

Transformation eines fahrerlosen Transportfahrzeugs zu einem autonomen mobilen Roboter

Autonome mobile Roboter: Zukunft der Bau-Fertigung

S. Mansour, A. Lechler, A. Verl

ZUSAMMENFASSUNG Fahrerlose Transportfahrzeuge (FTF) sind essenziell für den automatisierten Transport von Gütern in der Industrie. Übliche FTF stoßen jedoch an Grenzen, weshalb autonome mobile Roboter (AMR) als flexiblere Weiterentwicklung genutzt werden. Statt neue AMR zu beschaffen, können bestehende FTF nachgerüstet werden. Ein Forschungsprojekt zur Vorfertigung und Automatisierung im Bauwesen zeigt die Umwandlung eines konventionellen FTF in ein AMR und erörtert dessen erweiterte Funktionalität.

STICHWÖRTER

Automatisierung, Flexible Fertigungssysteme, Logistik

Autonomous mobile robot: Future of manufacturing in construction

ABSTRACT Automated guided vehicles (AGVs) are essential for the automated transportation of goods in industrial applications. However, conventional AGVs are reaching their limits, which is why autonomous mobile robots (AMRs) are being used as a more flexible alternative. Instead of procuring new AMRs, existing AGVs can be retrofitted. A research project on prefabrication and automation in construction demonstrates the conversion of a conventional AGV into an AMR and discusses its extended functionality.

1 Einleitung

1.1 Motivation und Hintergrund

Fahrerlose Transport Systeme (FTS) haben sich zu einem essenziellen Bestandteil der industriellen Infrastruktur entwickelt. Es handelt sich um ein System, das aus mehreren fahrerlosen Transportfahrzeugen (FTF) besteht, die für den Materialtransport, beispielsweise in der Intralogistik, eingesetzt werden [1]. Obwohl FTS bereits in Branchen wie der Automobilindustrie und der Lagerlogistik etabliert sind, werden sie bisher in der Bauindustrie nur in begrenztem Umfang angewendet [2]. Diese Tatsache lässt sich unter anderem auf die generell geringe Automatisierung im Bauwesen zurückführen [3].

Es wird jedoch intensiv daran gearbeitet, die Automatisierung und Digitalisierung in der Bauindustrie voranzutreiben, und es gibt Bereiche, in denen der Einsatz mobiler Roboter erforderlich ist [3]. Ein wesentlicher Aspekt ist die Tatsache, dass die Bauindustrie zu den Hauptverursachern des Treibhauseffekts zählt [4]. Zudem ist mit Zunahme der Urbanisierung die Nachfrage nach erschwinglichem Wohnraum signifikant angestiegen und es wird prognostiziert, dass dieser Anstieg weiter zunimmt [5]. Dies belastet die Bauweisen und Ressourcen zusätzlich. Der Trend in der Bauindustrie zielt somit auf eine nachhaltige Transformation ab, um die Emissionswerte zu reduzieren und die Effizienz zu steigern. An der Universität Stuttgart wurde im Jahr 2019 das Exzellenzcluster „Integrative Computational Design and Construction for Architecture“ (IntCDC) gegründet, das sich dieser Thematik widmet. Hier arbeiten Forscher verschiedener Fachrichtungen zusammen, um integrative Prozesse und Methoden für den Fortschritt der Bauindustrie zu entwickeln [6].

Im IntCDC existieren diverse Forschungsprojekte, von denen sich mehrere mit dem Holzbau beschäftigen. Im Rahmen einer dieser Initiativen wurden cyber-physische Roboterplattformen konzipiert, die für die Vorfertigung im Holzbau vorgesehen sind. Die in **Bild 1** dargestellten Plattformen sind darauf spezialisiert, Holzbauteile vor dem Transport zur Baustelle zu fertigen.

Der damit einhergehende Vorteil liegt in der Minimierung des Bauaufwands an der Baustelle sowie in der erhöhten Flexibilität und Kapazität der Fertigung. Die diskreten, automatisierten Prozesse ermöglichen es, projektbasiert unterschiedliche Bauteile mit minimalen Änderungen an der Anlage zu fertigen. Die Roboterplattformen können in Containern transportiert und bei einem Hersteller abseits oder in unmittelbarer Nähe der Baustelle aufgestellt werden. Zudem ist vorgesehen, dass zahlreiche Plattformen für verschiedene Fertigungsschritte in Kombinationen platziert werden.

1.2 Zielsetzung

Die Ziele des IntCDC, nämlich die Entwicklung integrativer Prozesse und Methoden für den Fortschritt der Bauindustrie, stellen besondere Herausforderungen an den Transport und die Werkstückhandhabung. Die Disposition der Bauteile zwischen den Roboterplattformen sowie zur Baustelle ist eine wesentliche Komponente des Prozesses. Um den Transfer in automatisierter Form zu gewährleisten, empfiehlt sich der Einsatz einer mobilen Transportplattform. Die beschriebenen Ziele der Vorfertigung im IntCDC setzen voraus, dass die mobile Transportplattform in einer variablen Umgebung in einer Halle oder im Freien navigieren kann. Klassische FTF-Methoden stoßen in diesem Kontext an



Bild 1. Cyber-physische Roboter-Vorfertigungsplattformen (links). Zu sehen ist auch das bestehende FTF (fahrerloses Transportfahrzeug) im beladenen (links) und unbeladenen Zustand (rechts). Foto: IntCDC

ihre Grenzen. Dies ist auf die limitierte Flexibilität und Navigationsfähigkeit zurückzuführen, vor allem in dynamischen und unstrukturierten Umgebungen. Diese Faktoren sind unzureichend, um die angestrebte Autonomie und Effizienz zu gewährleisten.

Um diesem Problem zu begegnen, besteht die Möglichkeit, das FTF in einen autonomen mobilen Roboter (AMR) umzuwandeln, indem es mit zusätzlichen Sensoren, einer erweiterten Steuerungsarchitektur und fortgeschrittenen Navigationsalgorithmen ausgestattet wird. Diese Anpassungen ermöglichen eine autonome und adaptive Bewegungsplanung. Dadurch ist das Fahrzeug in der Lage, sich selbstständig an wechselnde Bedingungen anzupassen und komplexe Transportaufgaben effizient zu bewältigen. Dieser Beitrag stellt eine Möglichkeit vor, wie eine solche Transformation von einem FTF zu einem AMR durchgeführt werden kann.

1.3 Forschungsfrage

Wie kann ein bestehendes FTF so transformiert werden, dass es als AMR flexibel an verschiedenen Orten der Vorfertigung im Baubetrieb, einschließlich Outdoorbereichen, eingesetzt werden kann? Welche Anpassungen oder Erweiterungen an Hardware und Software sind dafür erforderlich? Welche Anforderungen ergeben sich an die Definition einer neuen Steuerungsarchitektur?

2 Stand der Technik

FTF und AMR sind Schlüsseltechnologien im Kontext der Industrie 4.0 und zunehmend in modernen Produktions- und Logistikumgebungen anzutreffen [7, 8]. Beide Systeme dienen der automatisierten Bewegung von Gütern innerhalb definierter Räume, unterscheiden sich jedoch grundlegend hinsichtlich ihrer Technologie, Flexibilität und Intelligenz.

2.1 FTF

Im Rahmen der Entwicklung eines FTF kommt es zur Annäherung seiner Fähigkeiten im Vergleich zu einem AMR. In diesem Kapitel wird daher auf klassische FTF Bezug genommen. Im Bereich der Logistik werden solche für den Transport von Material und Gütern eingesetzt [9]. In der Regel sind sie in ein FTS integriert, wobei eine zentrale Steuerung die Planung übernimmt [9]. Charakteristisch für solche Systeme sind Bahnen, denen die Navigation folgt. Diese Bahnen werden durch Magnetstreifen, Linien, visuelle Bodenmarker oder Schienen definiert, welche durch Sensoren erkennbar sind. Der Bahn zu folgen heißt also der Markierung zu folgen.

