

Pfadplanung für robotergeführte Ultraschallprüfung auf Basis lokaler geometrischer Eigenschaften

Sebastian ZAMBAL¹, Kastor FELSNER¹, Klaus SCHLACHTER¹,
Christian EITZINGER¹

¹ PROFACOR GmbH, Im Stadtgut D1, Steyr-Gleink, Österreich

Kontakt E-Mail: sebastian.zambal@profactor.at

Kurzfassung. Pfadplanung für die robotergeführte Inspektion von Bauteilen mit komplexer Geometrie stellt in vielen Anwendungsbereichen eine Herausforderung dar. In der Regel müssen viele Randbedingungen berücksichtigt werden: Optimale Positionierung der Sensorik relativ zum Bauteil, möglichst kurze Pfade (Optimierung der Inspektionszeiten), maximale Abdeckung des Bauteils, Vermeidung von Kollisionen, etc.

Wir schlagen eine Methode zur automatischen Bewegungsplanung eines Roboters für die Inspektion von komplex geformten Bauteilen vor. Die Methode basiert auf der Berechnung eines Vektorfeldes von lokalen Krümmungen der Bauteilgeometrie. Teilbewegungen der robotergeführten Sensorik werden zunächst entlang von Strömungslinien dieses Vektorfeldes geplant. Einzelne Teilbewegungen werden dann unter Berücksichtigung von Kollisionsvermeidung und Abdeckung zu einem kompletten Pfad kombiniert. Am Beispiel von Ultraschallprüfung werden die Ergebnisse für verschiedene Bauteilgeometrien vorgestellt und im Detail erläutert.

1. Einführung

Für die Inspektion von komplex geformten Freiformflächen unterschiedlicher Produkte oder Halbzeuge in der Industrie, eignet sich in vielen Fällen der Einsatz eines Industrieroboters. Dieser übernimmt die Positionierung eines Sensors relativ zum Bauteil während gleichzeitig Sensordaten aufgenommen werden. Ausgehend von einer möglichst genauen 3-dimensionalen Modellierung von Bauteil, Roboter und Sensor kann die notwendige Bewegungsplanung für den Roboter virtuell erfolgen. Abhängig vom Bauteil sind hunderte oder tausende Einzelpositionen für die gesamte Inspektion eines Bauteils notwendig. Damit ist die Planung eines passenden Pfades für den Roboter keine triviale Aufgabe.

In vielen Fällen kommt für die Planung eines Inspektionspfades eine einfache Rasterung der Bauteiloberfläche zum Einsatz [1]. Dabei wird die Oberflächengeometrie durch mehrere parallele Ebenen geschnitten und die Schnittlinien werden zu einem meanderförmigem Pfad zusammengesetzt.

Einfache Rasterung der Bauteiloberfläche stößt dann an ihre Grenzen, wenn stärkere Krümmungen vorhanden sind bzw. die Bauteilgeometrie insgesamt komplexer wird. In diesen Fällen ergeben sich unregelmäßige Abstände auf der Bauteiloberfläche zwischen den einzelnen Bahnen. Darüber hinaus wird es bei steigender Komplexität der Prüfgeometrien immer schwieriger, den Sensor entlang des Pfades zu positionieren ohne dabei einen



Konfigurationswechsel des Roboters notwendig zu machen. Die Mitberücksichtigung von Konfigurationswechseln in der Pfadplanung erhöht die Komplexität für die Pfadplanung stark. Bei komplexeren Geometrien wird die Frage nach dem optimalen Inspektionspfad zunehmend komplex. Es ergeben sich kombinatorische Problemstellungen (Roboterkonfigurationen, Reihenfolge von Einzelpositionen, etc.), die sich angesichts vieler Inspektionspositionen nicht analytisch lösen lassen. Praktikable Lösungen sind daher auf Heuristiken oder Vereinfachungen des Problems angewiesen.

