

Prof. Dr.-Ing. Jörg Krüger, Berlin
Prof. Dr. h. c. Dr.-Ing. Eckart Uhlmann, Berlin
Prof. Dr.-Ing. Alexander Verl, Stuttgart
Prof. Dr.-Ing. Oliver Rieder, Stuttgart
Prof. Dr.-Ing. Christian Brecher, Aachen (Hrsg.)

Produktion 2030 – Wandel in der Automatisierungstechnik

Fortschritt-Berichte VDI

Reihe 2

Fertigungstechnik

Prof. Dr.-Ing. Jörg Krüger, Berlin
Prof. Dr. h. c. Dr.-Ing. Eckart Uhlmann, Berlin
Prof. Dr.-Ing. Alexander Verl, Stuttgart
Prof. Dr.-Ing. Oliver Rieder, Stuttgart
Prof. Dr.-Ing. Christian Brecher, Aachen (Hrsg.)

Nr. 699

Produktion 2030 -
Wandel in der
Automatisierungstechnik

VDI verlag

Jörg Krüger, Eckart Uhlmann, Alexander Verl, Oliver Rieder, Christian Brecher (Hrsg.)

Produktion 2030 – Wandel in der Automatisierungstechnik

Fortschr.-Ber. VDI Reihe 2 Nr. 699. Düsseldorf: VDI Verlag 2019.

160 Seiten, 81 Bilder, 4 Tabellen.

ISBN 978-3-18-369902-5, ISSN 0178-9406,

€ 57,00/VDI-Mitgliederpreis € 51,30.

Für die Dokumentation: Automatisierung – Mensch-Technik-Schnittstelle – Funkkommunikation – Rapid Engineering – Dienste-basierte Architekturen – Condition Monitoring – Cyber-physische Systeme – Rekonfiguration – Kollisionsvermeidung – Intelligente Zentrierung

Die produktionstechnischen Institute aus Aachen (WZL/IP), Berlin (IWF/IPK) und Stuttgart (ISW) tragen mit Ihren Forschungsarbeiten dazu bei, die von Industrie 4.0 gesetzten Impulse aufzugreifen und mit neuen disruptiven Ansätzen aus der Informations- als auch Kommunikationstechnik weiterzuentwickeln. Der vorliegende Band dokumentiert die Arbeiten und Aktivitäten zum Thema „Produktion 2030 – Wandel in der Automatisierungstechnik“ und gibt einen Überblick zu Konzepten und Visionen sowie bereits realisierten Implementierungen. Diese Ausgabe stellt Forschungsarbeiten zukünftiger Systeme auf den Gebieten Mensch-Technik-Schnittstelle in cyber-physischen Produktionssystemen, Condition Monitoring effizienter Drehprozesse, Intelligente Kollisionsvermeidung fahrerloser Transportsysteme, Rapid Engineering mittels Funkkommunikation, Servicearchitekturen für Robotersysteme, Smart Centering rotations-symmetrischer Körper und Live-Rekonfiguration von Seilrobotern dar.

Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

© VDI Verlag GmbH · Düsseldorf 2019

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9406

ISBN 978-3-18-369902-5

Vorwort

In den letzten Jahren brachte das Thema Industrie 4.0 sowohl in der Forschungs- als auch in der industriellen Praxis neue Impulse für Konzepte und Lösungsvorschläge im Bereich der Produktionstechnik hervor. Dabei führten technologische Entwicklungen im Bereich der Informations- als auch Kommunikationstechnik zu disruptiven Ansätzen in der Produktionstechnik, welche die Betrachtungsweise zukünftiger Automatisierungs- und Produktionstechnischer Lösungen als auch Paradigmen maßgeblich veränderten. Dieser Betrachtungswechsel als auch das Zusammenkommen verschiedener Domänen befindet sich dabei immer noch im Anfangsstadium und bedarf weiterer Versuche, um das komplette Innovationspotential und Spektrum an Möglichkeiten auszunutzen. An dieser Stelle übernehmen die Universitäten eine wichtige Rolle als unabhängige Vordenker und Katalysatoren zur Schaffung und Transfer von Wissen, um so einen langfristigen technologischen Nutzen für einen weiten Interessentenkreis zu schaffen. Die zentrale Aufgabe besteht dabei nicht nur in der Konzeption disruptiver und innovativer Ansätze, sondern ebenfalls in der Validierung und wissenschaftlichen Bewertung dieser bis hin zur industriellen, applikationsspezifischen Realisierung.

Das jährliche Treffen der drei Institute für Produktionstechnik aus Berlin, Stuttgart und Aachen soll den Wissenstransfer unterstützen, indem ein Austausch zu aktuellen Themenstellungen anhand von Beiträgen der einzelnen Institute stattfindet. Das Treffen aus dem Jahr 2018, dessen Beiträge Sie nun in der Hand halten, fand unter dem Leitthema „Produktion 2030 – Wandel in der Produktionstechnik“ statt. Hierbei fokussierte das Treffen die Themen:

- Mensch-Technik-Schnittstelle in cyberphysischen Produktionssystemen
- Condition Monitoring effizienter Drehprozesse
- Intelligente Kollisionsvermeidung fahrerloser Transportsysteme
- Rapid Engineering mittels Funkkommunikation
- Servicearchitekturen für Robotersysteme
- Smart Centering rotationssymmetrischer Körper
- Live-Rekonfiguration von Seilrobotern

Die Ausführungen der einzelnen Themen finden Sie in den nachfolgenden Kapiteln. An dieser Stelle gilt ein besonderer Dank den Autoren der Beiträge, welche sich in der täglichen Arbeit an den Instituten mit den jeweiligen Themen beschäftigen.

Abschließend hoffen wir, dass der Wissenstransfer zwischen den Instituten auch Ihnen als Leser nützliche Impulse liefert und das Themenfeld um Industrie 4.0 weiter beleuchtet.

Berlin 2018 J. Krüger, E. Uhlmann

Stuttgart 2018 A. Verl, O. Riedel

Aachen 2018 C. Brecher (Herausgeber)

Inhalt

1	INTEGRATION DER MENSCH-TECHNIK-SCHNITTSTELLE IM CYBERPHYSISCHEN PRODUKTIONSSYSTEM	1
2	PROZESSÜBERWACHUNG UND -REGELUNG FÜR EFFIZIENTE DREHPROZESSE.....	14
3	VERMEIDUNG VON NOTHALTEN VON FAHRERLOSEN TRANSPORTFAHRZEUGEN DURCH LOKALE BAHNPLANUNG.....	34
4	RAPID PROCESS PROTOTYPING – KONZEPT ZUR DRAHTLOSEN INBETRIEBNAHME UND PROZESSENTWICKLUNG AUF BASIS VON BLUETOOTH LOW ENERGY.....	58
5	Dienstebasierte Architekturen für Robotersysteme... ..	89
6	SMART CENTERING – METHODIK ZUR INTELLIGENTEN ZENTRI ERUNG ROTATIONSSYMMETRISCHER KÖRPER FÜR DIE NULL-FEHLER- FERTIGUNG IN MEHRSTUFIGEN PRODUKTIONSSYSTEMEN	110
7	EIN KONZEPT ZUR IN-BETRIEB-REKONFIGURATION VON SEILROBOTERN.....	133

1 Integration der Mensch-Technik-Schnittstelle im cyberphysischen Produktionssystem

K. Schilling und C. Brecher, WZL RWTH Aachen

1.1 Motivation und Ausgangssituation

Die Rolle des Menschen und die Entwicklung der Mensch-Technik-Interaktion werden entscheidend beeinflusst durch den Wandel der industriellen Produktion zu einem intelligenten Netzwerk der Dinge und Dienste bzw. dem sog. Internet of Things (IoT) [1, 2]. Das Potential der Automatisierungstechnik im cyberphysischen Produktionssystem (CPPS) und der künstlichen Intelligenz (KI) besteht in der Beherrschung komplexer Prozesse; die große Stärke der Informationstechnologie (IT) ist ihre Leistungsfähigkeit im Rahmen definierbarer Randbedingungen. Bei der Beherrschung unvorhersehbarer Verhaltens erweist sich dagegen der Mensch als flexibler und kreativer in der strategischen Problemlösung. [3, 4] Der Mitarbeiter spielt daher auch bei steigender Automatisierung und der zunehmenden Anwendung von KI weiterhin eine zentrale Rolle im Produktionsgeschehen.

Als Akteur im automatisierten Produktionssystem übernimmt er neben nicht-automatisierbaren Tätigkeiten, die Überwachung, Anpassung und Instandhaltung des Systems [5]. Dabei stehen den mit der zunehmenden Vernetzung verbundenen Anforderungen die Problemfelder des menschlichen Handelns gegenüber: Beherrschung von Komplexität, Dynamik und Intransparenz (vgl. [6]). Um Menschen vom technischen System nicht auszuschließen, sondern ihm dieses zur Erfüllung seiner Aufgaben zu jeder Zeit transparent und beherrschbar zu machen, bedarf es (in Ergänzung zu den spezialisierten Werkzeugen wie Maschinensteuerung, Roboterprogrammierung etc.) einer intuitiven, ganzheitlichen Mensch-Technik-Schnittstelle für das gesamte System.

Interaktionstechnologien wie mobile Geräte und Augmented Reality (AR) finden im aktuellen Wandel immer mehr Anwendung. Ihr Einsatz als Arbeitsmittel bietet neue Möglichkeiten zur Effizienzsteigerung, bringt aber auch besondere Herausforderungen mit sich. Oft ist es nicht eine einzelne Technologie, die eine ganze Arbeitsaufgabe optimal begleitet, sondern die Gesamtheit einzelner Elemente, die zusammen eine geräteübergreifende Mensch-Technik-Schnittstelle bilden. Die Vision einer interaktiven Umgebung, in der die digitale und reale Welt miteinander verschmelzen, wurde von WEISER unter dem Begriff „Ubiquitous Computing“ [7] geprägt. In diesem Beitrag wird ein Basisframework vorgestellt, das die Grundlage für eine solche Bedienschnittstelle im Kontext einer intelligenten Produktionsumgebung bereitstellt. Darauf aufbauen können konkrete Anwendungen, z.B. mit AR oder für mobile Geräte, im Nutzungskontext integriert werden.

1.2 Entwicklung der Mensch-Maschine-Interaktion im Kontext der Produktionsautomatisierung

Die Mensch-Technik-Schnittstelle im Produktionssystem hat die Aufgabe, die Transparenz der Prozesse und des Systemzustands herzustellen und die Zugänglichkeit von Funktionen, Daten, Information und Wissen bedarfsgerecht zu gewährleisten. Der Nutzungskontext wird (in Anlehnung an [8]) durch die Menschen, die zu erfüllende Aufgabe und das technische System gebildet (vgl. Abbildung 1). Die Rolle des Menschen wird durch seine Kompetenzen und Fähigkeiten, sowie die zu erfüllende Aufgabe beeinflusst. Das hierfür notwendige Wissen wird durch die Aufgabe und das technische System bestimmt. Durch die Bedienschnittstelle hat der Mitarbeiter die Möglichkeit, auf das technische System einzuwirken.



Abbildung 1 Aspekte des Nutzungskontexts in Anlehnung an das ABC-Modell nach [8]

Mit dem Wandel in der Gestaltung von Produktionssystemen wird es für den Mitarbeiter immer wichtiger, das System als Ganzes zu beherrschen und Abhängigkeiten zu erkennen. Die Bedienschnittstelle entwickelt sich daher vom einzelnen Bediengerät für konkrete Anwendungen zur Schnittstelle zum gesamten System. Die Grundlage bilden hierfür verschiedene Interaktionstechnologien, Geräte und Design Konzepte, die im Kontext der Produktionsautomatisierung adaptiert und integriert werden.

1.2.1 Technologien und Geräte

Abhängig von der jeweiligen Arbeitsaufgabe bieten die verschiedenen Bediengeräte individuelle Vor- und Nachteile. Geräte aus dem Konsumgütermarkt haben inzwischen

ihren Weg zum Arbeitsmittel für die Informationsbereitstellung und einfache Bedienungsaufgaben gefunden. Allerdings gilt es Herausforderungen, wie die Integration in die bestehende IT-Landschaft, sowie Anforderungen an Ergonomie und Sicherheit, zu lösen. [9]

1.2.1.1 Mobile Geräte und Wearables

Mobile Geräte als Human Machine Interface (HMI) erweitern den Wirkungsbereich des Mitarbeiters sowohl lokal als auch global. Räumlich unbegrenzt kann sich der Mitarbeiter bspw. um eine große Maschine herum frei bewegen; das digitale System rückt so näher an den physischen Prozess. Auch außerhalb der unmittelbaren Nähe zur Anlage bleibt das HMI eines Systems mit mobilen Geräten immer in Reichweite. Der Mitarbeiter kann die Informationen „mitnehmen“ und auch aus der Ferne auf Daten zugreifen. [10]

Tablets bieten eine große Interaktionsfläche, sie können gut einem Arbeitsplatz oder einer Aufgabe zugeordnet werden. Die Bereitstellung der Geräte am Arbeitsplatz (Befestigung, Stromversorgung, Zugriff) muss dabei mit der Arbeitsaufgabe abgestimmt werden. Smartphones fungieren gut als persönliches Gerät, das der Nutzer immer bei sich trägt. Die meisten Anwender sind mit dem Interaktionskonzept gut vertraut, für umfangreiche Inhalte ist die Interaktion allerdings nicht geeignet. Datenbrillen bzw. sog. Head Mounted Devices (HMD) ermöglichen das freihändige arbeiten. Die Interaktion unterscheidet sich von Gerät zu Gerät (z.B. Gestensteuerung, Sprachinteraktion, Touchpad, Buttons) und muss zunächst erlernt werden. Zudem sind die am Markt verfügbaren Geräte heute insbesondere bzgl. der Anforderungen an Ergonomie und Sicherheit immer noch nicht für einen längeren Einsatz am Arbeitsplatz geeignet. Weitere mobile Geräte wie Smartwatches haben als ergänzendes HMI ebenfalls Potential, z.B. als Alarm durch Vibration, für minimale Interaktionen (z.B. Quittier-Button) oder für eine reduzierte Informationsbereitstellung.

1.2.1.2 Augmented und Virtual Reality

Augmented Reality (AR) bezeichnet die Ergänzung der realen Umgebung durch computergenerierte Inhalte und wird derzeit intensiv für die Anwendung im Produktionskontext beforscht. Anwendungsfelder sind insbesondere die Unterstützung bei Montageaufgaben, Remoteassistentz, Trainingsszenarien und Logistikanwendungen [11]. Technisch lassen sich drei Formen von Augmented Reality unterscheiden: Bei der Video-basierten AR werden die virtuellen Inhalte in das mit einer Kamera aufgenommene Videobild integriert und z.B. auf Tablet oder Smartphone ausgegeben. See-Through AR bezeichnet die Ergänzung der realen Umgebung durch ein transparentes Display, z.B. in Form einer Datenbrille. Die in-situ Projektion meint eine Projektion der Inhalte z.B. mit einem Beamer, wobei die Darstellung die realen Objekte überlagert.

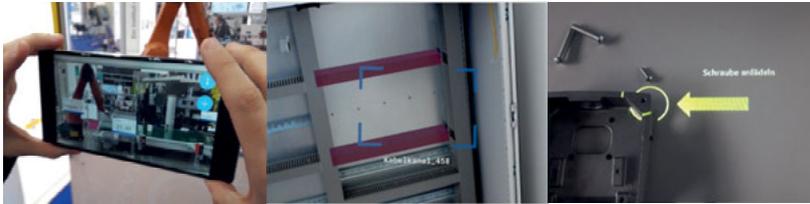


Abbildung 2 Anwendungsbeispiele für Video-basierte AR, See-Through AR und in-situ Projektion (v.l.n.r.)

Virtual Reality (VR) bezeichnet die Darstellung einer vollständig virtuellen Welt, z.B. durch HMD oder als CAVE (Cave Automatic Virtual Environment). Anwendungsfälle für VR sind u.a. in der Anlagen- und Fabrikplanung oder für Trainings zu finden [12]. VR-Anwendungen zielen darauf ab, den Nutzer in eine simulierte Situation zu versetzen, in die er durch eine immersive Darstellung eintauchen kann. Die immersive Gestaltung von AR-Anwendungen lassen die virtuellen Inhalte mit der realen Welt verschmelzen, so dass der Nutzer mit beiden Welten gleichermaßen interagieren kann.

Immersion bezeichnet das Eintauchen des Nutzers in die virtuelle Welt von VR/AR-Anwendungen oder allgemeiner verstanden: Der Nutzer interagiert so natürlich mit dem technischen System, dass die Bediengeräte als Hilfsmittel der Interaktion in den Hintergrund treten, bis sie schließlich nahezu verschwinden und der Nutzer (scheinbar) unmittelbar mit dem Interaktionsobjekt agiert. Technologien wie Gestensteuerung oder Sprachinteraktion unterstützen die natürliche Interaktion, bei der die Abstraktion zwischen Nutzer und Technologie immer weiter reduziert wird (vgl. Abbildung 3). [13]

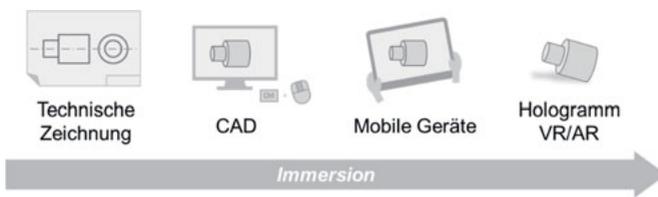


Abbildung 3 Immersion im Kontext von Produktdaten am Beispiel der Ansicht eines Geometriemodells aus verschiedenen Blickwinkeln

Wenn man sich von der klassischen Definition des AR-Begriffs (vgl. [14, 15]) löst, so könnte man auch die Ergänzung der physischen Welt durch ein in die Umgebung eingebettetes Informationssystem als „erweiterte Realität“ bezeichnen: Durch die Gesamtheit aus einzelnen Interaktionsgeräten, Lichtelementen, akustischen Signalen

und haptischem Feedback ließe sich die digitale Welt in die physische Umgebung einbetten, so dass ein immersiver Eindruck des digitalen Produktionssystems entsteht, mit dem der Nutzer auf natürliche Weise interagieren kann.

1.2.1.3 Multimodale Interaktion

Neben der grafischen Oberfläche als am weitesten verbreitete Bedienschnittstelle, kann der Nutzer auch über andere Modalitäten in Kontakt mit dem technischen System treten: Die visuelle Wahrnehmung als wichtigster Sinneseindruck für die Informationsaufnahme werden ergänzt durch akustische Signale bis hin zur Sprachein- und -ausgabe. Nicht zu vergessen ist die Bedeutung des Tastsinns: Vibrationen als taktiler Feedback können z.B. in lauten Umgebungen eingesetzt werden, insbesondere tastbare Steuerelemente sind trotz vermehrtem Einsatz von Touchscreens nicht zu vernachlässigen. Redundante Informationskanäle ermöglichen zudem eine Interaktion auch in besonderen Situationen oder für beeinträchtigte Nutzer. Die Modalität ist dabei der Umgebung (z.B. Lautstärke, Beleuchtungssituation, etc.) anzupassen.

