

# Autonome Navigation vierbeiniger Roboter für Bauwerksinspektionen

Aditya Tandon<sup>ORCID</sup> und Jan Stührenberg<sup>ORCID</sup>

Institut für Digitales und Autonomes Bauen, Technische Universität Hamburg, Blohmstraße 15,  
21079 Hamburg

E-Mails: aditya.tandon@tuhh.de, jan.stuehrenberg@tuhh.de

**Abstract:** Robotersysteme werden für automatisierte Inspektionen von Infrastrukturbauwerken eingesetzt, um die Inspektionsgenauigkeit zu erhöhen und die betriebliche Effizienz zu verbessern. Vierbeinige Roboter verfügen über ausgefeilte Fortbewegungstechniken, die eine große Stabilität, Manövrierfähigkeit und Anpassungsfähigkeit in komplexen bzw. schwierig zugänglichen Gebäuden ermöglichen. Für den zuverlässigen autonomen Betrieb von vierbeinigen Robotern sind robuste Navigations- und Regelungssysteme erforderlich, die jedoch mit zunehmender Komplexität der Bauwerke nicht optimal funktionieren oder die von vierbeinigen Robotern gebotene Flexibilität nicht ausnutzen. Um den autonomen Betrieb von vierbeinigen Robotern zu erleichtern, wird in diesem Beitrag ein Framework zur Bewegungsplanung vorgestellt, der aus einem sampling-basierten globalen Planungsansatz, den Batch Informed Trees\*, und dem lokalen Planungs- und Regelungsansatz Timed Elastic Band besteht. Das vorgeschlagene Framework beinhaltet zusätzlich den Cartographer-Algorithmus zur Lokalisierung, der einen erfolgreichen autonomen Betrieb und die Navigation in Innenräumen ermöglicht. Das Framework wird auf vierbeinigen Robotern unter Verwendung einer verteilten Softwarearchitektur eingesetzt, die die Planung und Ausführung von Inspektionsaufgaben mit Hilfe einer Basisstation und autonomen Navigationsfunktionen in Echtzeit auf den mobilen Robotern ermöglicht. Zur Validierung des Frameworks werden Inspektionsmissionen in einer Büroumgebung mit Hindernissen geplant und ausgeführt. Die Ergebnisse zeigen, dass das Framework robuste Echtzeit-Hindernisvermeidungsfunktionen bereitstellt und eine zuverlässige Pfadplanung und Lokalisierung mit geringer Latenzzeit ermöglicht. *Keywords:* Vierbeinige Roboter, autonome Navigation, Simultaneous Localization and Mapping (SLAM), Bewegungsplanung, Inspektionsplanung



Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13500  
© 2024 Das Copyright für diesen Beitrag liegt bei den Autoren. Verwendung erlaubt unter Creative Commons Lizenz Namensnennung 4.0 International.

## 1 Einleitung

Der Verfall von Infrastrukturbauwerken stellt ein erhebliches Risiko für die öffentliche Sicherheit dar. Infrastrukturbauwerke erfordern ein kontinuierliches Monitoring und regelmäßige Inspektionen, um

einen sicheren Betrieb zu gewährleisten [1]. Bestehende manuelle Monitoring- und Inspektionsverfahren können jedoch zeitaufwändig, arbeitsintensiv, gefährlich, und teuer sein. Die Einführung robotergestützter, automatisierter Monitoring- und Inspektionsverfahren hat das Potenzial, die Produktivität zu steigern, die Sicherheit zu erhöhen und die Kosten zu senken [2].