Mehrere vorgeschlagene Lösungen für die Pfadplanung zur Umsetzung von Oberflächeninspektion basieren auf dem Zusammenführen von einzelnen Positionen zu einem Gesamtpfad [2][3]. Bei dieser Formulierung des Problems kommen oft Algorithmen zur Lösung des „Travelling Salesman Problems“ (TSP) zum Einsatz.

In dieser Arbeit schlagen wir einen alternativen Ansatz basierend auf der Berechnung von Strömungslinien vor. Strömungslinien wurden in der Vergangenheit bereits mehrfach für Roboterpfadplanung eingesetzt. Allerdings betrifft das in erster Linie Problemstellungen für Bewegungen von mobilen Robotern [4][5]. Entsprechende Ansätze basieren auf der Erstellung eines Vektorfeldes in der Ebene.

In Kapitel 2 erläutern wir das Konzept von Strömungslinien allgemein. In Kapitel 3 beschäftigen wir uns mit der flächendeckenden Berechnung von Strömungslinien und mit der Zusammenführung von Einzelpfaden zu einem Inspektionspfad für das gesamte Bauteil. Abschließend stellen wir in Kapitel 4 einige ausgewählte Ergebnisse vor.

2. Strömungslinien

Allgemein wird eine Strömung durch ein Vektorfeld definiert. An jedem Punkt legt das Vektorfeld Richtung und Größe der Strömung fest. Ausgehend von einer Startposition beschreibt eine Strömungslinie die Bewegung eines Partikels durch die Strömung. Dabei entwickelt sich die Position des Partikels über die Zeit so, dass die momentane Bewegung dem Vektorfeld der Strömung entspricht.

Ein Inspektionspfad kann allgemein als Bewegung relativ zu der Oberfläche eines inspizierten Bauteils aufgefasst werden. Die Darstellung der Bewegung kann damit von einer Kurve auf der Bauteiloberfläche abgeleitet werden. Eine Strömungslinie ist definiert als

$$\frac{d\vec{s}_t}{dt} \times \vec{v}(\vec{s}_t) = 0$$

wobei \vec{v} das zugrunde liegende Vektorfeld bezeichnet. Die Parameterdarstellung der Streamline $\vec{s}_t = \vec{s}_t(t)$ als Kurve ist abhängig vom Parameter t . Das Vektorkreuzprodukt wird durch \times bezeichnet. Abgesehen von degenerativen Fällen (wenn entweder das Vektorfeld oder die Ableitung der Strömungslinie gleich dem Nullvektor sind) bedeutet das, dass Vektorfeld und Ableitung parallel oder anti-parallel verlaufen. Beide Vektoren unterscheiden sich also lediglich um die Größe ihres Betrages:

$$\frac{ds_x}{dv_x} = \frac{ds_y}{dv_y} = \frac{ds_z}{dv_z}$$

wobei $\vec{s}_t = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix}$ und $\vec{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$. Abb. 1 zeigt ein Beispiel für ein Vektorfeld in 2D. Der schwarze Punkt markiert den Ursprung ($t = 0$) der Strömungslinie.

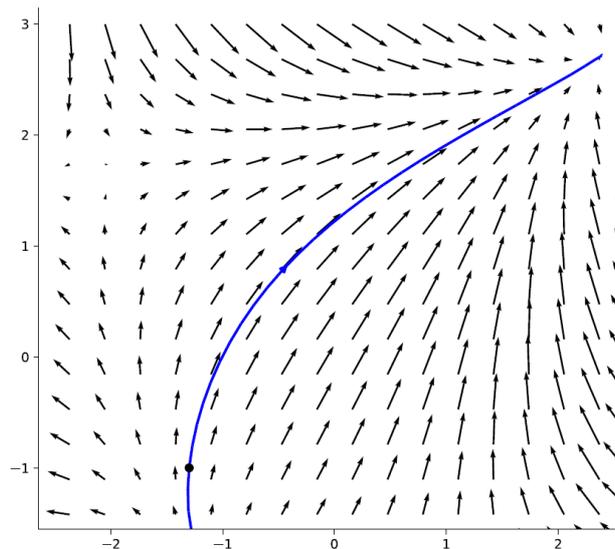


Abb. 1: Beispiel für eine Strömungslinie in einem 2-dimensionalen Vektorfeld.