1.2.2 Gestaltungskonzepte

Die Gestaltung der Bedienschnittstelle definiert das Erscheinungsbild des technischen Systems. Immer mehr Interaktionskonzepte aus der Unterhaltungselektronik (z.B. Touch statt WIMP vgl. [6]) werden auch für die Bedienung industrieller Maschinen und Anlagen adaptiert. Die gebrauchstaugliche Gestaltung der Bedienoberflächen (vgl. [16, 17]) bildet die Grundlage einer stress- und fehlerfreien Bedienung und spiegelt zugleich den Eindruck hoher Qualität des gesamten Systems wieder. Die Bedienschnittstelle ist dabei nicht nur auf das Graphical User Interface (GUI) beschränkt, sondern impliziert multimodale und natürliche Schnittstellen (sog. Natural/Tangible User Interface), die der Bediensituation angepasst sind.

1.2.2.1 Kontextualisierung

Die kontextsensitive Anpassung des HMI unterstützt die Bedienung durch eine Reduktion der Informationsdichte und Interaktionsmöglichkeiten. Entitäten werden dabei bzgl. Identität, Ort und Zeit (Primärkontext) beschrieben, Attribute beschreiben Zustand bzw. Aktivitäten im System (Sekundärkontext). [18] Die Bedürfnisse des Nutzers werden so exakt adressiert, passen sich automatisch an die Situation an (adaptiv) oder werden manuell beliebig skaliert (adaptierbar). Durch semantische Beschreibungskonzepte (z.B. Umgebungs-, Nutzer- und Ressourcenmodell) ist eine feinstufige Anpassung der Informationsdichte auch in Abhängigkeit mehrerer Einflussgrößen möglich.

1.2.2.2 Calm Design

Erfahrene Mitarbeiter können den Zustand einer klassischen Werkstattumgebung bereits durch den Gesamteindruck aus Geräuschbild, Bewegungsverhalten oder sogar Geruch intuitiv einordnen - oft noch bevor ein tatsächlicher Störfall explizit erkennbar

wird. In der Produktionsumgebung verschwindet durch die Digitalisierung zunehmend das sinnlich wahrnehmbare Gesamtbild; insbesondere informationstechnischen Prozesse dominieren die Systemsteuerung, sind jedoch natürlicherweise nicht erkennbar.

Calm Design bezeichnet die Gestaltung von technischen Systemen und Alltagsgegenständen, so dass ihr Zustand und nicht sichtbare Prozesse durch kleine Signale wahrnehmbar werden, ohne dass sie die explizite Aufmerksamkeit des Nutzers erfordern. Das kann beispielsweise eine kurze Tonfolge beim positiven bzw. negativen Beenden eines automatisierten Arbeitsschrittes oder das Aufleuchten einer LED beim Start einer Aktion sein. Dem Nutzer wird Feedback gegeben, ohne dass die Interaktion ihn von seiner eigentlichen Tätigkeit ablenkt. [19, 20]

Durch solche eine Systemgestaltung werden automatisierte Prozesse wahrnehmbar, ohne die explizite Aufmerksamkeit des Mitarbeiters zu fordern. Abbildung 3 zeigt ein Beispiel für das CPPS: Durch integrierte Leuchtelemente wird angezeigt, ob die Anlage gerade arbeitet, auf ein Produkt wartet oder inaktiv ist.



Abbildung 3 Beispiel für Calm Design in der Werkstatumgebung: Integrierte Leuchtelemente in der Anlage

1.2.2.3 Multi Device Interaction

Abhängig von der jeweiligen Arbeitsaufgabe bieten die verschiedenen Bediengeräte individuelle Vor- und Nachteile. Für viele Tätigkeiten ist eine gegenseitige Ergänzung der Geräte sinnvoll. Die durch den Nutzer wahrgenommene Bedienschnittstelle setzt sich daher aus verschiedenen Geräten zusammen, z.B. von der Smartwatch über Brille oder Tablet bis zum stationären Bedienpanel. Damit der Mitarbeiter sich auf die eigentliche Tätigkeit und das Umfeld konzentrieren kann, sollten die verschiedenen Geräte innerhalb der Anwendung integriert sein. Das bedeutet einerseits, dass das

Erscheinungsbild der Oberfläche und die elementaren Bedienelemente die Wiedererkennung optisch und funktional fördern. Andererseits ist die Konsistenz der zugrundeliegenden Daten zu beachten, damit der Nutzer auch inhaltlich nicht auf den Wechsel zwischen verschiedenen Geräten achten muss.

Die Anpassung der Darstellung von Inhalten auf verschiedenen Geräten ist im Bereich des Webdesigns unter Begriffen wie „Responsive Design“ oder „Fluid Layout“ bekannt. Dabei werden die Inhalte (meist als Blöcke zusammengefasst) der Größe des Ausgabegeräts (vom Bildschirm bis zum Smartphone) in der Größe und Anordnung angepasst. Diese Gestaltungskonzepte werden genauso auch für industrielle Anwendungen entwickelt werden. Abbildung 4 zeigt, wie sich mit den Geräten einerseits die Informationen bis auf ein einzelnes Signal reduzieren, durch den Geräteverbund sich andererseits ein ganzheitlicher Informationsraum (vgl. Abschnitt 1.2.2.2) bildet.

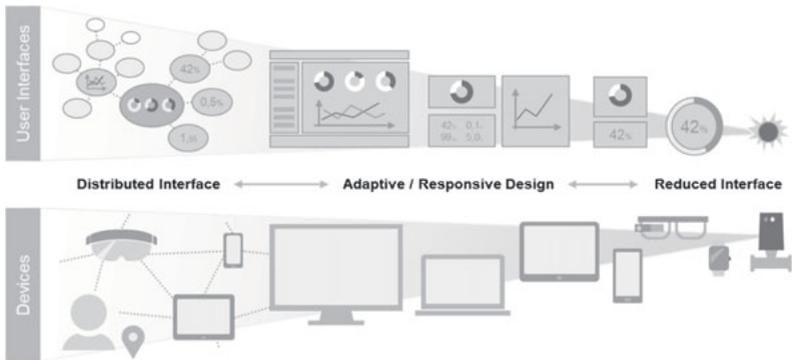


Abbildung 4 Skalierung der Informationsdichte mit den Interaktionsmöglichkeiten der Geräte

1.3 Basisframework für die Mensch-Maschine-Schnittstelle

Um durch die Bedienschnittstelle im vernetzten Produktionssystem eine natürliche Interaktion zu ermöglichen, bei der die digitale und die reale Welt miteinander verschmelzen, müssen einzelne Anwendungen innerhalb des Nutzungskontext verknüpft werden können. Dazu bedarf es eines zugrundeliegenden Frameworks, das verteilte Systeme und Inhalte bedarfsgerecht zu einem Interaktionsraum verbindet. Die Grundlage zur konzeptionellen wie auch technischen Gestaltung bilden folgende Ansätze:

- Das System bildet eine ganzheitliche, aggregierte Bedienschnittstelle aus vernetzten Komponenten.

- Bediengeräte werden analog zur Automatisierungstechnik als Assets im Internet of Things (IoT) betrachtet.
- Die Gestaltungsgrundlage bildet das Paradigma einer losen Kopplung der Dinge und Dienste, d. h. die Anwendungslogik wird vom Bediengerät getrennt, Inhalte werden von der konkreten Darstellung abstrahiert und einzelnen Anwendungen werden bedarfsgerecht orchestriert.

Das Gesamtsystem wird in Anlehnung an den Aufbau wissenschaftsbasierter Assistenzsysteme (vgl. [12]) in drei Ebenen strukturiert (Abbildung 5).

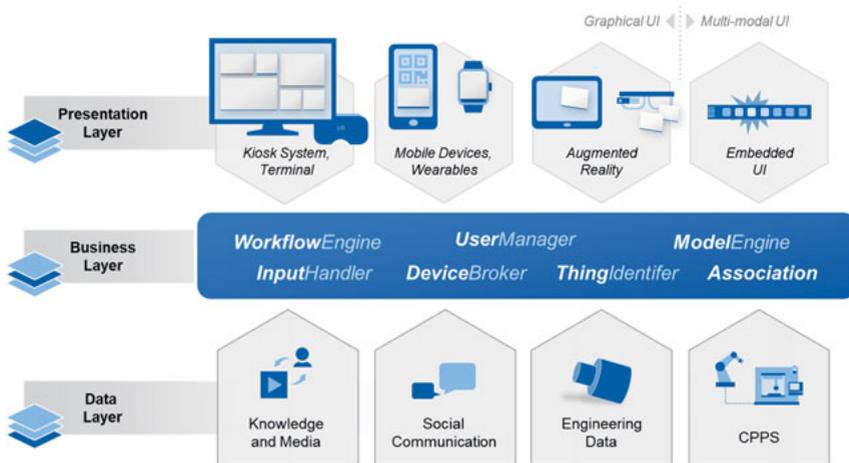


Abbildung 5 Gliederung des Basisframeworks in drei Ebenen: Data Layer, Business Layer und Presentation Layer

1.3.1 Data Layer

Die unterste Ebene beschreibt die Schnittstelle zu den Daten aus der Produktentwicklung, dem CPPS und Wissensressourcen. Dies sind insbesondere Dokumente und Informationen aus der Produktentwicklungsphase und aktuelle Prozessdaten, aber auch der Nutzungskontext und die semantische Beschreibung von Ressourcen. Durch die Digitalisierung des Produktentwicklungsprozesses und der Vernetzung des Produktionssystems sind Daten, Dokumente und Wissen grundsätzlich verfügbar. Die Herausforderung besteht darin, diese dem Nutzer auch einfach zugänglich zu machen und dabei auf die existierenden Schnittstellen aufzusetzen. Dies sind insbesondere etablierte Datenformate zur Beschreibung von strukturierten Daten und Standarddateiformate, sowie Kommunikationsschnittstellen des IoT wie MQTT und REST.

1.3.2 Business Layer

Auf der mittleren Ebene wird die Infrastruktur zur Assoziation von Ressourcen untereinander und mit Kontextinformationen definiert. In Informationsobjekte werden so Gegenständen, Personen, Situationen oder Orten zugewiesen und miteinander verknüpft. Weiterhin wird die Technologie zur Vernetzung einzelner Geräte und Anwendungen zu einem Geräteverbund bzw. einem verteilten Interaktionssystem bereitgestellt. In Abhängigkeit der konkreten Anwendung werden die Dienste für eine spezifische Tätigkeit implementiert oder als übergreifende Basis bereitgestellt. Das Beispiel eines Assistenzsystems wird in Abschnitt 1.4 vorgestellt.

1.3.3 Presentation Layer

Die oberste Ebene umfasst die Visualisierung der Daten und Interaktion mit dem System. Dabei werden die Informationen an verschiedene Artige Ausgabetechnologien in ihrer Darstellung angepasst. Die konkreten Anwendungen können im gesamten Spektrum der Interaktionstechnologien realisiert werden. Damit sie dennoch eine gemeinsame Bedienschnittstelle bilden, wird der Kontext (Workflow, Datenmodelle etc.) aus der jeweiligen Applikation in den Business Layer übertragen. Die Anwendung selbst bindet die Daten über die entsprechenden Dienste ein und überträgt sie in eine konkrete Darstellung.

1.4 Anwendungsbeispiel: Assistenzsystem für manuelle Montagetätigkeiten

Die Unterstützung von Montagetätigkeiten am Arbeitsplatz durch Daten aus der Produktion ist ein häufig nachgefragter Anwendungsfall für die Digitalisierung in der Produktion. Insbesondere Augmented Reality gilt hier als vielversprechende Technologie. Es hat sich allerdings gezeigt, dass neben der konzeptionellen Machbarkeit eines solchen Systems, die tatsächliche Nutzbarkeit einen sehr entscheidenden Einfluss auf die Einsatzfähigkeit eines solchen Assistenzsystems hat. So haben z.B. AR-Brillen viele Potentiale, es fehlt für diese Anwendungsfälle jedoch noch immer an langfristigen einsetzbaren Geräten, die den Anforderungen einer Industrieumgebung gerecht werden.

Da jeder Arbeitsplatz individuelle Besonderheiten und Anforderungen mit sich bringt, gibt es nicht „die eine“ Technologie für eine optimale Informationsbereitstellung. Vielmehr ergibt sich der Mehrwert durch eine Kombination verschiedener Komponenten, die bedarfsgerecht in den Tätigkeitsablauf integriert werden. Zur Anzeige von Montageinformationen am Arbeitsplatz sind dies beispielsweise (Touch-)Display oder Tablet, Head-Mounted-Displays bzw. verschiedene AR-Brillen oder Projektoren. Für die Eingabe gibt es neben Maus und Tastatur auch BarcodeScanner und Kamerasysteme, oder die Sprachinteraktion. Zur Integration der verschiedenen Interaktionselemente

wurden auf Basis des in Abschnitt 1.3 vorgestellten Frameworks innerhalb des Anwendungskontextes die Dienste implementiert:

- **ModelEngine:** Vermittelt zwischen einem oder mehreren Modellen und der Anwendung, indem Produkt- und Prozessdaten für verschiedene Systemelemente vorgehalten werden. Dazu gehört das Abrufen der Dateien oder Datensätze aus verschiedenen Quellen, dem Lesen der Daten aus dem jeweiligen Datenformat und die Extraktion der benötigten Informationen.
- **InputHandler:** Gibt Eingaben von verschiedenen Geräten (z.B. Tastatur, Barcode-Scanner, Spracheingabe) an alle betroffenen Teilnehmer weiter. Die Eingabe wird von der Eingabemethode abstrahiert, verschiedene Methoden können parallel existieren.
- **DeviceBroker:** Kennt die verfügbaren Geräte im Nutzungskontext und vermittelt Inhalte zwischen diesen. Dies ermöglicht es, Inhalte von einem zum anderen Gerät weiterzugegeben.
- **SpacialAgent:** Vermittelt zwischen verschiedenen Referenzräumen und Lokalisierungsmethoden; bildet die Basis für eine Positionsbestimmung von Objekten und der Handhabung von (Teil-)Interaktionsräumen.
- **ContextManager:** Bringt Ressourcen miteinander in Beziehung und beschreibt Relationen zwischen informationstechnischen Elementen (z.B. Daten, Dokumente, Medien), physischen Objekten oder Referenzen (z.B. Orte oder Situationen).
- **UserManager:** Verwaltet Nutzerinformationen zur Zuordnung eines individuellen Nutzungskontextes, sowie zur Organisation von Rollen, Zugriffsberechtigungen oder persönlichen Einstellungen.

Die Dienste stellen notwendige Daten zu Verfügung, mit denen die konkrete Anwendung auf der Brille, dem Tablet oder dem stationären PC angezeigt werden. In der Applikation selbst ist dabei nur die notwendigste Interaktion implementiert, Anwendungsübergreifende Eingaben oder Zustandsänderungen werden an die Dienste weitergegeben und stehen so allen Geräten zur Verfügung. Abbildung 6 zeigt den beschriebenen Aufbau des Kernsystems.

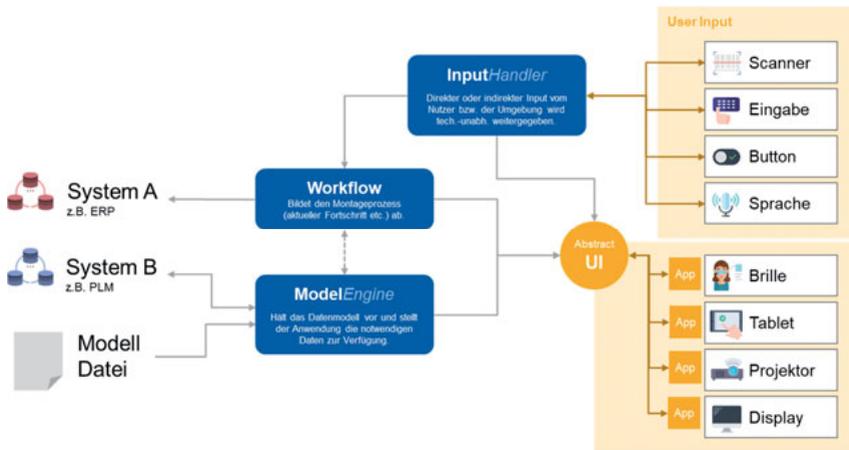


Abbildung 6 Kernelemente für die Anwendung des Frameworks auf die Gestaltung von Assistenzsystemen für Montagetätigkeiten

1.5 Zusammenfassung und Ausblick

Die Rolle des Menschen, die Anforderungen und Bedingungen dazu zur Verfügung stehenden Arbeitsmittel verändern sich unter dem Einfluss des technischen und organisatorischen Paradigmenwechsels nach der Vision Industrie 4.0 maßgeblich. Diesbezüglich entstehen ineinandergreifende konzeptionellen bis zur technischen Ebene, um den Mitarbeiter in diesem Kontext optimal zu befähigen und zu unterstützen. Dieses Basisverständnis der veränderten Bedingungen in der Produktion von Methoden und Konzepten für die intuitive Mensch-Technik-Interaktion, sowie der Gestaltung geeigneter Werkzeuge und (offener) Schnittstellen.

Ein übergreifendes Basisframework ermöglicht es, diese Nutzungskontext über eine Applikation hinaus abzubilden. So lassen sich verschiedene Technologien und Interaktionsformen miteinander kombinieren. Das Mensch-Technik-System lässt sich so den individuellen Bedürfnissen einer Arbeitsaufgabe anpassen.

Die Kernelemente dieses Frameworks wurden bereits auf einige Problemstellungen aus der Industrie angewendet und werden als Demo-Anwendungen getestet. Mit dem Transfer auf weitere Anwendungsfälle wird der Kern um weitere Dienste erweitert werden müssen. Durch die modulare Gestaltung der Dienste kann das Basisframework bedarfsgerecht wachsen und neue Funktionen können kontinuierlich ergänzt werden.

gungen seiner Aufgaben sowie dern sich unter dem Einfluss des s nach der Vision Industrie 4.0 ende Lösungskonzepte von der itarbeiter in diesem Kontext opren auf einem nutzerfokussierten uktionsumgebung, der Entwickensch-Technik-Interaktion, sowie ystemschnittstellen.

n Nutzungskontext über eine Applikation hinaus abzubilden. So lassen sich verschiedene Technologien und Interaktionsformen miteinander kombinieren. Das Mensch-Technik-System lässt sich so den individuellen Bedürfnissen einer Arbeitsaufgabe anpassen.

Die Kernelemente dieses Frameworks wurden bereits auf einige Problemstellungen aus der Industrie angewendet und werden als Demo-Anwendungen getestet. Mit dem Transfer auf weitere Anwendungsfälle wird der Kern um weitere Dienste erweitert werden müssen. Durch die modulare Gestaltung der Dienste kann das Basisframework bedarfsgerecht wachsen und neue Funktionen können kontinuierlich ergänzt werden.