Mobile, mit Sensoren ausgestattete Roboter wurden bereits erfolgreich für Inspektionsaufgaben eingesetzt [3], wobei bisher spezialisiertes Personal wie Gebäudeinspektoren und Roboterbediener erforderlich waren. Jüngste Fortschritte in der Robotik haben den Einsatz mobiler Roboter zur automatischen Durchführung visueller Inspektionen vorangetrieben [4]. Insbesondere werden vermehrt vierbeinige Roboter eingesetzt, die Hindernisse überwinden, Treppen steigen und schwere Nutzlasten tragen können. Um robotergestützte automatische Inspektionen zu ermöglichen, müssen vierbeinige Roboter in der Lage sein, zuverlässig autonom zu arbeiten. Dafür werden in der Regel Bewegungsplanungssysteme eingesetzt, die die Umgebung in Echtzeit wahrnehmen, den Roboter lokalisieren, Pfade planen, Hindernisse vermeiden und die Roboter so auf kollisionsfreien Trajektorien zu Zielpositionen bewegen. Bestehende Bewegungsplanungssysteme für vierbeinige Roboter nutzen häufig Ganzkörperregelungen für die Planung des Fußkontakts und Erreichbarkeitsüberprüfungen [5], um eine umfassende Bewegungsplanung im Gelenkraum zu ermöglichen. Das Bewegungsproblem in Ganzkörperregelungen kann jedoch hochdimensional und nichtlinear sein, was die Lösung des Problems in Echtzeit komplex und rechenaufwändig macht. Die Komplexität rechenintensiver Ganzkörperregelungen ist in herkömmlichen Innenräumen kaum gerechtfertigt, da die Berechnung präziser Fußkontakte weniger wichtig ist. Stattdessen kann eine Gangstrategie, die das Gleichgewicht des Roboters aufrechterhält, ausreichend sein. In diesem Beitrag wird ein Framework zur Bewegungsplanung vierbeiniger Roboter für automatisierte Inspektionen präsentiert. Das Framework entkoppelt das Bewegungsplanungsproblem in (i) ein High-Level-Aufgabenraumproblem und (ii) ein Low-Level-Gelenkraumproblem, um einen effektiven Echtzeitbetrieb von vierbeinigen Robotern zu gewährleisten. Innerhalb der Frameworks wird ein Missionsplanungsmodul verwendet, um die Planung und Ausführung von Inspektionsaufgaben zu ermöglichen.

Der Beitrags ist wie folgt gegliedert. Abschnitt 2 beschreibt den Entwurf und die Implementierung des Frameworks für automatische Inspektionen. In Abschnitt 3 werden die zur Validierung des Frameworks durchgeführten Tests sowie deren Ergebnisse vorgestellt. Abschnitt 4 schließt mit einer Diskussion der Ergebnisse, einer Zusammenfassung und einem Ausblick auf mögliche zukünftige Arbeiten.

## **2 Ein Framework zur Bewegungsplanung für automatische Bauwerksinspektionen**

Das Framework zur Bewegungsplanung für automatische Bauwerksinspektionen ist in Abbildung 1 dargestellt. Da Roboter in einer dynamischen Welt operieren, muss eine zuverlässige Echtzeitleistung der Module des Bewegungsplanungsframeworks gewährleistet sein, um Kollisionen

zu vermeiden. Daher basiert das Framework auf einer verteilten Robot Operating System (ROS)-Architektur, die es ermöglicht, Module auf Roboter und Computer zu verteilen. Die Auslagerung von Modulen auf Computer reduziert die Rechenlast der Roboter und erleichtert den Echtzeitbetrieb. Das Missionsplanungsmodul, das für den Echtzeitbetrieb nicht entscheidend ist, wird daher auf einem Computer implementiert. Zusätzlich wird das Framework in ein High-Level-Aufgabenraumproblem und ein Low-Level-Gelenkraumproblem entkoppelt. Die Entkopplung der Probleme führt zu zwei Problemen mit reduzierter Komplexität, die unabhängig voneinander gelöst werden können. Das High-Level-Task-Space-Problem befasst sich mit der Planung kollisionsfreier Pfade für den Roboter und der Erzeugung von Zielgeschwindigkeiten im Task-Space. Die Low-Level-Bewegungsausführung verwendet eine Gangstrategie, um die Zielgeschwindigkeiten unter Beibehaltung des Gleichgewichts und der Stabilität des Roboters zu erreichen, wobei eine Hochfrequenzkommunikation verwendet wird. Da die High-Level-Bewegungsplanung unter Verwendung vollständiger 3D-Darstellungen in Echtzeit auf eingebetteten Plattformen nach wie vor sehr rechenaufwändig ist [5], wird eine hybride Strategie angewandt, die die 3D-Umgebung in 2D approximiert. Die Bewegungsplanung auf oberer Ebene im Aufgabenraum wird daher durch die Darstellung des vierbeinigen Roboters als holonomer Roboter im SE(2)-Zustandsraum mit einem rechteckigen Roboterfußabdruck entsprechend der Größe des Roboters erreicht. Durch die Entkopplung kann der Low-Level-Controller mit hohen Frequenzen arbeiten und so eine robuste und stabile Fortbewegung in unebenem Gelände und bei Vorhandensein von externen Störungen gewährleisten. Als Basis der Implementierung wird der ROS-Navigationsstack genutzt. In den folgenden Abschnitten werden die Kernmodule des Frameworks vorgestellt.