Für die Pfadplanung mittels Strömungslinien wird also zunächst ein Vektorfeld benötigt. Als Basis dafür schlagen wir die minimale lokale Krümmung auf der Oberfläche des inspizierten Bauteils vor. Für 3D Oberflächen existieren in jedem Punkt unendlich viele Tangentiallinien. Die Krümmung der Oberfläche kann jeweils als Abweichung der Oberfläche entlang der Tangentiallinie aufgefasst werden. Allgemein gibt es eine Tangentiallinie mit minimaler lokaler Krümmung k_{\min} und eine mit maximaler lokaler Krümmung k_{\max} . Diese beiden Krümmungen nennt man auch Hauptkrümmungen. Die Tangentiallinien der Hauptkrümmungen stehen senkrecht aufeinander. Abb. 2 veranschaulicht die beiden Hauptkrümmungen auf einer konvexen Oberfläche in R^3 . Entsprechend der Hauptkrümmungen und dem Normalvektor lässt sich ein lokales Koordinatensystem definieren mit:

- Tangentiallinie mit maximaler Krümmung (grüner Pfeil) als X-Achse,
- Tangentiallinie mit minimaler Krümmung (roter Pfeil) als Y-Achse, und
- Oberflächennormale (blauer Pfeil) als Z-Achse.

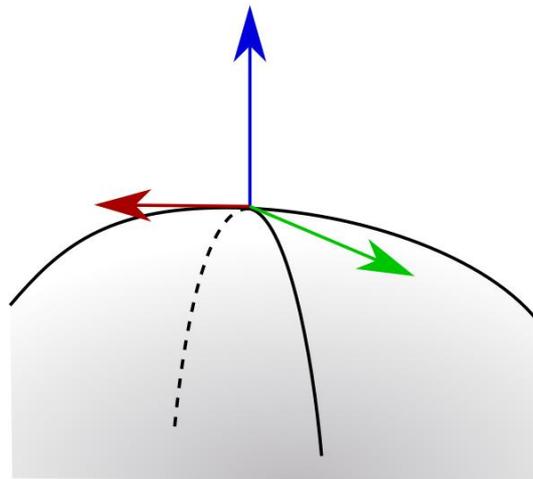


Abb. 2: lokale Krümmung in 2 Hauptrichtungen (roter und grüner Pfeil). Die Ultraschall-Sensorik wird so positioniert, dass die Elemente des Phased Array in Richtung der Stärksten Krümmung (grüner Pfeil) ausgerichtet sind. Die Bewegungsrichtung des Sensors richtet sich an der minimalen Krümmung (roter Pfeil) aus.

Für diskretisierte Oberflächen (z.B. Dreiecksnetze) müssen die relevanten geometrischen Eigenschaften lokal bestimmt werden. Um diese Berechnung vorab durchzuführen, nähern

wir zunächst die Oberfläche durch ein Gitter an Punkten an. Für jeden Punkt werden dann Krümmung und Normalvektor berechnet. Zur Bestimmung der Hauptkrümmungsrichtungen wird eine Singulärwertzerlegung auf Basis der Normalen in einer bestimmten Kugelumgebung durchgeführt. Bei der Berechnung der Strömungslinien selbst kann dann auf diese vorberechneten Werte effizient zugegriffen werden.

Prinzipiell erfolgt die Berechnung einer kompletten Streamline nach dem Prinzip einer numerischen Integration. Ausgehend von einem Startpunkt wird die Strömungslinie parallel und anti-parallel zur lokalen Tangentiallinie mit maximaler Krümmung integriert. Die Berechnung von Startpunkten, Abbruchkriterien und spezielle Randbedingungen für die Nutzung zur Pfadplanung werden im nächsten Kapitel behandelt.