Es existieren jedoch auch modernere Systeme, wie das verwendete System von Stäubli, welche virtuelle Bahnen aufweisen, die auf der Karte definiert sind. Diese virtuellen Bahnen müssen bei der Inbetriebnahme des Systems implementiert werden und werden anschließend durch die Lokalisierung mittels Sensoren detektiert. Die festgelegten Bahnen bedingen, dass die FTF nicht von diesen abkommen können. Dies kann zu Problemen führen, wenn Hindernisse die Bahn blockieren oder stark verengen. Obwohl diese Flächen oft klar definiert oder sogar abgegrenzt sind, kann es in hochdynamischen Umgebungen dennoch zur Blockade kommen. Bei einem klassischen FTF kann eine Blockierung der Bahn zu einem Stillstand führen, da keine Planung um ein Hindernis herum möglich ist. In einem solchen Fall muss darauf gewartet werden, dass das Hindernis entfernt wird. Eine weitere Option besteht darin, die Bahn zurückzufahren und einen alternativen Weg zu nehmen, sofern zuvor eine alternative Bahn zum Ziel definiert wurde. [9]

2.2 AMR

Das Ziel der AMR besteht darin, in unvorhersehbaren und zum Teil unbekannten Umgebungen einen autonomen Betrieb zu ermöglichen. Zu diesem Zweck können moderne Sensorik und Methoden eingesetzt werden, um die Umgebung dynamisch zu erfassen und zu aktualisieren. Diese Technologie wird als Simultaneous Localization and Mapping (SLAM) bezeichnet. Während einer Fahrt generiert der AMR eine Karte, in der er sich selbst lokalisiert. Das Resultat ist eine automatisch generierte Karte, die während des Betriebs aktualisiert werden kann. So ist es möglich, neue Hindernisse zu erfassen und in die Planung zu integrieren. Ein weiterer Vorteil ist die Unabhängigkeit von festen Bahnen. Stattdessen wird durch einen Bahnplanungsalgorithmus eine Bahn vom Start zum Zielpunkt gefunden. Dadurch ist es dem Planer möglich, auch um neue Hindernisse herum zu planen, falls diese unerwartet auftauchen. Zudem ist die kollaborative Durchführung von SLAM durch mehrere AMR möglich [10]. Damit ist eine effizientere und regelmäßige Abdeckung der gewünschten Fläche möglich. [7]

3 Anforderungen im Bauwesen

Dieses Kapitel stellt die Anforderungen an die Transportplattform vor. Zuerst werden die Allgemeinen Anforderungen an das System für den Einsatz im Bauwesen erörtert. Diese begründen die Beschaffung des bestehenden FTF. Danach folgen die

Anforderungen an die Transformation des bestehenden FTF zu einem AMR.

3.1 Allgemein

Die Transportplattform soll flexibel und effizient sowohl in verschiedenen Bereichen der Vorfertigung (zum Beispiel LCRL Waiblingen, Holzbaubetrieb) als auch im Außenbereich, etwa auf Baustellen, eingesetzt werden können. Die Kompatibilität mit dem Fabrikationsrahmen ist eine grundlegende Anforderung, wie sie in Bild 1 auf dem FTF dargestellt ist. Dieser Rahmen findet nicht nur im Holzbau, sondern auch in anderen IntCDC-Verfahren Anwendung.

Gemäß Anforderungen soll die Plattform eine Nutzlast von mindestens 10 000 kg aufnehmen können, wobei die Bauteile ein Gewicht von mindestens 7500 kg und der Rahmen ein Gewicht von mindestens 2500 kg aufweisen müssen. Die resultierenden Fahrzeugabmessungen zeigen eine Spannweite von mindestens $5000 \times 2000 \times 450$ mm und maximal 6000 mm Länge, 2500 mm Breite und 650 mm Höhe. Die Navigation soll automatisiert erfolgen, ohne dass externe Sensoren oder im Boden eingelassene Leitsysteme erforderlich sind (Konturen-Navigation). Da sich der Arbeitsbereich mit dem der Menschen überschneidet, sind gemäß DIN EN 1525 [11] und DIN EN ISO 3691 [12] Personenschutzscanner vorzusehen. Unter Berücksichtigung der spezifischen Bedingungen in holzverarbeitenden Fertigungen, die auch die Staubbelastung durch zerspanende Tätigkeiten umfassen, ist es von entscheidender Bedeutung, das System möglichst wartungsarm betreiben zu können. Das Ziel besteht darin, omnidirektionale Bewegungen zu ermöglichen, um die Manövrierbarkeit auch in begrenzten Räumlichkeiten zu optimieren. Gemäß dem aktuellen Forschungsstand ist eine adäquate Federung der Räder zur Gewährleistung der Fahrsicherheit bei der Bewältigung geringer Bodenunebenheiten erforderlich.

Nach Analyse der am Markt verfügbaren Plattformen wurde das Stäubli „WFT PF210“-FTF (siehe Bild 1 rechts) als überlegenes System identifiziert. Aus softwarebezogener Perspektive erfolgt die Ausstattung mit der Navigationssoftware „Navitrol“ von Navitec Systems [13]. Dies umfasst die Erstellung von Karten, die Lokalisierung und Navigation. Darüber hinaus existieren Funktionalitäten zur Integration mehrerer FTF in ein FTS. Die von Navitec Systems entwickelte Software findet in der Industrie bei einer Vielzahl von Herstellern Anwendung.

Obwohl der Hersteller der Navigationssoftware in industriellen Umgebungen Robustheit zu kleineren Änderungen in der Umgebung verspricht, ist dies für die geplante Anwendung nicht ausreichend. Dies ist darauf zurückzuführen, dass das FTF sich in einer hochdynamischen Umgebung bewegt. In [13] beschreibt der Hersteller Navitec, dass die Positionierung in hochdynamischen Anlagen schwierig sein kann. In der Vorfertigung im Bau, in der mehrere robotergestützte Fertigungsstationen, unvorhergesehene Hindernisse und die unmittelbare Nähe zur Baustelle eine Rolle spielen, stoßen Navigationssysteme, die für kontrollierte Industrieumgebungen konzipiert wurden, an ihre Grenzen. Infolgedessen müssten die Karten in regelmäßigen Abständen manuell von einem zertifizierten Arbeiter neu aufgezeichnet, konfiguriert, mit Bahnen ergänzt und das FTF neu aufgespurt werden.

Bei der Fahrt im Freien ist die Sensorik den Wetterbedingungen ausgesetzt. Falls eine zuverlässige Fortbewegung bei Regen oder Schnee nicht gewährleistet werden kann, hat dies einen sig-

nifikanten Einfluss auf den Materialfluss. Die Beschränkung auf 2D-Laser-Scanner zur Navigation stellt dabei eine signifikante Einschränkung dar, welche die Zuverlässigkeit der Navigation bei Regen und Schnee erheblich beeinträchtigt [14]. Eine Möglichkeit, die Robustheit der Navigation zu erhöhen, ist die Erweiterung der Sensorik und deren Fusion [7].

Im Rahmen dieses Forschungsprojekts werden mitunter Änderungen am Verhalten und den Funktionen des FTF erwünscht. Dies ermöglicht die Erprobung und Evaluierung neuer Ansätze und Methoden. Außerdem besteht Bedarf an benutzerdefinierten Kommunikationsschnittstellen, die eine Interaktion mit den Roboterplattformen für die Automatisierung erlauben. Es ist jedoch zu bemerken, dass die Software des Herstellers und die externe Kommunikationsschnittstelle nicht für die aktive Entwicklung zur Verfügung steht, da es sich um ein geschütztes Produkt handelt. Für die Forschung ist daher die Entwicklung einer eigenen Software erforderlich.

3.2 Anforderung an die Transformation

Zunächst soll die Funktionalität des Originalsystems (parallel zur Erweiterungslösung) erhalten bleiben. Dies wird durch die Notwendigkeit begründet, die Funktionsfähigkeit in der Entwicklungsphase zu bewahren. Darüber hinaus kann das ursprüngliche Navigationssystem als Vergleichsbasis zum neu entwickelten AMR herangezogen werden. Ferner kann das Navigationssystem als Rückfalllösung dienen, falls während des Produktivbetriebs Fehler in den Erweiterungen auftreten. Die Erhaltung des Original-Navigationssystems ist somit ein zentraler Bestandteil der Software- und Hardware-Architektur.

Die Gewährleistung der Sicherheit von Menschen im Umfeld sowie von Maschinen stellt bei diesem Projekt eine prioritäre Herausforderung dar. Da der Fokus in diesem Forschungsprojekt nicht auf der Entwicklung eines Sicherheitskonzepts und dessen Validierung liegt, wurde eine weitere Anforderung definiert. Gemäß dieser Anforderung ist sicherzustellen, dass die Sicherheitsfunktionalitäten des Herstellers intakt bleiben. Diese Maßnahme vereinfacht nicht nur die Sicherheitsanforderungen an das entwickelte System, sondern reduziert auch den erforderlichen Entwicklungs- und Validierungsaufwand.