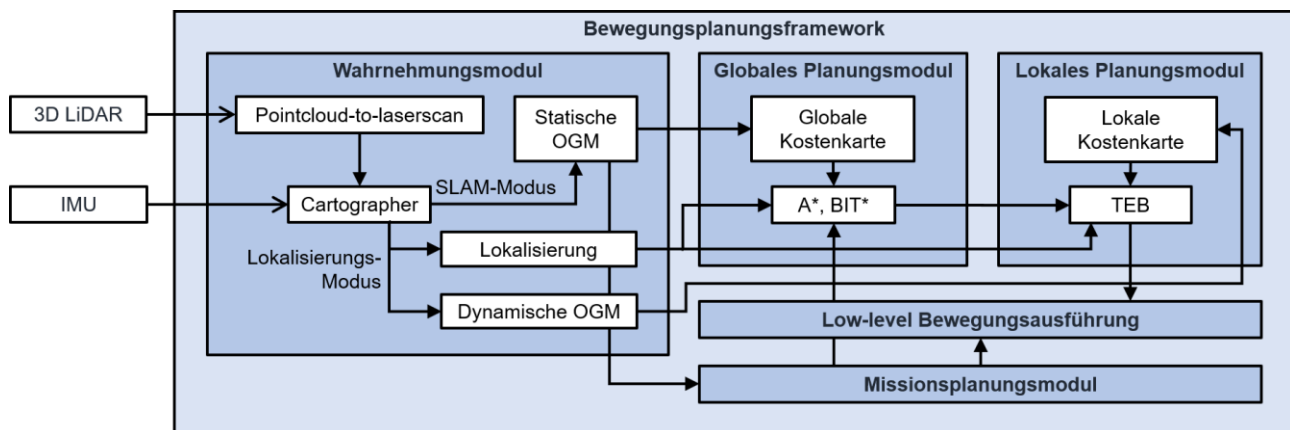


Abbildung 1: Framework zur Bewegungsplanung für automatische Bauwerksinspektionen.

## 2.1 Wahrnehmungsmodul

Die Bewegungsplanung erfordert die Wahrnehmung der Umgebung, um Karten der Umgebung zu erstellen, sich innerhalb der Karten zu lokalisieren und dynamische Hindernisse zu erkennen. In diesem Beitrag wird die Wahrnehmung durch einen LiDAR (Light Detection and Ranging)-Sensor ermöglicht, ergänzt durch eine Inertialmesseinheit (IMU). Da die Bewegungsplanung und Navigation im SE(2)-Zustandsraum durchgeführt wird, werden die 3D-Punktwolken durch Abbildung auf

die horizontale Ebene in 2D-Laserscandaten umgewandelt. Das simultane Lokalisierungs- und Kartierungssystem (SLAM) „Cartographer“ [6] nutzt die 2D-Laserscan- und IMU-Daten, um eine Karte der Umgebung zu erstellen und sich innerhalb der Karte zu lokalisieren. Cartographer operiert dabei in zwei Modi: Im SLAM-Modus zur Erzeugung einer statischen Occupancy Grid Map (OGM) und im Lokalisierungsmodus zur Positionierung des Roboters auf der statischen OGM.

