorgehen wird solange wiederholt, bis alle Punkte einem Cluster zugeordnet wurden. Abbildung 6.19 zeigt beispielhaft die ersten fünf Schritte solch einer Extraktion.

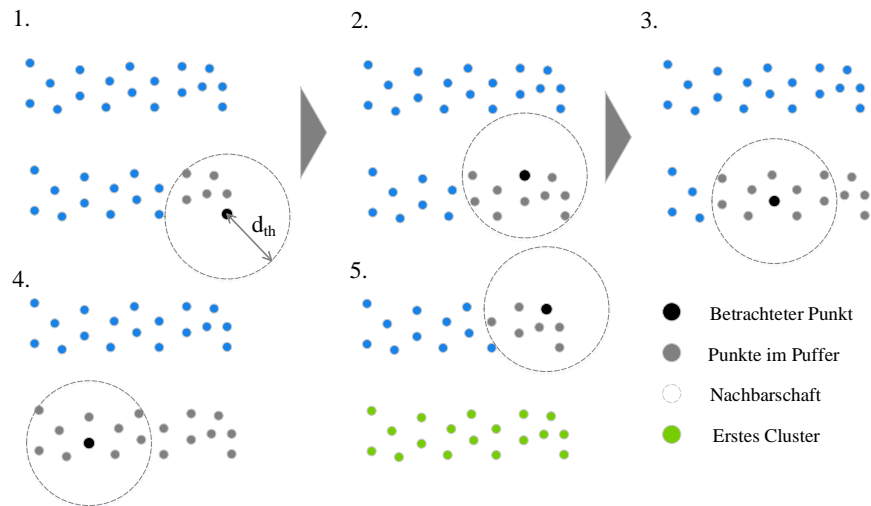


Abbildung 6.19.: 5 Schritte einer Euclidean-Cluster-Extraktion



Die extrahierten Cluster werden im Anschluss wieder in einem komplexen *Ebenen-Vereinigungs*-Algorithmus zu Ebenen zusammengefügt. Nach diesem Schritt repräsentieren die resultierenden Ebenen unterschiedliche Paneele des Bauteils. Im Anschluss an diesen Schritt werden von den resultierenden Paneelen unterschiedliche Parameter bestimmt, zu denen unter anderen die Position des Ebenenschwerpunktes, dessen Oberflächennormale genauso wie dessen Punktzahl gehören. Entsprechend dieser Parameter wird der betrachtete Pfadpunkt einer dieser Ebenen zugewiesen, wobei den Parametern entsprechend ihrer Relevanz ein Gewicht zugeordnet ist. Dem Paneel mit der höchsten Summe aus diesen gewichteten Parametern wird der Pfadpunkt zugeordnet. Entsprechend der Position und Orientierung des ermittelten Paneels werden Position und Orientierung des Punktes optimiert. In einem letzten Schritt wird die Punktwolke des selektierten Paneels zusammen mit dem optimierten Pfadpunkt in einer Containerstruktur abgespeichert und zurücktransformiert. So stehen diese Daten der anschließenden Adaption zur Verfügung. Den gesamten Vorgang der Optimierung veranschaulicht Abbildung 6.20: Rote Punkte zeigen Punkte, die im entsprechenden Schritt herausgefiltert werden; graue Punkte wurden in einem früheren Schritt herausgefiltert.

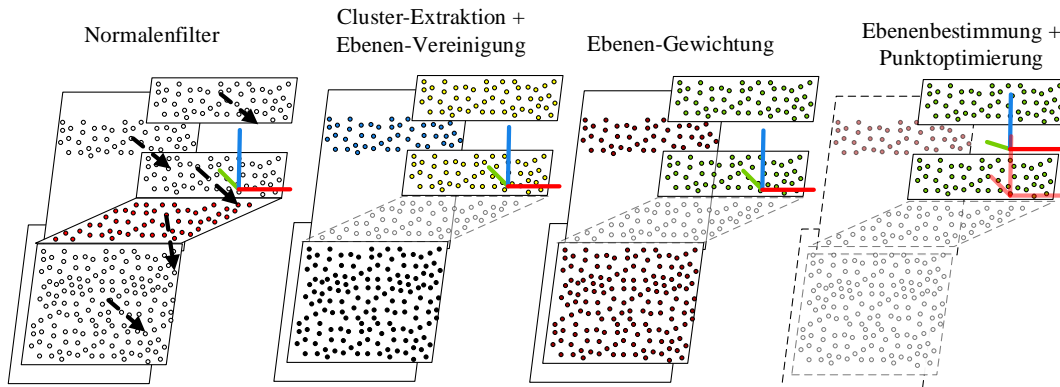


Abbildung 6.20.: Darstellung der benötigten Schritte zur globalen Optimierung der Pfadpunkte auf dem Schema eines Bauteilsegmentes.

### 6.3.2.2. Adaption der Messpunkte an die individuelle Bauteilgeometrie

In dem Schritt der Pfadadaption wird aus dem lokal geplanten und global optimierten Roboterpfad ein vollständiger Inspektionspfad. Dabei wird wieder sukzessive jeder Pfadpunkt einzeln betrachtet. Im ersten Schritt der Adaption wird der Pfadpunkt entsprechend der aktuellen Messbereichsbreite  $s_b$  und der relativen Messbereichsüberlappung  $s_d$  zwischen zwei Aufnahmen in Prozent sowie dem Bauteilradius  $r$  in der entsprechenden Bauteilhöhe um den gesamten Umfang des Bauteils vervielfältigt, indem er um ein Vielfaches des Inkrementes  $i_{eff}$  um das Drehzentrum gedreht wird. Das Inkrement berechnet sich wie folgt:

$$i_{eff} = \frac{2 \cdot \arcsin\left(\frac{s_b}{2 \cdot r}\right) \cdot 180}{\pi} \cdot (1 - 2 \cdot s_d) \quad (6.25)$$

Jeder der dabei entstehenden Pfadpunkte repräsentiert einen späteren Messpunkt für das WLI in der sich anschließenden Inspektion. Die Orientierung bezüglich der Oberfläche bleibt aufgrund der Anforderung einer einheitlichen Roboterpose pro Messring erhalten. Sind die Punkte vervielfältigt, wird für jeden einzelnen davon ein zugehöriger Adaptionwert bestimmt. Dazu wird das in der Optimierung gefilterte Bauteilpaneel herangezogen und an der jeweiligen Position, wie in Abbildung 6.21, einer Boxfilterung unterzogen. Das Ergebnis dieses weiteren Filterschrittes sind die Punkte des Bauteils auf dem richtigen Paneel, die innerhalb des Messbereichs eines Messpunktes liegen. In diesen Punkten wird in Richtung der Adaption ein Maximum und ein Minimum bestimmt. Die Mitte zwischen diesen beiden Werten ist dann der resultierende Adaptionwert für diesen Punkt. Die Bestimmung von Maximum und Minimum führt zu einer weiteren wichtigen Information, die nicht direkt in der Adaption verwendet wird. Die beiden Werte können dazu verwendet werden, eine für jeden Messpunkt individuelle Verfahrrweite für den WLI-Sensor zu errechnen. Dies führt zu einer optimalen Messtiefe, bei der nicht unnötig leerer Raum gemessen wird, sondern nur die Oberfläche selbst. Dieser Schritt erhöht die Effizienz einer WLI-Messung und führt zu einer erheblich geringeren Durchlaufzeit für die gesamte Inspektion. Werden für einen Messpunkt bei der Boxfilterung keine innenliegenden Punkte ermittelt, liegt dieser Punkt entweder vor einer gewollten Fehlstelle wie einem Loch oder einer Bohrung oder einer großen Beschädigung. Für diese Punkte wird der Adaptionwert auf NaN (engl. Not a Number) gesetzt, sodass bei der Inspektion an dieser Stelle keine Messung ausgelöst und weitere Durchlaufzeit eingespart wird.

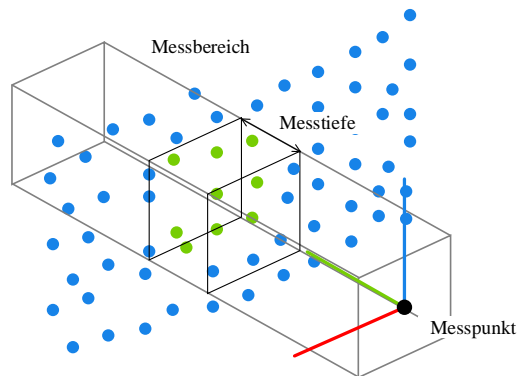


Abbildung 6.21.: Bestimmung des Adaptionwertes und der Messtiefe

### 6.3.2.3. Erweiterung der Adaption auf zwei Dimensionen

Das zuvor geschilderte Vorgehen mit einer eindimensionalen Adaption über die WLI-Achse funktioniert gut für Oberflächenbereiche, die annähernd der Hohlkegelstumpfgeometrie entsprechen. Das bedeutet, dass die Normalen von dem geometrischen Körper und dem Bauteil annähernd übereinstimmen. Das schematische Profil des Bauteils weist allerdings auch Bereiche auf, die orthogonal zu dieser Geometrie ausgerichtet sind. In diesen Bereichen reicht eine Dimension bei der Adaption nicht aus, weswegen ein alternatives Vorgehen benötigt wird. Bei dem alternativen Vorgehen soll der IR verwendet werden, um die Adaption in zwei Dimensionen zu ermöglichen. Aus Gründen der Schwingungsvermeidung kann der IR allerdings nicht analog wie die WLI-Achse eingesetzt werden, um jeden Messpunkt einzeln zu adaptieren. Stattdessen ist dieses Vorgehen mit in den Optimierungsschritt integriert, indem Position und Orientierung der Messpunkte des Eingangspfades aus der Pfadplanung manipuliert werden. Für die Messpunkte in diesen besonderen Bereichen ähneln sich die einzelnen Schritte bei der Optimierung bis hin zur Clusterextraktion. Im Anschluss an diesen Schritt wird der optimierte Pfadpunkt allerdings nicht nur in die Z-Richtung auf den Ebenenschwerpunkt verschoben, sondern auch in Y-Richtung. Dies bedeutet, dass der Messpunkt nach der Rücktransformation in der Mitte des jeweiligen Clusters liegt. Im Anschluss wird wiederum mit einem Boxfilter überprüft, ob das gesamte Cluster mit diesem Punkt inspiziert werden kann. Ist dies nicht der Fall, werden wie in Abbildung 6.22 solange sukzessiv weitere Pfadpunkte hinzugefügt, bis das gesamte Cluster inspiziert werden kann.

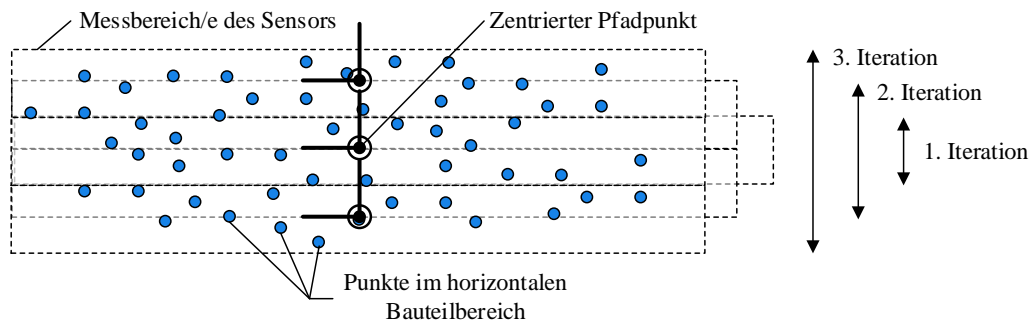


Abbildung 6.22.: Zentrierung und iterative Vervielfachung des Pfadpunktes im horizontalen Bauteilbereich

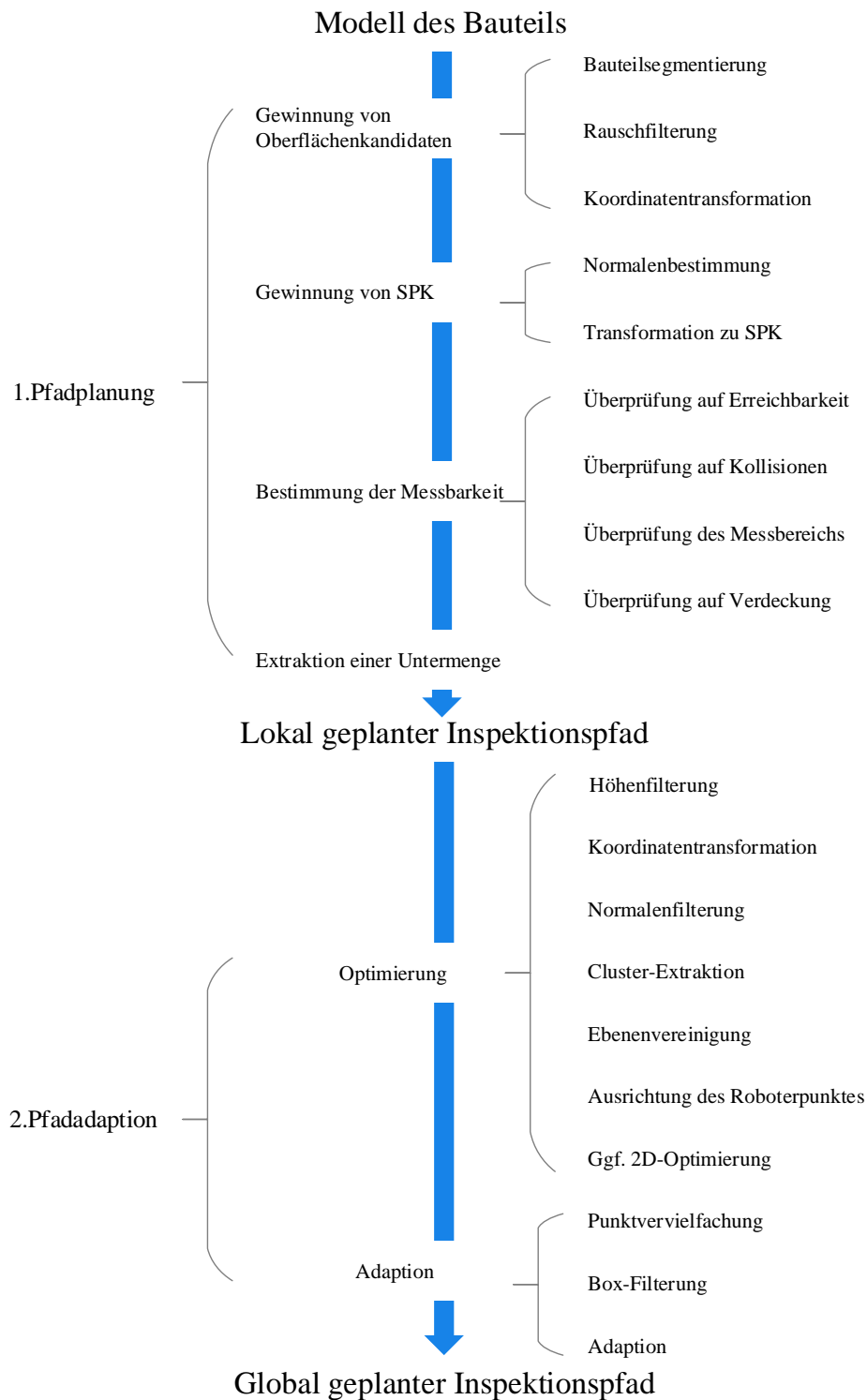


Abbildung 6.23.: Detaillierte Übersicht aller Prozessschritte der Inspektionsplanung

## 6.4. Automatische Rissinspektion mit einem Weißlichtinterferometer

Die Umsetzung des Konzeptes für die Inspektion umfasst aus Sicht der Handhabung aufgrund der umfangreichen Off-Line-Berechnungen in dem Pfadplanungsschritt ausschließlich das sukzessiven Anfahren der einzelnen Ringe des Inspektionspfades und das iterative Drehen des Bauteils auf der zusätzlichen Drehachse. Die Steuerung der eigentlichen Akquirierung der dreidimensionalen Oberflächendaten liegt in der Verantwortung der WLI-Steuerung. Im Netzwerk wurde aufgezeigt, dass diese über TCP/IP mit *AutoXIV* verbunden ist. Hat die Drehachse einen Messpunkt erreicht, wird dieser Steuerung das Kommando zum Auslösen der Messung gegeben. An diesen Befehl werden noch weitere Daten wie die zugehörige Pose, die aktuelle Ring- und Aufnahmeummer sowie die entsprechenden Werte für Adaption und Messtiefe angehängt. Mit dem Adaption- und Messtiefenwert parametrisiert die WLI-Steuerung das entsprechende Verfahrenprofil der Linearachse des Weißlichtinterferometers für die Aufnahme. Die Position sowie die Ring- und Aufnahmeummer werden an die Rissauswertung weitergegeben. Mit diesen Informationen werden die einzelnen Aufnahmen zu einem Gesamtmodell vereinigt. Der Abschluss einer Aufnahme meldet die WLI-Steuerung in Form eines weiteren Befehls zurück an *AutoXIV*. Ab diesem Zeitpunkt beginnt der beschriebene Zyklus wieder von vorn. Die genaue Sequenz bei einer Messung veranschaulicht das Sequenzendiagramm in Abbildung 6.24.