1.6 Literatur

- [1] H. Kagermann et al., Hg., „Smart Service Welt: Umsetzungsempfehlungen für das Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft“, acatech – Deutsche Akademie der Technikwissenschaften e.V., Mrz. 2014. Zugriff am: Mai. 22 2017.
- [2] H. Kagermann, W. Wahlster und J. Helbig, Hg., „Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0: Abschlussbericht des Arbeitskreises Industrie 4.0“, acatech, 2013.
- [3] M. Peissner und C. Hipp, „Potenziale der Mensch-Technik Interaktion für die effiziente und vernetzte Produktion von morgen“, Stuttgart art, 2013.
- [4] D. Gorecky, M. Schmitt und M. Loskyll, „Mensch-Maschine-Interaktion im Industrie 4.0-Zeitalter“ in Industrie 4.0 in Produktion, Automatisierung und Logistik, T. Bauernhansl, M. ten Hompel und B. Vogel-Heuser, Hg., Wiesbaden: Springer Fachmedien Wiesbaden, 2014, S. 525–542.
- [5] K. S. Bibby, F. Margulies, J. E. Rijnsdorp, R.M. J. Withers und I. M. Makarov, „Man's Role in Control Systems“, IFAC Proceedings Volumes, Jg. 8, Nr. 1, S. 664–683, 1975.
- [6] D. Zühlke, Nutzergerechte Entwicklung von Mensch-Maschine-Systemen: Userware-Engineering für technische Systeme. Dordrecht: Springer, 2012.
- [7] M. Weiser, „The Computer for the 21st Century“, Scientific American, September, 1991.
- [8] M. Frese und F. C. Brodbeck, Computer in Büro und Verwaltung: Psychologisches Wissen für die Praxis. Berlin, Heidelberg: Springer Berlin Heidelberg, 1989.
- [9] M. Schmitt und D. Zühlke, „Smartphones und Tablets in der industriellen Produktion: Nutzerfreundliche Bedienung“ (de), atp edition - Automatisierungstechnische Praxis (ATP) 03/2013, Nr. 03, S. 58–65, 2013.
- [10] M. Schmitt, G. Meixner, D. Gorecky, M. Seissler und M. Loskyll, „Mobile Interaction Technologies in the Factory of the Future“, IFAC Proceedings Volumes, Jg. 46, Nr. 15, S. 536–542, <http://www.sciencedirect.com/science/article/pii/S1474667016331159>, 2013.
- [11] P. Schaumlöffel, M. Talha, D. Gorecky und G. Meixner, „Augmented Reality Applications For Future Manufacturing“ in Proceedings of the International Conference on Manufacturing Science and Education. International Conference on Manufacturing Science and Education (MSE-11), 5th, June 2-5, Sibiu, Romania 2011.
- [12] D. Gorecky, Entwicklung einer Methodik zur informationstechnischen Integration von virtuellem Training innerhalb der Digitalen Fabrik. Dissertation. Kaiserslautern: Techn. Univ, 2014.

- [13] T. Salter, Introducing ARCore: Augmented Reality at Android Science (Online) Verfügbar unter: <https://www.youtube.com/watch?v=rFbcOGuDMPk>.
- [14] P. Milgram, H. Takemura, A. Utsumi und F. Kishino, „Augmented reality: A class of displays on the reality-virtuality continuum“, Telemanipulator and Telepresence Technologies, Jg. 2351, 1994.
- [15] R. T. Azuma, „A Survey of Augmented Reality“, Presence: Teleoperators and Virtual Environments, Jg. 6, Nr. 4, S. 355–385, 1997.
- [16] Ergonomie der Mensch-System-Interaktion, 9241.
- [17] Gebrauchstaugliche Gestaltung von Benutzungsschnittstellen für technische Anlagen, VDI/VDE 3850, 2014.
- [18] D. Görlich, Laufzeit-Adaption von Benutzungsschnittstellen für Ambient-Intelligence-Umgebungen mittels raumbasierter Benutzungsmethoden. Dissertation. Kaiserslautern: Techn. Univ, 2009.
- [19] A. Case, Calm technology: Principles and patterns for non-intrusive design. Sebastopol, CA: O'Reilly Media, 2015.
- [20] M. Weiser und J. S. Brown, „The Coming Age of Calm Technology“, in: Beyond Calculation, P. J. Denning und R. M. Metcalfe, Hg., New York, NY: Springer New York, 1997, S. 75–85.

2 Prozessüberwachung und -regelung für effiziente Drehprozesse

T. Holznagel, E. Uhlmann

2.1 Einleitung

Wie in Abbildung 1 schematisch dargestellt, wirken verschiedene Störgrößen auf den Drehprozess. Diese beeinflussen maßgeblich die Qualitätsmerkmale Oberflächengüte und Aufmaß des gefertigten Bauteils, aber auch die Werkzeugbelastung, die Werkzeugstandzeit und die Prozesseffizienz. Aufgabe von Prozessregelungen ist es, den Einfluss dieser Störgrößen auf den Drehprozess durch Anpassung von Schnittparametern zu kompensieren. Prozessüberwachungen und -regelungen beziehungsweise sogenannte adaptive Regelungen von Drehprozessen sind seit längerem Gegenstand wissenschaftlicher Forschung und in begrenztem Maße auch industrieller Anwendung. Da in den meisten Fällen lediglich Schnittparameter, nicht aber die Zeitkonstanten oder der Verstärkungsfaktor der Regelkreise durch die Systeme angepasst werden, ist die Bezeichnung adaptive Regelung für die meisten in den folgenden Kapiteln dargestellten Systeme im regelungstechnischen Sinne nicht zureichend [1, 2]. Deshalb wird in dieser Arbeit der allgemeinere Begriff Prozessregelung verwendet.

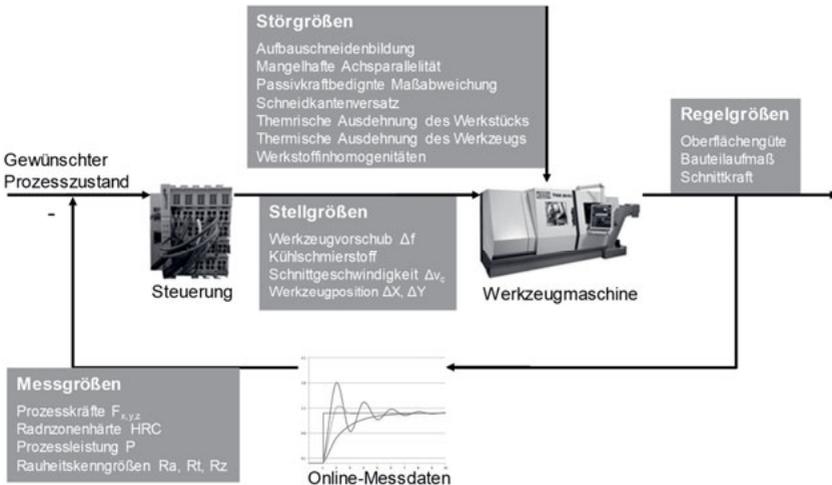


Abbildung 1 Prinzipieller Aufbau einer Prozessregelung für Drehprozesse mit beispielhaften Mess-, Regel-, Stell- und Störgrößen

Da beispielsweise der Schneidkantenversatz SV, die der Kolkverschleiß während eines Zerspanprozesses werden für nicht geregelte Drehprozesse konservativ Werkzeugwechsel nach geringer Schnittzeit durchgeführt in den geforderten Grenzen zu halten. Hier können die Erhöhung der Werkzeugstandzeit sowie der Effizienz bei gleichbleibender geregelter Oberflächengüte und Maßhaltigkeit erwirken [2].

In älteren Untersuchungen zu Prozessregelungen werden zur Umsetzung der online-Adaptierbarkeit von Schnittparametern an NC-Werkzeugmaschinen beispielsweise die Antriebsregler der Vorschubachsen beziehungsweise der Spindelachse durch externe Hardware angesteuert [3, 4]. Moderne Werkzeugmaschinensteuerungen bieten über Kommunikationsschnittstellen die Möglichkeit, maschineninterne Messdaten wie die Spindelleistung online auszulesen und Schnittparameter auf der numerischen Steuerung online und in Echtzeit zu schreiben. Zur Realisierung einer solchen Schnittstelle kann beispielsweise das Machine to Machine M2M-Kommunikationsprotokoll OPEN PLATFORM COMMUNICATIONS UNIFIED ARCHITECTURE (OPC UA) beziehungsweise OPC UA TIME SENSITIVE NETWORKING (TSN) der OPC FOUNDATION, Scottsdale, USA verwendet werden [5].

2.2 Forschungsarbeiten zur Prozessregelung für das Drehen

Prozessregelungen sind seit Anfang der 1970er Jahre Gegenstand intensiver wissenschaftlicher Forschung. Hierbei wurden unterschiedliche Ziele angestrebt, beziehungsweise Kenngrößen verwendet und Sensor-Systeme in die Werkzeugmaschinen integriert. Ein Überblick zu Prozessregelungen wird in Abbildung 2 gegeben.

Des Weiteren können die entwickelten Systeme in Adaptive Control with Constraints (ACC), Adaptive Control Optimization (ACO) und Geometric Adaptive Control (GAC) unterteilt werden [6]. Eine kurze Erläuterung zu den Begriffsdefinitionen sowie ein Überblick über die vorliegenden wissenschaftlichen Arbeiten für die unterschiedlichen Arten von Prozessregelungen wird im Folgenden gegeben.



Abbildung 2 Überblicksdarstellung zu Prozessregelungen nach WECK [6]

ACC's kompensieren den Einfluss von Werkzeugverschleiß und Werkstoffinhomogenitäten auf die Belastung, welche auf das Drehwerkzeug wirkt. Damit kann die Werkzeugbelastung weitestgehend konstant gehalten werden, wodurch Werkzeugbruch vermieden wird und sich das Zeitspannvolumen MRR erhöhen lässt. So beschreibt MASORY die Messung der Schnittkraft F_c während der Zerspaltung und die Anpassung des Werkzeugvorschubs f durch einen simplen Kennlinienregler mit konstanter Verstärkung [7]. In einer weiterführenden Arbeit wurde die Reglerverstärkung gemäß dem Prozesszustand angepasst, um Reglerinstabilitäten zu vermeiden und Prozessabweichungen schneller auszuregulieren [4]. Ein ähnliches System wurde von KESHMEND UND PAK als Model Reference Adaptive Control (MRAC) entwickelt [8]. BATAVA ET AL. detektieren aus einer Messung von Spindelleistung, Vibration, Körperschall und Luftschall über einen Fuzzy-Regler Prozessinstabilitäten beziehungsweise regeneratives Rattern und hielten die Spindelleistung P durch Anpassung des Werkzeugvorschubs f konstant [9]. In einer weiterführenden Arbeit wurde der Aufbau um eine Spanlängenerkennung und -regelung und um die Stellgröße Schnittgeschwindigkeit erweitert [10].

ACO's werden verwendet, um eine definierte Qualitätsfunktion während des Drehprozesses durch online-Schnittparameteranpassung zu optimieren. KEYAMA ET AL. optimierten beispielsweise unter Messung der Prozesskräfte und Abschätzung des Werkzeugverschleißes und der Werkzeugverschleißrate durch Anpassung der Schnittgeschwindigkeit v_c die Werkzeugstandzeit auf einen definierten Wert [3]. JENSEL realisierte die direkte Verschleißmessung mittels optischer Verfahren und eine Abschätzung der noch verbleibenden Werkzeugstandzeit zur Maximierung derselben [11]. AZOUZI UND GUILLOT entwickelten ein neuronales Netz, welches durch Messung von Prozesskräften und -vibrationen optimierte Schnittgeschwindigkeiten und Werk-

zeugvorschübe f bestimmt. Die Schnittwerte wurden zur Maximierung eines aus Zeitspannvolumen MRR, Oberflächenqualität R_a und Bauteilabweichung zusammengesetzten multikriteriellen Optimierungsproblems durch online-Optimierung einer offline trainierten Führungsgrößentrajektorie angepasst [12]. JESTER UND HEIGEL maximierten mithilfe eines Prozessmodells das Zeitspannvolumen MRR unter Beibehaltung der Randbedingungen Prozesskraft, Oberflächengüte und Standvolumen durch Messung der Spindellast P [13]. ORABY UND ALASKARY maximierten das Zeitspannvolumen unter der Nebenbedingung einer möglichst geringen Werkzeugverschleißrate durch Anpassung des Werkzeugvorschubs f und der Schnittgeschwindigkeit durch Messung der Prozesskräfte und -leistungen [14].

GAC's werden verwendet, um den Einfluss von Prozessschwankungen und Werkzeugverschleiß auf die gefertigte Oberflächengüte und das resultierende Bauteilmaß zu minimieren [15]. SHIRAIISHI UND SATO verwendeten hierfür eine Lasermessung, um über Anpassung der Werkzeugposition und mithilfe eines zweiten Werkzeugs sowohl Durchmesser als auch Oberflächengüte des Werkstücks zu regeln [16]. CHOUDHURY UND RAMESH nutzten einen photoelektrischen Wegsensor zur Messung der gefertigten Bauteilgeometrie. Aufgrund der online-Messungen wurde die Werkzeugposition in X-Richtung zur Kompensation des Werkzeugverschleißes nachgeführt [17]. COKER UND SHIN ermittelten die resultierende Oberflächengüte aus einer Ultraschallmessung und regelten die Oberflächengüte mit der Stellgröße Werkzeugvorschub f [18]. ZHANG ET AL. bestimmten die resultierende Oberflächengüte aus Vibrationsmessdaten mithilfe von verschiedenen künstlichen neuronalen Netzen und passeten den Werkzeugvorschub f entsprechend an [19]. GARCIA-PLACA ET AL. verwendeten Signalkennwerte wie Mittelwerte, Standardabweichung und Frequenz der Maximalamplitude einer Prozesskraftmessung, um über ein Prozessmodell die resultierende Oberflächenrauheit abzuleiten und den Werkzeugvorschub f mithilfe eines Kennfeldreglers anzupassen [20].

Derzeitige Forschungsbemühungen beschäftigen sich mit der definierten Einstellung von Randzoneeigenschaften durch Prozessregelungen. Hier kommen zum Beispiel die Randzonenhärte, Mikro- und Makroelastizitäten oder Gefügezusammensetzung als Zielgröße in Betracht [21, 22, 23]. Da die genannten Randzoneeigenschaften mit derzeitiger Messtechnik nicht ohne weiteres direkt während der Zerspanung gemessen werden können, wird die Entwicklung von Softsensoren beziehungsweise die indirekte Ermittlung der Randzoneeigenschaften durch Messung von Prozesskräften, Prozesstemperaturen, Prozessvibrationen, Körperschall und weiteren Größen angestrebt. Die Möglichkeit, Werkstoffeigenschaften in der Randzone während des Drehprozesses definiert einzustellen, kann nachfolgende stoffeigenchaftsändernde Prozesse wie das Randzonenhärten überflüssig machen und damit Prozessketten und Fertigungszeiten verkürzen [24, 25].

2.3 Prozessregelungen in der industriellen Anwendung

Es existieren am Markt Hard- und Softwarelösungen, mit denen eine Maschinensteuerung zur Realisierung von Prozessregelungen für Drehprozesse erweitert werden kann. Solche Erweiterungen wurden beispielsweise für die Werkzeugmaschinensteuerungen TNC der Firma HEIDENHAIN GMBH, Traunreut, und Sinumerik 840D sl der Firma SIEMENS AG, Berlin, aber auch für FANUC, OKUMA sowie BOSCH-REXROTH-Steuerungen entwickelt. So überwacht die Vorschubkontrolle ACM der Firma OMATIVE SYSTEMS GMBH, Villingen-Schwenningen, sowie iCut der Firma COMARA GMBH, St. Georgen, in-process die Spindellast. Die Messdaten werden verwendet, um eine geeignete Anpassung des Werkzeugvorschubs abzuleiten, welche reduzierte Bearbeitungszeiten bei gleichzeitigem Schutz vor Werkzeug-, Werkstück- und Spindelüberlastung garantieren soll [26, 27]. Ein ähnliches System, welches um diverse Überwachungsfunktionen erweitert werden kann, wird von der Firma MAFON ENGINEERING INC., Wells, USA für verschiedenste Steuerungstypen angeboten [28]. Die Prozessüberwachung CTM der Firma MARPOSS MONITORING SOLUTIONS GMBH, Egestorf, verwendet zusätzlich die in der Werkzeugmaschinensteuerung vorliegenden Antriebsdaten und gegebenenfalls externe zusätzliche Sensoren zur Bestimmung des Prozesszustandes [29].

Obwohl ein nicht zu vernachlässigender Forschungsaufwand bezüglich diverser, oben erläuteter Prozessregelungen, durchgeführt wurde, konnten sich diese nicht in der breiten industriellen Anwendung durchsetzen. So ist beispielsweise zu beachten, dass eine Änderung von Schnittparametern während der Bearbeitung Schwankungen der generierten Werkstoffeigenschaften wie beispielsweise der Randzonenhärte induzieren kann. Dies wiederum kann zu starken und schwer vorhersehbaren Schwankungen der Bauteillebensdauer führen [30]. Aus technischer Perspektive stellen regelungstechnische Stabilitätskriterien für adaptive Reglersyntheseverfahren teilweise bis heute nicht bekannt sind und die Regelkreisstabilität daher zu meist lediglich anhand von Simulationsstudien untersucht werden kann. Auch stellen die Identifikation eines Regelstreckenmodells und der Adaptionvorschrift, die notwendige Konvergenz der Reglerparameter sowie die Untersuchung des Störgrößenverhaltens weitere Herausforderungen dar [31]. Fehlende Robustheit der Sensorik gegen Kühlschmierstoff und Spanflug, regelungstechnische Stabilitätsprobleme und die hohe Komplexität der systemtheoretischen Modellierung des Zerspanprozesses für den jeweils vorliegenden Werk- beziehungsweise Schneidstoff werden ebenfalls als Gründe genannt [32, 33, 34, 2]. Weiterhin müssen durch die Systeme spezielle Prozesszustände wie Aufbauschneidenbildung und Rattern sowie Werkzeugbruch und die Bildung von Fließspan erkannt und kompensiert beziehungsweise verhindert werden [2]. Auch die fehlende Möglichkeit, Werkzeugverschleiß während der Bearbeitung präzise zu messen oder aus indirekten Messungen abzuleiten, wird als Hindernis

genannt [1, 2]. Die wichtigsten Störgrößen während der Zerspang beziehungsweise Herausforderungen für Prozessregelungen und ihre Einflüsse auf den Drehprozess werden im Folgenden dargestellt.

2.4 Störgrößen von Prozessregelungen in Drehprozess en

Verschiedene Störgrößen wirken während der Zerspang, deren Einfluss auf unter anderem Werkzeugstandzeit, Bauteilmaß und resultierende Oberflächengüte näher erläutert werden soll. Außerdem werden zu den Störgrößen beziehungsweise den Störgrößeneinflüssen mögliche Gegenmaßnahmen für den Drehprozess genannt. Diese können als Stellgrößen für Prozessregelungen Verwendung finden.