3. Pfadplanung

Für die eigentliche Berechnung eines Roboterpfades werden zunächst vollflächig Strömungslinien auf der Oberfläche des inspizierten Bauteils berechnet. Danach werden diese Strömungslinien unter Berücksichtigung von Roboterkinematik und Vermeidung von Kollisionen in einen Roboterpfad umgewandelt. Die einzelnen Teile des Algorithmus werden in den nachfolgenden Abschnitten im Detail erläutert.

3.1. Prüfbereiche definieren

Im ersten Schritt werden die zu prüfenden Bereiche des Bauteils definiert. Dies geschieht durch Selektion eines zufälligen Startpunktes auf der Oberfläche. Ausgehend von diesem Startpunkt werden zusammenhängende Bereiche selektiert. Übersteigt der Winkel zwischen zwei Flächen einen vorab definierten Schwellwert, so bricht die Selektion an dieser Stelle ab. Auf diese Weise wird das Prüfobjekt also zunächst in Teilbereiche zerlegt. Im Folgenden werden diese Teilbereiche separat betrachtet.

3.2 Diskrete Repräsentation der Abdeckung

Es wird eine Punktwolke generiert, die die Abdeckung der betrachteten Oberfläche repräsentiert. Dafür wird die Geometrie durch ein 3-dimensionales Raster unterteilt. Schnittpunkte zwischen Geometrie und Raster werden in die Punktwolke der diskretisierten Repräsentation eingefügt. Jeder Punkt repräsentiert lokal die Abdeckung der Oberfläche.

3.3 Startpunkt für erste Strömungslinie

Für die erste Strömungslinie wird zunächst ein erster Startpunkt benötigt. Zur Festlegung dieses Punktes kommen zwei Möglichkeiten in Frage: Der Startpunkt wird entweder (1) manuell durch den Benutzer definiert (durch Klick auf einen Punkt auf der Oberfläche) oder (2) automatisch als jener Punkt mit der maximalen Krümmung auf der Oberfläche. Falls mehrere Punkte mit maximaler Krümmung existieren, wird zufällig einer davon ausgewählt.

3.4 Expansion der Strömungslinien

Ausgehend vom Startpunkt werden Strömungslinien in zwei Richtungen expandiert – parallel und anti-parallel zur Richtung der kleinsten Krümmung am Startpunkt. Durch

numerische Integration werden schrittweise Punkte entlang der Strömungslinie hinzugefügt. Dafür wird entweder eine fixe Schrittweite verwendet, oder adaptiv die Schrittweite an die lokale Krümmung angepasst. Für jeden neuen Punkt entlang der Strömungslinie werden alle Punkte innerhalb eines bestimmten Radius, die Teil der diskretisierten Oberfläche sind als „gesehen“ markiert. Der entsprechende Radius wird adaptiv an die lokale Krümmung angepasst. Die Expansion der Strömungslinie schreitet so lange fort, bis eine der folgenden Abbruchbedingungen zutrifft:

- 1.) Die Strömungslinie überschreitet den Rand der betrachteten Oberfläche.
- 2.) Alle Punkte der diskretisierten Oberfläche in einer bestimmten Kugelumgebung wurden bereits als „gesehen“ markiert.
- 3.) Die Strömungslinie unterschreitet einen Mindestabstand zu einer benachbarten Strömungslinie oder zu sich selbst. Um diese Überprüfung effizient durchführen zu können werden alle Strömungslinien in einen KD-Baum eingetragen.

3.5 Neue Startpunkte auf Basis abgeschlossener Strömungslinien

Sobald eine Strömungslinie abgeschlossen ist, werden auf Basis ihrer Stützpunkte neue potentielle Startpunkte für weitere Strömungslinien berechnet. Dazu wird für jeden Stützpunkt in einem bestimmten Abstand normal zur Strömungslinie auf beiden Seiten ein neuer potentieller Startpunkt bestimmt.