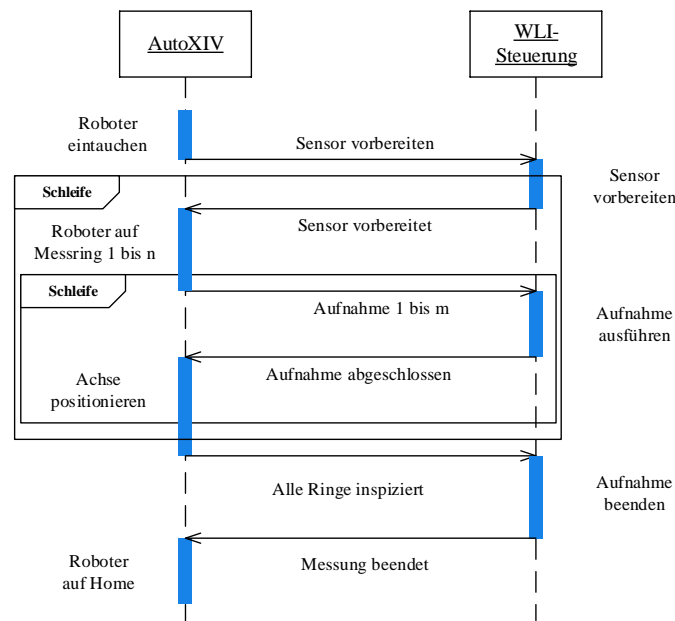


Abbildung 6.24.: Ablauf bei der Inspektion



## 7. Validierung des Inspektionssystems

In den vergangenen Kapiteln wurde ein automatisches Inspektionssystem basierend auf der Technologie der WLI anhand verschiedener Anforderungen konzipiert, ausgestaltet und dann später in Hard- und Software implementiert. Im Folgenden geht es um die Validierung der beschriebenen Anforderungen; die qualitative Funktionalität der Pfadplanung als Kern der Arbeit und der dazu benötigten Hilfsprozesse werden nachgewiesen; der Inspektionsaufbau mit den diversen Handhabungskomponenten auf Grundlage der in Kapitel 3 vorgestellten Werkzeuge wird verifiziert. Hierzu werden neben den vom IR abhängigen Eigenschaften wie Zugänglichkeit und Erreichbarkeit auch die mit dem Sensor erreichbare Messabdeckung untersucht. Alle Parameter zusammen repräsentieren die globale Anforderung der Vollständigkeit. Nur wenn jeder Teilschritt der gesamten Prozesskette die erwarteten Ergebnisse erbringt, kann am Ende des Prozesses eine Inspektion des betrachteten Bauteils durchgeführt werden. Die einzelnen Teilprozesszeiten summieren sich schlussendlich zu einer Gesamtdurchlaufzeit auf. Der Vorgabe folgend darf diese am Ende der Inspektion nicht mehr als 8 Stunden betragen. Tabelle 7.1 vermittelt einen Überblick sowohl über die globalen als auch über individuelle Anforderungen, die sich nur an einzelne Prozessschritte richten.

Globale Anforderungen	Anforderungen an einzelne Prozessschritte	
	Prozessschritt	Anforderung
Durchlaufzeit $\leq 8$ h	Kalibrierung	reproduzierbare Werte für Werkzeug-TCP und Drehachsbasis
Inspektion der vollständigen Bauteiloberfläche	Digitalisierung	Homogene Punktedichte
	Digitalisierung	Auflösung der Punktwolke $\leq 0.3$ mm
	Pfadplanung	Kompensation von Bauteilverformungen

Tabelle 7.1.: Übersicht der Anforderungen an den Inspektionsprozess

### 7.1. Inspektionsaufbau

Wie erläutert muss beim Inspektionsaufbau die Anforderung an eine vollständige Inspektion berücksichtigt werden. Die Parameter, die hierzu überprüft werden müssen, sind Erreichbarkeit, Zugänglichkeit und Messabdeckung. Um Kollisionen zwischen den realen Komponenten zu vermeiden, wurde die Validierung dieser Parameter ausschließlich simulativ durchgeführt. Hierbei kommen unterschiedliche Softwarewerkzeuge zum Einsatz.

### 7.1.1. Zugänglichkeit und Erreichbarkeit

Zur Validierung der Zugänglichkeit und Erreichbarkeit wurde das ausgewählte Handhabungskonzept virtuell nachmodelliert. Hierbei kam die CAD-basierte Off-Line-Robotersimulation *FAMOS* der Firma *carat robotic innovation* zum Einsatz. Für das betrachtete Inspektionssystem wurde der erste Konzeptaufbau, bestehend aus dem Robotermodell KUKA KR30, zusammen mit der Drehachse, dem WLI-Sensor und dem betrachteten Bauteil dem entwickelten Konzept entsprechend in die Simulation integriert (siehe Abbildung 3.2a). Die Software bietet dem Anwender unter anderem die Funktion, ein Volumenmodell mit einer Ebene zu schneiden. Durch solch einen Schnitt durch das rotationssymmetrische Bauteilmodell wird ein vertikaler Schnittgraph erzeugt (siehe Abbildung 7.1 links). Aufgrund der ausgewählten Konfiguration des Handhabungssystems mit der zusätzlichen Drehachse, kombiniert mit einer ringweisen Inspektionsstrategie, reicht es aus, die Zugänglichkeit und Erreichbarkeit für Roboterposen auf diesem Roboterpfad zu validieren, da der Roboter während des Prozesses nur auf dieser Schnittebene Posen anfährt. Dazu wird der erzeugte Schnittgraph in der Simulation durch eine Segmentierung in einem engen Intervall mit Messposen für den Roboter versehen und anschließend mit der vorliegenden Handhabungskonfiguration simuliert (siehe Abbildung 7.1 rechts).

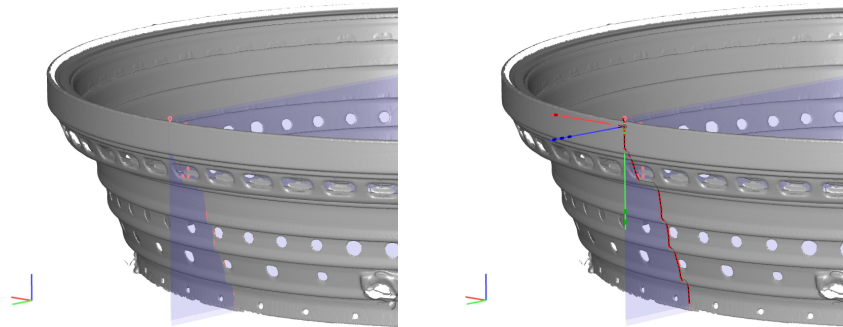


Abbildung 7.1.: Generierung eines Pfades aus einem Ebenenschnitt. Graph nach Ebenenschnitt mit Bauteil (links), Roboterpfad auf Schnittgraph (rechts)

Sind alle Posen auf dem Schnittgraph von dem Roboter erreichbar und gleichzeitig kollisionsfrei zugänglich, kann theoretisch das gesamte Bauteil aufgrund seiner Rotationssymmetrie inspiziert werden. Zu beachten ist dabei allerdings die maximal mögliche Winkelabweichung zwischen der Strahlrichtung des Sensors und der lokalen Oberflächennormale. Dieser Wert ist abhängig von der Reflektivität der Oberfläche. In einem praktischen Versuch wurde deswegen die durchschnittliche Intensität des auf dem Kamerachip auftreffenden Lichtes herangezogen, um zu bestimmen, welche Winkel noch zu einem validen Messergebnis führen. Die Intensität ist dabei einheitenlos. Die Orientierung des Sensors zur Oberflächennormale wurde hierbei inkrementell in Eingrad-



Schritten erhöht und der zugehörige Intensitätswert entsprechend abgelesen (siehe Abbildung 7.2).

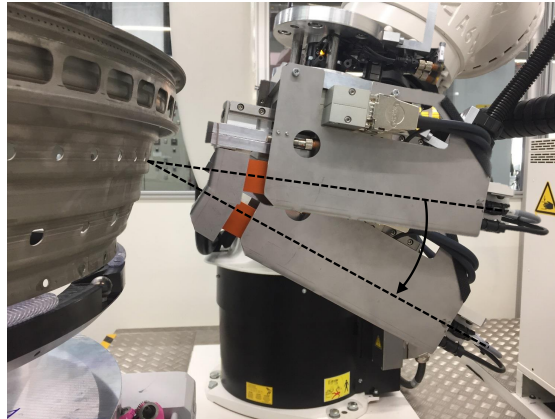


Abbildung 7.2.: Vermessung der reflektierten Lichtintensität unter verschiedenen Winkeln zur Bauteiloberfläche

Bei diesem Versuch wurden zwei unterschiedliche Oberflächen betrachtet. Beide Oberflächen treten in der Form auch während des Reparaturprozesses auf. Die erste Oberfläche ist matt. Diese entsteht in einer Wasserstrahlanlage (engl. Waterjet), in der Beschichtungen vom Brennkammerbauteil entfernt werden. Die andere Oberfläche ist aufgrund von Schleifprozessen hochglänzend. Das Ergebnis dieses Versuchs zeigen die zwei Graphen in Abbildung 7.3. Der orangefarbene Graph, der die Intensität auf dem Chip bei der matten, diffus reflektierenden Oberfläche zeigt, weist einen stabilen, aber gleichzeitig niedrigen Wert bis über  $60^\circ$  auf. Der blaue Graph, der die Ergebnisse der Messung auf einer hoch reflektiven Oberfläche kennzeichnet, zeigt für niedrige Winkel sehr hohe Werte für die Intensität. Diese fällt jedoch aufgrund des überwiegend vom Sensor wegreflektierten Lichtes schnell mit steigendem Messwinkel ab. Der Schnittpunkt beider Graphen, der angibt, ab welchem Winkel die Intensität auf beiden Oberflächen gleich ist, liegt bei etwa  $15^\circ$ . Auch wenn das Bauteil bei der überwiegenden Zahl der anfallenden Inspektionen eine matte Oberfläche aufweist, muss für die Simulation der Messwinkel für die hoch reflektive Oberfläche als „worst case“ herangezogen werden, damit in jedem Fall ein Minimum an Intensität sichergestellt ist.

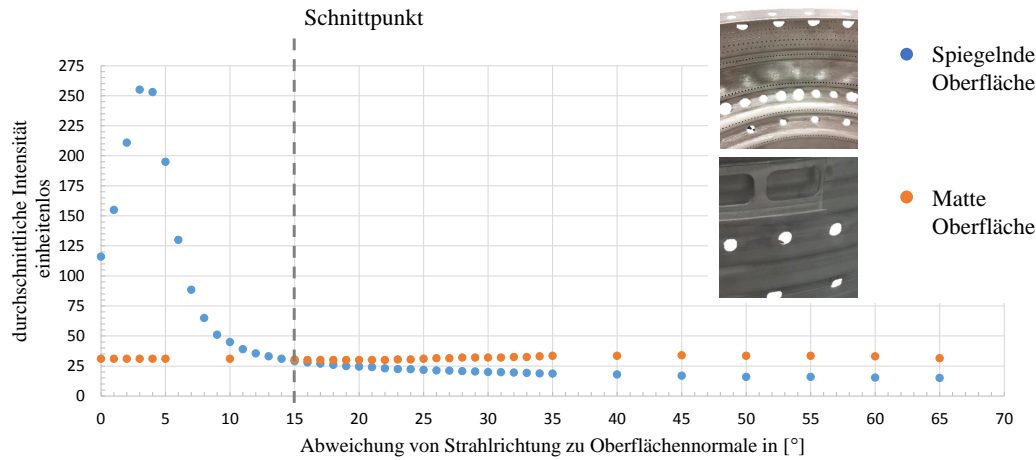


Abbildung 7.3.: Versuch zu Intensitäten über die Messwinkel

Unter den entsprechenden Winkelwerten konnte in der Simulation nachgewiesen werden, dass eine Erreichbarkeit und Zugänglichkeit für das gesamte Bauteil gewährleistet werden kann. Abbildung 7.4 zeigt exemplarisch die Situationen auf einigen speziellen Messposen. Über die genaue Messabdeckung bei der Inspektion kann in dieser Simulation keine Aussage getroffen werden.

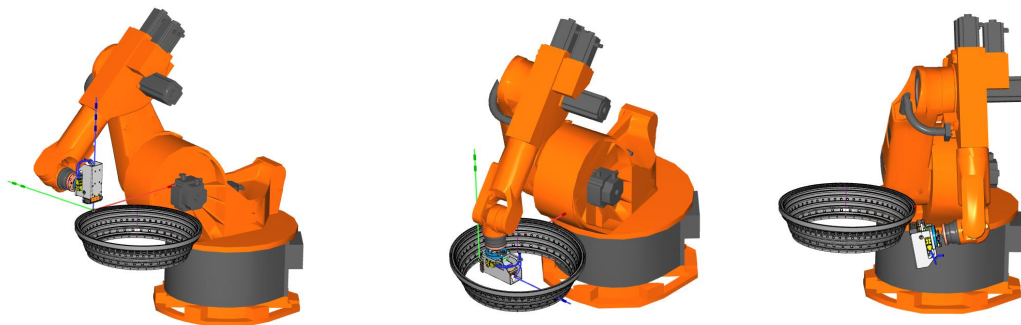


Abbildung 7.4.: Simulation der Zugänglichkeit mit dem konzipierten Aufbau

### 7.1.2. Messabdeckung

Der Nachweis der vollständigen Messabdeckung bei der Inspektion mit einem WLI in dem ausgewählten Aufbau wurde ebenfalls simulativ durchgeführt. Allerdings wurde für diese Aufgabe ein eigenes *MATLAB*-Script konzipiert und implementiert (siehe Kapitel 5.1). Neben den Roboterpfaden zur Inspektion, stehen bei dieser Simulation als weitere Ressource Oberflächenmodelle des gesamten Bauteilspektrums im STL-Format zur Verfügung. Die Ausgabe dieser Simulation ist wiederum ein STL-Modell der messbaren Oberfläche. Das Ergebnismodell kann in *FAMOS*

importiert werden und veranschaulicht in der Überlagerung mit seinem Ursprungsmodell die erreichte messbereichsscharfe Abdeckung bei der Inspektion. Die Ausführung dieses Scriptes ist aufgrund der gewählten Programmiersprache mit einem erheblichen Zeitaufwand verbunden. Dieser erhöht sich sogar noch, wenn die Auflösung der Modelle oder der verwendeten Suchstrukturen für diesen Prozess erhöht wird. Dieses Script wird allerdings nur zur Validierung eines manuell oder automatisch geplanten Roboterpfades verwendet und verlängert somit nicht die eigentliche Durchlaufzeit des Prozesses, weshalb diese Zeit keine große Rolle spielt. Das Ergebnis dieser Simulation für einen kleinen Messbereich zeigt Abbildung 7.5.