2.4.1 Aufbauschneidenbildung

Aufbauschneidenbildung bezeichnet die Ablagerung und Aufschweißung von Werkstoffpartikeln auf der Spanfläche des Drehwerkzeugs. Die damit einhergehende Veränderung der Werkzeuggeometrie führt zu höheren resultierenden Schnitttiefen und damit zu unerwünschten Maßabweichungen am Werkstück. Des Weiteren führt die Aufbauschneidenbildung zu einem stark positiven effektiven Spanwinkel und damit in der Regel zu geringeren Prozesskräften beziehungsweise erleichteter Spanbildung [35]. Außerdem kann die Werkzeugverschleißrate, vor allem bezüglich des Kolkverschleißes reduziert werden, da die Aufbauschneide als Schutzschicht wirkt [36]. Aufbauschneiden sind in der Regel unerwünscht, da sich die Werkzeuggeometrie und die Zerspangbedingungen durch zyklisches Aufbauen und späteres Abreißen der Aufbauschneide während des Zerspangprozesses nicht reproduzierbar verändert, was sich nachteilig auf die Oberflächengüte auswirkt [37]. Eine Aufbauschneide am Drehwerkzeug sowie die resultierenden Zerspangbedingungen sind in Abbildung 3 dargestellt.

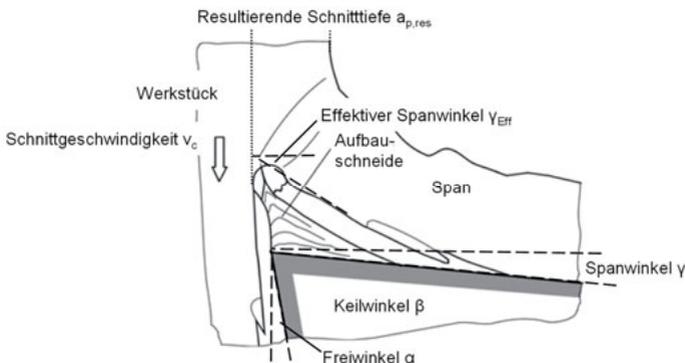


Abbildung 3 Aufbauschneidenbildung am Schneidkeil nach DENKENA [35]

Geeignete Gegenmaßnahmen zur Aufbauschneidenbildung sind die Erhöhung der Schnittgeschwindigkeit v_c , die Anpassung der Kühlschmierung, die Erhöhung des Spanungsquerschnitts sowie des Spanwinkels γ [38]. Zur Kompensation des Einflusses einer Aufbauschneide auf den Drehprozess kann die Werkzeugzustellung in X-Richtung angepasst werden.

2.4.2 Mangelhafte Achsparallelität

Aufgrund von Achskollisionen beziehungsweise mangelhafter Einmessung bei der Inbetriebnahme kann es zu Positionierabweichungen der Werkzeugachsen in X, Y und Z-Richtung kommen. Diese Abweichungen von der ideal achsparallelen Ausrichtung der Vorschubachse haben einen direkten Einfluss auf die gefertigte Bauteilgeometrie in Abhängigkeit von der Position des Werkzeuges in Z-Richtung. Mögliche Abweichungen von der ideal achsparallelen Vorschubachse sind in Abbildung 4 dargestellt.

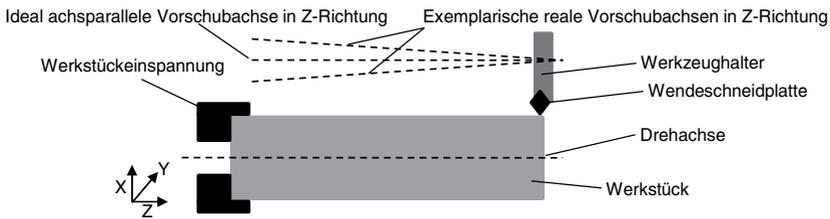


Abbildung 4 Mangelhafte Achsparallelität von Vorschubachse und Drehachse

Mangelhafte Achsparallelität stellt einen systematischen Fehler im Bearbeitungsprozess dar. Ihr Einfluss auf das Bearbeitungsergebnis kann durch geeignete Kompensation der Werkzeugposition in X- und Y-Richtung minimiert werden [39]. Wird Messtechnik auf der Vorschubachse zur online-Messung und Realisierung einer Prozessregelung synchron zum Werkzeugverfahren, ist der Einfluss der mangelhaften Achsparallelität auf das Messergebnis zu bestimmen und gegebenenfalls durch geeignete Kalibration zu kompensieren.

2.4.3 Passivkraftbedingte Maßabweichung

Während der Zerspaltung wirken Zerspankräfte auf den Schneidkeil, die sich in Schnittkraft F_c , Vorschubkraft F_f und Passivkraft F_p aufteilen lassen. Die Passivkraft wirkt dabei senkrecht zur Schnitt- und Vorschubrichtung und drückt das Werkzeug vom Werkstück weg. Aufgrund dieser Abdrängung kann es zu Maßabweichungen gemäß der Steifigkeit c_x der Werkzeugeinspannung kommen. Die Maßabweichung ΔD kann, lineares Systemverhalten vorausgesetzt, über (1) abgeschätzt werden.

$$\Delta D = 2 \frac{F_P}{c_x} \quad (1)$$

Aufgrund von steigendem Werkzeugverschleiß nimmt die Passivkraft F_P in der Regel zu. Die Abdrängung hat eine Verringerung der Schnitttiefe zur Folge. Der Zusammenhang von Verschleißmarkenbreite VB , Passivkraft F_P und Maßabweichung δ_x ist beispielhaft für unterschiedliche Achssteifigkeiten c_x in Abbildung 5 dargestellt.

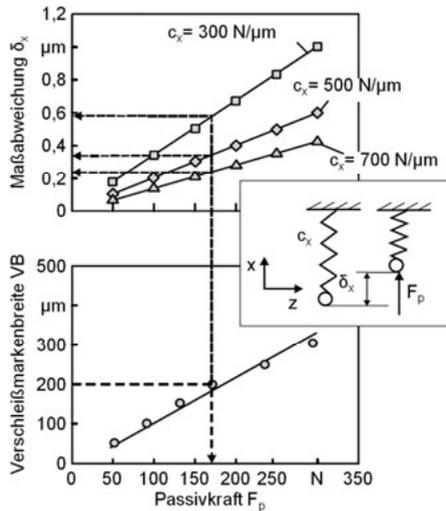


Abbildung 5 Zusammenhang von Passivkraft F_P und Maßabweichung δ_x während der Drehbearbeitung nach DENKENA [35]

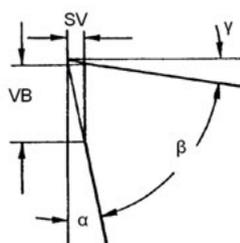
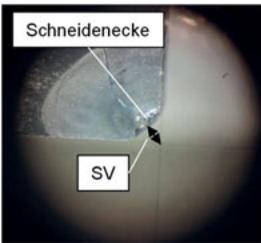
Eine geeignete Maßnahme zur Kompensation des Einflusses der passivkraftbedingten Maßabweichung ist die Adaption der Werkzeugposition in X-Richtung. Zur Verringerung der Passivkraft F_P kann beispielsweise ein stark positiver Spanwinkel γ gewählt werden [40].

2.4.4 Verschleißmarkenbreite VB und Schneidkantenversatz SV

Aufgrund von abrasivem Verschleiß an der Freifläche des Werkzeuges kommt es während des Drehprozesses zu einem zunehmenden Schneidkantenversatz SV . In konventionellen Drehprozessen werden Drehwerkzeuge meist bis zu einer bestimmten Verschleißmarkenbreite VB genutzt. Diese Verschleißmarkenbreite dient als Zeitkriterium und wird beispielsweise zu $VB = 200 \mu\text{m}$ festgelegt [40]. Gemäß den geometrischen Zusammenhängen am Drehwerkzeug lässt sich der Schneidkantenversatz SV mit dem Freiwinkel α , dem Spanwinkel γ und der Verschleißmarkenbreite VB durch (2) wie folgt bestimmen [41].

$$SV = VB \frac{\tan \alpha}{1 - \tan \alpha \tan \gamma} \tag{2}$$

Somit ergibt sich beispielsweise für eine SPHN-Wendeschneidplatte mit einem Freiwinkel $\alpha = 11^\circ$, einem Spanwinkel $\gamma = 0^\circ$ und einer Verschleißmarkenbreite $VB = 200 \mu\text{m}$ ein Schneidkantenversatz $SV = 38,9 \mu\text{m}$. Des Weiteren hat die zunehmende Verschleißmarkenbreite VB einen Einfluss auf die resultierende Bauteilrauheit. Der geometrische Zusammenhang von Schneidkantenversatz und Verschleißmarkenbreite am Drehwerkzeug ist in Abbildung 6 Geometrische Zusammenhänge am Drehwerkzeug nach PAUKSCH [41] dargestellt.



- α Freiwinkel
- β Keilwinkel
- γ Spanwinkel
- SV Schneidkantenversatz
- VB Verschleißmarkenbreite

Abbildung 6 Geometrische Zusammenhänge am Drehwerkzeug nach PAUKSCH [41]

Die Kompensation des Einflusses von Werkzeugverschleiß auf die Bauteilrauheit kann durch eine Anpassung des Werkzeugvorschubs f realisiert werden. Zur Kompensation von SV ist die Werkzeugposition in X-Richtung nachzustellen.

2.4.5 Thermische Ausdehnung des Werkzeugs

Die Wirkleistung beim Drehen P_e wird fast vollständig in Wärme umgesetzt. Mit den Schnittparametern Schnittgeschwindigkeit v_c und Vorschubgeschwindigkeit v_f sowie den Prozesskräften Schnittkraft F_c und Vorschubkraft F_f kann die Wirkleistung P_e nach (3) abgeschätzt werden [35].

$$P_e = P_c + P_f = F_c v_c + F_f v_f \tag{3}$$

Bei der Trockenzerspannung wird die Wärme über Span, Werkstück und Werkzeug aus der Zerspanzone geleitet, wobei bei der Nasszerspannung noch der Wärmeabtransport durch den Kühlschmierstoff hinzukommt. Die resultierende Verteilung der Wärmeströme in Span, Werkstück, Kühlschmierstoff und Werkzeug ist abhängig von den Schnittparametern und Werkstoff-/Schneidstoffeigenschaften und kann über Wärme-

verteilungskoeffizienten grob analytisch abgeschätzt werden [42, 43, 44, 45]. Die Aufteilung der im Zerspanprozess umgesetzten Wirkleistung in Wärmeströme, die über das Werkstück, den Span und das Werkzeug abgeführt werden, ist in Abbildung 7 dargestellt.

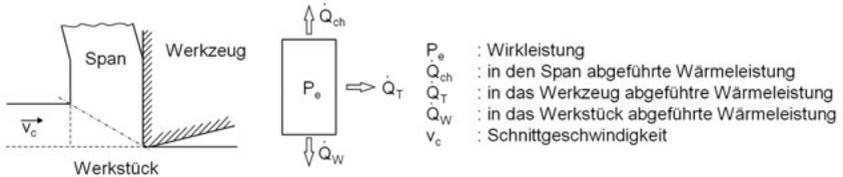


Abbildung 7 Aufteilung der Wirkleistung in über Werkstück, den Span und das Werkzeug abgeführte Wärmeströme nach DENKENA [35]

Aufgrund des Wärmestroms in das Werkzeug \dot{Q}_T kommt es zu einer Aufheizung und damit entsprechender thermischer Ausdehnung Δl des Werkzeugs. Bei einer Ausgangslänge des Drehwerkzeugs von $l_0 = 12$ mm kann die thermische Ausdehnung Δl mit dem thermischen Ausdehnungskoeffizienten von Wolframcarbid $\alpha_{WC} = 7,3 \cdot 10^{-6} \text{ K}^{-1}$ und einer konservativ angenommenen Aufheizung der gesamten Wendeschneidplatte um $\Delta T = 200$ K nach (4) berechnet werden [35].

$$\Delta l = l_0 \alpha \Delta T = 17,5 \text{ } \mu\text{m} \quad (4)$$

Die Aufheizung des Werkzeugs kann durch den Einsatz von Kühlschmierstoff und die Anpassung von Schnittparametern wie Schnittgeschwindigkeit und Werkzeugvorschub f verringert werden. Der Einfluss der thermischen Werkzeugausdehnung auf den Drehprozess kann durch Adaption der Werkzeugposition in X-Richtung kompensiert werden.

2.5 Versuchsaufbau der entwickelten neuartigen Prozessregelung basierend auf online-Lasermessung

Der Neuheitswert der entwickelten Prozessregelung besteht in der direkten Messung der resultierenden Oberflächengüte und des Bauteilmaßes mittels Laser. Somit ist keine gesonderte Kalibration der Prozessregelung für unterschiedliche Werk- beziehungsweise Schneidstoffe notwendig. Auch konnte eine prozesssichere online-Messung trotz Kühlschmierstoffeinsatz und Spanflug realisiert werden. Zur Beschreibung der für diese Arbeit entwickelten Prozessregelung basierend auf online-Lasermessung soll zunächst die Auswahl der Mess- und Stellgrößen dargelegt werden. Darauf aufbauend werden die verwendete Mess- und Steuerungstechnik sowie der Versuchsauf-

bau und dessen Kalibration erläutert. Abschließend werden die Prozessverbesserungen, die durch Anwendung der entwickelten Prozessregelung erzielt wurden präsentiert und diskutiert.

2.5.1 Messtechnik, Steuerungstechnik und Werkzeugmaschine

Die Experimente zur Entwicklung und Validierung von Prozessregelungen wurden auf einem CNC-Drehzentrum TNX 65 der Firma TRAUB GMBH & Co. KG, Reichenbach an der Fils, durchgeführt. Dabei wurde der Edelstahl X 5CrNi18-10 und 42CrMo4+QT mit unbeschichteten Wendeschneidplatten SPHN120108 aus Hartmetall vom Grad K10 in einem Werkzeughalter CSBPL2525M12 zerspannt. Oberflächengüte und Bauteildimensionen wurden in-process mit dem Lasertriangulationssensor LK-H082 der Firma KEYENCE CORPORATION, Osaka, Japan gemessen. Der Versuchsaufbau ist in Abbildung 8 dargestellt.

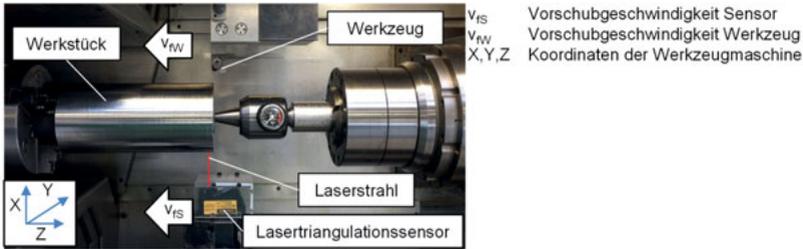


Abbildung 8 Aufbau der Messtechnik in der Werkzeugmaschine Traub TNX 65 für die Prozessregelung

Zum Schutz vor Spänen und dem Eindringen von Kühlschmierstoff (KSS) wird der Laser eingehaust. Um den Messpunkt trotz Nassbearbeitung von KSS freizuhalten, werden mehrere Düsen mit 8 bar Druckluft eingesetzt. Die Sensoreinhausung, der Laser, das Werkstück, der Messpunkt und die Druckluftzuführung ist in Abbildung 9 dargestellt.

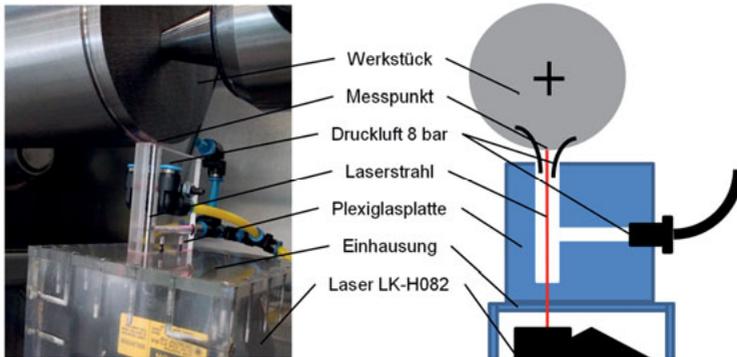


Abbildung 9 Einhausung des Lasertriangulationssensors LK-H082 zum Schutz gegen Spanflug und KSS

Die Vorschubgeschwindigkeit des Sensors v_{is} wird dabei zur Aufzeichnung einer Höhenlinie und späteren Ableitung der Rauheitskenngrößen auf dem Werkstück doppelt so hoch wie die Vorschubgeschwindigkeit des Werkzeuges gewählt. Zur Sicherstellung einer robusten Messung wird ein Medianfilter mit einer Kernelgröße von 31 verwendet. Alle Messdaten wurden mit einem Industrie-PC C6525-0050 der Firma BECKHOFF GMBH, Verl. mit einer Abtastrate von 1 kHz aufgezeichnet. Die Bauteilrauheit wurde nach jedem Schnitt mit dem taktilen Rauheitsmessgerät Etamicd45 der Firma JENOPTIK AG, Jena mit einer Tastspitze mit $2\ \mu\text{m}$ Tastspitzenradius und einer Grenzwellenlänge von $\lambda_c = 0,25\ \text{mm}$ vermessen. Der Werkzeugverschleiß wurde mit dem Mikroskop Edge Digital AM7915MZT der Firma ANMO ELECTRONICS CORPORATION, New Taipei City, Taiwan an Schneidkante, Hauptfreifläche und Nebenfriefläche nach jedem Einzelprozess vermessen. Über eine Datenschnittstelle konnte das Schreiben von Vorschub- und Werkzeugpositionsoffsets mithilfe von 16-Bit Signalen der Beckhoffsteuerung in die Steuerung der Werkzeugmaschine realisiert werden. Diese können somit während der Ausführung von CNC-Programmen online angepasst werden. Auch das Triggern der Lasermessung durch die Werkzeugmaschinensteuerung lässt sich so realisieren.

2.5.2 Kalibration der Lasermessung

Die Messwerte des Lasertriangulationssensors werden von der Keyence-Steuerung GK-G5001PV als $\pm 10\ \text{V}$ -Signal in 24 Bit von der 2-Kanal-Analog-Eingangsklemme EL3602 der Firma Beckhoff digitalisiert. Zur Kalibration des Messsignals wird der Laser zunächst auf die Durchmesserlinie des Werkstücks positioniert und anschließend für unterschiedliche Abstände die Lasermessung mit dem Koordinatensystem der Werkzeugmaschine abgeglichen. Dieses Vorgehen zur Kalibration des Lasers ist in Abbildung 10 dargestellt.

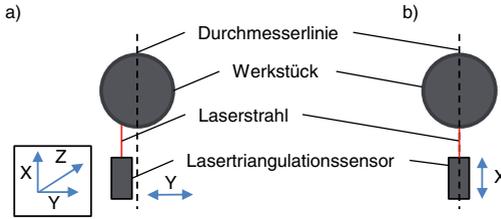


Abbildung 10 Kalibrierung der Lasermessung; a) Positionierung des Lasers auf der Durchmesserlinie des Werkstücks; b) Anfahren von mehreren Abständen zum Werkstück durch Verfahren der Laserpositionierung in X-Richtung

Durch Referenzmessungen mit einer Koordinatenmessmaschine Prismo 7 Vast HTG der Firma CARL ZEISS AG, Jena mehrerer Wellendurchmesser konnte eine Abweichung von der idealen Achsparallelität der Laservorschubachse von maximal $9\mu\text{m}$ festgestellt werden. Die Abweichung von Laser- und taktiler Durchmessermessung ist in Abbildung 11 dargestellt.