4 Systemumrüstung: Vom FTF zum AMR

Die Transformation eines FTF zu einem AMR vollzieht sich auf multiplen Ebenen. Im nächsten Abschnitt werden die Modifikationen, welche in der Hardware und Software vorgenommen werden, erörtert.

4.1 Steuerungsarchitektur

Die Steuerungsarchitektur des FTF (Bild 2) kann wie folgt beschrieben werden: Die Steuerung der Antriebe wird durch den Sicherheitslenkcomputer (SLC) übernommen, während auf der Speicherprogrammierbaren Steuerung (SPS) das Sicherheitsprogramm ausgeführt wird und auf dem Navigationsrechner die Navigationssoftware Navitrol läuft. Die zwei diagonalen Laserscanner werden für die Navigation und alle vier werden für die Sicherheitsüberwachung benutzt.

Unter Berücksichtigung der in Kapitel 3.2 genannten Anforderungen wurde die neue Steuerungsarchitektur definiert. Dazu

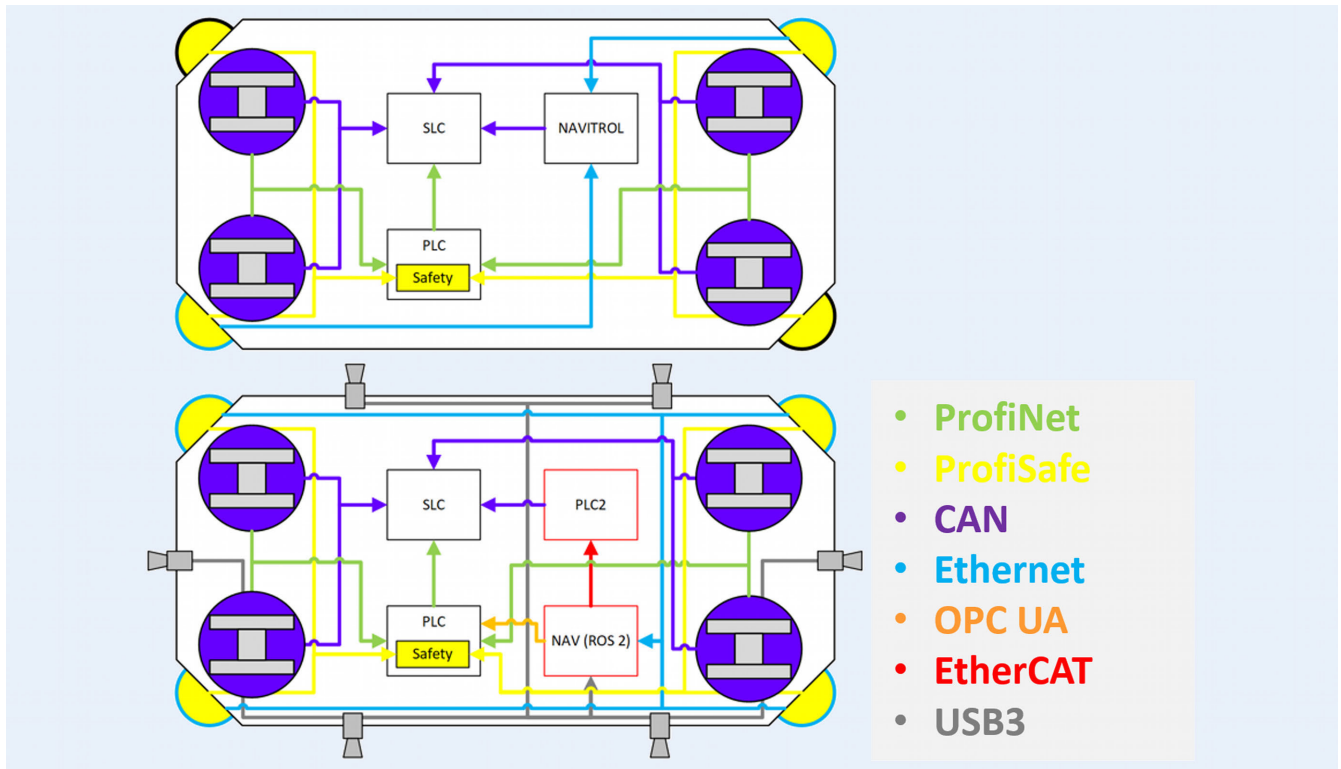


Bild 2. Steuerungsarchitektur des Staubli „WFT PF210“ vor (oben) und nach (unten) der Transformation zu einem autonomen mobilen Roboter (AMR).
Grafik: ISW

wurde der Navigationsrechner durch einen neuen ersetzt, der mit dem Robot Operating System 2 (ROS2) [15] aufgesetzt ist. ROS ist in der Forschung sehr verbreitet, da es ein strukturiertes und basisfunktionsreiches Rahmenwerk bietet [16].

Der ROS2-Navigationsrechner soll fähig sein, die Daten der Sensoren sowie den Status der SLC und Sicherheits-SPS auszuwerten und Befehle sowie Statusmeldungen zu übermitteln. Eine Analyse der internen Kommunikation des FTF ergab, dass der Anschluss an den Controller-Area-Network (CAN)-Bus ausreicht, um den ROS2-Navigationsrechner einzubinden. Um eine Schnittstelle von ROS2 zum CAN-Bus zu ermöglichen, ist die Implementierung einer weiteren SPS vorgesehen. Obwohl es auch Umsetzungen gibt, die eine direkte CAN-Integration mit ROS2 anbieten, fällt hier die Entscheidung für die Verwendung einer SPS mit CAN-Anschluss. Die Verwendung einer solchen SPS erlaubt eine präzisere Konfiguration der Schnittstelle durch ein spezifisches SPS-Programm (mehr dazu in Kapitel 4.2.2).

Die resultierende Architektur ist in Bild 2 dargestellt, wo der ROS2-Navigationsrechner mit der CAN-SPS und damit indirekt mit dem CAN-Bus verbunden ist. Es besteht jederzeit die Möglichkeit, den ursprünglichen Navigationsrechner mit dem CAN-Bus zu verbinden und die neue CAN-SPS zu entfernen, um die vorgenommenen Änderungen zu verwerfen. Des Weiteren ist festzuhalten, dass eine Verbindung aller Laserscanner mit dem ROS2-Navigationsrechner über ProfiNet und aller Kameras über USB 3 besteht. Die Schnittstelle der Sicherheits-SPS mit dem SLC bleibt unangetastet, um, wie in den Anforderungen definiert, die Sicherheitsfunktionen nicht zu beeinträchtigen.

4.2 Hardware-Erweiterungen/-Ersetzungen

In diesem Kapitel werden die Modifikationen der Hardware einer detaillierten Erörterung unterzogen. Die erörterten Hardware-Geräte sind in Bild 3 dargestellt. In dieser Ausrichtung erfolgt auch der Einbau des Schaltschranks, der in das Innere der Plattform integriert wird.

4.2.1 ROS2-Navigationsrechner

Für den ROS2-Navigationsrechner wird ein kompakter Industrie-PC (IPC) benötigt, der für den Schaltschrank-Einbau geeignet ist und eine hohe Rechenleistung bietet, vor allem eine leistungsstarke CPU (Central Processing Unit). Zudem ist eine Vielzahl an USB-Anschlüssen (mindestens USB 3.1 Generation 1) nötig, um die Anbindung von Kameras und Sensoren zu gewährleisten. Ein Ethernet-Anschluss ist für die Verbindung zu einem Switch ausreichend.

Im Rahmen der Auswahl wurde der Beckhoff „C6030“ IPC präferiert, der sich durch sein kompaktes Design sowie eine Konfiguration mit einem Intel Core i7 auszeichnet und somit den Anforderungen von ROS2 gerecht wird. Aufgrund der Lüftungstechnik ist lediglich eine Schutzart IP20 gegeben, jedoch wird der IPC durch einen entsprechend konstruierten Schaltschrank vor Staub und Feuchtigkeit geschützt.