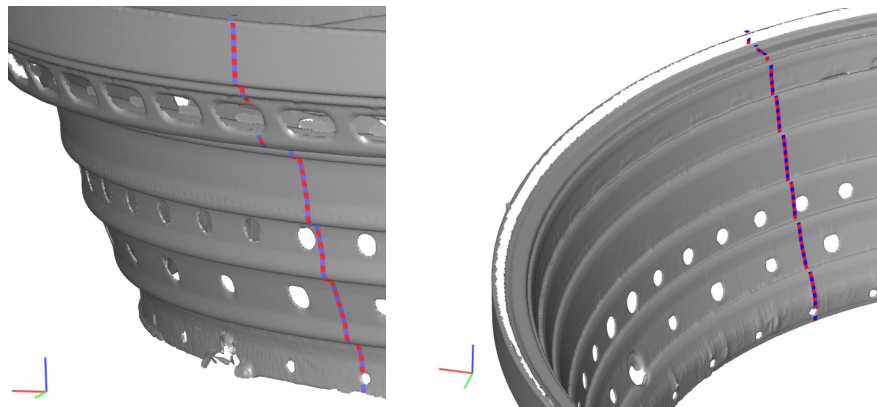


Abbildung 7.5.: Messabdeckung außen (links) und innen (rechts) beim betrachteten Brennkammerbauteil

Wie in der Konzeption vorgestellt, muss nur die vom Hersteller spezifizierte Bauteiloberfläche vollständig inspiziert werden. Dies konnte durch die Simulation validiert werden. Trotzdem sind im Rahmen dieser Simulation Bereiche auf der Oberfläche des Bauteils aufgefallen, die mit dieser Technologie nicht inspiziert werden können. Das liegt daran, dass diese Bereiche in Hinterschnitten liegen und durch andere Oberflächen verdeckt werden. Zwei solcher Beispiele zeigen die orangefarbenen Bereiche in Abbildung 7.6. Für diese Stellen des Bauteils müssen Endoskope zur manuellen Nachinspektion herangezogen werden.

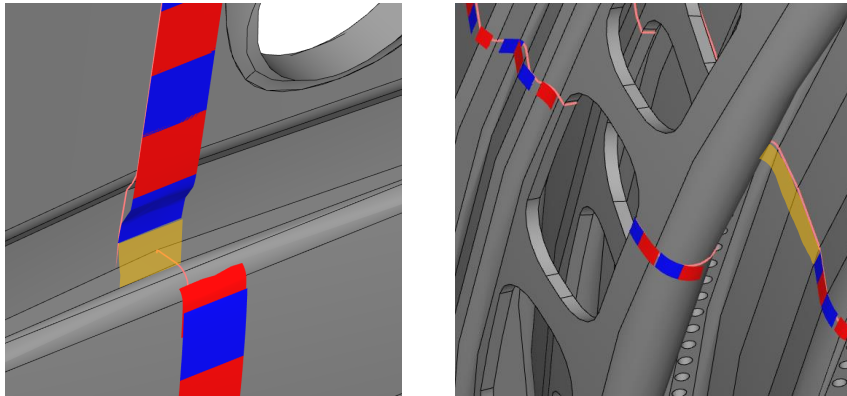


Abbildung 7.6.: Verdeckte Bereiche der Oberfläche in Hinterschnitten des Bauteils (orange)

## 7.2. Validierung des Prozessfunktionalität

In diesem Abschnitt wird die Funktionalität der einzelnen Prozessschritte des Inspektionsprozesses mit WLI validiert. Dazu wird ein Testbauteil sukzessive durch die gesamte Prozesskette geführt, um entsprechend alle prozessspezifischen Anforderungen an ihm zu überprüfen.

### 7.2.1. Kalibrierung

#### 7.2.1.1. Kalibrierung der Werkzeugspitzen

Für die Validierung der Kalibrierung werden für den WLI-Sensor zwölf Datensätze und für den LLS-Sensor elf Datensätze herangezogen, die aus archivierten Kalibrierprotokollen, verteilt über einen Zeitraum von einem Monat, stammen. Hieraus werden die kalibrierten Werte für die TCP der beiden Sensoren extrahiert und statistisch ausgewertet. Das TCP besteht aus den X-,Y- und Z-Koordinaten für die Position und den drei RX-, RY- und RZ-Werten für die Orientierung. Wie in der Konzipierung in Kapitel 4.2 vorgestellt, geht die TCP-Orientierung bei der Berechnung der TCP-Position mit ein. Stimmt die TCP-Position von zwei verschiedenen Kalibrierungen überein, muss folglich auch deren TCP-Orientierung übereinstimmen. Im eigentlichen Kalibrierprozess wird diese Abweichung aus Sicherheitsgründen trotzdem überprüft. Für die Validierung wird sie allerdings an dieser Stelle nicht berücksichtigt. Die wichtigen Bereiche der Kalibrierprotokolle befinden sich im Anhang A.1.

Bei der Untersuchung wurde das arithmetische Mittel der TCP-Position  $P_{DS}$  aller Datensätze der beiden Sensoren ermittelt. Dieses ergibt sich aus der Teilung der Summe der jeweiligen Koordinaten durch die Anzahl der genutzten Datensätze. Im Anschluss wird  $P_{DS}$  herangezogen, um den euklidischen Abstand  $d_{1...n}$  jeder einzelnen Position zu diesem Wert zu bestimmen. Aus diesen Abstandswerten werden wiederum die mittlere euklidische Abweichung  $d_{DS}$  und die

zugehörige einfache Standardabweichung  $s_d$  ermittelt. In der folgenden Tabelle 7.2 können die vorgestellten Werte für beide Sensoren abgelesen werden:

Sensor	Variable	Wert in mm
WLI	$\vec{P}_{DS}$	$\begin{pmatrix} -169.862 \\ -29.764 \\ 179.105 \end{pmatrix}$
WLI	$d_{DS}$	0.0163
WLI	$s_d$	0.015
LR	$P_{DS}$	$\begin{pmatrix} -40.783 \\ -27.06 \\ 108.735 \end{pmatrix}$
LR	$d_{DS}$	0.264
LR	$s_d$	0.075

Tabelle 7.2.: Statistische Auswertung der TCP Kalibrierung

Die obige Tabelle zeigt, dass die Wiederholgenauigkeit beim Weißlichtinterferometer 0.016 mm beträgt. Dieser Wert ist erheblich geringer als die vom Hersteller spezifizierte Wiederholgenauigkeit des eingesetzten Robotertyps in der HA-Variante. Diese liegt bei 0.05 mm, ist jedoch gültig für den gesamten Arbeitsraum. Der erzielte Wert kann für einen robotergeführten Sensor folglich als sehr gut betrachtet werden und die Kalibrierung für das Weißlichtinterferometer ist mit diesem Wert validiert.

Bei dem LLS-Sensor liegt die Wiederholgenauigkeit der TCP-Kalibrierung eine Größenordnung oberhalb des gleichen Wertes bei der TCP-Kalibrierung des Weißlichtinterferometers. Es ist aber zu beachten, dass die Tiefenauflösung bei diesem Sensor mehr als eine Größenordnung schlechter ist und bei der Schätzung nicht über 80.000 Werte auf dem Kugeltarget herangezogen werden, sondern lediglich die 476 Werte einer Laserlinie. Außerdem wird von dem Radius einer ebenen Kreisscheibe auf den Radius der zugehörigen Kugel geschlossen, wobei ebenfalls Fehler entstehen können. Die guten Inspektionsergebnisse und eine niedrige Abweichung von WLI und LLS-Daten zeigen allerdings, dass der Wert trotzdem als gut bewertet werden kann. Die vorliegende Abweichung muss allerdings bei den nachfolgenden Prozessschritten in Form von Toleranzen und Sicherheiten beachtet werden.

#### 7.2.1.2. Kalibrierung der Drehachsbasis

Für die Validierung der Wiederholgenauigkeit bei der Kalibrierung der Drehachsbasis kann, wie einleitend erwähnt, entweder das WLI- oder der LLS-Sensor herangezogen werden. Mit Blick auf den besseren Wert für die Wiederholgenauigkeit bei der Kalibrierung der Werkzeugspitze ist der WLI-Sensor verwendet worden. Die konzipierte Kalibrierstrategie aus Kapitel 4.2.2 wurde sechs Mal wiederholt. Die folgende Tabelle 7.3 fasst die Versuchsergebnisse zusammen. Es

werden wiederum nur die translatorischen Werte betrachtet.  $P_{DS}$  beschreibt also wiederum das arithmetische Mittel der Position der Drehachsbasis im Roboterkoordinatensystem, den mittleren euklidischen Abstand der einzelnen Ergebnisse zu dem arithmetischen Mittel  $d_{DS}$  und wiederum die einfache Standardabweichung  $s_{DS}$ .

Variable	Wert in mm
$\vec{P}_{DS}$	$\begin{pmatrix} 643.242 \\ 1298.583 \\ 629.941 \end{pmatrix}$
$d_{DS}$	0.173
$s_{DS}$	0.065

Tabelle 7.3.: Statistische Auswertung bei der Kalibrierung der Werkstückbasis

Die Wiederholgenauigkeit bei der Kalibrierung der Werkstückbasis beträgt 0.173 mm im Mittel mit einer einfachen Standardabweichung von 0.065 mm. Bei der begrenzten Wiederholgenauigkeit des zum Einsatz kommenden Industrieroboters, in Kombination mit der ebenfalls mit einer Toleranz behafteten Werkzeugspitze des WLI, ist dieses Ergebnis als gut zu bewerten. Welchen Einfluss dieser Wert auf die anderen Prozessschritte hat, wird in den folgenden Abschnitten betrachtet und diskutiert.

### 7.2.2. Digitalisierung

Bei der Digitalisierung beziehen sich die, in der Konzipierung formulierten, Anforderungen auf die *Durchlaufzeit* und eine *homogene Punktedichte* des Modells der realen Bauteiloberfläche. Beide Anforderungen sind allerdings eng miteinander verknüpft. Eine Erhöhung der Prozessgeschwindigkeit durch eine schnellere Winkelgeschwindigkeit der Drehachse bei gleichbleibender Messfrequenz mit dem LLS-Sensor, führt zu einer geringeren Punktedichte auf dem Bauteilmodell in horizontaler Richtung. In der Implementation wurde ein Wert für die minimale Bauteilauflösung angenommen, der aus der späteren Merkmalsextraktion für die Rissdetektion stammt. Dieser führt zu einer Winkelgeschwindigkeit von 60 % ihrer Maximalgeschwindigkeit. Mit dieser Geschwindigkeit wurde eine Digitalisierung auf dem realen Brennkammerbauteil durchgeführt. Die zugehörigen Pfade für die Innen- und Außenseite wurden auf der Robotersteuerung manuell angelernt und in *AutoXIV* Off-Line nachbearbeitet. Zur besseren Einschätzung der Oberflächenhomogenität zeigt Abbildung 7.7 das resultierende Modell in Gesamt- und Nahansicht.

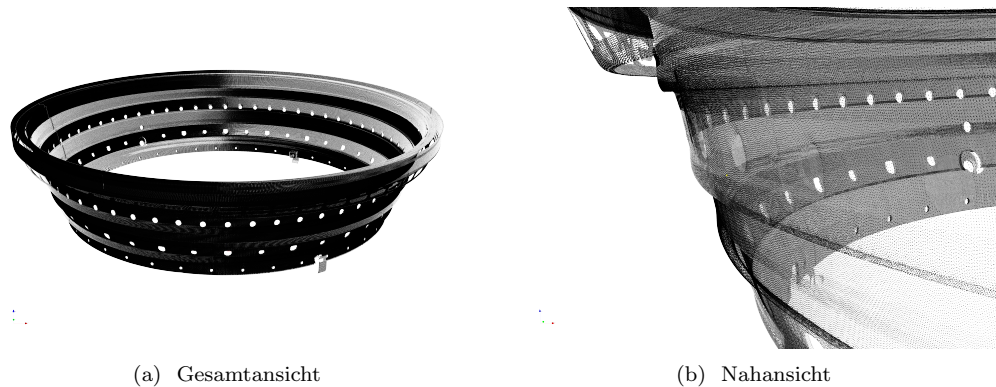


Abbildung 7.7.: Modell als Resultat der Digitalisierung mit dem LLS-Sensor

Die Analyse ergibt, dass die Maßnahmen zur Steigerung der Oberflächenhomogenität wie die asynchrone Datenaufnahme, die triggerbasierte Kommunikationsarchitektur und die Extrapolation der Drehachswinkel funktionieren. Das Bauteil weist keine Lücken aufgrund fehlender Daten auf und zeigt horizontal wie vertikal eine homogene Punktedichte. Damit konnte die Anforderung an die Homogenität bei der Digitalisierung nachgewiesen werden. Die Durchlaufzeit für die Digitalisierung der Innen- und Außenseite zusammen mit allen notwendigen Umsetzbewegungen des Roboters zwischen zwei Ringen beträgt mit dem entsprechenden Parametersatz insgesamt deutlich unter fünfzehn Minuten und erfüllt damit ebenfalls die Anforderung an eine niedrige Durchlaufzeit. Die überlappenden Bereiche von zwei aufeinanderfolgenden Ringen liegen in einer Ebene übereinander (siehe Abbildung 7.8). Diese Beobachtung spricht für die hohe Qualität der Kalibrierung der Werkzeugspitze des LLS-Sensors. Da alle formulierten Anforderungen an diesen Prozess erfolgreich validiert werden konnten, ist gleichzeitig der gesamte Teilprozess der Digitalisierung validiert.

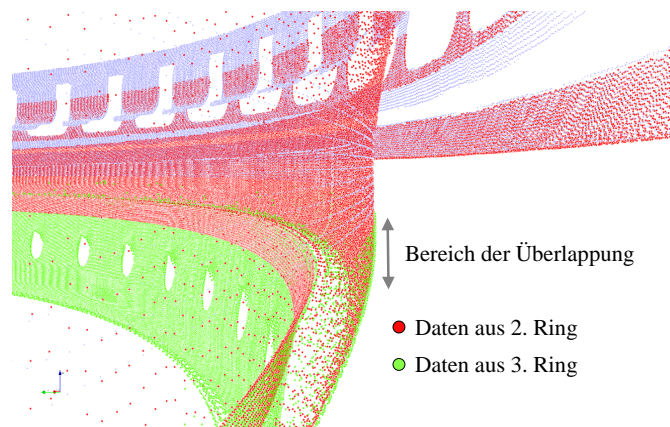


Abbildung 7.8.: Überlappung der Daten von zwei aufeinanderfolgenden Messringen bei der Digitalisierung

### 7.2.3. Pfadplanung

In diesem Abschnitt wird das in *AutoXIV* implementierte zweistufige Konzept zur Pfadplanung für die Inspektion validiert. Die Anforderungen an diesen Prozess entstammen der beschädigten und verformten Bauteilgeometrie und ihren einzigartigen Bauteilmerkmalen. Beide müssen bei der Pfadplanung berücksichtigt und entsprechend kompensiert werden, um wiederum die globale Anforderung der vollständigen Inspektion sicherzustellen. In diesem Abschnitt werden beide Stufen, also die modellbasierte und die adaptive, gemeinsam betrachtet. Die realen Oberflächendaten liegen in Form von Punktwolkenmodellen der Bauteil Innen- und Außenseite vor, die im vorangegangenen Schritt der Digitalisierung akquiriert wurden.

#### 7.2.3.1. Modellbasierte Pfadplanung

Der modellbasierte Teil der Pfadplanung kann aufgrund der vorliegenden Rotationssymmetrie des Bauteils auf einem reduzierten Bauteilsegment lokal durchgeführt werden. Die Planung des Inspektionspfades umfasst die folgenden vier Prozessschritte:

##### Schritte bei der modellbasierten Pfadplanung

1. Generierung der Objektkandidaten
2. Ableitung von Sensorpositionskandidaten
3. Überprüfung der Messbarkeit
4. Extraktion einer Untermenge

Im Folgenden werden diese Schritte auf die Außenseite des Testmodells angewendet und deren Funktionen exemplarisch veranschaulicht.

**Generierung der Objektkandidaten:** Das Konzept bei der Generierung von Objektkandidaten sieht bei der Pfadplanung drei Teilschritte vor. Zuerst wird aus dem Bauteilmodell auf dem Umfang ein  $10^\circ$  Segment extrahiert (Abbildung 7.9a). Dieses Segment wird anschließend mit der in Kapitel 6.3.1.1 konzipierten Abbildungsregel in den Zylinderraum transformiert, sodass des Weiteren anstatt in Zylinder- in kartesischen Koordinaten gerechnet werden kann (Abbildung 7.9b). Im Anschluss an die Koordinatentransformation werden alle Punkte des Segments auf die neue  $y'z'$ -Ebene abgebildet (Abbildung 7.9c).