## 2.2 Globales Planungsmodul

Das globale Planungsmodul ermöglicht Robotern, durchführbare, sichere und optimale Pfade zu Zielposen zu planen, indem es mögliche Kollisionen mit Hindernissen in der statischen Karte überprüft und Entfernung, Zeit und/oder Energie minimiert. Für die Suche werden graphenbasierte oder sampling-basierte Bahnplanungsalgorithmen verwendet. Das globale Planungsmodul verwendet die NavfnROS-Implementierung des graphbasierten Planungsalgorithmus A\*. A\* arbeitet auf einer diskretisierten und statischen Belegungsraasterkarte und garantiert mit einer zulässigen Heuristik eine optimale Lösung. Sampling-basierte Pfadplanungsalgorithmen nehmen zufällige Stichproben im Konfigurationsraum und konstruieren einen Graphen, indem sie Punkte mit Kanten verbinden, wenn sie kollisionsfrei sind. ROS-kompatible C++-Implementierungen von sampling-basierten Planungsalgorithmen werden aus dem globalen Planungsmodul OmpfPlanner [7] verwendet. Die Implementierungen der Planungsalgorithmen Rapid-Exploring Random Trees (RRT\*) und Informed-RRT\* werden unverändert verwendet, während der OmpfPlanner um den Planungsalgorithmus Batch-Informed Trees (BIT\*) erweitert wurde.

## 2.3 Lokales Planungsmodul

Das lokale Planungsmodul nutzt die vom globalen Planungsmodul geplanten Pfade. Um Hindernissen Rechnung zu tragen, die während der Kartierung möglicherweise nicht erfasst wurden, passt das lokale Planungsmodul Trajektorien für ein kurzes Zeitfenster in der unmittelbaren lokalen Umgebung des Roboters mit hoher Frequenz an, um eine kollisionsfreie Navigation zu ermöglichen. Das lokale Planungsmodul nutzt Sensordaten vom Wahrnehmungsmodul um eine lokale Karte inklusiver dynamischer und statischer Hindernisse zu erstellen, die sich um den Roboter herum befinden. Um minimale Abstände zwischen Robotern und Hindernissen zu gewährleisten, werden Kostenkarten eingesetzt. Kostenkarten ergänzen die OGMs, indem sie den benachbarten belegten Zellen Reisekosten zuweisen. Das lokale Planungsmodul verwendet den TEB (Timed Elastic Bands)-Ansatz für die lokale Planung [8]. Der TEB-Planungsansatz modifiziert bestehende globale Pläne, indem er die virtuellen Anziehungskräfte der globalen Pläne und die Abstoßungskräfte von Hindernissen ausgleicht. Der TEB-Planungsansatz generiert zeitoptimale Trajektorien, wobei kinodynamische Beschränkungen eingehalten und minimale Abstände um Hindernisse eingehalten werden. Da vierbeinige Roboter holonomische Roboter sind, wird die Strafe für Querbewegungen im TEB-Planungsansatz abgeschwächt. Dadurch wird die Generierung von Trajektorien mit Geschwindigkeiten und Beschleunigungen in allen Richtungen im SE(2)-Zustandsraum gefördert.

## 2.4 Missionsplanungsmodul

Um automatisierte Bauwerksinspektion zu ermöglichen, wird ein Missionsplanungsmodul eingesetzt. Das Missionsplanungsmodul umfasst eine Benutzerschnittstelle für die sequentielle Planung, Ausführung und Verwaltung von Inspektionsaufgaben und -missionen und verwaltet die roboterbezogene Sicherheit und Funktionalität. Inspektionsmissionen sind als hierarchische endliche Zustandsautomaten modelliert. Inspektionsmissionen bestehen aus Missionsaufgaben, die als einzelne Zustände modelliert sind, und Übergänge zwischen den Zuständen, die auf der Grundlage des Ausführungsstatus der jeweiligen Aufgaben definiert werden. Die Inspektionsmissionen werden mit Hilfe der SMACH-Bibliothek im ROS-Framework implementiert [9] und lassen sich intuitiv mit einer grafischen Benutzeroberfläche im ROS-Visualisierungstool RViz erstellen. Zielposen werden durch das Markieren von Pfeilen auf der statischen OGM zu einer Inspektionsmission hinzugefügt. Die grafische Benutzeroberfläche ermöglicht das Speichern, Laden und Ausführen von Inspektionsmissionen. Zusätzlich kann über die grafische Benutzeroberfläche der Betriebs- und Bewegungsstatus des Roboters eingesehen und kontrolliert werden, um beispielsweise die Ausführung eines Auftrags zu unterbrechen.