4.2.2 SPS mit CANopen

Eine SPS ermöglicht eine erhöhte Freiheit und Zuverlässigkeit bei der Programmierung der CAN-BUS-Schnittstelle. Dies ist zum einen darauf zurückzuführen, dass SPS für Echtzeitanwen-

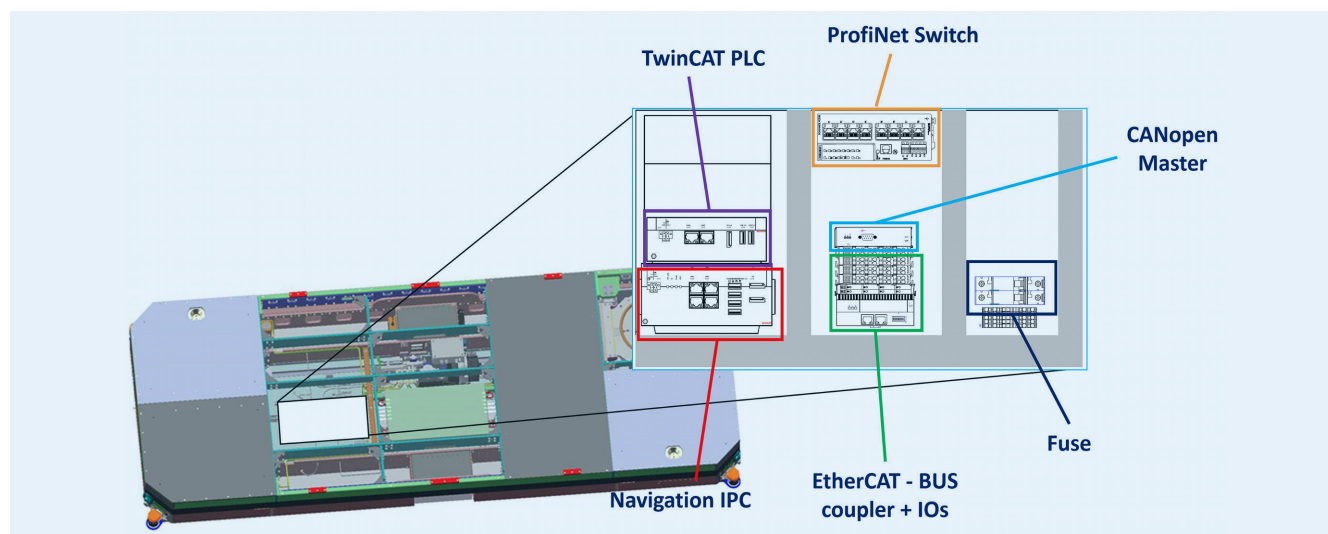


Bild 3. Hardware-Erweiterung für die Transformation zum einem AMR. Grafik: ISW

dungen gut geeignet sind und zum anderen darauf, dass die CAN-Bus-Anbindung einer SPS bereits in der Industrie erprobt ist [17]. Im Gegensatz dazu sind Implementierungen für die CAN-Bus Kommunikation in ROS weniger erprobt.

SPS steuern industrielle Hardware mit analogen und digitalen E/A. Die E/A-Module werden durch Klemmen über ein Bus-system modular erweitert [17]. Für das FTF war eine CANopen-Klemme nötig, da es das CANopen-Protokoll nutzt.

In diesem Fall findet ein IPC des Herstellers Beckhoff Automation Verwendung, der sich durch eine besonders geringe Größe auszeichnet und mit der Software TwinCAT [18] als SPS betrieben wird. Für den Einsatz von E/A-Modulen wird ein EtherCAT-Buskoppler verwendet. An diesen Buskoppler kann dann die CANopen-Klemme angeschlossen werden.

4.2.3 GNSS/INS

Ein weiterer Bestandteil der Aufrüstung ist die Sensoreinheit „VN-200“ von Vektornav. Der VectorNav VN-200 ist ein hochpräzises inertiales Navigationssystem (INS) [19] mit integriertem Global Navigation Satellite System (GNSS)-Empfänger [20], das für die präzise Positions- und Orientierungsbestimmung in mobilen Robotersystemen eingesetzt wird. In der Navigation eines mobilen Roboters nutzt der VN-200 eine Kombination aus Trägheitsmesseinheit (IMU) [21] und GNSS-Datenfusion, um eine kontinuierliche und genaue Bestimmung der Position, Geschwindigkeit und Orientierung des Roboters zu ermöglichen. Durch den Einsatz von MEMS-Sensoren für Beschleunigung und Winkelgeschwindigkeit sowie eines Kalman-Filters zur Sensorfusion werden auch in Umgebungen mit kurzzeitigem GNSS-Signalverlust robuste Navigationsdaten bereitgestellt. Diese Eigenschaften sind besonders vorteilhaft für autonome mobile Roboter, die sich in dynamischen oder strukturierten Umgebungen bewegen. [22]

4.2.4 Kameras

Die Kombination von Kamera-Sensortechnologie eröffnet ein Spektrum an Erweiterungsmöglichkeiten. Roboter, welche über Kamera-Eingänge verfügen, können Methoden aus den Bereichen der Bildverarbeitung und der künstlichen Intelligenz anwenden.

Zu den entsprechenden Methoden zählen die Marker-Erkennung zur Identifikation und Lokalisierung [23], das 3D-SLAM [24] und die Objekterkennung [25].

Tiefenkameras erfassen Umgebungsinformationen in der Robotik. Sie nutzen zwei Bildsensoren, um durch Verschiebung und Punktkorrelation Tiefeninformationen zu berechnen. So erhält jeder Pixel einen Abstandswert. Ein IR-Projektor kann die Genauigkeit durch ein projiziertes Muster verbessern. [26]

Die beschriebenen Stereo-Tiefen-Kameras mit Infrarotprojektion werden von der Intel RealSense „D400-Serie“ angeboten. Die Intel RealSense D400-Serie nutzt Stereo-Tiefenkameras mit Infrarotprojektion und verarbeitet die Daten sensorseitig zu einem Tiefenbild. Ein RGB (Rot-Grün-Blau)-Sensor liefert Farbbilder. Dank globalem Shutter werden, selbst bei Bewegung, Bilder mit niedriger Verzerrung erfasst, was besonders in der mobilen Robotik vorteilhaft ist. [26]

Insgesamt acht Kameras werden durch Gehäuse mit Magneten am äußeren Bereich des AMR befestigt. Die vorderen und hinteren Kameras (links und rechts in Bild 2 unten) sind in Hauptfahrtrichtung ausgerichtet und erfassen Hindernisse in mittlerer Entfernung, da der AMR in diesen Richtungen am schnellsten fährt. Die seitlichen Kameras befinden sich näher an Hindernissen, vor allem beim Einfahren unter Tische (Bild 1), und sollten daher auf Objekte in geringerer Distanz fokussieren, da seitliche Bewegungen meist langsamer erfolgen.

Der ideale Abstandsbereich der Intel RealSense „D455“ beträgt 0,6 m bis 6 m [26]. Daher wurden vier Intel RealSense D455 für die Vorder- und Rückseite festgelegt. Die Intel RealSense „D435“ weist hingegen einen idealen Abstandsbereich von 0,3 m bis 3 m auf [26]. Aus diesem Grund wurden vier Intel RealSense D435 für die seitliche Anbringung ausgewählt.

4.3 Software und Navigation mit ROS2 und Nav2

Dieses Kapitel erörtert die Software-Seite der Transformation. Dazu wird zunächst ROS2 und dessen Einsatz in der Software erläutert. Das Vorgehen sowie die entwickelte Software werden vorgestellt. Dies umfasst die Verwendung von „ros2_control“ für die Hardwareabstraktion, „Nav2“ für die Navigation und „Gazebo“ als Simulationsumgebung.

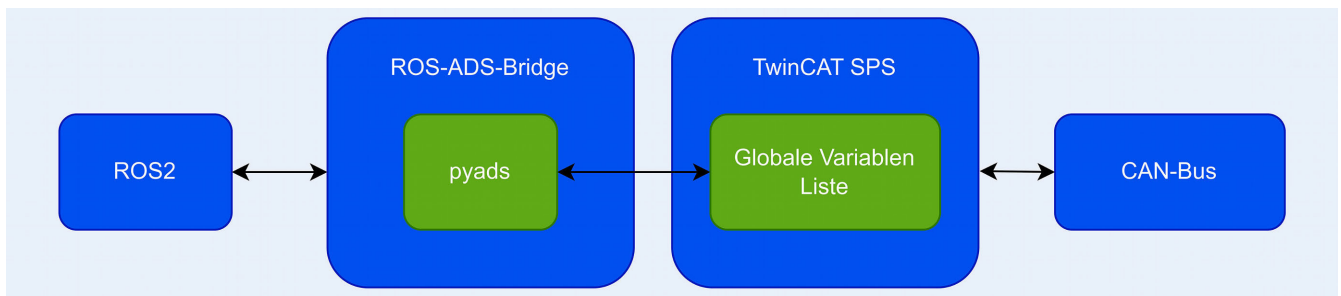


Bild 4. Die ROS-ADS-Bridge-Verbindung. Grafik: ISW

4.3.1 Software mit ROS2

ROS ist eine Reihe von Softwarebibliotheken und Werkzeugen für die Entwicklung von Roboteranwendungen [27]. Dabei werden Entwicklern von Robotikprojekten Open-Source-Werkzeuge und Algorithmen zur Verfügung gestellt. Um den Herausforderungen moderner Roboter-Systeme zu begegnen, wurde ROS2 entwickelt, wobei die zusammenhängenden Projekte von Grund auf neu konzipiert wurden [15].