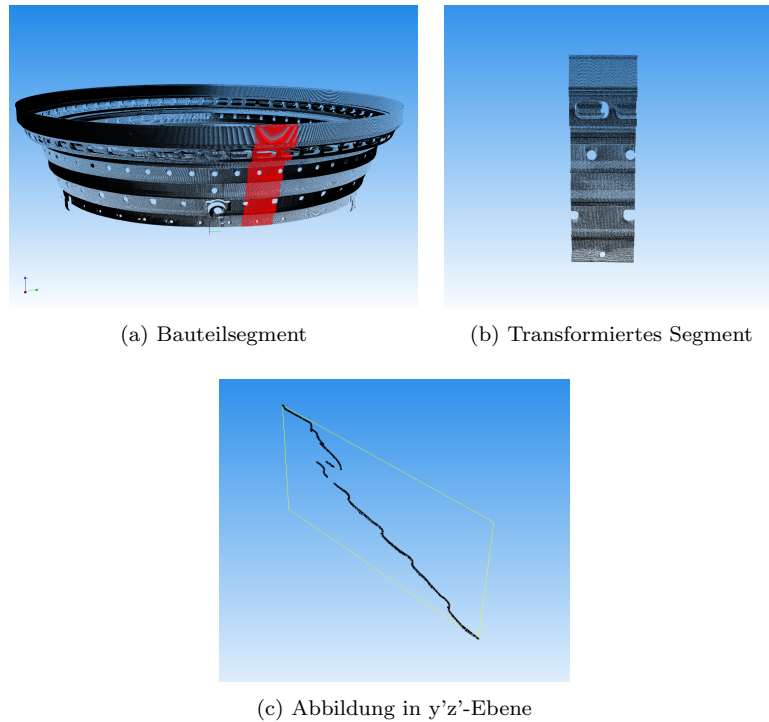


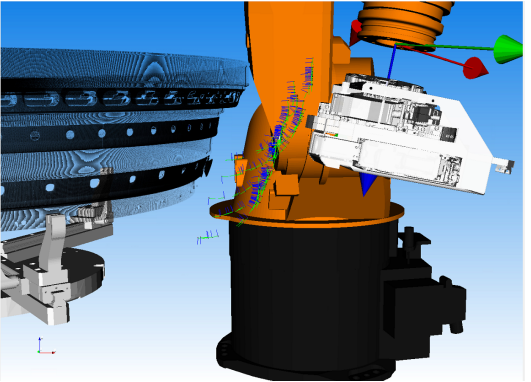
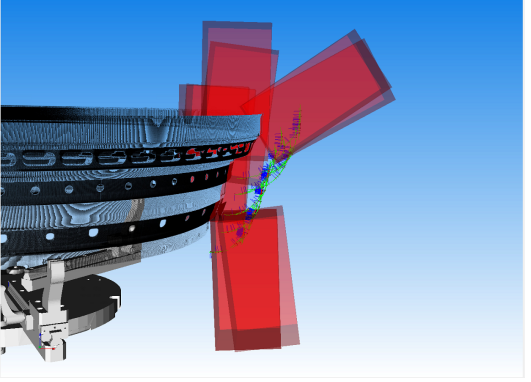
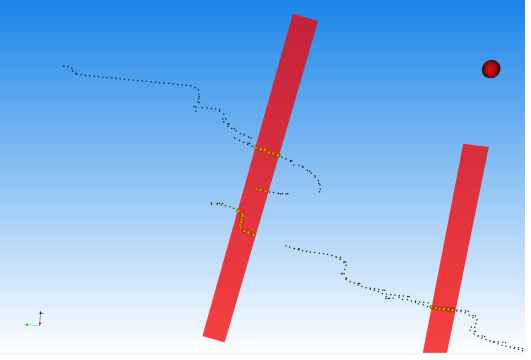
Abbildung 7.9.: Teilergebnisse bei der Generierung von Objektkandidaten

Die Abbildungen zeigen, dass ein vollständiges Segment (rot gekennzeichnet) aus dem kompletten Bauteilmodell extrahiert, transformiert und in einer Ebene abgebildet wird. Die resultierenden, zweidimensionalen Punkte sind sehr gut als potentielle Objektkandidaten geeignet, da sie die zugehörige Bauteiloberfläche gut annähern.

**Ableitung von Sensorpositionskandidaten:** Bei der modellbasierten Pfadplanung für die Inspektion mit einem WLI-Sensor ist der Schritt der Ableitung von SPK aus den entsprechenden Objektkandidaten trivial. Anders als zum Beispiel bei einem LLS-Sensor befindet sich der Nullpunkt in der Mitte des Sensor-Messbereichs. Zur optimalen Ausnutzung des gesamten Messbereichs wird dieser direkt auf die Bauteiloberfläche geplant. Dies bedeutet, dass die herangezogenen Objektkandidaten nicht verändert werden müssen und stattdessen für die folgenden Schritte direkt als SPK übernommen werden können. Für alle Sensoren bei denen der Nullpunkt im Sensor liegt, kann das konzipierte Verfahren über die Normalenverschiebung angewendet werden.

**Überprüfung der Messbarkeit:** Mithilfe der in Kapitel 6.3.1.3 implementierten Messbarkeitskriterien wird die Komplexität der zentralen Struktur der MM reduziert. Matrixeinträge bei denen eines der Kriterien negativ ausfällt, werden auf null gesetzt. Dadurch wird die Rechenkomplexität bei den nachfolgenden Kriterien sowie bei der finalen Ableitung eines Inspektionspfades

erheblich reduziert. Tabelle 7.4 zeigt die einzelnen Kriterien noch einmal kurz auf und veranschaulicht deren Funktionalität.

<p><b>Messbarkeitskriterium Erreichbarkeit</b></p> <p>Da die SPK den Objektkandidaten entsprechen, wurden sie in der Abbildung zur Veranschaulichung manuell entlang ihrer Z-Achse negativ verschoben. Diejenigen Punkte, die vom Roboter erreichbar sind, werden als Pfadpunkt visualisiert. In dem vorliegenden Fall sind alle Punkte erreichbar. Das bedeutet: Es werden in diesem Schritt keine Matrixeinträge entfernt.</p>	
<p><b>Messbarkeitskriterium Kollision</b></p> <p>Auch das zweite Kriterium „Kollision“ wird in Zylinderkoordinaten überprüft. An der Position der Punkte, an denen es zu einer Kollision kommt, wird die OBB des Werkzeugmodells in rot visualisiert. Die zugehörigen Punkte aus der Menge der SPK und alle entsprechenden Einträge aus der MM werden entfernt.</p>	
<p><b>Messbarkeitskriterium Messbereich &amp; Messwinkel</b></p> <p>Ein exemplarischer Messbereich des WLI-Sensors wird als lange Box dargestellt. Die zugehörigen Messpunkte in der Mitte des Messbereichs sind durch gelbe Kugeln hervorgehoben. Es ist zu erkennen, dass nach diesem Kriterium noch Punkte als messbar gekennzeichnet werden, die auf einer eigentlich verdeckten Oberfläche liegen.</p>	

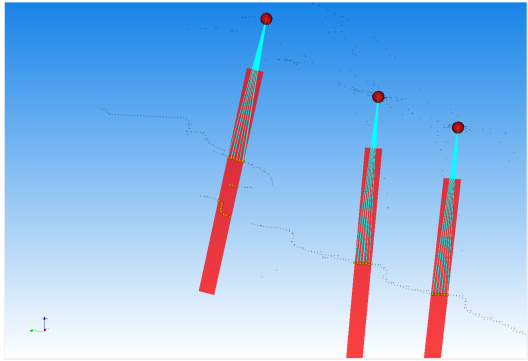
Messbarkeitskriterium Verdeckung	
<p>Die verdeckten Punkte innerhalb des Messbereichs werden vom letzten Kriterium „Verdeckung“ erfolgreich herausgefiltert. Die Abbildung zeigt mit türkisfarbenen Linien die unverdeckte Verbindung zwischen Objektkandidaten und ihrem korrespondierenden SPK</p>	

Tabelle 7.4.: Unterschiedliche Kriterien bei der Überprüfung der Messbarkeit

Zur Validierung und übersichtlichen Visualisierung der Funktion und Auswirkung der einzelnen Messbarkeitskriterien auf die Objekt- sowie SPK wird nach jedem Kriterium der Zustand der zentralen Struktur der MM farbcodiert als Bild gespeichert. Dieses kann im Anschluss dargestellt werden. Das Pixel  $P_{ij}$  eines solchen Bildes repräsentiert einen Matrixeintrag  $m_{ij}$ , also die Messbarkeit des Objektkandidaten  $O_i$  vom SPK  $S_j$ . Das verwendete Farbspektrum reicht dabei von rot, stellvertretend für eine „1“, bis blau, stellvertretend für eine „0“. Die folgende Abbildung 7.10 zeigt, wie die Matrix als Ergebnis der Überprüfung der unterschiedlichen Kriterien ausgedünnt wird. Ist ein Eintrag „0“ beziehungsweise blau, dann wird er im nächsten Kriterium nicht mehr berücksichtigt. Das erste Bild zeigt die Matrix nach dem Anwenden des Kriteriums „Erreichbarkeit“. Die Matrix ist unverändert rot, also weiterhin vollständig mit Einsen gefüllt. Das zweite Bild verdeutlicht, dass bei einigen der SPK beim zweiten Kriterium Kollisionen detektiert werden, zu erkennen an den mit Nullen gefüllten blauen Spalten in der MM nach diesem Schritt. Nach dem dritten Messbarkeitskriterium, in dem parallel Messbereich und Messwinkel überprüft werden, ist die MM sichtbar ausgedünnt und weist über das ganze Farbspektrum verteilte Pixel auf. Dies bedeutet, dass die Einträge der Matrix durch den Messwinkel auf der Skala von 0 bis 1 gewichtet worden sind. Im vierten Bild sind auch noch die im letzten Kriterium erkannten, verdeckten Objektkandidaten auf „0“ gesetzt worden. Nach der Überprüfung der vier Kriterien wird aus der MM eine Untermenge an Sensorpositionen extrahiert, die dann der Adaption übergeben wird.

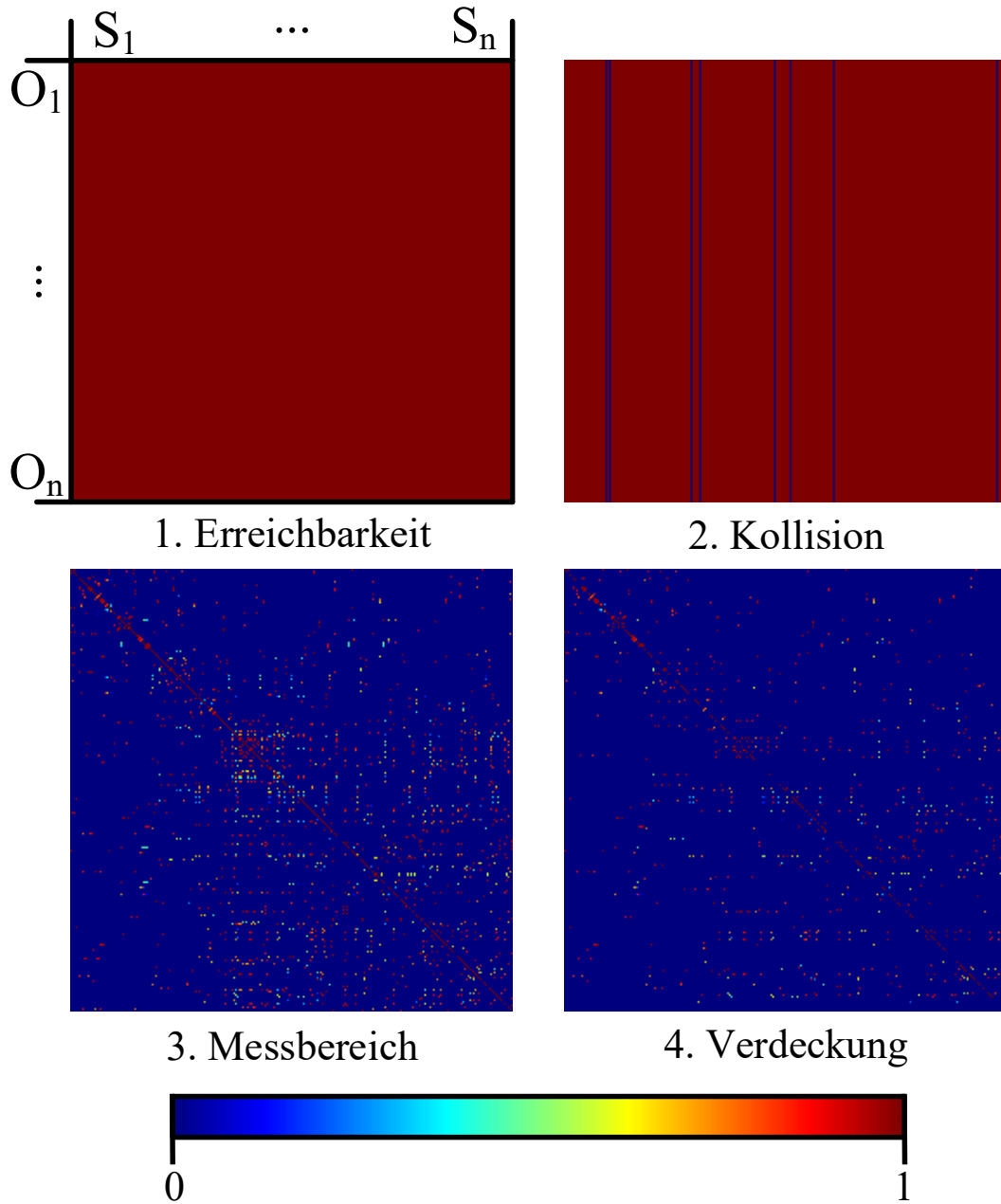


Abbildung 7.10.: Messbarkeitsmatrix mit Sensorpositions-kandidaten  $S_1 - S_n$  und Objektkandidaten  $O_1 - O_n$  farbcodiert als Bild nach Anwendung der einzelnen Messbarkeitskriterien

**Extraktion einer Untermenge:** Nachdem alle Kriterien auf die SPK angewendet worden und dadurch die meisten Einträge der eingangs sehr großen MM auf null gesetzt worden sind, sieht die Pfadplanungsstrategie vor, dass in der verbleibenden Menge der SPK mit der Bestimmung einer Untermenge begonnen wird. Hierzu legt das in Kapitel 6.3.1.4 umgesetzte Konzept fest, dass pro Iteration jeweils der momentan beste Sensorpositions-kandidat bestimmt und extrahiert wird und im Anschluss die zugehörigen Zeilen der Objektkandidaten aus der Matrix entfernt werden. Dieser Prozess wird solange wiederholt, bis mindestens eine der formulierten Terminierungsbedingungen erfüllt ist. Ein Ergebnis dieses Prozessschrittes zeigt Abbildung 7.11, in der die aus der MM selektierten Sensorpositionen durch eine rote Kugel visualisiert und mit ihren korrespondierenden Objektkandidaten durch türkisfarbene Strahlen verbunden sind. Die selektierten Sensorpositionen werden im Anschluss zu Zylinderkoordinaten zurücktransformiert und in einen Roboterpfad gewandelt (Abbildung 7.12), der dann die Grundlage für den sich anschließenden 2. Pfadplanungsschritt der Adaption bildet.