## 3 Validierungstests und Ergebnisse

Das in diesem Beitrag präsentierte Framework wird in einer Büroumgebung validiert, um die Funktionalität und Echtzeitleistung zu bewerten. Zunächst wird das Wahrnehmungsmodul des Frameworks validiert, indem eine statische OGM der Umgebung mit dem SLAM-Modus von Cartographer erstellt und die Lokalisierung in der statischen OGM mit dem Lokalisierungsmodus von Cartographer getestet wird. Anschließend werden mit dem Missionsplanungsmodul Inspektionsmissionen in der statischen OGM geplant, um das globale Planungsmodul und das lokale Planungsmodul des Frameworks zu validieren. Das Framework wird auf einem vierbeinigen Roboter vom Typ „Intelligent Documentation Gadget“ (IDOG) implementiert [10]. Die Büros sind mit beweglichen Möbeln ausgestattet, die die räumliche Anordnung der kartierten Umgebung verändern können und somit als statische Hindernisse fungieren. Personen, die sich in den Büros bewegen, stellen dynamische Hindernisse dar.

Zur Validierung des Wahrnehmungsmoduls wird Cartographer verwendet, um eine statische OGM der Umgebung zu erstellen und anschließend den Roboter auf der statischen OGM zu lokalisieren. Der LiDAR wird mit einer Frequenz von 10 Hz und die IMU mit 100 Hz betrieben. Der IDOG wird manuell gesteuert, während der SLAM-Modus von Cartographer die OGM mit einer Auflösung von 0,05 m/Pixel erstellt. Die Karte hat eine Länge von ca. 780 Pixeln und eine Breite von 400 Pixeln, was 39 m × 20 m entspricht. Nach dem Start des Lokalisierungsmodus von Cartographer wird die zunächst unbekannte Position des IDOG in der Karte innerhalb von Sekunden bestimmt.

Zur Validierung des globalen Planungsmoduls werden Inspektionsmissionen in der statischen OGM erstellt, die als Missionen mit mehreren Wegpunkten angelegt sind. Das globale Planungsmodul plant Pfade zwischen den Wegpunkten, die anhand von Pfadlänge und Planungszeit verglichen

werden. Wie in Abbildung 2 dargestellt, werden in der statischen OGM drei Wegpunkte ausgewählt, die verschiedene Herausforderungen simulieren. Die Validierung wird für die Algorithmen BIT\*, Informed-RRT\*, RRT\* und A\* durchgeführt.



Abbildung 2: Inspektionsmission zur Validierung des globalen Planungsmoduls.

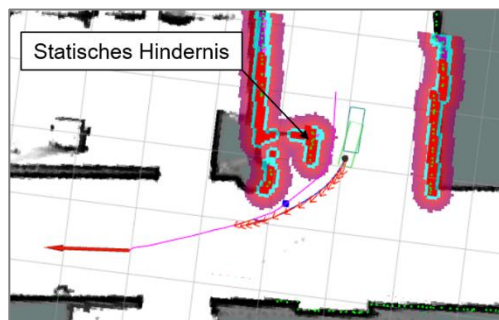
Die Validierungstests zeigen, dass bis auf RRT\* alle globalen Planungsalgorithmen ausschließlich kollisionsfreie Pfade zwischen den Wegpunkten in den Tests erzeugen. Daher werden weitere Pfadplanungstests mit RRT\* aufgegeben. Die Ergebnisse sind in Tabelle 1 dargestellt. Die von A\*, BIT\* und Informed-RRT\* geplanten Pfade haben vergleichbare Längen. Der A\*-Algorithmus weist eine um eine Größenordnung geringere Planungszeit auf als die anderen stichprobenbasierten Planungsalgorithmen mit einem Planungsbudget von einer Sekunde. Die Ergebnisse in dieser Studie deuten darauf hin, dass die globale Bewegungsplanung in 2D und in der vergleichsweise kleinen Validierungsumgebung schneller vom graphenbasierten A\*-Algorithmus gelöst wird. Für künftige Bahnplanungen in 3D und in großen Umgebungen wird BIT\*, der den Stand der Technik darstellt, zusätzlich zum A\*-Algorithmus in das Bewegungsplanungssystem integriert. Der A\*-Algorithmus wird für die Planung in 2D und kleinen Umgebungen verwendet.