Im Zuge der Arbeit für das AMR wurde ROS2 als Rahmenwerk für die Softwareentwicklung genutzt. Dadurch bestand die Möglichkeit, Software-Komponenten modular zu gestalten und die Verwaltung der Abhängigkeiten zu integrieren. Die Node-Architektur von ROS2 bildet die Grundlage dieser Vorgehensweise. Die einzelnen Nodes sind jeweils einem einzigen modularen Zweck zuzuordnen. Die Kommunikation zwischen den Nodes erfolgt über Publisher und Subscriber, die Topics ausschreiben oder verfolgen. Einzelne Ausschreibungen eines Topics werden Messages genannt. Dazu hat ein Node die Fähigkeit, einen benötigten Prozess als Service bereitzustellen. Dabei stellt der Service Client die Anfrage, und der Service Server antwortet. Letztlich können eine Vielzahl an Nodes über Launch-Konfigurationen gestartet werden. [27]

Für die Sensorik werden separate Nodes geschrieben. Für die Laserscanner mussten zunächst die ROS2-Treiber übernommen werden. Die Laserscanner sind vom Typ „outdoorScan3“ des Herstellers Sick AG, der einen ROS2-Treiber zur Verfügung stellt. Um jedoch die Werte der Laserscanner für die Navigation und SLAM nutzen zu können, darf nur ein Scan Topic verwendet werden. Zu diesem Zweck wird eine Fusion der vier Laserscanner mit der Funktion „ira_laser_merger“ der ira_laser_tools-Toolbox [28] durchgeführt. Das Endergebnis ist ein einziges Scan Topic, in dem die gesamte Umgebung des AMR abgebildet ist.

Für die Integration der Kameras in ROS2 stellt Intel einen Treiber zur Verfügung. Damit können Video- und Bilddaten direkt in einem ROS2-Node verarbeitet werden. Letztlich verfügt die INS Einheit auch über eine ROS2-Schnittstelle.

4.3.2 ROS-ADS Brücke

Um eine Kommunikation mit der Sicherheits-SPS sowie der SLC zu gewährleisten, ist die Schaffung einer Schnittstelle von ROS2-Messages zum CAN-Bus-Protokoll erforderlich. Da die CAN-SPS (siehe Kapitel 4.2.2) die Umwandlung zum CANopen-Protokoll übernimmt, fehlt nur die Schnittstelle der ROS2-Messages auf dem ROS2-Navigationsrechner zu der CAN-SPS. Diese ist in **Bild 4** dargestellt.

Die CAN-SPS wurde vom Hersteller Beckhoff entwickelt und läuft mit TwinCAT [18]. Die Kommunikation kann bei Beckhoff durch die Definition von „Automation Device Specification“ (ADS)-Geräten eingerichtet werden. Der letzte Baustein ist eine Brücke, die den Navigationsrechner als ADS-Gerät definiert und die ROS2-Messages in SPS-Variablen umformuliert. Die pyads-Bibliothek [29] stellt einen Python-Wrapper für die TwinCAT-ADS-Bibliothek bereit. Für die Anbindung an ROS2 wird pyads in einen ROS2-Node integriert, um die ADS-Verbindung zu legen.

4.3.3 Hardware Abstraktion mit ros2_control

ros2_control ist ein modular aufgebautes Rahmenwerk innerhalb des ROS2, das zur Echtzeitsteuerung von Robotersystemen entwickelt wurde. Es ermöglicht eine einheitliche Schnittstelle zur Ansteuerung verschiedener Hardwarekomponenten, indem es eine Trennung zwischen Steueralgorithmen und Hardwareimplementierung bietet. Dies wird durch die Verwendung von Hardware-Abstraktionsschichten erreicht, die eine standardisierte Kommunikation mit Motoren, Sensoren und anderen Aktoren über Hardware Interfaces ermöglichen. Die Architektur von ros2_control basiert auf Controller-Managern, die verschiedene Regelungs- und Steueralgorithmen verwalten und zur Laufzeit dynamisch laden oder entladen können. Dies gewährleistet eine hohe Flexibilität bei der Entwicklung von Robotersystemen. Vor allem in Kombination mit dem ros2_controllers-Paket lassen sich gängige Regelungsverfahren effizient umsetzen. [30, 31]

Die Hardwareschnittstelle wurde so konfiguriert, dass alle vier Antriebseinheiten und das Hubsystem in den Hardwarestatus des AMR aufgenommen werden. Dadurch wird der Status aller Antriebseinheiten über Drehwinkelsensoren und des Hubsystems über Seilzuggeber als Wegmesssysteme ständig aktualisiert. Zusätzlich können Befehle an die Aktorik definiert werden, um etwa die Position oder Geschwindigkeit eines Motors zu setzen. Die Sensor- und Aktorwerte werden über die ROS2-ADS-Brücke übertragen. Die Status- und Kommandoschnittstellen der Hardware-Schnittstelle werden mit dem gesamten ros2_control Rahmenwerk geteilt.

Im Falle eines AMR mit omnidirektionaler Bewegung erfolgt die Bahnregelung in ROS2 in einer kartesischen Geschwindigkeitsrepräsentation relativ zum Roboterkoordinatensystem. Dabei werden typischerweise die translatorischen Geschwindigkeiten v_x , v_y und die Rotationsgeschwindigkeit ω_z um die Hochachse verwendet. Da der SLC jedoch eine Formulierung in Bezug auf den Momentanpol (MP) erfordert, müssen diese Größen entsprechend transformiert werden. Die MP-Darstellung ist in **Bild 5** ersichtlich. Die Berechnung des MP erfolgt unter Berücksichtigung

der gegebenen Geschwindigkeiten, wobei eine detaillierte Analyse und Herleitung in [32] zu finden ist. Die resultierende Steuergrößenformulierung gibt den MP als Punkt relativ zum Roboter sowie die Zielgeschwindigkeit entlang der Kreisbahn um diesen Punkt an.

4.3.4 Navigation mit Nav2

Das Nav2-Framework [16] bietet eine modulare Lösung für die autonome Navigation mobiler Roboter in ROS2, die grundlegende Funktionen wie Pfadplanung, Lokalisierung und Hindernisvermeidung umfasst. Nav2 basiert auf einer serviceorientierten Architektur, die eine flexible Kombination von Komponenten wie Behavior Trees, globalen und lokalen Planern sowie Sensorfusion ermöglicht. Für die Pfadplanung werden Algorithmen wie A* oder RRT verwendet, die Trajektoriengenerierung erfolgt zum Beispiel über DWB. Die enge Integration mit `ros2_control` erlaubt eine flexible Steuerung und Kompatibilität mit verschiedenen Robotersystemen. [16]

Für die Navigation muss zuerst eine Karte erstellt werden. Hierfür wird SLAM Toolbox [33] verwendet. Die SLAM Toolbox ist ein Softwarepaket in ROS2, das für das SLAM-Verfahren für mobile Roboter konzipiert wurde. Sie bietet eine robuste und skalierbare Lösung für die Erstellung, Speicherung und kontinuierliche Aktualisierung von Karten in Echtzeit. Der Algorithmus basiert auf Graph-SLAM, einer optimierungsbasierten Methode, die Sensordaten verarbeitet, um eine konsistente und genaue Umgebungskarte zu erstellen. Zudem unterstützt das System Pose-Graph Optimization (PGO), was eine effiziente Korrektur von Mapping-Fehlern durch Loop-Closure-Techniken ermöglicht. Die Integration der SLAM Toolbox mit Nav2 und `ros2_control` erlaubt eine nahtlose Einbindung in autonome Navigationsanwendungen. Bei dem entwickelten AMR wird beim ersten Aufsetzen manuell die Strecke abgefahren, um die Karte aufzuzeichnen, wobei der Ausblick ist, dass spätere Aktualisierungen automatisiert eingerichtet werden. [33]