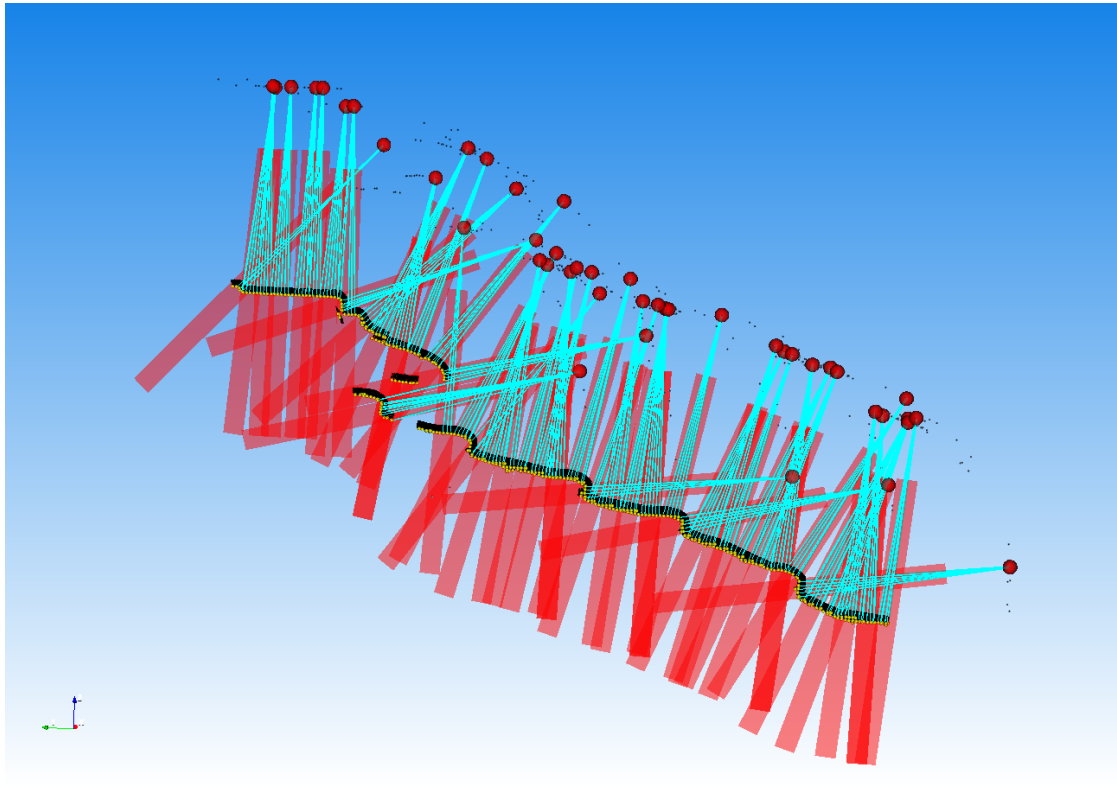


Abbildung 7.11.: Aus der MM gefilterte Sensorpositionen (rote Kugeln), verbunden mit ihren messbaren Objektpunkten

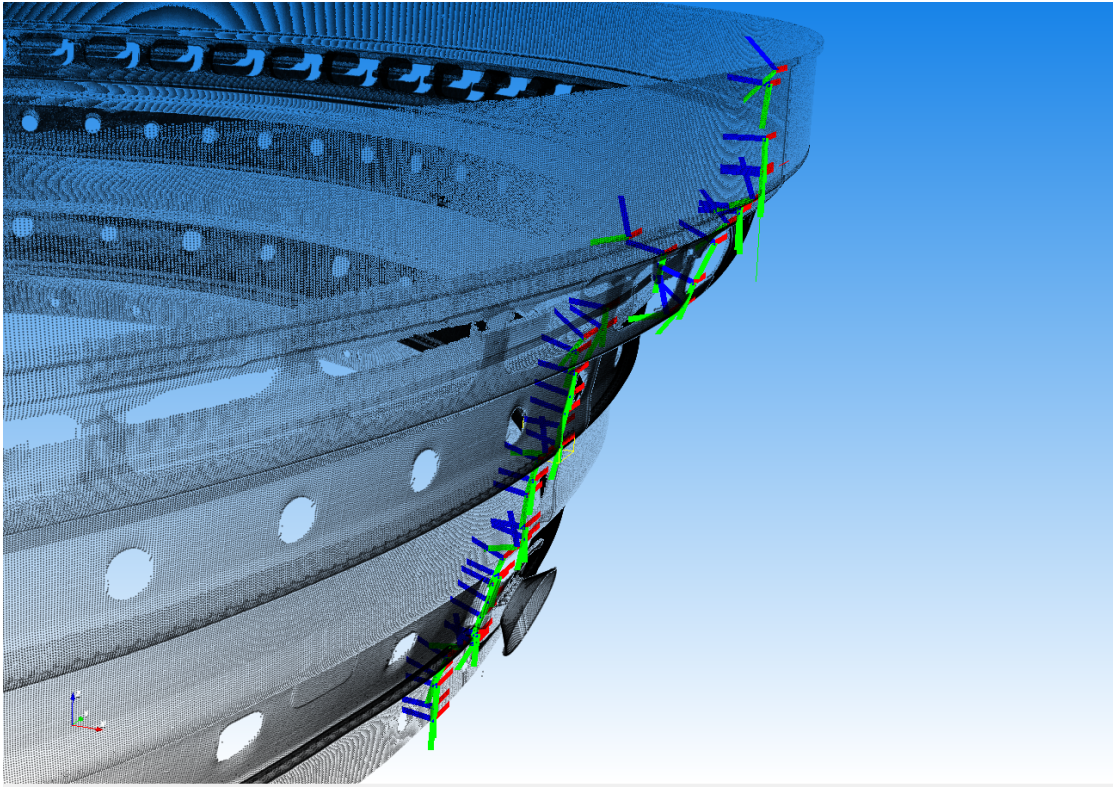
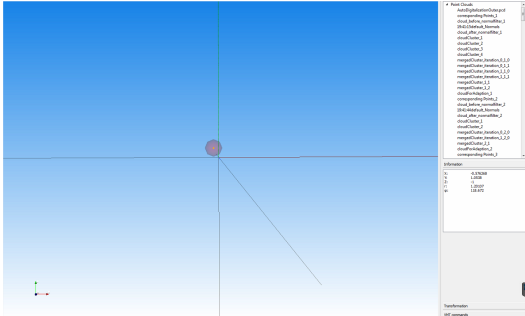
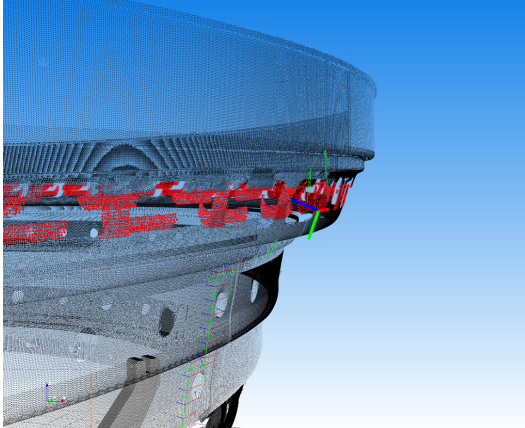
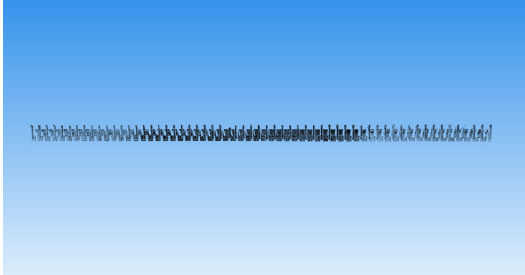


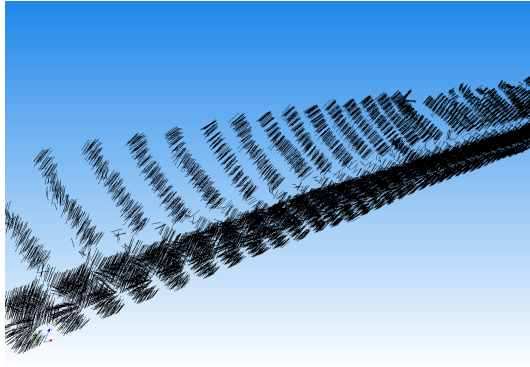
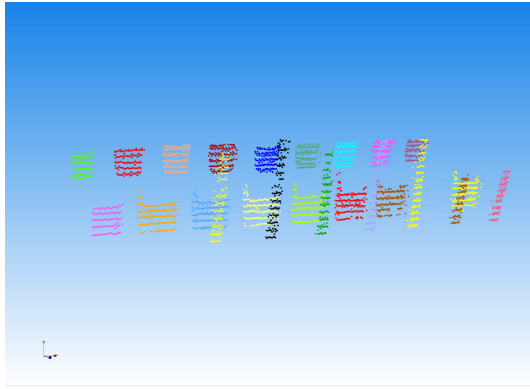
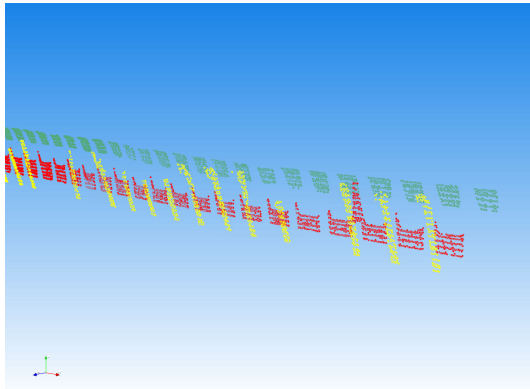
Abbildung 7.12.: Die Transformation der Sensorpositionen zu Zylinderkoordinaten führt zu dem Roboterpfad als Ergebnis der modellbasierten Pfadplanung

#### 7.2.3.2. Optimierung des Eingangspfades

Die Validierung der Umsetzung des Konzeptes der Pfadoptimierung wird in diesem Abschnitt an dem Testbauteil und einem ausgewählten Pfadpunkt aufgezeigt. Dieser Pfadpunkt stammt aus dem Ergebnispfad der modellbasierten Pfadplanung und liegt in einer für die Adaption besonders anspruchsvollen Bauteilregion. Hier weist das Bauteil große Löcher auf, durch die hindurch Oberflächen digitalisiert werden, die eigentlich verdeckt wären. Diese Oberflächen sollen später nicht inspiziert werden. Jedes Teilresultat der einzelnen Prozessschritte wird wiederum in einer Tabelle kurz erläutert und bildlich veranschaulicht. Hierdurch kann wiederum die Funktion der einzelnen Algorithmen validiert werden.

1. Bestimmung der Exzentrizität	
<p>Bestimmung der Bauteilexzentrizität relativ zur Werkstückaufspannung in der XY-Ebene der Drehachsbasis. Der Abstand der dargestellten Kugel in der Mitte zum Koordinatenkreuz repräsentiert die ermittelte Verschiebung des Bauteilzentrums.</p>	
2. Z-Filterung	
<p>Filterung der Punktwolke bezüglich der Z-Koordinate zur Reduktion des Rechenaufwandes in den folgenden Prozessschritten. Obere und untere Grenze werden hierbei durch den, um einen Sicherheitsfaktor vergrößerten, Messbereich des Sensors vorgegeben.</p>	
3. Koordinatentransformation	
<p>Transformation in kartesische Koordinaten: Transformation („Abrollen“) der extrahierten Punktwolke von Zylinderkoordinaten auf kartesische Koordinaten</p>	



4. Normalenfilterung	
<p>Normalenfilterung der Punktwolke durch lokale Ebenenschätzung. Punkte mit zu großer Normalenabweichung zu der Pfadpunktorientierung werden in diesem Schritt ausgeschlossen</p>	
5. Euklidische Clusterung	
<p>Zuordnung der einzelnen Punkte der Punktwolke in Cluster. Punkte eines Clusters weisen in der Abbildung rechts eine gemeinsame Farbe auf.</p>	
6. Ebenenvereinigung	
<p>Vereinigung der Cluster zu gemeinsamen Ebenen. Die Vielzahl von erkannten Clustern wurden insgesamt zu drei verschiedenen Ebenen vereinigt. Zu einer Ebene vereinigte Cluster sind wiederum über eine gemeinsame Farbe gekennzeichnet.</p>	



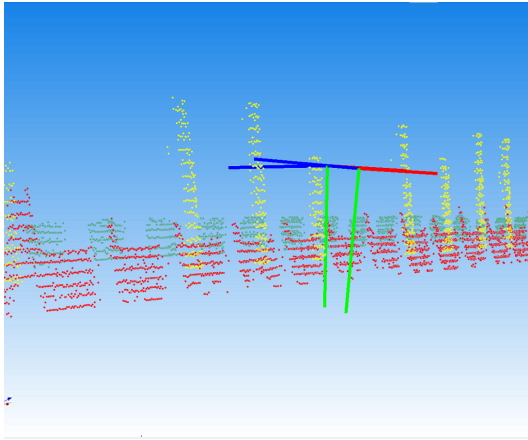
7.Pfadpunktoptimierung	
<p>Finale Optimierung der Position und Orientierung des entsprechenden Pfadpunktes an eine Ebene. Die entsprechende Ebene wird dabei ordnungsgemäß über unterschiedlich gewichtete Parameter wie die Punkteanzahl oder den Abstand zur Pose bestimmt.</p>	

Tabelle 7.5.: Teilschritte der Pfadoptimierung als 1.Stufe der Pfadadaption auf die globale Bauteilgeometrie

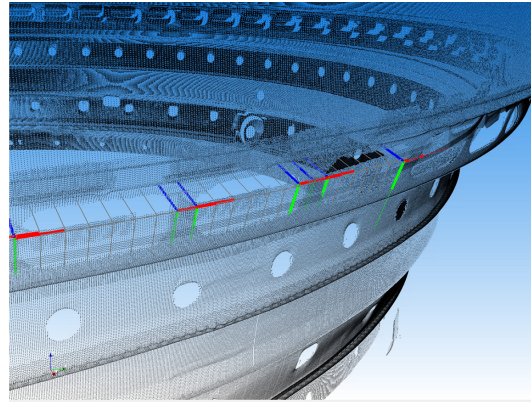
Die einzelnen Teilprozesse der Pfadoptimierung erzielen solide das angestrebte Resultat. Die korrespondierenden Ebenen auf den Oberflächen werden aus extrahierten Clustern zusammengesetzt und anschließend wird die entsprechende Ebene ordnungsgemäß selektiert. Der Pfadpunkt wird korrekt auf diese Ebene verschoben und orientiert.

#### 7.2.3.3. Adaption der Messpunkte

Als Eingangsdaten hat die Adaption den global optimierten Pfad sowie das digitalisierte Modell der äußeren Bauteiloberfläche. Es dient der Umsetzung des Konzeptes der Pfadadaption, das als Ergebnis die Messringe für die anschließende Inspektion erzeugt. Da auch für die 2. Stufe der Adaption das Konzept schon ausführlich vorgestellt worden ist, wird in der Validierung dieses Schrittes analog der vorangegangenen Prozesse, das Testbauteil betrachtet. Die einzelnen Teilprozesse werden in 7.6 kurz erläutert und das entsprechende Ergebnis auf dem Testbauteil mit Abbildungen dargestellt.

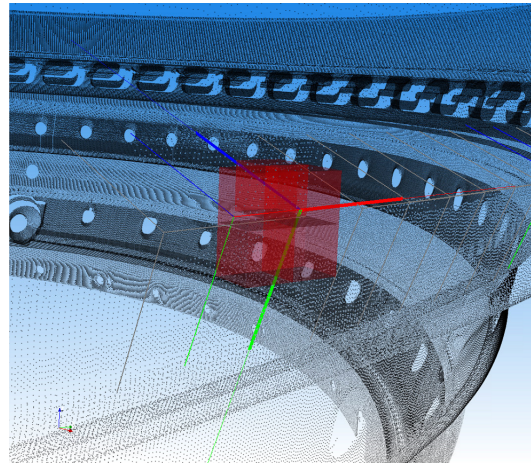
### 7. Punktmultiplikation

Vervielfachung des Pfadpunktes zu einem vollständigen Messring unter Betrachtung des Messbereichs und des zugehörigen lokalen Bauteilradius. Löcher im Bauteil werden zuverlässig erkannt und die Messpunkte innerhalb dieses Loches werden aus der Inspektion ausgeschlossen. (graue Einfärbung des Koordinatenkreuzes)



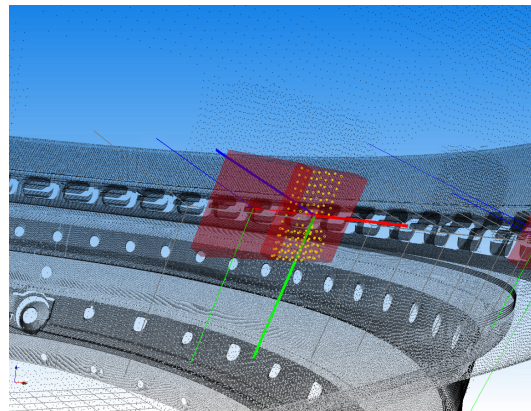
### 8. Oberflächenfilterung

Bestimmung der nächsten Nachbarn zum Pfadpunkt über das Schneiden eines Strahls mit der Octree Suchbaumstruktur des Bauteilmodells. Im betrachteten Fall werden alle vom Strahl geschnittenen Octreevoxel in der Abbildung veranschaulicht.