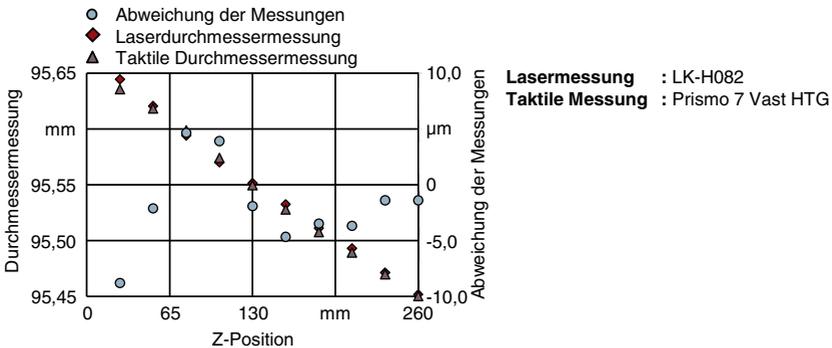


Abbildung 11 Abweichung zwischen taktiler Durchmessermessung und Laserdurchmessermessung

2.5.3 Kalibrierung der Rauheitsmessung

Aus den Messwerten der Lasermessung wird während des Drehprozesses ein Höhenprofil konstruiert, aus dem sich die Rauheitskenngrößen R_a , R_t und R_z ableiten lassen. Zur Kalibrierung der Lasermessung wurden daher eine Vielzahl unterschiedlichen Rauheiten durch Verwendung verschiedener Schnittparameter gefertigt und die Oberflächengüten online mit dem Laser sowie offline mit dem taktilen Rauheitsmessgerät vermessen. Aus den Korrelationsgraphen lassen sich mit Hilfe linearer Regression Kalib-

rationsgleichungen ableiten, die einen Zusammenhang zwischen taktiler und online-Rauheitsmessung herstellen. Die Korrelationsgraphen für die Rauheitskenngrößen Ra, Rt und Rz sind in Abbildung 12 dargestellt.

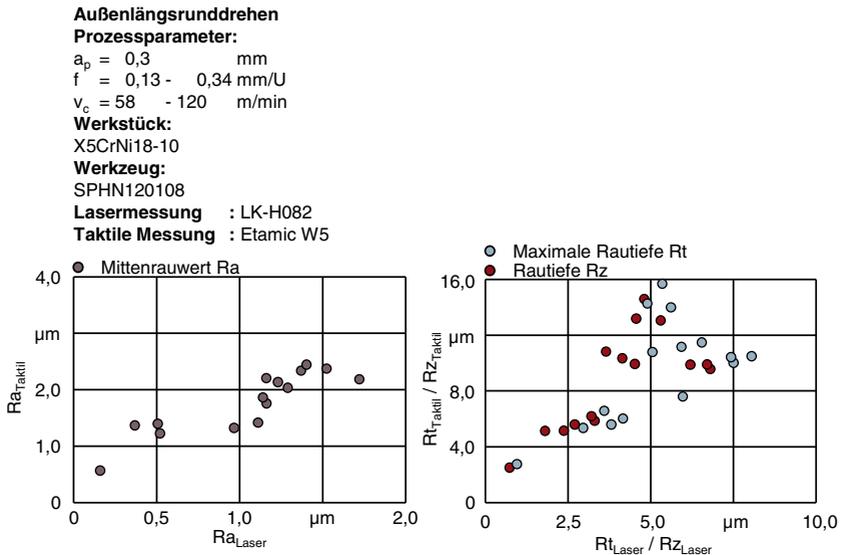


Abbildung 12 Korrelationsgraphen für die online-Messung der Rauheitskenngrößen Ra, Rt und Rz

Die mit den unterschiedlichen Messgeräten aufgetragenen Rauheitskennwerte können dabei durch Werkstückvibrationen sowie aufgrund unterschiedlicher Filterkennwerte, dem Einfluss des Tastspitzenradius beziehungsweise des Laserpunktdurchmessers auf die Rauheitsmessung voneinander abweichen. Durch diese Kalibrationsgleichungen kann eine prozesssichere und überprüfbare online-Messung der Oberflächenrauheit für die Prozessüberwachung und -regelung garantiert werden. Die Kalibrationsgleichungen für die Mittenrauheit Ra, die maximale Rautiefe Rt und die Rautiefe Rz sind in (5), (6) und (7) dargestellt.

$$Ra_{\text{Taktil}} = 1,05 Ra_{\text{Laser}} + 0,67 \quad (5)$$

$$Rt_{\text{Taktil}} = 1,22 Rt_{\text{Laser}} + 3,15 \quad (6)$$

$$Rz_{\text{Taktil}} = 1,43 Rz_{\text{Laser}} + 2,95 \quad (7)$$

2.6 Prozesswissen zur Ableitung der Adaptionvorschrift

Im Rahmen der vorliegenden Arbeit wurden zwei Prozessregelungen entwickelt. Eine zur Regelung der Oberflächengüte und eine zur Regelung der Bauteildimension. Das notwendige Prozesswissen zur Ableitung der Adaptionvorschriften wird im Folgenden vorgestellt.

2.6.1 Adaptionvorschrift für Prozessregelung der Oberflächengüte

Gemäß (8) kann die kinematische Rauheit R_{th} mit dem Schneideckenradius r_ϵ und dem Werkzeugvorschub f unter Vernachlässigung des Werkzeugverschleißes V_B wie folgt bestimmt werden [35].

$$R_{th} = r_\epsilon - \frac{f^2}{8r_\epsilon} \quad (8)$$

Die Richtigkeit dieses analytisch abgeleiteten Zusammenhanges konnte in mehreren experimentellen Untersuchungen gezeigt werden [46, 47, 23]. Prozessstellgrößen wie die Schnittgeschwindigkeit v_c oder der Spanwinkel γ haben ebenfalls einen Einfluss auf die resultierende Bauteilrauheit. Ihr Einfluss kann jedoch in den meisten Fällen vernachlässigt werden und lässt sich nicht über einfache lineare Methoden modellieren [48, 40]. Durch Ableitung von (8) ergibt sich (9), mit der die Änderung der Oberflächengüte δR_z durch Anpassung des Werkzeugvorschubs f bestimmt werden kann.

$$\delta R_z \approx \delta R_{th} = \frac{f}{4r_\epsilon} \delta f \quad (9)$$

Bei einem Schneideckenradius von $r_\epsilon = 0,8$ mm und einem Werkzeugvorschub von $f = 0,2$ mm/U kann die Rauheitsänderung für eine Vorschubanpassung von $\delta f = 0,01$ mm/U abgeschätzt werden zu $\delta R_z = 0,625$ μ m. Zur Stabilisierung der Regelung wird die Schnittparameteranpassung um eine Hysterese erweitert. Die Adaptionvorschrift ist in (10) dargestellt. Als Hysterese wurde der Wert $0,2$ μ m gewählt.

$$\Delta f = \begin{cases} 0,01 & \Leftrightarrow R_{z,Soll} - R_{z,Gemessen} > K_1 \\ 0 & \Leftrightarrow |R_{z,Soll} - R_{z,Gemessen}| < K_1 \\ -0,01 & \Leftrightarrow R_{z,Soll} - R_{z,Gemessen} < -K_1 \end{cases} \quad (10)$$

2.6.2 Adaptionvorschrift für die Prozessregelung der Bauteildimension

Ähnlich der Prozessregelung für die Oberflächengüte wird eine Hysterese für die Anpassung der Werkzeugposition ΔX definiert. Die Adaptionvorschrift für die Prozessregelung der Bauteildimension kann (11) entnommen werden. Als Hysterese wurde der Wert $K_2 = 5 \mu\text{m}$ gewählt.

$$\Delta X = \begin{cases} 0,001 & \Leftrightarrow D_{\text{Soll}} - D_{\text{Gemessen}} > K_2 \\ 0 & \Leftrightarrow |D_{\text{Soll}} - D_{\text{Gemessen}}| < K_2 \\ -0,001 & \Leftrightarrow D_{\text{Soll}} - D_{\text{Gemessen}} < -K_2 \end{cases} \quad (11)$$

2.7 Erreichte Prozessverbesserungen durch Prozessregelung

Die beschriebenen Prozessregelungen wurden für einen Drehprozess realisiert. Es konnte gezeigt werden, dass die auf online-Lasermessung basierende Prozessregelung in der Lage ist, sowohl Durchmesserabweichungen als auch Oberflächengüten prozesssicher einzustellen. Die Gegenüberstellung von konventioneller und geregelter Zerspanung bezüglich der Bauteildurchmesserabweichung ist in Abbildung 13 dargestellt.

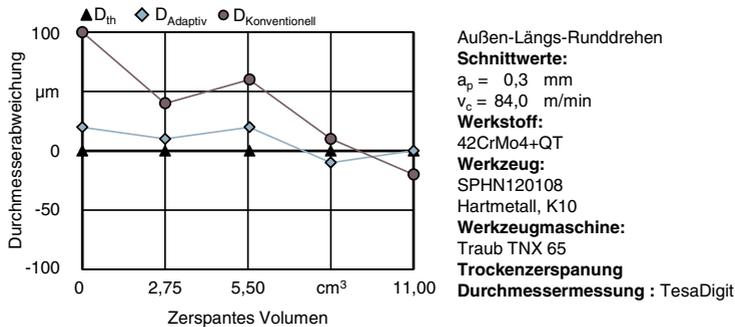


Abbildung 13 Gegenüberstellung der Bauteildurchmesserabweichung bei konventioneller und geregelter Zerspanung

Für die Durchmesserabweichung zeigt sich eine deutlich geringere Abweichung vom Sollwert. Die Störgrößen konnten somit ausgeregelt werden. Die Gegenüberstellung von konventioneller und geregelter Zerspanung in Hinblick auf die Bauteilrauheit in Abbildung 14 zeigt, dass geforderte Oberflächengüten durch die Prozessregelung trotz Werkzeugverschleiß für deutlich höhere Schnittzeiten garantiert werden können.

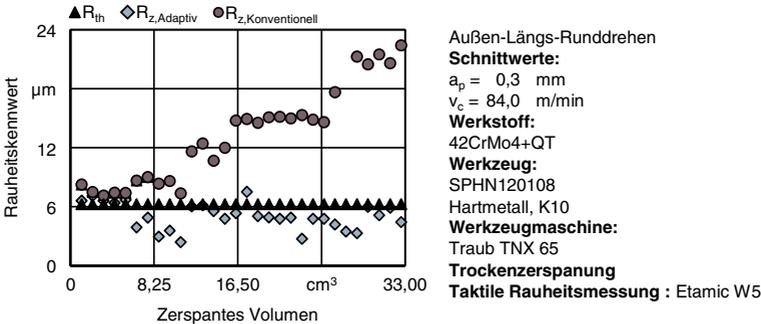


Abbildung 14 Gegenüberstellung der Oberflächengüte bei der konventionellen und der geregelten Zerspanung

2.8 Zusammenfassung und Schlussfolgerungen

Der vorliegende Artikel gibt einen Überblick über w issenschaftliche Vorarbeiten zum Thema Prozessregelung von Drehprozessen und stellt die industriellen Anwendungen vor. Des Weiteren werden Störgrößen auf den Drehprozess, deren Einfluss durch die Prozessregelungen kompensiert werden soll, aufgeführt und erläutert. Die Inbetriebnahme und Kalibration einer Lasermessung zur Messung der Oberflächengüte und Bauteildimension während der Drehbearbeitung wird erläutert und es werden Adaptionsvorschriften für entsprechende Prozessregelungen hergeleitet. Abschließend werden die mittels Prozessregelung erzielten Prozessverbesserungen vorgestellt und diskutiert. Aus dieser Studie können folgende Schlussfolgerungen gezogen werden:

- Die online-Messung mit einem Lasertriangulationssensor ist eine Möglichkeit, Oberflächengüten und Bauteilabweichungen prozesssicher während der Drehbearbeitung aufzuzeichnen.
- Prozessregelungen für die Oberflächengüte und die Bauteilabweichung können die Werkzeugstandzeit sowie die Bauteilqualität verbessern.
- Prozessregelungen haben das Potential, die Effizienz der Zerspanung von schwer zerspanbaren Werkstoffen drastisch zu erhöhen.

In weiteren Arbeiten sollten folgende Ansätze zur Weiterentwicklung der vorgestellten Prozessregelungen untersucht werden:

- Integration weiterer Messtechnik in den Maschinenraum zur Verbesserung der online-Messung von Bauteilrauheit und Bauteilqualität.
- Untersuchung weiterer Stellgrößen wie der Schnittgeschwindigkeit und dem Spanwinkel γ sowie deren Einfluss auf die Werkzeuglebensdauer.

2.9 Literatur

- 1 Ulsoy, A. G.; Koren, Y.: Applications of Adaptive Control to Machine Tool Process Control. IEEE Control Systems Magazine 9 (1989) 4, S. 33 - 37.
- 2 Wang, L.; Gao, R. X.: Condition monitoring and control for intelligent manufacturing. London: Springer, 2006.
- 3 Takeyama, H.; Honda, T.; Sekiguchi, H.; Takada, K.: Adaptive control of tool life in turning. The Government Mechanical Laboratory (1970) 8, S. 1 - 6.
- 4 Koren, Y.; Masory, O.: Adaptive Control with process estimation. Annals of the CIRP (1981) S. 373 - 376.
- 5 Schleipen, M.: Praxishandbuch - OPC UA. Vogel Business Media, 2018.
- 6 Weck, M.; Brecher, C.: Werkzeugmaschinen 3. Heideberg: Springer, 2006.
- 7 Masory, O.: Adaptive Control System for Turning. Annals of the CIRP (1980) S. 281 - 284.
- 8 Daneshmend, L. K.; Pak, H. A.: Model reference adaptive Control of feed force in turning. Journal of dynamic systems, measurement and Control 108 (1986) S. 215 - 222.
- 9 Ratava, J.; Rikkonen, M.; Ryyänen, V.; Leppänen, J.; Lindh, T.; Varis, J.; Sihvo, I.: An adaptive fuzzy control system to maximize rough turning productivity and avoid the onset of instability. International Journal of Advanced Manufacturing Technology 53 (2011) 1 - 4, S. 71 - 79.
- 10 Ratava, J.; Lohtander, M.: Fuzzy feed rate and cutting speed optimization in turning. International Journal of Advanced Manufacturing Technology 62 (2018) 1 - 4, S. 59.
- 11 Hänsel, W.: Beitrag zur Technologie des Drehprozesses im Hinblick auf Adaptive Control. Berichte aus dem Laboratorium für Werkzeugmaschinen und Betriebslehre der RWTH Aachen. Hrsg.: König, W. Dissertation, RWTH Aachen. Aachen Fakultät Maschinenwesen, 1974.
- 12 Azouzi, R.; Guillot, M.: Online Optimization of the turning process using an inverse process neurocontroller. Journal of Manufacturing Science and Engineering 120 (1998) S. 101 - 108.
- 13 Ivester, R. W.; Heigel, J. C.: Smart machining systems: robust optimization and adaptive control optimization for turning operations. Transactions of NAMRI/SME 35 (2007) S. 505 - 512.
- 14 Oraby, S.; Alaskary, A.: Adaptive Control Program for Rough Turning Machining Process. Conference on Advances in Engineering Sciences and Applied Mathematics 6 (2016).
- 15 Abellan, J. V.; Romero, F.; Siller, H. R.; Estruch, A.; Vila, C.: Adaptive Control Optimization of Cutting Parameters for High Quality Machining Operations based on Neural Networks and Search Algorithms. Advances in Robotics, Automation and Control (2008) S. 472 - 492.
- 16 Shiraishi, M.; Sato, S.: Dimensional and Surface Roughness Controls in a Turning Operation. Transactions of ASME, Journal of Manufacturing Science and Engineering 112 (1990) S. 77 - 83.
- 17 Choudhury, S. K.; Ramesh, S.: Online tool wear sensing and compensation in turning. Journal of Materials Processing Technology 49 (1993) S. 247 - 254.
- 18 Coker, S. A.; Shin, Y. C.: In-process control of surface roughness due to tool wear using a new ultrasonic system. International Journal of Machine Tools and Manufacturing 36 (1996) 3, S. 411 - 422.

- 19 Zhang, J. Z.; Chen, J. C.; Kirby, E. D.: The development of an in-process surface roughness adaptive control system in turning operations. *Journal of Intelligent Manufacturing* 18 (2007) 3, S. 301 - 311.
- 20 García-Plaza, E.; Núñez, P. J.; Martín, A. R.; Sanz, A.: Adaptive control of surface finish in automated turning processes. In: *Band 4 C* adiz, Spain. AIP, 2012, S. 351 - 359.
- 21 Saldana, C.; Swaminathan, S.; Brown, T. L.; Moscoso, W.; Mann, J.B.; Compton, W.D.; Chandrasekar, S.: Unusual Applications of Machining: Controlled Nanostructuring of Materials and Surfaces. *Journal of Manufacturing Science and Engineering* (2010) 132, S. 309081 - 3090812.
- 22 Caruso, S.; Umbrello, D.; Outeiro, J. C.; Filice, L.; Micari, F.: An experimental investigation of residual stresses in hard machining of ALSI 52100 steel. *Procedia Engineering* 1 (2011) 19, S. 67 - 72.
- 23 Denkena, B.; Grove, T.; Maiss, O.: Influence of the cutting edge radius on surface integrity in hard turning of roller bearing inner rings. *Production Engineering Research Development* 9 (2015) 3, S. 299 - 305.
- 24 Brinksmeier, E.; Brockhoff, T.: Spanende Kurzzerteilmetallurgie. *wt Werkstattstechnik online* 87 (1997) 9/10, S. 463 - 466.
- 25 Aurich, J.; Schneider, F.; Mayer, P.; Kirsch, B.; Hasse, H.: Oberflächenerzeugungsmorphologie-Eigenschafts-Beziehungen. *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 111 (2016) 4, S. 213 - 216.
- 26 Comara: iCut. St. Georgen. Firmenschrift. 2018.
- 27 Omatic Systems: ACM - Adaptive Control & Monitoring. Villingen-Schwenningen. Firmenschrift. 2018.
- 28 Caron Engineering Inc.: TMAC. Firmenschrift. Wells, Maine, USA. 2018.
- 29 Artis Marposs: CTM - Passgenaue Werkzeug-, Prozess- und Maschinenzustandsüberwachung. Egestorf. 2018.
- 30 Guo, Y. B.; Warren, A. W.; Hashimoto, F.: The basic relationships between residual stress, white layer, and fatigue life of hard turned and ground surfaces in rolling contact. *CIRP Journal of Manufacturing Science and Technology* 2 (2010) S. 129 - 134.
- 31 Unbehauen, H.: Regelungstechnik III. Vieweg, 2011.
- 32 Ulsoy, A. G.; Koren, Y.; Rasmussen, F.: Principal Development in the adaptive control of machine tools. *Journal of dynamic systems, measurement and Control* 105 (1983) S. 107 - 112.
- 33 Liu, Y.; Cheng, T.; Zuo, L.: Adaptive Control Constraint of Machining Processes. *The International Journal of Advanced Manufacturing Technology* 17 (2001), S. 720 - 726.
- 34 Liang, S. Y.; Hecker, R. L.; Landers, R. G.: Machining Process monitoring and Control: the State-of-the-art. *International Mechanical Engineering Congress & Exposition* (2002) S. 1 - 12.
- 35 Denkena, B.; Tönshoff, H. K.: Spanen. Berlin, Heidelberg, New York: Springer, 2011.
- 36 Kümmel, J.: Detaillierte Analyse der Aufbauschneidendenbildung bei der Trockenzerspannung von Stahl C45E mit Berücksichtigung des Werkzeugverschleißes. Schriftenreihe des Instituts für angewandte Materialien. Hrsg.: Gibmeier, E. Dissertation, Karlsruher Institut für Technologie. Karlsruhe: KIT Scientific Publishing, 2016.