Für die Navigation wird die Karte durch Costmaps ergänzt. In Nav2 werden Costmaps verwendet, um die Umgebung eines mobilen Roboters zu modellieren und sichere Navigationsentscheidungen zu ermöglichen. Costmaps sind rasterbasierte Darstellungen des Raums, in denen Hindernisse, befahrbare Bereiche und Kosten für verschiedene Regionen kodiert sind. Die Basis für eine Costmap bildet eine statische Karte, auf der dynamische Hindernisse berücksichtigt werden können. Die Konfiguration der globalen Costmap umfasst eine Reihe wichtiger Parameter, etwa die Auflösung, die Update-Rate sowie die Inflationsradius-Einstellung, welche bestimmt, inwiefern sich Hinderniskosten in befahrbare Bereiche ausbreiten. Die Einrichtung der Costmap erfolgte unter Verwendung der Simulationsumgebung Gazebo. In diesem Rahmen wurde evaluiert, welche Ausmaße der Inflationsradius um ein Hindernis aufweisen muss, um eine zu nahe Annäherung des AMR zu verhindern oder Kollisionen zu vermeiden. Es wurde festgelegt, dass ein Inflationsradius von 3,8 m mit einer Kostenskalerung von 0,9 als angemessen erachtet wird. [16, 34]

Mittels einer Karte und einer Costmap ist es einem Planer nun möglich, einen globalen Pfad zu einem Ziel zu generieren. Da es sich um einen omnidirektionalen AMR handelt, ist der „Smac State Lattice Planner“ [16] von Nav2 eine geeignete Option. Der Smac State Lattice Planner implementiert einen generischen globalen Zustandsgitter-Pfadplaner, der vorberechnete Mindestkon-

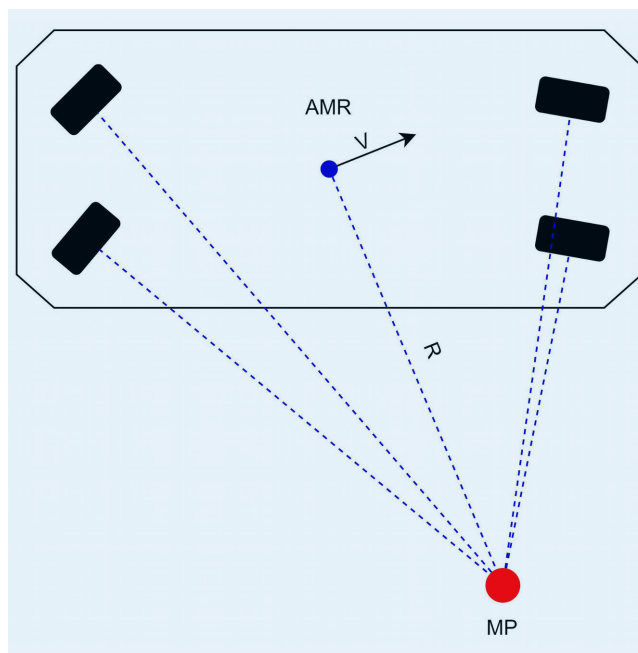


Bild 5. MP (Momentanpol). Darstellung der AMR-Bewegung. Alle vier Antriebseinheiten sind senkrecht zum MP ausgerichtet. Der Radius geht vom Zentrum des AMR zum MP aus und die Geschwindigkeit ist senkrecht auf den Radius. Grafik: ISW

trolsätze verwendet. Das Paket umfasst außerdem ein Python-Dienstprogramm zur Berechnung minimaler Kontrollsätze basierend auf minimalen Krümmungsbeschränkungen, Drehungen an Ort und Stelle sowie omnidirektionalen Bewegungen. Der Smac Lattice Planner vereint die Vorzüge gitterbasierter Planungsverfahren mit einer zufälligen Sampling-Methode, wodurch er lokal optimale Trajektorien ermitteln kann. Diese berücksichtigen sowohl den aktuellen Zustand des Roboters als auch Hindernisse in der Umgebung. Die Optimierung von gesampelten Pfaden erlaubt die Anpassung der Optimierungskriterien, um das gewünschte Verhalten zu erreichen. Die Werte wurden in der Simulation durch Testverfahren so modifiziert, dass der AMR die Vorwärtsrichtung minimal bevorzugt, Rückwärtsfahrt nicht unnötig vermieden wird und reine Rotation nicht erwünscht ist, da sie als ineffizient gilt. Diese Änderungen wurden durch die spezifischen Penalty-Werte `non_straight_penalty`, `reverse_penalty` und `rotation_penalty` vorgenommen.

Die Generierung des globalen Pfades durch den Planer bedingt, dass der AMR diesen während der Fahrt folgt. Nach der Trajektorienplanung mit dem Smac Lattice Planner übernimmt der MPPI-Controller die Steuerung des Roboters entlang des berechneten Pfades. Der MPPI-Controller basiert auf einem modellprädiktiven Ansatz, bei dem zukünftige Bewegungen des Roboters über mehrere Trajektorien simuliert und die Steuerung optimiert wird, um bestimmte Kriterien (zum Beispiel Abweichung vom Pfad) zu minimieren. Die Berücksichtigung von Unsicherheiten und Störungen erlaubt eine präzise und robuste Pfadverfolgung, die eine schnelle Anpassung des Roboters an Änderungen in der Umgebung gewährleistet und die Navigation in dynamischen Szenarien stabilisiert. Darüber hinaus kann das Verhalten des AMR durch die Anpassung der Gewichtungen der Kriterien erheblich modifiziert werden. [16]

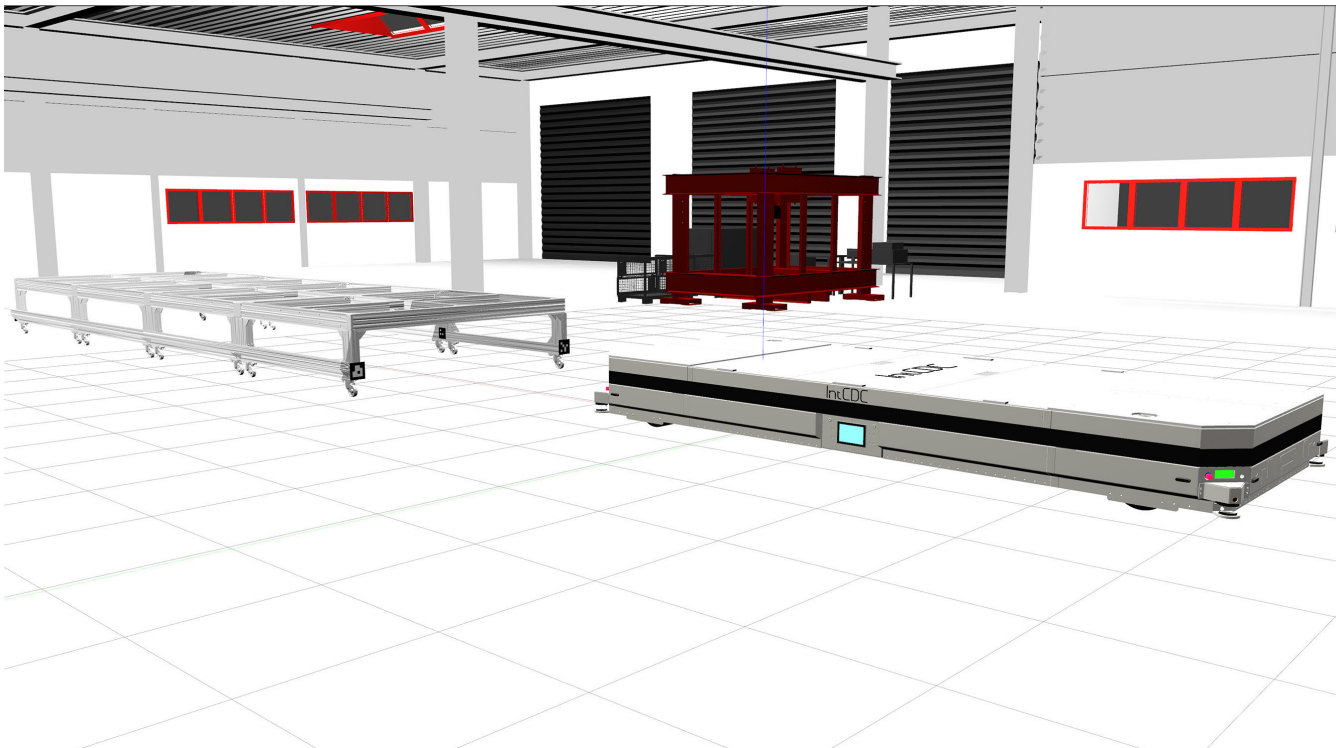


Bild 6. Der AMR in der Gazebo-Simulation. Grafik: ISW

4.3.5 Simulation mit Gazebo

Die Simulation des AMR mit ROS2 in Gazebo [35] bietet eine leistungsfähige Methode, um das Verhalten und die Interaktionen des Roboters in einer virtuellen Umgebung zu testen, bevor er in der realen Welt eingesetzt wird. Über ROS2 in Kombination mit Gazebo, einer weit verbreiteten Simulationssoftware, können verschiedene Szenarien, wie Navigation, Hindernisvermeidung und Interaktion mit der Umgebung, unter realistischen Bedingungen simuliert werden. Die Integration von Gazebo in ROS2 erfolgt über das `ros_gz`-Paket, welches die Kommunikation zwischen den beiden Systemen ermöglicht. Die Modellierung des AMR erfolgt durch das Universal Robotic Description Format (URDF) sowie die 3D-Dateien des Körpers der Plattform. Die Integration von Sensoren wie Laserscannern, Kameras und IMU erfolgt mit Modellen in Gazebo und wird dann mit ROS2 verbunden. [15]