### 9. Box-Filterung

Extraktion der Punkte aus der Wolke, die innerhalb des boxförmigen Messbereichs des entsprechenden Pfadpunktes liegen, und Schätzung der lokalen Bauteiloberfläche, die diese Punkte repräsentieren, sowie der notwendigen Messtiefe.



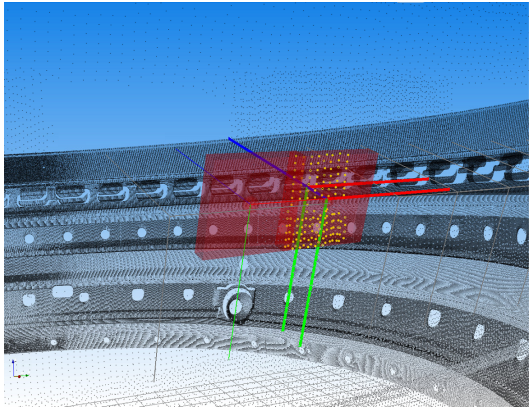
10. Messpunktadaption	
<p>Verschiebung des Pfadpunktes in Strahlrichtung auf die vorher geschätzte Oberfläche.</p>	

Tabelle 7.6.: Teilschritte der Pfadadaption als 2.Stufe der der übergeordneten Adaption auf die globale Bauteilgeometrie

Auch für die Teilschritte der Pfadadaption als zweite Stufe der Pfadplanung, konnte die entsprechende Funktionalität validiert werden. Die Pfadpunkte der einzelnen Messringe wurden vervielfacht und die zugehörigen Oberflächen bestimmt. Dadurch ist es möglich die Punkte entlang ihrer Z-Achse zu adaptieren.

#### 7.2.4. Automatische Rissinspektion mit einem Weißlichtinterferometer

Die einzige Funktionalität, die bei der Validierung des letzten Prozessschrittes der Inspektion isoliert von den anderen Prozessschritten betrachtet werden kann, ist die Kommunikation zur WLI-Steuerung, wie sie in Kapitel 6.4 vorgestellt worden ist. Sollten beim Senden Befehle verloren gehen oder auf einer der beiden Seiten missinterpretiert werden, würden folgende Konsequenzen auftreten:

- Abbruch des Prozesses wegen inkonsistenter Daten
- Lücken im Modell aufgrund fehlender Aufnahmen
- Das nebeneinander Auftreten von ursprünglich nicht benachbarten Aufnahmen

In allen beschriebenen Fällen würde der Prozess der Inspektion nicht bis zum Ende durchgeführt werden. Der Prozessablauf ist bis zum Abschluss dieser Arbeit in über hundert praktischen Versuchen erfolgreich durchgelaufen. Auch die Oberfläche des Inspektionsresultates war in diesen Versuchen vollständig, sodass Kommunikationsfehler im Prozess ausgeschlossen werden können (siehe Abbildung 7.13)

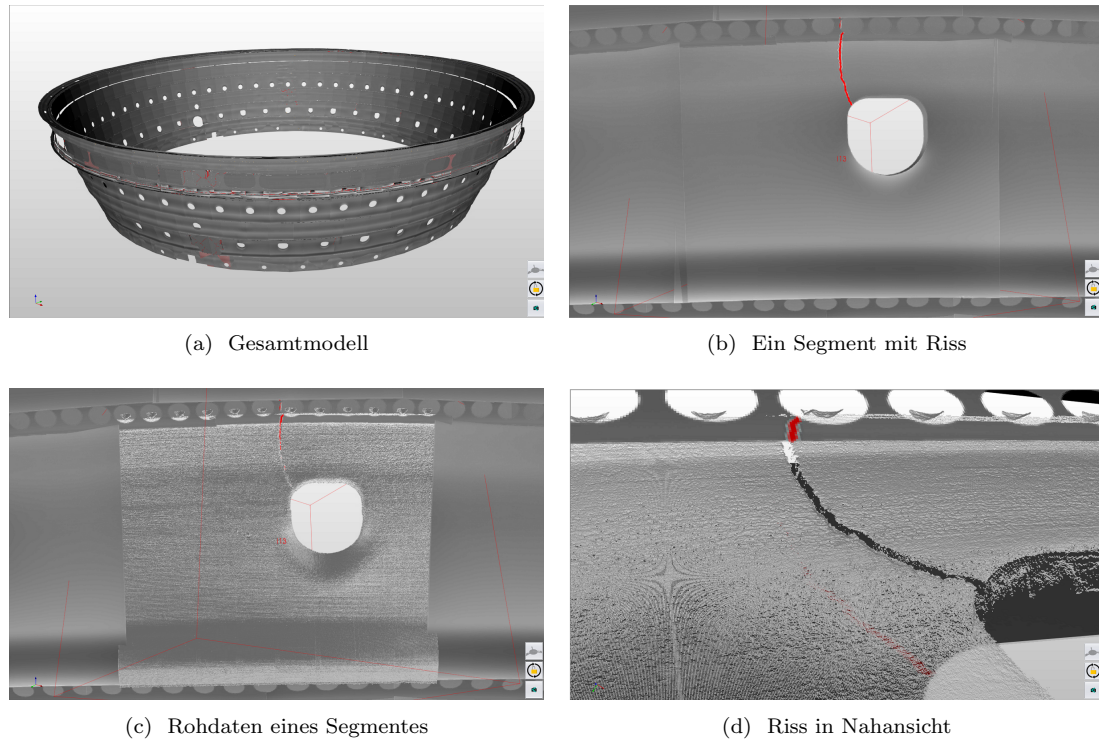


Abbildung 7.13.: verschiedene Ansichten des aufgenommenen Modells nach der automatischen Inspektion, dargestellt in der Software *AutoProcess*<sup>1</sup>

Das finale Modell kann außerdem herangezogen werden, um die qualitative Funktion der Prozesskette als Ganzes praktisch zu validieren. Sollte nur einer der Teilprozesse fehlerhaft sein, würden die Auswirkungen auf diesem Modell als Endergebnis sichtbar sein. Bei fehlerhafter oder nicht hinreichend genauer Kalibrierung würden die Daten der beiden Sensoren aufgrund der falschen Positions- oder Orientierungsinformationen nicht miteinander überlappen, sondern im schlimmsten Fall zufällig im Raum liegen. Dies würde eine Registrierung der Aufnahmen bei der Rissdetektion unmöglich machen und das Modell würde eine große Menge an Falschanzeigen aufweisen.

Bei einem fehlerbehafteten Digitalisierungsprozess würde das Modell Lücken oder andere Inhomogenitäten aufweisen. Lücken im digitalisierten Modell würden in der späteren Pfadplanung als gewollte Fehlstellen interpretiert werden und dann im endgültigen Modell ebenfalls zu Löchern führen, die im realen Bauteil nicht vorliegen.

Eine nicht funktionsfähige Pfadplanung wiederum führte dazu, dass anstatt auf der Bauteiloberfläche im leeren Raum vor oder hinter der Oberfläche Daten aufgenommen werden würden - mit der Konsequenz, dass diese Aufnahmen ausschließlich aus Rauschwerten bestehen würden, was im endgültigen Modell wiederum leicht identifiziert werden kann. Keines dieser Phänomene

<sup>1</sup>Software der Firma VMT Bildverarbeitungssysteme GmbH

tritt bei dem betrachteten Modell auf. Seine Oberfläche ist homogen und weist keine Lücken auf. Ebenso treten Aufnahmen, die vollständig aus Rauschwerten bestehen, nicht auf. Dies validiert qualitativ sowohl die Funktionalität aller Prozessschritte als auch deren Verkettung zu einem Gesamtprozess.

Durch diesen letzten Prozessschritt werden auch die in der Zielsetzung formulierten globalen Anforderungen validiert. Die in dem Forschungsprojekt spezifizierte Vollständigkeit bei der Inspektion der Bauteiloberfläche, die für die anschließende Rissdetektion Voraussetzung ist, konnte eingehalten werden. Für den gesamten Prozess wird bei dieser Messabdeckung eine Durchlaufzeit von unter acht Stunden benötigt, wodurch auch die globale Anforderung an die Durchlaufzeit eingehalten wird.



# 8. Zusammenfassung und Ausblick

## 8.1. Zusammenfassung

Aus Gründen der Sicherheit schreiben Luftfahrtbehörden die Instandhaltung von Flugzeugen in engen Intervallen vor. Diese trägt daher einen erheblichen Teil zu den Kosten bei, die im Laufe eines Flugzeuglebens anfallen. Dies gilt insbesondere für die hochbelastete Baugruppe der Triebwerke, die bei der Instandhaltung gesondert betrachtet wird. Viele Bauteile aus dem Triebwerk stellen teure Investitionsgüter dar, weswegen sie in den meisten Fällen nicht ausgetauscht, sondern stattdessen instandgesetzt werden. Im Bereich der Brennkammer sind typische Schäden Korrosion, Verbrennungen sowie Risse. Damit letztere zuverlässig detektiert werden, sind besondere Inspektionsverfahren nötig.

Im Rahmen eines Forschungsvorhabens sollte das lang etablierte, aber gleichzeitig Schwächen aufweisende Verfahren der FPI durch die optische Technologie der WLI substituiert werden. Damit diese effektiv eingesetzt werden kann, ist ein automatisches System zur Handhabung notwendig. Die Zielstellung dieser Arbeit ist die Entwicklung eines solchen automatischen Inspektionssystems. Hierfür mussten der Inspektionsaufbau und ein Inspektionsprozess konzipiert und später implementiert werden. Bei der Auswahl der Systemkomponenten mussten Erreichbarkeit, Zugänglichkeit, Messabdeckung sowie Schwingungen berücksichtigt werden. Die Herausforderungen im Inspektionsprozess entstammen der individuellen Geometrie der beschädigten Bauteile, die einen hohen Grad an Adaptivität bei dem System voraussetzen. Hierfür wird in dieser Arbeit ein neuartiger, zweistufiger Ansatz für die Pfadplanung vorgestellt. Dieser besteht aus einer modellbasierten Inspektionsplanung und einer sich anschließenden Pfadadaptation. Dabei wird zur Datenreduktion die Rotationssymmetrie des Bauteils ausgenutzt. Außerdem wurden die entsprechend benötigten intelligenten Softwarebausteine in einem Framework zu einer Software für die Off-Line-Programmierung sowie On-Line-Steuerung von Industrierobotern kombiniert. Für diesen off-Line ausgeführten Pfadplanungsansatz wird die reale Bauteiloberfläche benötigt. Diese wird in einem vorgelagerten Hilfsprozess mit einem zusätzlichen LLS-Sensor akquiriert und der Planung bereitgestellt. Alle Komponenten des Systems mussten außerdem zueinander kalibriert werden, um die notwendige Genauigkeit zu garantieren. Anhand eines realen Brennkammerbauteils wurde der gesamte Inspektionsprozess im Ganzen und jeder einzelne Hilfsprozess für sich qualitativ validiert. Schon mit der Entscheidung für die Substitution der FPI durch eine automatisierte WLI wurden die formulierten Anforderungen, was die Belastung von Mensch und Umwelt sowie die Reproduzierbarkeit betrifft, berücksichtigt. Zusätzlich generiert das in dieser Arbeit implementierte Inspektionssystem in einer vergleichbar kurzen Durchlaufzeit durchschnittlich 60.000 Einzelaufnahmen und 130 GB der vollständigen Bauteiloberfläche in Mikrometerauflösung, womit auch die weiteren quantitativen globalen Anforderungen erfüllt sind. Diese hoch aufgelösten Geometrieinformationen stellen eine wertvolle Erweiterung des Inspektionsresultates dar und stärken somit die Position des instandhaltenden Unternehmens gegenüber dem Hersteller



der betrachteten Bauteile. Der resultierende Forschungsdemonstrator (Abbildung 8.1) befindet sich derzeit in einer Industrialisierungsphase.



Abbildung 8.1.: Demonstratoranlage zur automatischen Inspektion mit Weißlichtinterferometrie

## 8.2. Ausblick

Das in dieser Arbeit entwickelte neuartige System zur roboterbasierten Rissinspektion von rotationssymmetrischen Triebwerksbauteilen mit einem Weißlichtinterferometer löst die an sie gestellte Aufgabenstellung vollumfänglich. Es weist allerdings einen hohen Grad an Spezialisierung hinsichtlich inspizierbarer Bauteile auf. Der implementierte Prozess lässt sich zwar mit relativ geringen Aufwand auf andere rotationssymmetrische Bauteile übertragen, darüber hinaus steigt jedoch der Aufwand für eine Portierung. Für Bauteile, die keine Symmetrie aufweisen, fällt die Möglichkeit zur Vereinfachung des Pfadplanungsproblems von drei auf zwei Dimensionen weg, sodass ein Großteil der in dieser Arbeit vorgestellten Algorithmen überarbeitet oder in manchen Fällen ausgetauscht werden müssten. Dies gilt insbesondere für die Strategie bei der Pfadadaptation als zweite Stufe der Pfadplanung. Trotz der genannten Schwierigkeiten weist der hier entwickelte Ansatz zur Inspektionsplanung ein großes Potential für die effektive Automatisie-



rung weiterer roboterbasierter Inspektions- oder Digitalisierungsprozesse auf. Dies gilt vor allem für Prozesse bei denen eine hohe Auflösung gefordert ist. Die Sensortechnologie ist noch immer Gegenstand von Forschungsprojekten, sodass auch in den kommenden Jahren mit einer weiteren Verbesserung zu rechnen ist. Ein vielversprechender Ansatz ist die kontinuierliche Vermessung mit Weißlichtinterferometrie. Diese Technologie würde das Anwendungsspektrum weiter erhöhen, da es eine erheblich höhere Messgeschwindigkeit erlaubt und aufgrund einer alternativen Messstrategie deutlich weniger Schwingungen generiert (BOD17).

Die Software *AutoXIV* bietet dem Anwender grundlegende Funktionen und Datenstrukturen zur Robotersteuerung, zur Datenvisualisierung sowie -manipulation, die auch für viele andere Aufgabenstellungen aus dem Bereich der industrierobotergetriebenen Automatisierung sowie Off-Line-Simulation eingesetzt werden können. Die Software baut auf Bibliotheken auf, die von einer großen Gemeinschaft gepflegt und weiterentwickelt werden, sodass sich das Spektrum an Funktionalitäten in der Zukunft ohne eigenen Aufwand kontinuierlich erweitern wird. Um die Portierbarkeit der Software noch weiter vereinfachen zu können, müsste der Grad ihrer Modularität erhöht werden. Nur so können andere Anwender ihre spezialisierten Toolboxes einfacher einbinden, ohne dass zentrale Teile der Architektur angepasst werden müssten.



# A. Anhang

## A.1. Kalibrierprotokolle WLI-Sensor

Für alle Koordinaten X,Y und Z sowie für den Radius gilt die Einheit [mm], für die Winkel RX,RY und RZ gilt die Einheit [°].