- 37 Kümmel, J.; Gibmeier, J.; Schulze, V.; Wanner, A.: Effect of built-up edge formation on residual stresses induced by dry cutting. *Materials Research* 996 (2014) S. 603 - 608.
- 38 Opitz, H.; Gappisch, M.: Die Aufbauschneidenbildung bei der spanabhebenden Bearbeitung. Köln: Westdeutscher Verlag, 1964.
- 39 Weck, M.; Brecher, C.: Werkzeugmaschinen 5. Heidelberg: Springer, 2006.
- 40 Dietrich, J.; Tschätsch, H.: Praxis der Zerspantechnik: Verfahren, Werkzeuge, Berechnung. Wiesbaden: Springer Vieweg, 2016.
- 41 Pauusch, E.; Holsten, S.; Linß, M.; Tikal, F.: Zerspantechnik. Wiesbaden: Vieweg + Teubner, 2008.
- 42 Gecim, B.; Winer, W. O.: Transient Temperatures in the Vicinity of an Asperity Contact. *Journal of Tribology* 1985 (1985) 107, S. 333 - 342.
- 43 Tian, X.; Kennedy, F. E.: Maximum and Average Flash Temperatures in Sliding Contacts. *Journal of Tribology* 116 (1994) 167, S. 167 - 174.
- 44 Grzeseik, W.; Nieslony, P.: Physics based modelling of interface temperatures in machining with multilayer coated tools at moderate cutting speeds. *International Journal of Machine Tools Manufacture* 2004 (2004) 44, S. 889 - 901.
- 45 Fahad, M.: A Heat Partition investigation of Multilayer Coated Carbide Tools for High Speed Machining through experimental studies and Finite Element Method. Hrsg.: Sheikh Dissertation. University of Manchester. Manchester Faculty of Engineering and Physical Sciences, 2012.
- 46 Rech, J.: Surface integrity in finish hard turning of case-hardened steels. *International Journal of Machine Tools & Manufacture* 43 (2003) S. 543 - 550.
- 47 Davim, J. P.: Surface Integrity in Machining. *Advanced Machining Processes* (2010) S. 37 - 66.
- 48 Jacobson, M.; Dahlman, P.; Gunnberg, F.: Cutting speed influence on surface integrity of hard turned bainite steel. *Journal of Materials Processing Technology* 128 (2002) S. 318 - 323.

3 Vermeidung von Nothalten von fahrerlosen Transportfahrzeugen durch lokale Bahnplanung

T. Heinemann, A. Lechler, O. Riedel, A. Verl, ISW U niversität Stuttgart

3.1 Einführung

Fahrerlose Transportfahrzeuge (FTF) sind logistisch e Betriebsmittel, die auf dem Fabrikboden operieren und durch ein geeignetes Leitsystem gesteuert werden. Sie übernehmen Transportaufgaben in vielfältigen Einsatzbereichen, z.B. in der Fertigung oder in Lagersystemen. Ullrich beschreibt sie in [1] als „probate[s] Organisationsmittel der modernen Intralogistik“. Dabei besteht ein großer Vorteil von FTF gegenüber fest installierten logistischen Anlagen wie z.B. Fließbändern darin, dass mit ihnen schnell auf sich ändernde Anforderungen in den logistischen Abläufen reagiert werden kann. Ebenso ist eine Flotte aus FTF im Vergleich zu fest installierten Logistiksystemen kostengünstiger skalierbar [2]. Gegenüber einer manuellen Logistik, in der die Bewegung durch Werker erfolgt, können durch den Einsatz von FTF die Personalbindung im Transport und damit auch die Personalkosten gesenkt werden. Die für den Einsatz der FTF notwendige Organisation schafft eine größere Transparenz der Logistikkabläufe. Hierzu gehört z.B. eine Strukturierung und Planung der Logistikaufgaben und das langfristige und konsequente Freihalten der Fahrwege [1].

Für den sicheren Betrieb von FTF ist es unerlässlich, dass die Fahrzeuge ihre Umgebung wahrnehmen und auf diese reagieren können. Bei ihrem Betrieb können Hindernisse vorkommen, die nicht auf einer Karte vorab gespeichert werden können wie z.B. menschliche Arbeiter, andere FTF oder temporär abgestellte Kisten oder Paletten. Im linken Teil der Abbildung ist ein FTF dargestellt, das exemplarisch mittels Kamera und Laserscanner eine Kiste und einen Menschen vor sich erkennt. Für jedes Objekt, das sich in der in der Nähe des FTF befindet, gibt es einen Bereich, in dem eine Kollision zwischen dem Objekt und dem FTF unvermeidbar ist, das heißt, dass das FTF nicht rechtzeitig bremsen kann. Bei bewegten Hindernisobjekten erstreckt sich dieser Bereich entlang des Bewegungsvektors des FTF. Schematisch sind solche Bereiche in Abbildung im mittleren Teil dargestellt. Ein FTF sollte also vermieden werden, in eine Kollisionssituation zu gelangen. Diese Aufgabe übernimmt eine Safety-SPS, in Kombination mit Sensoren, die das FTF vor einer drohenden Kollision sicher anhält.

Aus verschiedenen Gründen ist es sinnvoll, einen solchen Nothalt verzichtbar zu machen. Beispielsweise muss das FTF aus dem Nothalt-Zustand durch Personal wieder in funktionsfähigen Zustand versetzt werden. Das kostet Zeit und ist vor allem in spezialisierten Logistikkbereichen, in denen sich im Normalbetrieb kein Mensch aufhält, mit zusätzlichem Aufwand verbunden. Ein Nothalt lässt sich vermeiden, indem das FTF dem

Hindernis ausweicht und dabei stets außerhalb des kritischen Sicherheitsbereiches bleibt. Dazu muss der für das FTF vorgegebene Pfad entsprechend angepasst werden. Dies ist schematisch im rechten Teil der Abbildung zu sehen.

Die in diesem Beitrag vorgestellten Recherchen und erstellten Simulationen sind im Rahmen des Projektes SmartFTF entstanden. Im Projekt SmartFTF soll ein Nothalt während der Fahrt erkannten Hindernissen vermieden werden. Motivation ist neben den o.g. Argumenten, dass ein Bremsvorgang zeitlich und energetisch nicht optimal ist und zusätzlich Motoren und Bremsen belastet, die sich damit schneller abnutzen.

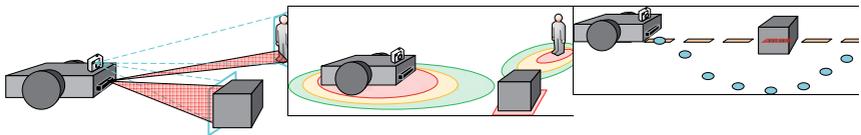


Abbildung 1: Links: ein FTF erkennt Objekte in seiner Umgebung. Mitte: Sicherheitszonen von bewegten und unbewegten Objekten. Rechts: Neuplanung eines Pfades, um einem Hindernis auszuweichen.

Um einen Nothalt zu vermeiden, sollen statische und dynamische Hindernisse umfahren werden. Die Trajektorie der zum Umfahren geplanten Bahn sollte dabei möglichst glatt sein, um hohe Belastungen der Bremsen und Antiriese wie auch ein Verrutschen der Ladung des FTF zu vermeiden. Um die globale Planung für alle FTF des Betriebes zu optimieren, soll ein FTF die erkannten Hindernisse an den Leitstand zurückmelden. So können blockierte Wege von allen anderen FTF in einer Flotte ebenso vermieden werden.

3.2 Inhalt dieser Arbeit und Kriterien zur Bewertung von lokalen Bahnplanungsalgorithmen

In dieser Arbeit werden zwei Aspekte des eben genannten Projekts SmartFTF beleuchtet. Zum einen betrifft dies die Theorie der Algorithmen zur lokalen Bahnplanung. Zum anderen wird die Erstellung einer Simulationsumgebung beschrieben, die zum Testen der Algorithmen verwendet wird.

In den folgenden Kapiteln werden Algorithmen zur Bahnplanung von Fahrzeugen vorgestellt. Dazu wird als Kontext beschrieben, wie unter Verwendung von Kartenmaterial ein Weg von einem Start- zu einem Zielpunkt gefunden werden kann. Solche Verfahren werden in dieser Arbeit als „globale Planung“ bezeichnet. Durch die Kenntnis der

kompletten Karte kann hier ein optimales Ergebnis global gesehen existiert. Diese globale Planung kann über auf Hindernisse während der Fahrt eingesetzt werden, da die Berechnungszeit für die Erstellung der optimalen Bahn zu groß ist. Einer der grundlegenden Algorithmen für diesen Anwendungsfall, der Dijkstra-Algorithmus [3], wird vorgestellt. Für das Projekt SmartFTF wird die globale Bahnplanung des FTF in Form einer Bahn vom Start- zum Zielpunkt als gegeben betrachtet.

Zusätzlich soll nun in Situationen, in denen ein statisches oder dynamisches Hindernis von den Sensoren des FTF erkannt wird, in geeigneter Weise auf diese Hindernisse reagiert werden. Eine solche Reaktion ist – unter dem Paradigma der Vermeidung eines Nothalts - gleichbedeutend mit einer Änderung des vorgegebenen globalen Weges. Eine solche lokale Bahnplanung kann aufgrund ihres Mangels an vollständiger Information keine optimalen Ergebnisse liefern. Dafür ist die Berechnungszeit geringer, da sich die Berechnung auf die direkte Umgebung des FTF und auf kleine Zeitintervalle beschränkt. Drei Algorithmen, mit denen diese situationsbezogene Neuplanung erreicht werden kann, werden beschrieben: der Elastic Band Algorithmus [4] der Dynamik Window Approach [5] und der Vector Field Histogram Algorithmus [6]. Diese drei Algorithmen werden in Bezug auf folgende Punkte bewertet und verglichen.

Berücksichtigung des globalen Pfades Im Anwendungsfall der FTF ist der global geplante Pfad des Fahrzeuges bekannt. Die lokale Bahnplanung kann von ihm profitieren, da er eine komplette Lösung des Wegs bis zum Zielpunkt beinhaltet. Es muss kein komplett neuer Weg zum Zielpunkt von der lokalen Bahnplanung gefunden werden, sondern ein Weg um das Hindernis herum und zurück auf den lokalen Pfad zum Zielpunkt.

Ausgabe von Geschwindigkeiten Die Motoren des FTF werden über Geschwindigkeitsvorgaben angesteuert. Wenn eine Bahnplanung geometrische Koordinaten ausgibt, müssen diese zunächst in dazu passende Geschwindigkeiten überführt werden.

Berücksichtigung der Fahrzeugdynamik Das Beschleunigungs- und Bremsverhalten, sowie der minimale fahrbare Kurvenradius des FTF bei einer bestimmten Ausgangsgeschwindigkeit sind begrenzt. Diese Grenzen müssen in der Vorgabe der Motorsollwerte berücksichtigt werden. Gibt der Algorithmus der Bahnplanung und die Umsetzung der Geschwindigkeiten einen nicht erreichbaren Wert aus, muss dieser zum nächstgelegenen realistischen Wert geändert werden. Wenn diese Grenzwerte bereits im Algorithmus beachtet

werden, wird möglicherweise ein geeigneterer Punkt ausgewählt.

Erzeugung glatter Trajektorien Glatte, d.h. C2-stetige Trajektorien vermindern den mechanischen Ruck auf das Antriebssystem des FTF und verringern dessen vorzeitigen Verschleiß. Zusätzlich wird verhindert, dass die Ladung verrutscht oder sich ungewollt vermischt und in Ordnung gerät.

Verwenden derselben Karte über mehrere Schritte im Algorithmus Die im Algorithmus verwendete Karte wird entweder in jedem Schritt neu erstellt oder eine bestehende Karte in jedem Schritt aktualisiert. Dabei bietet eine über mehrere Schritte wieder verwendete Karte neben einer kürzeren Berechnungszeit die Möglichkeit, während der Fahrt gesammeltes Wissen in Form der Karte zu speichern und weiterzugeben.

Umgang mit bewegten Hindernissen Die im Projekt angestrebte Lösung soll auch mit bewegten Hindernissen umgehen können.

Zusätzlich zur Vorstellung der Algorithmen und deren Vergleich wird auch der Aufbau der Simulationsumgebung für das FTF und der neuen Algorithmen beschrieben. Dabei wird ein dynamisches Modell erstellt, das im Rahmen des Projektes zur Massenschätzung des FTF verwendet werden soll. Dieses Modell kann auch dazu verwendet werden, die Bewegungen des FTF zu simulieren. Dies soll für die Tests der ausgewählten Algorithmen verwendet werden. Die Umgebung des FTF, sowie die Simulation von Sensorwerten während der Fahrt in der simulierten Umgebung wird mit dem Programm Gazebo [7] abgebildet. Dieses Programm lässt außerdem zu, mehrere FTF gleichzeitig zu simulieren, was beispielsweise bewegte Hindernisse ermöglicht. Die Steuerung des simulierten FTF soll mit dem Framework ROS [8] möglichst nahe an der originalen Steuerung der FTF nachgebaut werden.

3.3 Theoretische Grundlagen der Navigation und Modellierung von FTF und Vorstellung der Simulationsumgebung

Im Folgenden werden verschiedene Verfahren vorgestellt, die zur Navigation mobiler Systeme verwendet werden können. Ein Beispiel für eine globale Bahnplanung beleuchtet die Grundlagen der Navigation, auf denen die lokale Bahnplanung dann auf-

baut. Hier wird der Dijkstra-Algorithmus vorgestellt. Für die lokale Bahnplanung werden drei Algorithmen; Elastic Band, Dynamic Window und Vector Field Histogram; vorgestellt und nach den in Kapitel 2 genannten Kriterien verglichen [4 bis 6].

Außerdem wird beschrieben, wie die Dynamik eines FT F durch eine physikalische Beschreibung der wirkenden Kräfte modelliert werden kann.

Zuletzt werden die Softwaretools vorgestellt, die in dem Projekt Verwendung finden.

3.3.1 Globale Planung am Beispiel des Dijkstra-Algorithmus

Um initial einen Weg von einem bekannten Startpunkt, dem aktuellen Aufenthaltsort des FTF, und einem bekannten Zielpunkt zu bestimmen, sind in der Informatik mehrere Verfahren bekannt, z.B. der A* Algorithmus, der Bellman-Ford Algorithmus oder der Floyd-Warshall-Algorithmus [9], [10]. Beispielhaft für ein solches Verfahren sei hier die Suche nach einem kürzesten Pfad im Algorithmus nach Dijkstra vorgestellt [3]:

Der Algorithmus arbeitet auf einem ungerichteten Graphen mit positiven Kantengewichten. Start- und Zielknoten sind dabei bekannt. Für alle Knoten werden drei Eigenschaften betrachtet: der Vorgänger auf dem Pfad ausgehend vom Startknoten, die Summe der Kantengewichte auf dem bis zum Knoten betrachteten Pfadabschnitt und die Information, ob dieser Knoten bereits betrachtet wurde oder nicht.

Für den Startknoten ist der Vorgänger der Startknoten selbst, die Summe der Kantengewichte auf dem Pfad bis zu diesem Knoten ist Null und er wurde bereits betrachtet. Für alle anderen Knoten ist der Vorgänger zunächst unbekannt, die Summe der Kantengewichte auf der Strecke zum Startknoten unendlich groß und sie wurden bisher nicht betrachtet. Nun beginnt der Algorithmus damit, dass alle Knoten, die direkt mit dem Startknoten verbunden sind, einer Menge an Knoten hinzugefügt werden, die man als Rand des bekannten Pfades interpretieren kann. Ihre jeweiligen Kantengewichte werden ebenfalls gespeichert.

Unter den in dieser Randmenge befindlichen Knoten wird als nächstes derjenige Knoten ausgewählt, der das niedrigste Kantengewicht besitzt. Für diesen Knoten wird wie folgt verfahren: Sein Vorgänger ist der Startknoten. Die Entfernung zum Startknoten entspricht seinem Kantengewicht und er wird als „betrachtet“ markiert. All seine benachbarten Knoten, die sich noch nicht in der Randmenge befinden werden dieser Menge hinzugefügt. Der eben betrachtete Knoten wird aus der Randmenge entfernt.

Nun wird aus der Randmenge in der gleichen Weise wieder der Knoten mit der kürzesten Distanz (Summe der Kantengewichte) zum Startknoten entnommen und betrachtet. Dies geschieht so lange bis der Zielknoten erreicht wurde oder die Randmenge leer ist. Im ersten Fall wurde der kürzeste Pfad durch das Erreichen des Zielknotens gefunden, im zweiten Fall existiert kein Pfad zum Zielknoten. Sobald die Randmenge leer ist, wurden alle Knoten besucht, die vom Startknoten aus erreichbar sind.

In der Anwendung auf FTF muss für diesen Algorithmus die Karte der Umgebung des FTF entsprechend dargestellt werden. Beispielsweise das über die Karte gelegt wird. Jedes Kästchen im Graphen, die Distanzen zwischen benachbarten Knoten entsprechen der euklidischen Distanz zwischen ihnen.

3.3.2 Lokale Planung: Elastic Band

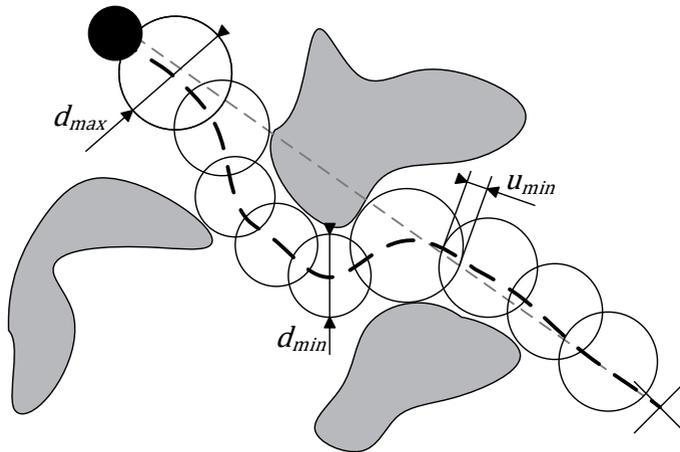


Abbildung 2: Elastic Band Algorithmus; Die Abweichung (dick gestrichelt) vom globalen Pfad (dünn gestrichelt) verläuft durch ein Band von Kreisen. Skizze nach [4].