Tabelle 1: Weglänge und Planungszeit der globalen Planungsalgorithmen.

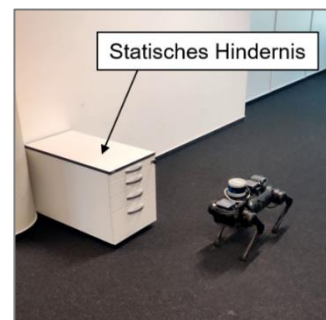
| Wegpunkt | A*        |          | BIT*      |          | Informed-RRT* |          | RRT*      |          |
|----------|-----------|----------|-----------|----------|---------------|----------|-----------|----------|
|          | Länge (m) | Zeit (s) | Länge (m) | Zeit (s) | Länge (m)     | Zeit (s) | Länge (m) | Zeit (s) |
| 1        | 18.878    | 0.082    | 18.795    | 1.037    | 19.732        | 1.035    | 26.747    | 1.034    |
| 2        | 11.544    | 0.288    | 11.621    | 1.051    | 11.535        | 1.085    | -         | -        |
| 3        | 08.815    | 0.085    | 08.816    | 1.032    | 08.811        | 1.039    | -         | -        |

Die Validierung der Funktionalität und Echtzeitleistung des lokalen Planungsmoduls wird durch mehrere Tests durchgeführt, die sowohl die An- und Abwesenheit von Hindernissen als auch die Durchquerung von offenen Räumen und schmalen Passagen beinhaltet. In Abwesenheit von Hindernissen zeigt sich, dass die tatsächliche Trajektorie des Roboters eng mit dem globalen Plan übereinstimmt. Lediglich in einigen Segmenten sind leichte Abweichungen vom globalen Plan zu beobachten. Statische Hindernisse, wie beispielsweise verschobene Möbel, werden durch vom Wahrnehmungsmodul empfangene Sensordaten der lokalen Kostenkarte des Roboters hinzugefügt, wie in Abbildung 3a dargestellt. Das lokale Planungsmodul plant um das Hindernis herum (blaue

Trajektorie), obwohl der globale Plan mit dem statischen Hindernis kollidiert (lila Trajektorie). Wie in Abbildung 3b zu sehen ist, ist der IDOG in der Lage, das Hindernis erfolgreich zu umgehen.



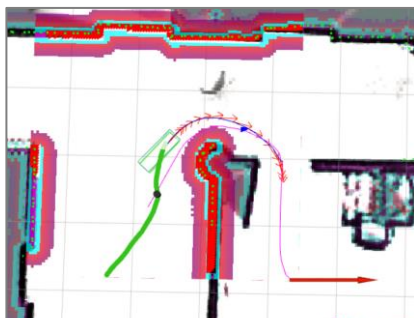
(a) Plan um statisches Hindernis herum



(b) IDOG umgeht das Hindernis

Abbildung 3: Ausführung eines lokalen Plans auf engem Raum und um Ecken.

Um die Navigation durch schmale Passagen zu validieren, werden Wegpunkte in zwei Räume platziert, die die Durchquerung von Türen mit einer Breite von 1,05 m und 0,8 m erfordern. Die Tests zeigen, dass der IDOG in der Lage ist, erfolgreich einen globalen Pfad zu planen, den Pfad lokal abzustimmen und in Echtzeit zu navigieren, während ein ausreichender Abstand zu den Türpfosten eingehalten wird. Während die autonomen Navigationsfähigkeiten um Hindernisse herum und durch enge Passagen zufriedenstellend sind, wird in schmalen Passagen ein oszillierendes Verhalten zwischen dem vom IDOG verfolgten Pfad und dem globalen Plan beobachtet (Abbildung 4), dennoch kann der IDOG den globalen Plan vervollständigen.



(a) Lokaler Plan um die Ecke.



(b) Durchgeführte Trajektorie um die Ecke.

Abbildung 4: Ausführung eines lokalen Plans auf engem Raum und um Ecken.