**Verwendete Positionen des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Rotation [X Y Z]:**

940.424 -109.554 567.615  
941.796 -112.531 591.657  
942.713 -91.488 591.936  
942.613 -91.200 568.934  
938.454 -98.928 568.882  
939.561 -99.293 590.886  
942.641 -85.913 581.099  
937.461 -111.971 580.833

**Verwendete Positionen des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Position [X Y Z]:**

935.409 -98.3401 570.768  
1046.66 -115.562 631.057  
1103.07 -227.908 591.397  
965.207 13.734 583.509

**Verwendete Orientierung des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Rotation [RX RY RZ]:**

179.279 0.018 -157.041

**Verwendete Orientierungen des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Position [RX RY RZ]:**

179.275 0.021 -157.042  
162.914 -25.443 -152.696  
151.198 -8.966 -115.002  
-146.97 -24.990 -157.465

**Messung 1:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.950 -11.820 18.664 25  
-12.392 12.367 17.132 25  
8.801 12.246 18.810 25  
8.581 -10.817 18.741 25  
0.334 -10.718 21.715 25  
0.521 11.348 20.951 25  
14.146 1.245 19.429 25  
-12.549 1.491 21.182 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.524 1.347 24.942 25  
-2.403 -7.262 23.745 25  
-1.723 -3.404 25.05 25  
-1.095 -4.224 24.836 25

Die errechneten Koordinaten der Werkzeugspitze in [X Y Z RX RY RZ]:

-169.89 -29.737 179.153 89.930 -0.431 -59.970

**Messung 2:**

Kugelzentren bei Kalibrierung der Orientierung in [X Y Z Radius]:

-9.931 -11.691 18.702 25  
-12.355 12.495 17.155 25  
8.825 12.372 18.817 25  
8.616 -10.695 18.764 25  
0.374 -10.593 21.749 25  
0.555 11.470 20.968 25  
14.181 1.373 19.437 25  
-12.515 1.627 21.207 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.489 1.481 24.957 25  
-2.375 -7.148 23.802 25  
-1.684 -3.303 25.075 25  
-1.123 -4.068 24.851 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

[X Y Z RX RY RZ]  
-169.859 -29.772 179.115 89.972 -0.44 -60.015

### Messung 3:

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.981 -11.697 18.673 25  
-12.408 12.496 17.128 25  
8.773 12.368 18.799 25  
8.57 -10.694 18.744 25  
0.32 -10.595 21.727 25  
0.504 11.469 20.944 25  
14.121 1.377 19.433 25  
-12.569 1.630 21.188 25

Kugelzentren bei Kalibrierung der Position in [X Y Z Radius]:

-0.541 1.485 24.94 25  
-2.423 -7.129 23.782 25  
-1.727 -3.276 25.081 25  
-1.16 -4.093 24.841 25

Die errechneten Koordinaten der Werkzeugspitze sind bzw. [X Y Z RX RY RZ]:

-169.867 -29.763 179.1 89.967 -0.436 -59.984

**Messung 4:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.937 -11.660 18.690 25  
-12.364 12.529 17.145 25  
8.817 12.405 18.811 25  
8.610 -10.661 18.760 25  
0.365 -10.556 21.745 25  
0.550 11.506 20.959 25  
14.169 1.41 19.428 25  
-12.524 1.66 21.206 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.5 1.519 24.954 25  
-2.381 -7.11 23.807 25  
-1.676 -3.269 25.093 25  
-1.152 -4.041 24.851 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.87 -29.766 179.105 89.97 -0.438 -60.004

**Messung 5:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.978 -11.609 18.67 25  
-12.403 12.588 17.131 25  
8.782 12.465 18.797 25  
8.584 -10.608 18.739 25  
0.328 -10.504 21.726 25  
0.511 11.568 20.945 25  
14.142 1.468 19.417 25  
-12.567 1.721 21.189 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.537 1.582 24.940 25  
-2.418 -7.051 23.825 25  
-1.707 -3.2 25.119 25  
-1.213 -4.015 24.842 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.857 -29.768 179.089 89.958 -0.433 -60.002

### **Messung 6:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.997 -11.739 18.659 25  
-12.422 12.446 17.119 25  
8.756 12.324 18.785 25  
8.558 -10.743 18.727 25  
0.303 -10.644 21.71 25  
0.489 11.427 20.928 25  
14.114 1.334 19.404 25  
-12.582 1.583 21.171 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.557 1.439 24.922 25  
-2.438 -7.16 23.737 25  
-1.768 -3.31 25.067 25  
-1.145 -4.14 24.83 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.863 -29.764 179.093 89.963 -0.433 -59.998

**Messung 7:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.984 -11.688 18.662 25  
-12.411 12.496 17.12 25  
8.766 12.370 18.785 25  
8.573 -10.696 18.723 25  
0.318 -10.593 21.713 25  
0.498 11.475 20.927 25  
14.124 1.381 19.414 25  
-12.572 1.633 21.178 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.546 1.492 24.931 25  
-2.434 -7.12 23.780 25  
-1.74 -3.275 25.089 25  
-1.17 -4.085 24.831 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.859 -29.773 179.097 89.962 -0.433 -60.003

**Messung 8:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.967 -11.732 18.679 25  
-12.392 12.464 17.141 25  
8.791 12.336 18.81 25  
8.595 -10.734 18.748 25  
0.34 -10.632 21.739 25  
0.524 11.437 20.954 25  
14.146 1.339 19.446 25  
-12.558 1.592 21.2 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:



-0.527 1.45 24.957 25  
-2.416 -7.167 23.778 25  
-1.743 -3.318 25.073 25  
-1.12 -4.108 24.847 25

[X Y Z RX RY RZ]  
-169.847 -29.759 179.085 89.953 -0.436 -59.982

**Messung 9:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-10.055 -11.744 18.626 25  
-12.482 12.447 17.085 25  
8.7 12.321 18.754 25  
8.49 -10.738 18.699 25  
0.242 -10.639 21.681 25  
0.426 11.428 20.895 25  
14.054 1.329 19.373 25  
-12.649 1.581 21.144 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.62 1.434 24.902 25  
-2.5 -7.145 23.722 25  
-1.833 -3.286 25.073 25  
-1.195 -4.178 24.793 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.852 -29.768 179.098 89.966 -0.435 -59.993

**Messung 10:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-10.03 -11.721 18.65 25  
-12.463 12.466 17.097 25  
8.720 12.338 18.773 25  
8.512 -10.718 18.719 25  
0.265 -10.621 21.705 25  
0.453 11.443 20.914 25  
14.078 1.348 19.391 25  
-12.624 1.598 21.173 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.598 1.459 24.927 25  
-2.48 -7.138 23.758 25  
-1.788 -3.293 25.101 25  
-1.193 -4.148 24.829 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.861 -29.7662 179.103 89.973 -0.435 -59.997

**Messung 11:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-9.971 -11.7 18.693 25  
-12.407 12.485 17.137 25  
8.777 12.359 18.808 25  
8.566 -10.696 18.76 25  
0.321 -10.594 21.742 25  
0.51 11.467 20.951 25  
14.116 1.368 19.459 25  
-12.568 1.62 21.205 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.542 1.476 24.959 25  
-2.427 -7.144 23.804 25  
-1.718 -3.288 25.094 25  
-1.16 -4.099 24.863 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.863 -29.762 179.116 89.988 -0.435 -59.982

### **Messung 12:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

-10.009 -11.721 18.663 25  
-12.434 12.466 17.121 25  
8.749 12.342 18.786 25  
8.541 -10.719 18.728 25  
0.289 -10.618 21.720 25  
0.475 11.442 20.933 25  
14.096 1.35 19.418 25  
-12.597 1.604 21.184 25

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

-0.574 1.457 24.941 25  
-2.455 -7.144 23.765 25  
-1.762 -3.296 25.097 25  
-1.18 -4.132 24.839 25

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-169.858 -29.765 179.104 89.962 -0.438 -59.997

## A.2. Kalibrierprotokolle LLS-Sensor

**Verwendete Positionen des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Rotation[X Y Z]:**

955.247 -51.969 503.173  
 957.897 -31.478 513.37  
 954.715 -59.224 512.869  
 958.467 -28.805 519.765  
 978.216 -61.960 512.136  
 981.726 -31.814 512.715  
 985.645 -59.241 516.227  
 988.097 -39.742 519.288

**Verwendete Positionen des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Position[X Y Z]:**

964.372 -45.613 509.622  
 1059.8 -58.556 587  
 1028.8 -189.994 536.3  
 957.537 -4.387 497.258

**Verwendete Orientierung des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Rotation [RX RY RZ]:**

178.555 -0.437 -156.241

**Verwendete Orientierungen des Roboterflansches bei der Bestimmung der Werkzeugspitzen-Position [RX RY RZ]:**

178.556 -0.438 -156.241  
 156.717 -32.036 -148.711  
 157.443 -4.711 -111.448  
 -160.371 -15.314 -158.126

**Messung 1:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

9.899 8.401 124.796 22.956  
-10.701 -1.51 124.57 22.594  
17.109 -1.415 124.56 18.228  
-13.696 -7.905 124.32 20.914  
17.356 -1.143 101.043 17.993  
-13.0399 -1.3 100.73 21.329  
13.755 -5.336 94.011 20.875  
-5.327 -8.195 93.829 24.425

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.482 -8.427 116.589 24.756  
2.982 -3.221 127.088 24.821  
7.435 2.351 142.78 23.868  
8.813 6.172 132.2 23.394

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.558 -27.194 108.987 -90.034 -0.427 119.782

**Messung 2:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.156 8.412 125.035 22.843  
-10.523 -1.5 124.787 22.677  
17.362 -1.382 124.815 17.987  
-13.361 -7.909 124.626 21.13  
17.590 -1.131 101.302 17.764  
-12.829 -1.277 100.946 21.457  
14.140 -5.314 94.148 20.616  
-5.776 -8.142 93.902 24.323

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.362 -8.436 116.898 24.772  
3.663 -3.0281 127.227 24.730  
7.496 2.441 143.122 23.849  
8.931 6.305 132.488 23.35

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.807 -26.848 108.894 -90.099 -0.358 119.954

### **Messung 3:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.113 8.358 125.041 22.863  
-10.501 -1.551 124.828 22.687  
17.303 -1.443 124.82 18.044  
-13.341 -7.972 124.692 21.142  
17.549 -1.181 101.267 17.804  
-12.88 -1.313 100.959 21.426  
13.979 -5.366 94.207 20.726  
-5.425 -8.228 93.982 24.404

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.609 -8.489 116.835 24.738  
3.686 -3.067 127.255 24.726  
7.403 2.391 143.111 23.878  
8.862 6.239 132.505 23.376

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.927 -26.919 108.626 -90.035 -0.443 119.941

**Messung 4:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.242 8.331 124.987 22.805  
-10.516 -1.587 124.785 22.680  
17.375 -1.474 124.78 17.974  
-13.396 -7.964 124.609 21.108  
17.583 -1.193 101.275 17.771  
-12.83 -1.329 100.944 21.456  
14.071 -5.370 94.166 20.663  
-5.111 -8.242 94.038 24.471

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.747 -8.481 116.813 24.717  
3.441 -3.086 127.294 24.762  
7.417 2.379 143.153 23.874  
8.918 6.235 132.481 23.355

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.717 -27.107 108.761 -90.028 -0.399 119.821

**Messung 5:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.048 8.342 125.068 22.891  
-10.457 -1.604 124.835 22.707  
17.323 -1.481 124.813 18.025  
-13.408 -7.991 124.64 21.1  
17.552 -1.215 101.277 17.802  
-12.792 -1.369 101 21.478  
13.984 -5.412 94.219 20.722  
-5.1474 -8.272 94.043 24.464

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.812 -8.495 116.814 24.707  
3.591 -3.088 127.294 24.740  
7.545 2.346 143.075 23.834  
8.932 6.187 132.476 23.349

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.626 -27.240 108.715 -90.042 -0.539 119.831

### **Messung 6:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.245 8.242 124.969 22.804  
-10.528 -1.661 124.772 22.674  
17.336 -1.542 124.791 18.012  
-13.337 -8.069 124.632 21.145  
17.545 -1.277 101.28 17.808  
-12.892 -1.423 100.902 21.419  
14.0488 -5.458 94.155 20.679  
-5.445 -8.314 93.952 24.399

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.868 -8.560 116.775 24.699  
3.523 -3.158 127.316 24.750  
7.356 2.300 143.183 23.893  
9.021 6.16 132.446 23.315

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.978 -26.947 108.727 -90.026 -0.354 119.921



**Messung 7:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.27 8.383 124.965 22.793  
-10.475 -1.532 124.776 22.699  
17.333 -1.401 124.848 18.014  
-13.212 -7.96 124.704 21.223  
17.571 -1.136 101.301 17.783  
-12.716 -1.294 100.999 21.524  
14.084 -5.317 94.170 20.654  
-5.632 -8.144 93.907 24.357

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.716 -8.435 116.818 24.722  
3.581 -3.061 127.238 24.742  
7.531 2.397 143.122 23.838  
8.848 6.297 132.511 23.381

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.887 -26.768 108.547 -90.00 -0.337 119.993

**Messung 8:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.190 8.347 124.995 22.828  
-10.495 -1.593 124.777 22.69  
17.359 -1.454 124.785 17.99  
-13.349 -7.985 124.622 21.135  
17.574 -1.193 101.285 17.780  
-12.812 -1.34 100.951 21.467  
14.010 -5.384 94.205 20.705  
-5.581 -8.209 93.920 24.369

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.724 -8.483 116.815 24.721  
3.515 -3.095 127.275 24.751  
7.387 2.357 143.163 23.883  
9.09058 6.227 132.41 23.288

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.994 -26.854 108.906 -90.057 -0.366 119.972

**Messung 9:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.164 8.386 125.007 22.84  
-10.495 -1.517 124.799 22.69  
17.283 -1.399 124.859 18.063  
-13.353 -7.939 124.654 21.134  
17.547 -1.162 101.291 17.809  
-12.8377 -1.282 100.957 21.452  
14.011 -5.348 94.185 20.704  
-5.135 -8.194 94.027 24.466

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.85 -8.432 116.796 24.701  
3.557 -3.055 127.242 24.745  
7.338 2.4 143.159 23.898  
8.925 6.254 132.459 23.352

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.832 -27.194 108.547 -89.962 -0.431 119.793

**Messung 10:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.085 8.368 125.071 22.875  
-10.446 -1.574 124.82 22.712  
17.365 -1.458 124.814 17.984  
-13.388 -7.955 124.612 21.112  
17.595 -1.193 101.284 17.759  
-12.74 -1.325 101.016 21.509  
14.065 -5.381 94.203 20.668  
-5.06 -8.25 94.059 24.482

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.946 -8.455 116.803 24.686  
3.593 -3.074 127.275 24.74  
7.517 2.377 143.112 23.842  
8.85 6.251 132.536 23.38

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.6223 -27.2881 108.639 -90.0535 -0.533077 119.781

**Messung 11:**

Kugelzentren bei Kalibrierung der Orientierung [X Y Z Radius]:

10.243 8.362 124.983 22.805  
-10.425 -1.563 124.803 22.722  
17.402 -1.434 124.772 17.949  
-13.411 -7.939 124.573 21.098  
17.597 -1.169 101.3 17.757  
-12.593 -1.325 101.079 21.596  
14.117 -5.34 94.164 20.632  
-4.977 -8.22 94.045 24.499

Kugelzentren bei Kalibrierung der Position [X Y Z Radius]:

3.763 -8.45 116.827 24.715

3.734 -3.069 127.222 24.719

7.551 2.372 143.124 23.832

9.029 6.275 132.445 23.312

Die errechneten Koordinaten der Werkzeugspitze sind [X Y Z RX RY RZ]:

-40.658 -27.284 108.733 -90.046 -0.428 119.792

# Literaturverzeichnis

- [Aub04] AUBIN, Bruce R.: *Aircraft maintenance: The art and science of keeping aircraft safe*. Warrendale, Pa. (400 Commonwealth Dr., Wallendale PA USA) : Society of Automotive Engineers, 2004 (Society of Automotive Engineers. Electronic publications)
- [Bau16] BAUERNHANS, Thomas (Hrsg.): *ROS-Industrial. Technologies, Trends, Applications*. 2016
- [BC96] BEASLEY, J.E ; CHU, P.C: A genetic algorithm for the set covering problem. In: *European Journal of Operational Research* 94 (1996), Nr. 2, S. 392–404
- [BD90] BOWYER, Kevin W. ; DYER, Charles R.: Aspect graphs: An introduction and survey of recent results. In: *International Journal of Imaging Systems and Technology* 2 (1990), Nr. 4, S. 315–328
- [Bey05] BEYER, Lukas: *Forschungsberichte aus dem Laboratorium Fertigungstechnik / Helmut-Schmidt-Universität, Universität der Bundeswehr Hamburg*. Bd. 1: *Genauigkeitssteigerung von Industrierobotern: Insbesondere mit Parallelkinematik: Helmut-Schmidt-Univ., Univ. der Bundeswehr, Diss.–Hamburg, 2004*. Aachen : Shaker, 2005
- [BH88] BIERINGER, Robert J. (Hrsg.) ; HARDING, Kevin G. (Hrsg.): *1988 Dearborn Symposium*. SPIE, 1988 (SPIE Proceedings)
- [BOB15] BELOBABA, P. ; ODONI, A. ; BARNHART, C.: *The Global Airline Industry*. Wiley, 2015
- [BOD17] BAHR, Sönke ; OTTO, Marc A. ; DOMASCHKE, Tomas: Continuous Digitalization of Rotationally Symmetrical Components With a Lateral Scanning White-Light Interferometer. In: *MHI-Fachkolloquium 2017*, 2017
- [BPF12] BEYERER, Jürgen ; PUENTE LEÓN, Fernando ; FRESE, Christian: *Automatische Sichtprüfung: Grundlagen, Methoden und Praxis der Bildgewinnung und Bildauswertung*. Berlin, Heidelberg : Springer, 2012
- [BRB88] BLAIS, Francois ; RIOUX, Marc ; BERALDIN, J.-Angelo: <title>Practical Considerations For A Design Of A High Precision 3-D Laser Scanner System</title>. In: BIERINGER, Robert J. (Hrsg.) ; HARDING, Kevin G. (Hrsg.): *1988 Dearborn Symposium*, SPIE, 1988 (SPIE Proceedings), S. 225–246
- [BWC13] BAUER, Wilfried ; WEBER, Mark ; CHANBAI, Sirichanok: White Light Interferometry. In: WANG, Q. J. (Hrsg.) ; CHUNG, Yip-Wah (Hrsg.): *Encyclopedia of Tribology*. Boston, MA, s.l. : Springer US, 2013, S. 4115–4127