Der Elastic Band Algorithmus [4] baut auf einem global geplanten Pfad auf. Entlang dieses Pfades erstellt er im zweidimensionalen Fall Kreise, die folgende Kriterien erfüllen:

- Kreisdurchmesser d_i : $d_{min} < d_i < d_{max}$. Dabei werden die Kreise so groß wie möglich gewählt, ohne dass sie ein Hindernis berühren.
- minimale Überlappung zweier benachbarter Kreise mit Mittelpunkt

$$\left(m_i + \frac{d_i}{2} \cdot (m_{i+1} - m_i)\right) - \left(m_{i+1} - \frac{d_{i+1}}{2} \cdot (m_{i+1} - m_i)\right) > u_{min} \quad (1)$$

- Es gibt keine redundanten Kreise, d.h. Kreise deren rechter und linker Nachbar die minimale Überlappung erfüllen, werden gelöscht

Diese Kreise beschreiben eine Teilmenge des Raumes, in dem sich das FTF frei bewegen kann und schränken so den Suchraum nach einem Pfad ein. Jeder Pfad, der sich ausschließlich innerhalb der Kreise befindet, kollidiert nicht mit einem Hindernis.

Sobald das FTF in seiner Umgebung Hindernisse detektiert, modifiziert der Algorithmus die Position der Kreise so, dass weiterhin keiner der Kreise mit einem Hindernis überlappt. Dies geschieht durch eine mathematische Formulierung der Hindernisse als abstoßende Kräfte und der benachbarten Kreismittelpunkte als anziehende Kräfte. Das Kräftegleichgewicht für jeden Kreis wird bestimmt und die Position der Kreismittelpunkte entsprechend angepasst. Die Kriterien für das Band werden daraufhin geprüft. Gibt es Lücken im Band, werden entsprechend Kreise eingefügt. Die Kreisgröße wird stets so groß gewählt, wie an der gegebenen Position möglich. Das führt dazu, dass an Kurven um Hindernisse herum kleinere und mehr Kreise positioniert werden als an geraden Strecken, der Pfad also an den Stellen genauer detailliert wird, an denen diese Notwendigkeit besteht.

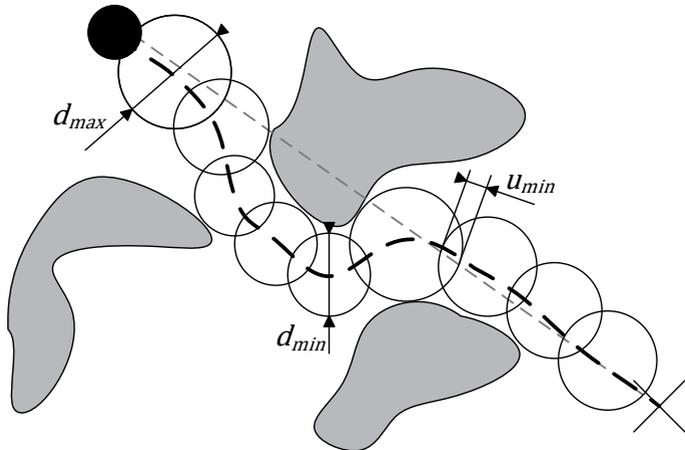


Abbildung 2 stellt schematisch ein solches Band an Kreisen und einen Pfad dar. Durch die minimale Größe der Kreise kann ein Sicherheitsabstand von Hindernissen gewährleistet werden.

3.3.3 Lokale Planung: Dynamic Window

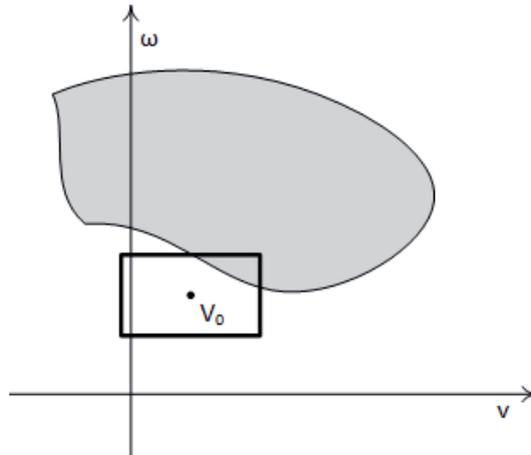


Abbildung 3: ω - v -Ebene des Dynamic Window Approach. Hindernisse sind grau hinterlegt dargestellt, der mit v_0 bezeichnete Punkt stellt die aktuelle FTF-Geschwindigkeit dar. Das Rechteck um diese Geschwindigkeit herum ist das dynamische Fenster: alle Geschwindigkeiten, die in einem Zeitintervall erreicht werden können. Skizze nach [5].

Der Dynamic Window Approach ist ein lokaler Bahnplanungsalgorithmus, der sich vor allem dadurch auszeichnet, die dynamischen Einschränkungen des FTF direkt in die lokale Bahnplanung zu integrieren. Für diesen Algorithmus wird kein vorher geplanter globaler Pfad benötigt, wie dies beim Elastic Band Algorithmus der Fall ist. Allerdings wird z.B. in der Implementierung des Dynamic Window Approach aus den verfügbaren Paketen der Roboter-Middleware ROS ein globaler Pfad berücksichtigt [11]. Hier werden Wegpunkte aus der Suche der globalen Bahnplanung als Zielvorgabe für die lokale Bahnplanung gesetzt. So kann die lokale Bahnplanung von der globalen Planung profitieren.

Im Dynamic Window Approach werden die Hindernisse nicht mit ihrer Position kartographiert, sondern in ihrer Konfiguration aus translatorischer Geschwindigkeit und Winkelgeschwindigkeit ω . In der entsprechenden $v - \omega$ - Ebene werden dann Trajektorien für das Fahrzeug gesucht. Dazu werden zunächst alle Geschwindigkeiten markiert, bei denen das FTF vor einem Hindernis rechtzeitig bremsen oder ausweichen kann. Sie kommen als mögliche Trajektorien infrage. Als nächstes wird das sogenannte „dynamische Fenster“ bestimmt. Es ist durch die maximale Beschleunigung des FTF begrenzt und beinhaltet alle Konfigurationen in Geschwindigkeit und Winkelgeschwindigkeit, die das FTF bis zum nächsten Rechen schritt erreichen kann [5].

Abbildung 3 zeigt eine Skizze der $v - \omega$ – Ebene, in der die Konfiguration des FTF als Punkt dargestellt ist. Das dynamische Fenster wird durch das Rechteck um die Position des FTF dargestellt. Die grau dargestellte Fläche stellt die Geschwindigkeitskonfigurationen dar, in denen das FTF mit einem Hindernis kollidieren würde. Innerhalb des „dynamischen Fensters“ wird nun nach der geeignetsten Steueranweisung für das FTF innerhalb des nächsten Zeitintervalls gesucht. Für diese Suche findet eine Kostenfunktionsanwendung, die die Einflüsse von Ziel-treue, Geschwindigkeit und Abstand von Hindernissen miteinander in Bezug setzt. Von Fox, Burgard, und Thrun wird Gleichung (2) als Kostenfunktion angegeben [5].

$$G(v, \omega) = \sigma (\alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{vel}(v, \omega)) \quad (2)$$

Dabei sind α , β und γ Gewichtungsfaktoren. Die Funktion σ glättet die gewichtete Summe der drei Komponenten und schafft größere Abstände der generierten Trajektorie zu Hindernissen [5].

3.3.4 Lokale Planung: Vector Field Histogram

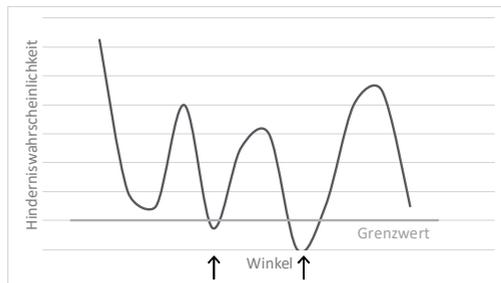
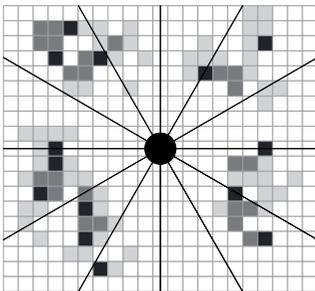


Abbildung 4: Links: Darstellung einer gerasterten Karte mit Hinderniswahrscheinlichkeiten in Graustufen. Rechts: Histogramm der Hinderniswahrscheinlichkeit. Beide Abbildungen nach [6].

Einen anderen Ansatz der lokalen Bahnplanung verfolgt der „Vector Field Histogram“ Algorithmus [6]. Dieser Algorithmus besteht aus zwei wesentlichen Teilen, der Datenkompression und der Wegsuche.

Bei jeder Abfrage der Scanner wird in einer im Fahrzeug vorgehaltenen Karte, die in Planquadrate aufgeteilt ist, für eines der Quadrate die gemessene Hinderniswahrscheinlichkeit eingetragen. So kann eine schnelle Laufzeit dieses Schrittes garantiert werden, die allerdings zu Lasten der Abbildungsgenauigkeit geht. Durch wiederholte

sensorische Scans und die entsprechenden Eintragungen in der Karte können die Hinderniswahrscheinlichkeiten der Umgebung trotzdem realistisch dargestellt werden. Der linke Teil in

Abbildung 4 zeigt eine Skizze einer solchen gerasterten Karte. Die Färbung der Quadrate stellt hier die Hinderniswahrscheinlichkeit dar: je dunkler ein Quadrat eingefärbt ist, desto wahrscheinlicher befindet sich dort ein Hindernis.

Diese rechteckige Karte wird in einem nächsten Schritt in konzentrische Sektoren mit einem gleichbleibenden Winkel um das FTF aufgeteilt. In jedem Sektor wird die Dichte an Hindernissen aus den Hinderniswahrscheinlichkeiten der Karte eingetragen. Danach wird die so entstandene Funktion geglättet.

Durch das Errechnen der Hindernisdichte werden die dreidimensionalen Daten bestehend aus Position in der Ebene und Hinderniswahrscheinlichkeit zu einem zweidimensionalen Datensatz reduziert, in dem die Hindernisdichte über dem Winkel steht.

Diese Darstellung wird Vektorfeldhistogramm genannt. Eine Skizze dieses Histogramms ist in

Abbildung 4 im rechten Teil zu sehen.

In diesem Histogramm werden nun alle Werte, die über einem Grenzwert liegen als nicht befahrbare Bereiche markiert. In den verbleibenden „Candidate Valleys“, also infrage kommenden Tälern, kann das FTF verfahren. Im rechten Teil der

Abbildung 4 sind diese Täler mit Pfeilen markiert.

Die Auswahl unter den potentiellen Tälern erfolgt unter Betrachtung der minimal notwendigen Durchgangsbreite und der Position des Ziels. Die Positionierung innerhalb des Tales orientiert sich, vor allem bei breiten Tälern, an der dem Ziel zugewandten Wandseite.

3.3.5 Vergleich der lokalen Verfahren

Im Folgenden werden die Algorithmen „Elastic Band“, „Dynamic Window Approach“ und „Vector Field Histogram“ basierend auf Informationen aus der Literatur verglichen.

Tabelle 3-1 stellt die aus Kapitel 3.2 bekannten bewerteten Kriterien gegenüber.

Tabelle 3-1: Vergleich der Algorithmen

	Elastic Band	Dynamic Window	Vector Field Histogram	
Berücksichtigung des globalen Pfades	○	◐	◐	Legende ○ erfüllt
Ausgabe von Geschwindigkeiten	●	○	○	

Berücksichtigung der Dynamik des FTF	◐	○	●	◐ zum Teil erfüllt ● nicht erfüllt
Erzeugung glatter Trajektorien	◐	●	●	
Die verwendete Karte kann über mehrere Algorithmusschritte verwendet werden	○	●	○	
Wurde erfolgreich mit dynamischen Hindernissen getestet	◐	●	◐	

Eine Berücksichtigung des globalen Pfades findet in dingen wird nur im Elastic Band Algorithmus dieser globalen Bahnplanung verwendet und jeweils angepasst erhalten, während gleichzeitig eine Reaktion auf vor hene Hindernisse stattfindet [4]. Im Dynamic Window togram wird die globale Bahnplanung verwendet, um den lokalen Bahnplanungsalgorithmus zu generieren. weilige Algorithmus dann zu. Borenstein und Koren n lichkeit, den Grenzwert für die Hinderniserkennung rithmus mit Informationen aus der globalen Bahnplan Fahrten kann dieser Wert auf einen niedrigen Wert g der Hinderniserkennung erhöht. Fährt das FTF jedoch kann der Grenzwert temporär angehoben werden, wodurch leichter gefunden werden kann.

Von den drei Algorithmen geben der Dynamic Window A Histogram Algorithmus zusätzlich zu den Weginformat gabe für das FTF eine Sollgeschwindigkeit/-Beschleu Window Approach ist dieser Geschwindigkeitswert dir enthalten, da eine Optimierung im Geschwindigkeitsr Histogram Algorithmus wird stattdessen basierend au keit in Blickrichtung des FTF eine Reduzierung der nommen. Der Elastic Band Algorithmus gibt keine Vor

Die spezielle Dynamik und –Kinematik des FTF kann v zum Teil verwendet werden. Quinlan und Khatib schla zwei translatorischen und einem rotatorischen Freih

allen drei Algorithmen statt. Allerlobale Pfad als Ganzes in der lo- So bleibt die globale Information m globalen Planer unvorhergese- Approach und Vector Field His- Ziel- oder Zwischenpunkte für Auf diesen Punkt steuert der je- ennen in [6] zusätzlich die Mög- des Vector Field Histogram Algo- ung zu beeinflussen. Für normale elegt werden, was die Sicherheit durch einen schmalen Korridor, ch ein Weg durch die Vereng-

pproach und der Vektor Field ionen bzw. der Richtungsvo- nigung vor. Im Falle des Dynamic ekt im Ergebnis des Algorithmus aum gelöst wird. Im Vector Field f der Hinderniswahrscheinlich- Maximalgeschwindigkeit vorge- gabegeschwindigkeiten an.

om Elastic Band Algorithmus gen vor, für ein Fahrzeug mit eitsgrad eine dreidimensionale

Repräsentation als Kugeln der Kreise aus dem zweidimensionalen Fall zu verwenden. Es werden jedoch keine kinematischen oder dynamischen Gegebenheiten des FTF explizit modelliert. [4] Der Vector Field Histogram-Algorithmus berücksichtigt die Kinematik/Dynamik des Fahrzeugs nicht. Im Dynamic Window Approach ist hingegen das dynamische Fenster den maximalen Beschleunigungen des Fahrzeugs angepasst. Auch ein unterschiedliches Beschleunigungsverhalten bei verschiedenen Ausgangsgeschwindigkeiten kann abgebildet werden. [5]

Im SmartFTF-Projekt soll die Trajektorie für das FTF möglichst glatt sein. Hierüber geben nur Quinlan und Kathib für den Elastic Band Algorithmus direkt Auskunft: Sie erwähnen die Generierung von Splines mit minimaler Verzerrungsenergie, die durch die im Algorithmus verwendeten Kreise/Kugeln gelegt werden. Diese Splines sind ableitungsstetig und erlauben dadurch dem FTF ein Abfahren der vorgegebenen Bahn ohne Halt. Der Dynamic Window Approach betrachtet in jedem Algorithmusschritt einen Kreisbogen, definiert durch konstante translatorische und rotatorische Geschwindigkeit im entsprechenden Zeitintervall. Da Übergängen nicht ableitungsstetig sind, wird davon ausgegangen, dass dieser Algorithmus keine glatte Trajektorie ausgibt. Auch der Vector Field Histogram Algorithmus hat keinen Fokus auf glatte Trajektorien. Hier wird in jedem Schritt eine neue Fahrtrichtung für das FTF bestimmt.

Beim Dynamic Window Approach fällt auf, dass die Karte des Suchraums in jedem Algorithmusschritt neu gebildet werden muss. Die Berechnung der Geschwindigkeitskonfigurationen, in denen eine Kollision mit einem Hindernis stattfindet, hängt direkt von der aktuellen Konfiguration ab. Bei den anderen beiden Algorithmen wird eine kartesische Karte verwendet, hier besteht die Möglichkeit einen Ausschnitt der globalen Karte auf dem FTF vorzuhalten und mit Sensorinformationen zu aktualisieren. Die eigene Position des FTF kann auf dieser Karte ebenfalls in jedem Rechenschritt aktualisiert werden.

Von den drei betrachteten Algorithmen wird bei zwei erwähnt, dass sie auch bei dynamischen Hindernissen Kollisionen erfolgreich vermeiden konnten. Ein bewegtes Hindernis im Elastic Band Algorithmus bewirkt die Verschiebung des Bandes. Hier kann es jedoch vorkommen, dass sich der Pfad durch das bewegte Hindernis so weit deformiert, dass keine Lösung für einen kollisionsfreien Pfad mehr gefunden werden kann, selbst wenn ein solcher existiert. Für kleine Veränderungen in der Umgebung liefert der Algorithmus jedoch gute Ergebnisse [4]. Auch der Vector Field Histogram Algorithmus wurde in einer Umgebung mit langsam umherlaufenden Menschen erfolgreich getestet [6].

Diese drei Algorithmen lösen das Problem der lokalen Bahnplanung jeweils mit einem anderen Ansatz. Der Elastic Band Algorithmus betrachtet stets den globalen Pfad und

justiert diesen während der Fahrt. Der Dynamic Windmik des FTF direkt in die lokale Bahnplanung mit ei der berechneten Werte auf die Möglichkeiten des FTF Histogram Algorithmus legt einen großen Fokus auf d der Eingangssignale.

Für jeden dieser Algorithmen muss die Wegsuche sowo naten, ohne Kollisionen mit Hindernissen, als auch geschwindigkeitsvorgaben an das Fahrzeug, durchgeführt algorithmus beinhaltet diese zeitliche Komponente im Algorithmen nicht.

Vor Allem in Bezug auf glatte Trajektorien müssen d werden. Hierzu kommt z.B. eine nachträgliche Glättu

Gerade im Umgang mit dynamischen Hindernissen muss performantes Ergebnis nicht Veränderungen oder Zusä notwendig sind. Hier gibt es die Möglichkeit, auf B wegten Hindernisses eine Art „Sicherheitsschatten“ gültige Lösung für den Algorithmus zu betrachten. Z möglich, wie beispielsweise für den Elastic Band AI Band nachträglich auf der anderen Seite um ein Hind eine zu starke Deformation zu vermeiden.

3.3.6 Erstellung eines dynamischen Modells

Für die lokale Bahnplanung sind die dynamischen Eig Bedeutung. Sie legen fest, mit welchen Beschleunigu FTF fahren kann und bilden damit die Bedingungen, d halten muss [12].