## 4 Zusammenfassung und Schlussfolgerungen

Der Verfall baulicher Infrastruktur erfordert ein kontinuierliches Monitoring und regelmäßige Inspektionen, um einen sicheren Betrieb zu gewährleisten. Um ein robotergestütztes, automatisiertes Monitoring und Inspektionen mit vierbeinigen Robotern zu erreichen, ist die Verwendung von Bewegungsplanungssystemen unerlässlich. Das in dieser Studie vorgestellte Framework entkoppelt das Bewegungsplanungsproblem in ein High-Level-Aufgabenraumproblem und ein Low-Level-Gelenkraumproblem für den Echtzeitbetrieb von vierbeinigen Robotern. Der sampling-basierte globale Planungsalgorithmus BIT\*, der A\*-Algorithmus und der lokale Planungsansatz TEB werden für die Steuerung von Robotern und die Vermeidung von Hindernissen unter Einsatz einer verteilten

Software-Architektur auf dem vierbeinigen IDOG-Roboter und einem Computer implementiert. Inspektionsmissionen wurden in einer Büroumgebung mit statischen und dynamischen Hindernissen geplant und in Validierungstests vom IDOG erfolgreich ausgeführt. Die Ergebnisse zeigen, dass das Bewegungsplanungsframework in Echtzeit arbeiten kann, statische und dynamische Hindernisse erfolgreich umgehen kann und schmale Türen durchqueren kann, obwohl hier ein unerwünschtes oszillierendes Verhalten der Roboterbewegung zu beobachten ist. Zusammenfassend lässt sich festhalten, dass das Framework eine nützliche Grundlage für die automatisierte Inspektion von baulicher Infrastruktur mit vierbeinigen Robotern darstellt. Zukünftige Arbeiten werden das oszillierende Verhalten des Bewegungsplanungsframeworks beim Durchqueren schmaler Passagen untersuchen und abmildern. Darüber hinaus werden spezifische Inspektionsaufgaben implementiert werden.

## Referenzen

- [1] Smarsly, K., Law, K. H., & König, M., 2011. Autonomous structural condition monitoring based on dynamic code migration and cooperative information processing in wireless sensor networks. In: Proceedings of the 8th International Workshop on Structural Health Monitoring 2011. Stanford, CA, USA, 09/13/2011.
- [2] Smarsly, K. & Dragos, K., 2024. Advancing civil infrastructure assessment through robotic fleets. Internet of Things and Cyber-Physical Systems, 4(2024), S. 138-140.
- [3] Smarsly, K., Dragos, K., Stührenberg, J., & Worm, M., 2023. Mobile structural health monitoring based on legged robots . Infrastructures, 8(9), 136.
- [4] Halder, S. & Afsari, K., 2023. Robots in inspection and monitoring of buildings and infrastructure: A systematic review. Applied Sciences, 13(4), 2304.
- [5] Wellhausen, L. & Hutter, M., 2022. Artplanner: Robust legged robot navigation in the field. Field Robotics, 3(1), S. 413-434, 2023.
- [6] Hess, W., Kohler, D., Rapp, H., & Andor, D., 2016. Real-time loop closure in 2D lidar slam. In: Proceedings of the 2016 Int. Conf. on Robotics and Automation. Stockholm, Schweden, 5/16/2016.
- [7] Sucas, I. A., Moll, M., & Kavraki, L. E., 2012. The open motion planning library. IEEE Robotics & Automation Magazine 19(4), S. 72-82.
- [8] Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F., & Bertram, T., 2012. Trajectory modification considering dynamic constraints of autonomous robots. In: Proceedings of ROBOTIK 2012; 7th German Conference on Robotics. München, Deutschland, 5/21/2012.
- [9] Bohren, J. & Cousins, S., 2010. The smach high-level executive. IEEE Robotics & Automation Magazine, 17(4), S. 18-20.
- [10] Johann, S., Stührenberg, J., Tandon, A., Dragos, K., Bartholmai, M., Strangfeld, C., & Smarsly, K., 2024. Implementation and validation of robot-enabled embedded sensors for structural health monitoring. In: Proceedings of the 11th European Workshop on Structural Health Monitoring. Potsdam, 06/10/2024.