Sobald eine Strömungslinie abgeschlossen ist und neue Startpunkte bestimmt wurden, wird wieder mit der Expansion der nächsten Strömungslinie fortgesetzt (wie in Abschnitt 3.4 erläutert). Abwechselndes Hinzufügen von Strömungslinien und potentiellen Startpunkten wird so lange fortgesetzt, bis das Hinzufügen von neuen Strömungslinien nicht mehr möglich ist.

3.6 Einbindung von Kinematik und Kollisionsvermeidung

Im letzten Schritt werden aus bestehenden Strömungslinien entsprechende Roboterpfade generiert. Zunächst werden für einen Endpunkt alle Lösungen der inversen Kinematik berechnet. Für jede dieser Lösungen werden entlang der Strömungslinie – sofern möglich – Lösungen mit derselben Roboterkonfiguration bestimmt. Gibt es keine durchgehende Lösung ohne Konfigurationswechsel so wird die Streamline an der entsprechenden Stelle geteilt. Falls überhaupt keine Lösung der inversen Kinematik existiert, müssen die entsprechenden Inspektionspositionen komplett entfernt werden. Das gleiche gilt für Abschnitte auf Strömungslinien, die auf Grund von Kollisionen ausgeschlossen sind.

Zuletzt werden Pfade für einzelne Strömungslinien zu einem Gesamtpfad kombiniert. Dafür wird ein Strömungslinienpfad nach dem anderen auf Basis von zusammenpassenden Roboterkonfigurationen hinzugefügt. Die Bewegung des Roboters von einer Strömungslinie zur nächsten wird dabei mittels „Rapidly-exploring Random Tree“ (RRT) Punkt-zu-Punkt Pfadplanung gelöst.

4. Ergebnisse

Im Folgenden sind die Ergebnisse für die Berechnung von Strömungslinien dargestellt. Abb. 3 stellt die Strömungslinien auf der Oberfläche von vier ausgewählten Geometrien dar. Die

ersten drei Geometrien stammen aus einer frei verfügbaren Sammlung von Daten¹, die ursprünglich für roboterbasierte Greifprozesse verwendet wurden [6]. Die letzte Geometrie (rechts unten) stellt einen kohlefaserverstärkten Bauteil (bzw. einen Ausschnitt davon) aus der Luftfahrt dar.