In Bild 6 wird der AMR in der Simulation dargestellt. Die Umgebung wird mit Elementen befüllt, die in der Halle vorzufinden sind. Links vom AMR ist der Tisch erkennbar, der für den Transport geeignet ist. Der AMR ist in der Lage, unter diesen Tisch zu fahren und ihn anzuheben.

Mit dem vorangehenden Kapitel wurde der gesamte Umfang der Transformation zum AMR manifestiert. Die Implementierung dieser Erweiterungen in Hard- und Software ermöglichte die erfolgreiche Transformation der Transportplattform zu einem AMR. Die Validierung der Funktionalität erfolgte durch erste Testfahrten in einer Maschinenhalle mit dem „PF210“.

6 Evaluation und Ergebnisse

Die vorliegende Arbeit resultiert in der Entwicklung einer Transportplattform, welche die erforderliche Hard- und Software eines AMR aufweist. Es sei darauf hingewiesen, dass der AMR

eine signifikante Bedeutung als Forschungsplattform besitzt, da die Integration neuer Funktionalitäten und Prozesse direkt ermöglicht wird.

In diesem Kontext wurden bereits mehrere Funktionalitäten implementiert. Es kann bereits mithilfe einer Kombination aus Laserscannern, Rad-Odometrie (durch die Rad-Encoder) und der IMU der VectorNav Sensoreinheit SLAM betrieben werden. Für die Untersuchung der Funktionalitäten wurden Testverfahren in einer Simulation durchgeführt. Dazu wurde ein virtuelles Modell der Fertigungshalle des IntCDC erstellt und sämtliche Sensoren in Gazebo eingerichtet. Im Rahmen einer Fahrt besteht die Möglichkeit, wie in Kapitel 4.3.4 dargelegt, eine Karte zu erstellen und diese anschließend zu speichern. Es konnte nach einer Übertragung auf den realen AMR bestätigt werden, dass die SLAM-Methode einwandfrei einsetzbar ist.

Im Rahmen der Lokalisierung wurde eine Adaptive Monte Carlo Localization (AMCL) [16] evaluiert. Zu diesem Zweck wurde die Navigationssoftware zusammen mit der aufgezeichneten Karte gestartet und eine Startpose vorgegeben. Eine empirische Untersuchung ergab, dass eine globale Lokalisierung durch eine Testfahrt gewährleistet werden kann. Mittels aktiver Lokalisierung besteht die Möglichkeit, ein Navigationsziel auf der Karte einzufügen. Nach Erhalt des Ziels wurde seitens des Planers ein Pfad für die Tests am realen AMR generiert. Die Analyse ergab, dass der Pfadfolgeregler erfolgreich implementiert wurde. Die Tests wurden zunächst mit reduzierter Geschwindigkeit und auf großflächigen Bereichen der Halle durchgeführt.

Im Rahmen der Simulation fand die Verwendung der Kameras statt, um visuelle Marker zu erkennen und zu identifizieren. Der Einsatz von Markern ermöglicht die vorab erfolgte Hinterlegung von Punkten auf der Karte. Es erfolgte keine Evaluierung der konkreten Anwendung des Kamerasystems.

Im Hinblick auf eine Fahrt im Außenbereich vor der Halle ist die Sicherheit des Straßenverkehrs sowie der Fußgänger zu gewährleisten. Daher wird die Erprobung der GNSS-Funktionalität sowie der Navigation im Freien eine Verzögerung erfahren.

7 Diskussion

Die in dieser Arbeit entwickelte Transportplattform ist eine wesentliche Grundlage für weiterführende Forschungsarbeiten im Bereich autonomer mobiler Robotik. Durch die vollständige Integration der nötigen Hard- und Softwarekomponenten konnte ein funktionsfähiger Prototyp realisiert werden, der als flexible Testplattform für verschiedene Anwendungsfälle dient. Besonders hervorzuheben ist die modulare Systemarchitektur, welche zukünftige Erweiterungen – sei es durch zusätzliche Sensorik, alternative Lokalisierungsansätze oder neue Navigationsstrategien – ohne grundlegende Änderungen am Gesamtsystem erlaubt.

Die durchgeführten Simulationen und anschließenden realen Tests verdeutlichen die Praxistauglichkeit der entwickelten SLAM- und Lokalisierungskomponenten. Der Einsatz von Laserscannern in Kombination mit Odometrie- und IMU-Daten hat sich im Indoor-Bereich als robuste Methode zur Kartenerstellung erwiesen. Ein wesentlicher Vorteil liegt in der Möglichkeit, mit den im Simulationsmodell gewonnenen Erkenntnissen direkt auf die reale Plattform zu übertragen – ein Vorgehen, das Entwicklungszeiten deutlich reduziert und iteratives Testen erleichtert.

Die Evaluierung des AMCL-Ansatzes zur Lokalisierung zeigte, dass eine initiale Positionsbestimmung sowie eine zuverlässige Navigation im bereits kartierten Bereich möglich ist. Die empirische Analyse belegt, dass die generierten Trajektorien zuverlässig abgefahren werden können, auch wenn derzeit noch mit reduzierter Geschwindigkeit operiert wird. Diese Einschränkung ist in erster Linie sicherheitsbedingten Erwägungen geschuldet und eröffnet gleichzeitig Perspektiven für künftige Optimierungen im Hinblick auf die Fahrdynamik und Regelgüte.

Der Einsatz visueller Marker in der Simulation weist auf weiteres Potenzial der Plattform hin, insbesondere im Kontext einer präzisen Objekt- und Zielidentifikation. Die Integration solcher Marker bietet Möglichkeiten zur semantischen Anreicherung der Karte und zur Kontextualisierung von Navigationszielen. Da eine systematische Validierung dieser Funktionalität bislang aussteht, ist dies ein relevantes Ziel für zukünftige Arbeiten.

Ein kritischer Punkt ist derzeit die eingeschränkte Nutzbarkeit des Systems im Außenbereich. Die Abhängigkeit von zuverlässigen GNSS-Daten und die Anforderungen an die Verkehrssicherheit machen weiterführende Tests notwendig, deren Umsetzung derzeit noch aussteht. Die damit verbundenen Herausforderungen, vor hinsichtlich der Sensorfusion unter variablen Umweltbedingungen, erfordern zusätzliche Entwicklungsarbeit, bieten jedoch gleichzeitig ein vielversprechendes Anwendungsfeld für den erweiterten Einsatz der Plattform.

Es lässt sich festhalten, dass die Transportplattform bereits wesentliche Grundfunktionen eines AMR erfüllt und als vielseitige Forschungs- und Entwicklungsbasis dient. Die bisherigen Ergebnisse belegen sowohl die Funktionstüchtigkeit als auch das Erweiterungspotenzial des Systems. Zukünftige Arbeiten sollten sich auf das erweiterte Testen der Navigation in der Halle, die Weiterentwicklung der Außenbereichsnavigation, die Integration semantischer Umgebungsinformationen sowie auf eine Verbesserung

der Systemrobustheit unter realen Betriebsbedingungen konzentrieren.

8 Fazit und Ausblick

In diesem Beitrag wird die Transformation eines FTF des Typs Stäubli WFT PF210 zu einem AMR präsentiert. Dabei erfolgte eine Erweiterung um Kamera- und GNSS/INS-Sensoren, während die Software durch ROS2 aufgerüstet und partiell ersetzt wurde. Die Testverfahren umfassten sowohl Simulationen als auch reale Tests mit dem finalen System.