- [CK88] COWAN, C. K. ; KOVESI, P. D.: Automatic sensor placement from vision task requirements. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10 (1988), Nr. 3, S. 407–416
- [CLK11] CHEN, S. ; LI, Y. ; KWOK, N. M.: Active vision in robotic systems: A survey of recent developments. In: *The International Journal of Robotics Research* 30 (2011), Nr. 11, S. 1343–1377
- [CLR07] CORMEN, Thomas H. ; LEISERSON, Charles E. ; RIVEST, Ronald L.: *Introduction to algorithms*. 2nd. ed., 10th printing. Cambridge (Massachusetts) and London and Boston [etc.] : MIT Press and McGraw-Hill, 2007
- [CTF00] CAPRARA, Alberto ; TOTH, Paolo ; FISCHETTI, Matteo: Algorithms for the set covering problem. In: *Annals of Operations Research* 98 (2000), Nr. 1/4, S. 353–371
- [CW08] CLAUSING, Nils ; WULFSBERG, Jens: Locally Flexible Measuring Robot for Quality Control. In: *Proceedings of the 2nd CIRP Conference on Assembly Technologies and Systems*, 2008, S. 59–66
- [Dem06] DEMTRÖDER, Wolfgang: *Experimentalphysik*. 4., überarb. und erw. Aufl. Berlin : Springer, 2006 (Springer-Lehrbuch)
- [Dil06] DILTHEY, Ulrich: *Schweißtechnische Fertigungsverfahren 1: Schweiß- und Schneidtechnologien*. 3rd ed. Dordrecht : Springer, 2006 (VDI-Buch)
- [DIN00] DIN: *Grundlagen der Instandhaltung*. 2012-09-00
- [Eri08] ERICSON, Christer: *Real-time collision detection*. [Nachdr.]. Amsterdam : Elsevier Morgan Kaufmann, 2008 (Morgan Kaufmann series in interactive 3D technology)
- [For15] FORSBERG, Dick: *Aircraft Retirement and Storage Trends: Economic Life Analysis Reprised and Expanded*. 2015
- [GB03] GEOFFREY BIGGS ; BRUCE MACDONALD: A Survey of Robot Programming Systems. In: *Proceedings of the Australasian Conference on Robotics and Automation*, 2003
- [GC13] GALCERAN, Enric ; CARRERAS, Marc: A survey on coverage path planning for robotics. In: *Robotics and Autonomous Systems* 61 (2013), Nr. 12, S. 1258–1276
- [GG06] GEVATTER, H. J. ; GRÜNHaupt, U.: *Handbuch der Mess- und Automatisierungstechnik in der Produktion*. Springer, 2006
- [Gmb16] GMBH, 3D-Shape ; 3D-SHAPE (Hrsg.): *Beschleunigte Qualitätskontrolle: Höchste Leistungsfähigkeit in der Weißlichtinterferometrie*. [http://www.3d-shape.com/broschueren/Brochure\\_Metal\\_KORAD%203D\\_DE\\_2016-04\\_low.pdf](http://www.3d-shape.com/broschueren/Brochure_Metal_KORAD%203D_DE_2016-04_low.pdf). Version: 2016

- [GR70] GOLUB, G. H. ; REINSCH, C.: Singular value decomposition and least squares solutions. In: *Numerische Mathematik* 14 (1970), Nr. 5, S. 403–420
- [GT11] GAN, Zhongzue ; TANG, Qing: *Visual sensing and its applications: Integration of laser sensors to industrial robots*. Hangzhou and Heidelberg : Zhejiang University Press Heidelberg and Springer, 2011 (Advanced topics in science and technology in China)
- [Har07] HARIHARAN, P.: *Basics of interferometry*. 2nd ed. Amsterdam, Boston : Elsevier Academic Press, 2007
- [Hay08] HAYNES INTERNATIONAL: *HIGH-TEMPERATURE ALLOYS: Hastelloy X Alloy*. Kokomo, 2008
- [HC11] HARRIS, Adam ; CONRAD, James M.: Survey of popular robotics simulators, frameworks, and toolkits. In: *SOUTHEASTCON 2011*, 2011, S. 243–249
- [Hin13] HINSCH, Martin: *Industrielles Luftfahrtmanagement: Technik und Organisation luftfahrttechnischer Betriebe*. 2., überarb. und erw. Aufl. 2013. Berlin, Heidelberg : Springer, 2013
- [HSE14] HOSSEININAVEH, Ali ; SARGEANT, Ben ; ERFANI, Tohid: Towards fully automatic reliable 3D acquisition: From designing imaging network to a complete and accurate point cloud. In: *Robotics and Autonomous Systems* 62 (2014), Nr. 8, S. 1197–1207
- [Hug14] HUGHES, John F.: *Computer graphics: Principles and practice*. Third edition. Upper Saddle River, New Jersey : Addison-Wesley, 2014
- [ICS05] IMPOCO, G. ; CIIGNONI, P. ; SCOPIGNO, R.: A Six-Degrees-of-Freedom Planning Algorithm for the Acquisition of Complex Surfaces. In: *International Journal of Shape Modeling* 11 (2005), Nr. 01, S. 1–23
- [IFF96] IERUSALIMSKY, Roberto ; FIGUEIREDO, Luiz Henrique d. ; FILHO, Waldemar C.: Lua—An Extensible Extension Language. In: *Software: Practice and Experience* 26 (1996), Nr. 6, S. 635–652
- [Jac07] JACKSON, J.: Microsoft robotics studio: A technical introduction. In: *IEEE Robotics & Automation Magazine* 14 (2007), Nr. 4, S. 82–87
- [JD11] JIANG, Yi ; DING, Wenhui: Recent developments in fiber optic spectral white-light interferometry. In: *Photonic Sensors* 1 (2011), Nr. 1, S. 62–71
- [Kim04] KIM, J. Y.: CAD-based automated robot programming in adhesive spray systems for shoe outsoles and uppers. In: *Journal of Robotic Systems* 21 (2004), Nr. 11, S. 625–634

- [Kou16] KOUBAA, Anis (Hrsg.): *Studies in Computational Intelligence*. Bd. 625: *Robot Operating System (ROS): The Complete Reference (Volume 1)*. 1st ed. 2016. Cham, s.l. : Springer International Publishing, 2016
- [LG04] LI, Yadong ; GU, Peihua: Free-form surface inspection techniques state of the art review. In: *Computer-Aided Design* 36 (2004), Nr. 13, S. 1395–1417
- [LK08] LARSSON, Sören ; KJELLANDER, J.A.P.: Path planning for laser scanning with an industrial robot. In: *Robotics and Autonomous Systems* 56 (2008), Nr. 7, S. 615–624
- [LKR15] LI, Yinan ; KÄSTNER, Markus ; REITHMEIER, Eduard: Development of a compact low coherence interferometer based on GPGPU for fast microscopic surface measurement on turbine blades. In: LEHMANN, Peter (Hrsg.) ; OSTEN, Wolfgang (Hrsg.) ; ALBERTAZZI GONÇALVES, Armando (Hrsg.): *SPIE Optical Metrology*, SPIE, 2015 (SPIE Proceedings), S. 95250R
- [LR06] LENNARCIC, Jadran (Hrsg.) ; ROTH, B. (Hrsg.): *Advances in Robot Kinematics: Mechanisms and Motion*. Dordrecht : Springer, 2006
- [MB12] MOUTCHHO, Tekouo ; BRICE, William: *A New Programming Approach for Robot based Flexible inspection Systems*. München, Technische Universität München, Dissertation, 2012
- [MF98] MASSIOS, N. A. ; FISHER, R. B.: A Best Next View Selection Algorithm Incorporating a Quality Criterion. In: NIXON, Mark (Hrsg.) ; LEWIS, Paul (Hrsg.) ; CARTER, John (Hrsg.): *British Machine Vision Conference 1998*, 1998, S. 78.1–78.10
- [MGBC05] MARTINS, Fernando António R. ; GARCÍA-BERMEJO, Jaime G. ; CASANOVA, Eduardo Z.: Automated 3D surface scanning based on CAD model. In: *Mechatronics* 15 (2005), Nr. 7, S. 837–857
- [MSTL06] MEHDI-SOUZANI, Charyar ; THIÉBAUT, François ; LARTIGUE, Claire: Scan Planning Strategy for a General Digitized Surface. In: *Journal of Computing and Information Science in Engineering* 6 (2006), Nr. 4, S. 331
- [Mun11] MUNKELT, Christoph: *Aktive daten- und modellbasierte Sensorpositionierung zur 3-D Vermessung*. Jena, Friedrich-Schiller Universität, Dissertation, 21.10/2011
- [NJ95] NEWMAN, Timothy S. ; JAIN, Anil K.: A Survey of Automated Visual Inspection. In: *Computer Vision and Image Understanding* 61 (1995), Nr. 2, S. 231–262
- [PH13] PORDEL, Mostafa ; HELLSTRÖM, Thomas: *Robtic architecture frameworks, available tools and further requirements*. Umeå, Umeå University, Report, 2013



- [PHS08] PULKKINEN, Topi ; HEIKKILA, Tapio ; SALLINEN, Mikko: 2D CAD based robot programming for processing metal profiles in short series manufacturing. In: *2008 International Conference on Control, Automation and Systems (ICCAS)*, 2008, S. 156–162
- [PLB03] PRIETO, Flavio ; LEPAGE, Richard ; BOULANGER, Pierre: A CAD-based 3D data acquisition strategy for inspection. In: *Machine Vision and Applications* 15 (2003), Nr. 2, S. 76–91
- [PPL12] PAN, Zengxi ; POLDEN, Joseph ; LARKIN, Nathan: Recent progress on programming methods for industrial robots. In: *Robotics and Computer-Integrated Manufacturing* 28 (2012), Nr. 2, S. 87–94
- [QCG09] QUINGLEY, Morgan ; CONLEY, Ken ; GERKEY, Brian P.: ROS: an open-source Robot Operating System. In: *ICRA Workshop on Open Source Software* Bd. 3, 2009
- [RMH13] ROSSANO, Gregory F. ; MARTINEZ, Carlos ; HEDELIND, Mikael: Easy robot programming concepts: An industrial perspective. In: *2013 IEEE International Conference on Automation Science and Engineering (CASE 2013)*, 2013, S. 1119–1126
- [RSF13] ROHMER, Eric ; SINGH, Surya P. N. ; FREESE, Marc: V-REP: A versatile and scalable robot simulation framework. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, 2013, S. 1321–1326
- [Rus13] RUSU, Radu B.: *Springer Tracts in Advanced Robotics*. Bd. 85: *Semantic 3D Object Maps for Everyday Robot Manipulation*. Berlin and Heidelberg : Springer, 2013
- [Sco09] SCOTT, William R.: Model-based view planning. In: *Machine Vision and Applications* 20 (2009), Nr. 1, S. 47–69
- [Sen93] SEN, Sandip: Minimal cost set covering using probabilistic methods. In: DEATON, Ed (Hrsg.) ; GEORGE, K. M. (Hrsg.) ; BERGHEL, Hal (Hrsg.) ; HEDRICK, George (Hrsg.): *the 1993 ACM/SIGAPP symposium*, 1993, S. 157–164
- [SFW14] SCHNEIDER, Max ; FEY, Dietmar ; WENZEL, Kay: A Generic Approach for Analysis of White-Light Interferometry Data via User-Defined Algorithms. In: MURGANTE, Beniamino (Hrsg.): *Computational science and its applications – ICCSA 2014* Bd. 8582. Cham : Springer, 2014, S. 522–537
- [SHM00] SCHWARTE, Rudolf ; HÄUSLER, Gerd ; MALZ, Reinhard W.: Three-Dimensional Imaging Techniques. In: *Computer Vision and Applications*. Elsevier, 2000, S. 177–208
- [SKL03] SON, Seokbae ; KIM, Seungman ; LEE, Kwan H.: Path planning of multi-patched freeform surfaces for laser scanning. In: *The International Journal of Advanced Manufacturing Technology* 22 (2003), Nr. 5-6, S. 424–435

- [SRR01] SCOTT, William R. ; ROTH, Gerhard ; RIVEST, Jeremy: View Planning for Multi-Stage Object Reconstruction. In: *Vision Interface* (2001), S. 64–71
- [SRR03] SCOTT, William R. ; ROTH, Gerhard ; RIVEST, Jean-François: View planning for automated three-dimensional object reconstruction and inspection. In: *ACM Computing Surveys* 35 (2003), Nr. 1, S. 64–96
- [STD09] SANSONI, Giovanna ; TREBESCHI, Marco ; DOCCHIO, Franco: State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation. In: *Sensors (Basel, Switzerland)* 9 (2009), Nr. 1, S. 568–601
- [SY04] SIVRIKAYA, F. ; YENER, B.: Time synchronization in sensor networks: A survey. In: *IEEE Network* 18 (2004), Nr. 4, S. 45–50
- [TAT95] TARABANIS, K. A. ; ALLEN, P. K. ; TSAI, R. Y.: A survey of sensor planning in computer vision. In: *IEEE Transactions on Robotics and Automation* 11 (1995), Nr. 1, S. 86–104
- [TG95] TARBOX, G. H. ; GOTTSCHLICH, S. N.: Planning for Complete Sensor Coverage in Inspection. In: *Computer Vision and Image Understanding* 61 (1995), Nr. 1, S. 84–111
- [TLJ02] TING, Y. ; LEI, W. I. ; JAR, H. C.: A path planning algorithm for industrial robots. In: *Computers & Industrial Engineering* 42 (2002), Nr. 2-4, S. 299–308
- [TUWR97] TRUCCO, E. ; UMASUTHAN, M. ; WALLACE, A. M. ; ROBERTO, V.: Model-based planning of optimal sensor placements for inspection. In: *IEEE Transactions on Robotics and Automation* 13 (1997), Nr. 2, S. 182–194
- [Wec05] WECK, Manfred: *Werkzeugmaschinen 1*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2005
- [Zan14] ZANGEMEISTER, Christof: *Nutzwertanalyse in der Systemtechnik: Eine Methodik zur multidimensionalen Bewertung und Auswahl von Projektalternativen: Teilw. zugl.: Berlin, Univ., Diss., 1970*. 5., erw. Aufl. Norderstedt : Books on Demand, 2014
- [ZQC13] ZHU, Weidong ; QU, Weiwei ; CAO, Lianghong: An off-line programming system for robotic drilling in aerospace manufacturing. In: *The International Journal of Advanced Manufacturing Technology* 68 (2013), Nr. 9-12, S. 2535–2545

## B. Lebenslauf

Name	Domaschke
Vorname	Tomas
Staatsangehörigkeit	deutsch
Geburtsdatum	17.09.1984
Geburtsort,-land	Hamburg, Deutschland

1991-1995	Grundschule Barsbüttel
1995-2004	Gymnasium im Schulzentrum Glinde
2004-2005	Wehrpflicht im 162. Logistikbatallion, Seeth
2005-2011	Studium zum Diplom der Elektrotechnik an der Technischen Universität Hamburg-Harburg
2011-2017	Wissenschaftlicher Mitarbeiter am Institut für Flugzeug-Produktionstechnik der Technischen Universität Hamburg-Harburg