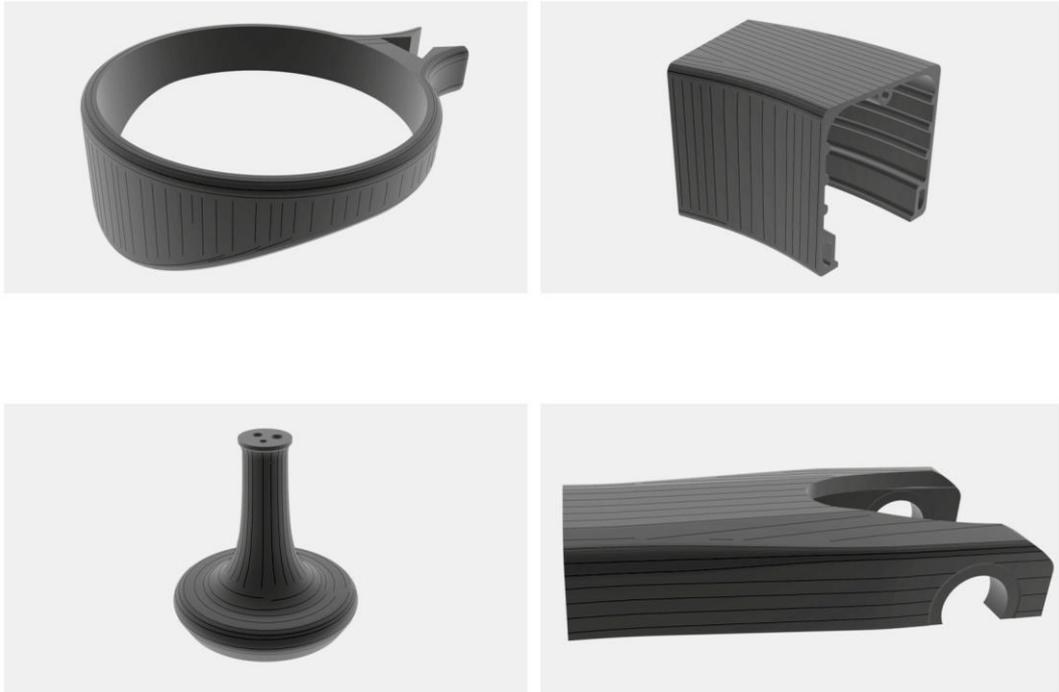


Abb. 3: Ergebnisse für die Berechnung von Strömungslinien basierend auf lokaler Krümmung für 4 unterschiedliche Bauteilgeometrien.

Abb. 4 zeigt einen Konkreten Roboterpfad der auf Basis der entsprechenden Strömungslinien des Bauteils links oben in Abb. 3 berechnet wurde.



Abb. 4: Roboterpfad zur Inspektion eines Bauteils mit komplexer Geometrie. Der Pfad ist vollständig von entsprechenden Strömungslinien entlang der Bauteiloberfläche abgeleitet. Zusätzlich sind hier Roboterkinematik und Kollisionsvermeidung mitberücksichtigt.

¹ Thingiverse pruned meshes: https://berkeleyautomation.github.io/dex-net/#dexnet_4 – Raw Data, Experiments and Analysis Code – raw_data – synthetic_data – object_models – thingiverse-pruned-meshes.zip

5. Zusammenfassung und Ausblick

Wir stellen eine Methode für die Berechnung von Roboterpfaden für Inspektionsaufgaben – speziell Ultraschallprüfung – vor. Wir gehen dabei grundlegend von der Annahme aus, dass der Bewegungsablauf des Roboters bzw. die Positionierung des Sensors durch lokale geometrische Eigenschaften des inspizierten Bauteils vorgegeben wird. Zunächst wird über unsere Methode eine Strömungslinie entlang der Bauteiloberfläche generiert. Diese verläuft entlang der betragsmäßig kleineren der beiden Hauptkrümmungsrichtungen. Unser Ansatz vermeidet das Lösen umfangreicher kombinatorischer Probleme („Travelling Salesman Problem“) durch die Annahme, dass die Positionierung entlang von Strömungslinien effiziente Prüfpfade ermöglicht.

Acknowledgements

This work has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 831830. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union.

Referenzen

- [1] M. Morozov, S.G. Pierce, C.N. MacLeod, C. Mineo, R. Summan, Off-line scan path planning for robotic NDT, *Measurement*, Volume 122, 2018, Pages 284-290, ISSN 0263-2241, <https://doi.org/10.1016/j.measurement.2018.02.020.t>
- [2] Yamin Li, Long Zeng, Kai Tang, Chun Xie, Orientation-point relation based inspection path planning method for 5-axis OMI system, *Robotics and Computer-Integrated Manufacturing*, Volume 61, February 2020
- [3] Bo Li, Pingfa Feng, Long Zeng, Chao Xu, Jianfu Zhang, Path planning method for on-machine inspection of aerospace structures based on adjacent feature graph, *Robotics and Computer-Integrated Manufacturing*, Volume 54, 17-34, 2018, 10.1016/j.rcim.2018.05.006.
- [4] Robert Daily and David M. Bevly, Harmonic Potential Field Path Planning for High Speed Vehicles, *American Control Conference*, 2008
- [5] Han-Jung Chou, Pei-Li Kuoand, Jing-Sin Liu, Numerical streamlinepath planning based on log-space harmonic potential function: a simulation study, *IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 535-542. 2017
- [6] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, Ken Goldberg, Learning ambidextrous robot grasping policies, *Science Robotics*, Volume 4, no. 26, 2019