Der AMR wird nun als Forschungsplattform für das IntCDC-Exzellenzcluster genutzt, wobei die Anwendung eines AMR in der Vorfertigung für den Bau zum Einsatz kommt. Die Erwartung ist, dass die Roboterplattformen und das AMR über ein gemeinsames Netzwerk kommunizieren und folglich kollaborieren. Zudem werden in Zukunft Tests durchgeführt, um die Kombinationen von Sensoren und den entsprechenden Sensorfusion-Algorithmen zu ermitteln, die in der Halle und außerhalb die beste Leistung aufweisen. Schließlich wird angestrebt, das Sicherheitskonzept zu erweitern, indem die Hinderniserkennung mittels Kameras, die Objekte erkennen können, welche für den Laserscanner nicht sichtbar sind, integriert wird.

FÖRDERHINWEIS

Gefördert durch die Deutsche Forschungsgemeinschaft (DFG) im Rahmen der deutschen Exzellenzstrategie – EXC 2120/1 – 390831618

LITERATUR

- [1] Ullrich, G.; Albrecht, T.: Die Welt des FTS. In: Ullrich, G.; Albrecht, T.: Fahrerlose Transportsysteme. Wiesbaden: Springer Vieweg 2023, S. 1–32
- [2] Zhang, J.; Yang, X.; Wang, W. et al.: Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering. *Automation in Construction* 146 (2023), #104699
- [3] Feldmann, F. G.: Towards Lean Automation in Construction – Exploring Barriers to Implementing Automation in Prefabrication. *Sustainability* 14 (2022) 19, #12944
- [4] Sizirci, B.; Fseha, Y.; Cho, C.-S. et al.: A Review of Carbon Footprint Reduction in Construction Industry, from Design to Operation. *Materials* 14 (2021) 20, #6094
- [5] United Nations Department of Economic and Social Affairs (eds.): *World Urbanization Prospects: The 2018 Revision*. Stand: 2019, doi.org/10.18356/b9e995fe-en
- [6] Knippers, J.; Kropp, C.; Menges, A. et al.: Integrative computational design and construction: Rethinking architecture digitally. *Civil Engineering Design* 3 (2021) 4, pp. 123–135
- [7] Ullrich, G.; Albrecht, T.: *Fahrerlose Transportsysteme*. Wiesbaden: Springer Vieweg 2023
- [8] Alati, M. B.; Hancke, G. P.: A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* 8 (2020), pp. 39830–39846
- [9] Ryck, M. de; Versteyhe, M.; Debrouwere, F.: Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems* 54 (2020), pp. 152–173
- [10] Atanasov, N.; Le Ny, J.; Daniilidis, K. et al.: Decentralized active information acquisition: Theory and application to multi-robot SLAM, 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 2015, pp. 4775–4782, doi.org/10.1109/ICRA.2015.7139863


- [11] DIN EN 1525:1997–12: Sicherheit von Flurförderzeugen – Fahrerlose Flurförderzeuge und ihre Systeme. Deutsche Fassung, Ausgabe 12 1997
- [12] DIN EN ISO 3691–1:2020–11: Flurförderzeuge – Sicherheitstechnische Anforderungen und Verifizierung – Teil 1: Motorkraftbetriebene Flurförderzeuge mit Ausnahme von fahrerlosen Flurförderzeugen, Staplern mit veränderlicher Reichweite und Lastentransportfahrzeugen (ISO_3691–1:2011 einschließlich Cor_1:2013_+ AMD.1:2020). Deutsche Fassung, Ausgabe 11 2020
- [13] Navitec Systems: Natural Feature Navigation – Basics and Benefits. Stand: 2023. Internet: navitecsystems.com/blog-post/natural-feature-navigation-basics-and-benefits/. Zugriff am 02.06.2025
- [14] Navitec Systems: Real 3D outdoor navigation. Infrastructure-free, all-weather solution. Internet: navitecsystems.com/outdoor-navigation/. Zugriff am 02.06.2025
- [15] Macenski, S.; Foote, T.; Gerkey, B. et al.: Robot Operating System 2: Design, architecture, and uses in the wild. Science Robotics 7 (2022) 66, eabm6074
- [16] Macenski, S.; Martin, F.; White, R. et al.: The Marathon 2: A Navigation System. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2020, pp. 2718–2725, doi.org/10.1109/IROS45743.2020.9341207
- [17] Linke, P.: Speicherprogrammierbare Steuerungen. In: Böge, A., Böge, W. (Hrsg.): Handbuch Maschinenbau. Wiesbaden Springer Vieweg 2021, S. 1497–1516
- [18] Beckhoff Automation: Beckhoff Information System. TwinCAT 3. Internet: infosys.beckhoff.com/index.php?content=content/1031/tc3_installation/179473291.html&id=. Zugriff am 02.06.2025
- [19] Advanced Navigation: Inertial Navigation Systems (INS) – An Introduction. Stand: 2023. Internet: www.advancednavigation.com/tech-articles/inertial-navigation-systems-ins-an-introduction/. Zugriff am 02.06.2025
- [20] Kaplan, E. D.; Hegarty, C.: Understanding GPS/GNSS: Principles and Applications. London: Artech House 2017
- [21] Advanced Navigation: Inertial Measurement Unit (IMU) – An Introduction. Stand: 2023. Internet: www.advancednavigation.com/tech-articles/inertial-measurement-unit-imu-an-introduction/. Zugriff am 02.06.2025
- [22] Vectornav: VN-200. GNSS / INS. Internet: www.vectornav.com/products/detail/vn-200. Zugriff am 02.06.2025
- [23] Bharadwaj R. K. Mantha; Borja García de Soto: Designing a Reliable Fiducial Marker Network for Autonomous Indoor Robot Navigation. 36th International Symposium on Automation and Robotics in Construction (ISARC 2019), 2019, doi.org/10.22260/ISARC2019/0011
- [24] Engelhard, N.; Endres, F.; Hess, J.; et al: Real-time 3D visual SLAM with a hand-held RGB-D camera. Stand: 2011. Internet: cvg.cit.tum.de/_media/spezial/bib/engelhard11euron.pdf. Zugriff am 02.06.2025
- [25] Civera, J.; Galvez-Lopez, D.; Riazuelo, L. et al.: Towards semantic SLAM using a monocular camera. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 2011, pp. 1277–1284, doi.org/10.1109/IROS.2011.6094648
- [26] Intel RealSense Technology: Intel® RealSense™ Product Family D 400 Datasheet. Stand: 2024. Internet: www.intelrealsense.com/wp-content/uploads/2024/10/Intel-RealSense-D400-Series-Datasheet-October-2024.pdf. Zugriff am 02.06.2025
- [27] Open Robotics: ROS 2 Documentation. Internet: docs.ros.org/en/humble/index.html. Zugriff am 02.06.2025
- [28] Ballardini, A. L.; Fontana, S.; Furlan, A. et al.: ira_laser_tools: a ROS LaserScan manipulation toolbox. Arxiv 2014, doi.org/10.48550/arXiv.1411.1086
- [29] Pyads: pyads documentation. Stand: 2015. Internet: pyads.readthedocs.io/en/latest/index.html. Zugriff am 02.06.2025
- [30] Github: ros2_control. Internet: github.com/ros-controls/ros2_control. Zugriff am 02.06.2025
- [31] ros2_control Development Team: ros2_control documentation. Stand: 2025. Internet: control.ros.org/humble/index.html. Zugriff am 02.06.2025
- [32] Connette, C. P.; Pott, A.; Hagele, M. et al.: Control of an pseudo-omnidirectional, non-holonomic, mobile robot based on an ICM representation in spherical coordinates. 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 2008, pp. 4976–4983, doi.org/10.1109/CDC.2008.4738958
- [33] Macenski, S.; Jambrecic, I.: SLAM Toolbox: SLAM for the dynamic world. Journal of Open Source Software 6 (2021) 61, #2783
- [34] Lu, D. V.; Hershberger, D.; Smart, W. D.: Layered costmaps for context-sensitive navigation. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 2014, pp. 709–715, doi.org/10.1109/IROS.2014.6942636
- [35] Koenig, N.; Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 2004, pp. 2149–2154

Siddieq Mansour, M.Sc.

siddieq.mansour@isw.uni-stuttgart.de

Tel. +49 711 685 82440

Dr.-Ing. Armin Lechler 

Univ.-Prof. Dr.-Ing. Alexander Verl 

Institut für Steuerungstechnik der Werkzeugmaschinen
und Fertigungseinrichtungen (ISW)
Universität Stuttgart
Seidenstr. 36, 70174 Stuttgart
www.isw.uni-stuttgart.de

LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons
Namensnennung 4.0 International (CC BY 4.0)