Das hier vorgestellte Modell wird im Projekt SmartF Fahrt des FTF die Masse der zugeladenen Nutzlast zu dieser Masse lassen sich dann präzise Vorgaben über für die lokale Bahnplanung ableiten. Eine Schätzung wicht der Nutzlast zwischen verschiedenen Fahrten v

Das FTF, das modelliert werden soll, besitzt zwei u beiden Seiten einer Plattform, die mit einem Hebeme gestattet ist. So kann es unter einen Tisch fahren transportieren und wieder absenken, um Damit das FTF nicht kippt, sind an den vier Ecken d lastrollen mit zwei rotatorischen Freiheitsgraden a drehen sich in die Fahrtrichtung des Fahrzeugs und kung der Bewegungsfreiheit des FTF.

ow Approach bezieht die Dyna und macht so eine Begrenzung überflüssig. Der Vector Field ie Darstellung und Verwendung

hl für die räumlichen Koordi in der Zeit, nämlich durch Ge t werden. Der Elastic Band AI- Gegensatz zu den anderen beiden

ie Algorithmen noch verbessert ng infrage.

überprüft werden, ob für ein tze zu einem der Algorithmen asis der Geschwindigkeit des be- des Hindernisses ebenfalls als un- udem sind Fallunterscheidungen gorithmus eine Möglichkeit, das ergebnis verlaufen zu lassen, um

enschaften des FTF von großer ngen und Geschwindigkeiten das ie die lokale Bahnplanung ein-

TF verwendet, um während der bestimmen. Unter Kenntnis die Grenzen der FTF-Dynamik ist notwendig, wenn Art und Ger- ariieren.

nabhängige angetriebene Räder zu chanismus für die Nutzlast aus- und diesen anheben, um ihn zu die Ladung a bstustellen. er rechteckigen Plattform Schwer- angebracht. Diese passiven Rollen bringen damit keine Einschrän-

Die beiden angetriebenen Räder des FTF werden unabhängig voneinander modelliert. Dazu gehört der Stromregler des Antriebs, der Motor und das Getriebe des jeweiligen Rades. In Abbildung 5 beschreiben die Blöcke „Antriebssystem rechts“ und „Antriebssystem links“ diese Modelle. Input für das jeweilige Antriebssystem ist die Soll-Winkelgeschwindigkeit für das entsprechende Rad. Output des Antriebssystems ist das Motormoment, das aufgebracht werden muss, um die entsprechende Geschwindigkeit zu erreichen.

Aus diesen Motormomenten können die Beschleunigungen und Geschwindigkeiten der beiden Räder errechnet werden. Hierzu wird die dynamische Bewegungsgleichung (3) in der Darstellung von Euler/Lagrange verwendet [13]:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial L}{\partial \mathbf{q}} = \mathbf{Q}^T \quad (3)$$

Dabei ist die Lagrange-Funktion als die Differenz aus kinetischer und potentieller Energie nach (4), \mathbf{q} der Vektor der verallgemeinerten Koordinaten und \mathbf{Q} der Vektor der verallgemeinerten Kräfte und Momente.

$$L = E_{kin} - E_{pot} \quad (4)$$

Im speziellen Fall des FTF wird angenommen, dass das Fahrzeug sich nur in der Ebene bewegt. Unter dieser Annahme ist die potentielle Energie $\frac{\partial L}{\partial t} = 0$ [12].

Die kinetische Energie kann in drei Komponenten eingeteilt werden: die beiden getrennt betrachteten Energien der translatorischen und rotatorischen Bewegung sowie die Rotationsenergie der Motoren und Räder.

Für die verallgemeinerten Kräfte \mathbf{Q}^T werden die Motormomente des linken und rechten Rades sowie die viskose Reibung für jedes der beiden Räder angenommen.

Die Beschleunigungen und Geschwindigkeiten der beiden Räder sind in Abbildung 5 als Eingang für den Block „FTF-Kinematik“ gezeigt. Aus ihnen wird die Position, Geschwindigkeit und Beschleunigung des Fahrzeugs errechnet. Dies geschieht unter Verwendung der kinematischen Zusammenhänge am Fahrzeug, wie in (5), (6) und (7) dargestellt.

$$\boldsymbol{\omega} = \frac{r}{d} \cdot \mathbf{v}_w^T \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (5)$$

$$\mathbf{v}_t^T = \frac{r}{2} \cdot \mathbf{v}_w^T \cdot \begin{bmatrix} \cos(s_\varphi) & \sin(s_\varphi) \\ \cos(s_\varphi) & \sin(s_\varphi) \end{bmatrix} \quad (6)$$

$$\dot{\omega} = \frac{r}{2} \cdot \left(\mathbf{a}_w^T \cdot \begin{bmatrix} \cos(s_\varphi) & \sin(s_\varphi) \\ \cos(s_\varphi) & \sin(s_\varphi) \\ -\sin(s_\varphi) & \cos(s_\varphi) \\ -\sin(s_\varphi) & \cos(s_\varphi) \end{bmatrix} + \mathbf{v}_w^T \cdot \omega \right) \tag{7}$$

Die Winkelgeschwindigkeit ω hängt hier vom Quotienten aus Radius und Abstand zwischen den Rädern d , sowie der Differenz der beiden Radgeschwindigkeiten ab. Die Radgeschwindigkeiten (Winkelgeschwindigkeiten) sind dabei mit dem Vektor \mathbf{v}_w beschrieben.

Der Vektor der translatorischen Geschwindigkeiten \mathbf{v}_t ergibt sich aus der Summe der Radgeschwindigkeiten in Abhängigkeit vom Radradius und vom Winkel des Fahrzeugs zum Referenz-Koordinatensystem.

Für die Berechnung der Winkelbeschleunigung $\dot{\omega}$ wird zusätzlich noch die Beschleunigung der Räder, \mathbf{a}_w benötigt.

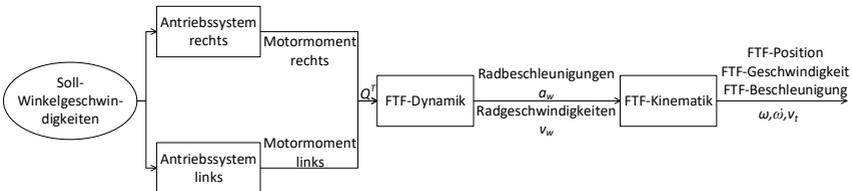


Abbildung 5: Schematische Darstellung des FTF Modells

Mit diesem Modell kann nun die Masse des FTF während der Fahrt bestimmt werden. Dazu werden die vom Modell für eine angenommene Masse berechneten Werte mit den vom FTF gemessenen Werten verglichen. Um die Genauigkeit weiter zu steigern sind Störfaktoren wie Reibung und Unebenheiten der Fahrbahn ebenfalls modelliert, in dieser Beschreibung werden sie der Übersichtlichkeit halber allerdings ausgeklammert.

In der hier vorgestellten Arbeit wird dieses Modell jedoch zu einem anderen Zweck benutzt. Mit dem Modell kann unter der Annahme einer bekannten Masse die Bewegung des FTF unter verschiedenen Vorgabegeschwindigkeiten simuliert werden.

3.3.7 Die Programme ROS und Gazebo

Im folgenden Kapitel wird die Modellierung des FTF als Softwaresimulation vorgestellt. Dabei kommen die open-source Programme ROS und Gazebo zum Einsatz.

ROS (Robot Operating System) bezeichnet sich als „Meta-Betriebssystem“ für Roboter. Es ist ein Framework, um die zum Betrieb eines Roboters notwendige Software zu nutzen und selbst zu schreiben [8]. ROS bietet in der Rolle einer Middleware eine Möglichkeit zur nachrichtenorientierten Kommunikation verschiedener Softwarekomponenten eines Roboters. Ebenfalls ist es über das Framework möglich, auf eine große Anzahl an roboterspezifischen Libraries zurückzugreifen. So muss nicht jeder einzelne Aspekt der Robotersteuerung selbst entwickelt werden.

Zusätzlich bietet ROS eine Reihe von Tools z.B. zur Orchestration einzelner Teilaufgaben der Robotersoftware, zum Aufzeichnen von Versuchen und Testen des Roboters oder zum Anzeigen der Sensordaten.

Dabei können mit ROS sowohl reale Roboter angesteuert werden als auch Robotersimulationen, für Software-In-The-Loop-Tests. ROS ist also eine Sammlung von Werkzeugen, Libraries und Konventionen, die es vereinfachen, ein komplexes aber dennoch funktionsfähiges Roboterverhalten zu generieren [8].

ROS selbst kann allerdings keine physikalischen Gegebenheiten des Roboters selbst und seiner Umgebung simulieren. Zu diesem Zweck ist ein Simulationsprogramm erforderlich.

Gazebo ist ein solches Simulationsprogramm, das im dreidimensionalen Raum arbeitet. Hier können Modelle von Robotern, anderen Gegenständen und Gebäuden oder Outdoor-Umgebungen erstellt werden. Der Simulation nutzt eine Physics Engine, um die wirkenden Kräfte einzubeziehen. Zudem können simulierte Sensoren und Kameras die simulierte Welt erfassen. In einer solchen Umgebung können dann Roboteralgorithmen getestet und Roboter entworfen werden [7].

3.4 Simulation des FTF der arculus GmbH

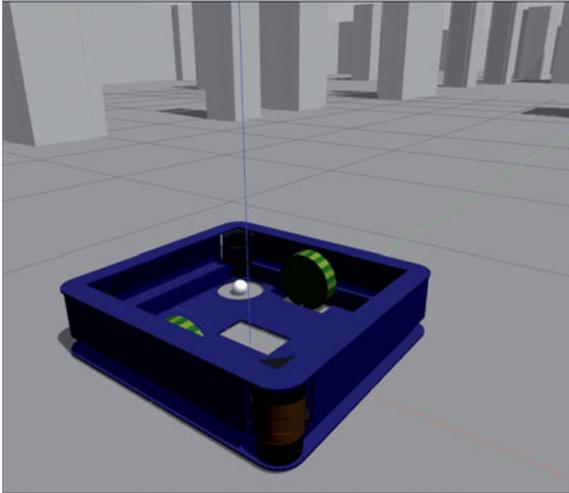


Abbildung 6: Modell des FTF in Gazebo.

Um die Laserscanner und die Kamera mit ihren Eingangsdaten zu simulieren, muss ein dreidimensionales Modell vorliegen, aus dem die erfassten Daten generiert werden können. Eine solche Simulation ist mit Gazebo möglich. In der Simulation werden die Sensoren und die Umgebung beschrieben. Es existiert eine Schnittstelle zu ROS, um die Sensordaten zu verarbeiten und den Roboter zu steuern.

In Gazebo, wie auch in ROS wird die Geometrie der Roboter-Einheiten beschrieben, die sie ausmachen. In der realen Welt wird die reale Konstruktion eine Vereinfachung getroffen, z. B. werden die Räder jeweils als Ganzes modelliert. Für jede solche geometrische Form wird eine Vereinfachung getroffen, z. B. wird ein Zylinder als Kugel etc.) oder groben Meshes und dient zur Vereinfachung der Berechnung von Kollisionen. Zur Darstellung können kompliziertere Meshes, Farben und Materialien angegeben werden. Abbildung 6 zeigt das Modell des FTF in Gazebo. Für die Berechnung der physikalischen Eigenschaften wird in Gazebo zusätzlich die Masse und Trägheitsmatrix der Körper angegeben. Diese Berechnung wird in Gazebo durchgeföhrt. Gelenke zwischen diesen einzelnen Baugruppen werden einzeln spezifiziert. Hier können die Freiheitsgrade passend eingeschränkt werden um z.B. einen Sensor mit dem Chassis fest zu verbinden, ein Rad an

gssignalen zu simulieren, muss erfassten Daten generiert werden. Dazu werden das FTF als Roboter existiert eine Schnittstelle zu ROS, um die Sensordaten zu verarbeiten und den Roboter zu steuern.

Roboter anhand der konstruktiven Vereinfachung wird im Vergleich zur tatsächlichen Konstruktion eine Vereinfachung getroffen, z. B. werden die Räder jeweils als Ganzes modelliert. Für jede solche geometrische Form wird eine Vereinfachung getroffen, z. B. wird ein Zylinder als Kugel etc.) oder groben Meshes und dient zur Vereinfachung der Berechnung von Kollisionen. Zur Darstellung können kompliziertere Meshes, Farben und Materialien angegeben werden. Abbildung 6 zeigt das Modell des FTF in Gazebo. Für die Berechnung der physikalischen Eigenschaften wird in Gazebo zusätzlich die Masse und Trägheitsmatrix der Körper angegeben. Diese Berechnung wird in Gazebo durchgeföhrt. Gelenke zwischen diesen einzelnen Baugruppen werden einzeln spezifiziert. Hier können die Freiheitsgrade passend eingeschränkt werden um z.B. einen Sensor mit dem Chassis fest zu verbinden, ein Rad an

Freiheitsgraden beweglich zu lassen. Motoren werden mit der entsprechenden Geschwindigkeit bewegt.

Für einzelne solche Komponenten können nun zusätzlich Plugins geladen werden. Diese ermöglichen die Funktionalität der Komponente und/oder ihre Kommunikationsschnittstelle mit dem Rest der Software. Ein Beispiel sind die Sensor Plugins: Diese legen fest, auf welche Art und Weise ein Sensor die simulierte Umgebung wahrnimmt und können über Parameter wie z.B. Sensorrauschen oder der Winkel, in dem sie aufnehmen, konfiguriert werden.

An dieser Stelle zeigt sich, inwiefern die Benutzung eines großen Frameworks Vorteile bietet: Für ROS und Gazebo sind viele Sensoren und andere Komponenten bereits modelliert und über das Internet verfügbar.

3.4.1 Einsatz des dynamischen Modells im Robotersimulator Gazebo

In den drei vorangegangenen Kapiteln wurde die Simulation der FTF-Dynamik mit MATLAB Simulink, der Aufbau einer FTF-Steuerung zu Testzwecken in ROS und eine Simulation des FTF mit seinen Sensoren und seiner Umgebung in Gazebo vorgestellt. Diese Simulationen werden im Projekt gemeinsam verwendet, um die lokalen Bahnplanungsalgorithmen und eine Objekterkennung mit dem Kamerasystem zu testen.

ROS und Gazebo werden häufig gemeinsam verwendet, eine entsprechende Schnittstelle ist vorhanden. Es ist mit geringem Aufwand möglich, Nachrichten zwischen ROS und Gazebo im ROS-Messaging System auszutauschen. So kann die FTF-Steuerung aus ROS direkt z.B. die Daten der simulierten Laser scanner aus Gazebo oder die Bilderkennung im ROS-Framework das simulierte Kamerabild aus Gazebo nutzen.

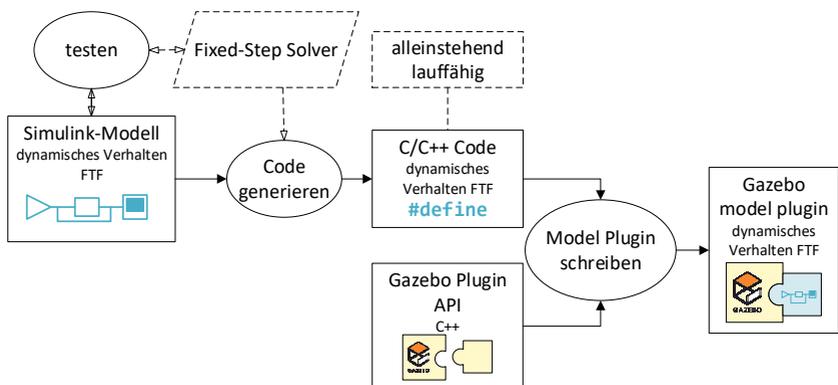


Abbildung 7: Schematische Darstellung der Erzeugung des Gazebo Plugins, welches die FTF-Dynamik abbildet.

Nun soll auch die Dynamiksimulation, die in Simulink Gazebo Modell genutzt werden. Über die Robotics System Toolbox über ROS-Messages und eine Schnittstelle für Gazebo auf beide Programme zugriff werden. Allerdings ist es über diese Schnittstelle, die Position und/oder Geschwindigkeit des simulierten Gazebo-Modells zu setzen. Stattdessen kommt Gazebo zum Einsatz. Im vorliegenden Fall wird ein Plugin entwickelt, das auf das Modell des FTF zugeschnitten ist z.B. „World Plugins“ möglich, die Einfluss auf die Simulation oder „Sensor Plugins“, die sich auf einen simulierten Sensor beziehen. Plugin Funktionen kann auf die simulierten physikalischen Eigenschaften des Modells zugriffen werden. Diese Möglichkeit wird genutzt, um die in Simulink bestimmten Geschwindigkeiten des FTF für das simulierte Modell zu realisieren.

Voraussetzung dafür, die Geschwindigkeit des simulierten Gazebo-Modells mit den in Simulink errechneten Werten zu repräsentieren ist zunächst, dass die Physics Engine in Gazebo nicht zusätzlich Einflüsse wie z.B. Reibung berücksichtigt. Diese werden im Simulink-Modell bereits mit in die Dynamik aufgenommen. Dazu gibt es die Möglichkeit, das Modell des FTF in der Physics Engine als sogenannten kinematischen Körper zu modellieren.

Mehrere Physics Engines unterscheiden drei Arten von Festkörpern, für die alle dynamischen Eigenschaften berechnet werden. Zusätzlich dazu gibt es statische Körper, die sich nie bewegen können. Andere Körper können aber mit ihnen kollidieren, die reaktive Kraft wirkt sich auch entsprechend aus. Und als drittes gibt es „Kinematische Körper“ [15]. Diese können sich explizit bewegen, allerdings geschieht dies nicht unter dem Einfluss der Physics Engine. Wenn sie mit dynamischen Körpern kollidieren, verdrängen sie diese, allerdings wirkt auf den kinematischen Körper keine Kraft aus der Kollision, die seine Bewegung beeinflusst. Kollisionen zwischen statischen und kinematischen Körpern werden erkannt, haben aber keinen Einfluss auf die beteiligten Körper [15].

Die Erstellung des Plugins ist in Abbildung 7 skizziert. Das Simulink-Modell ganz links kann für sich selbst auf Plausibilität und Genauigkeit getestet werden. Dabei ist wichtig, dass ein fixed-Step Solver eingesetzt wird. Ein variable-Step Solver kann nicht in den Simulationstakt von Gazebo eingebunden werden. Aus dem Simulink-Modell wird C oder C++-Code generiert. Dieser Code ist ohne MATLAB allein stehend lauffähig. Die Eingänge des Modells können gesetzt werden und Simulationsschritte können einzeln durchgeführt werden. Auf die Ausgänge des Simulationsmodells kann zugriffen werden.

Dieser C(++)-Code kann nun in einem Gazebo Model Plugin verwendet werden, das

implementiert ist, mit dem ROS-System Toolbox [14] kann in MATLAB Gazebo auf beide Programme zugriff werden. Allerdings ist es über diese Schnittstelle nicht ohne weiteres möglich, die Position und/oder Geschwindigkeit des simulierten Gazebo-Modells zu setzen. Stattdessen kommt Gazebo zum Einsatz. Im vorliegenden Fall wird ein Plugin entwickelt, das auf das Modell des FTF zugeschnitten ist z.B. „World Plugins“ möglich, die Einfluss auf die Simulation oder „Sensor Plugins“, die sich auf einen simulierten Sensor beziehen. Plugin Funktionen kann auf die simulierten physikalischen Eigenschaften des Modells zugriffen werden. Diese Möglichkeit wird genutzt, um die in Simulink bestimmten Geschwindigkeiten des FTF für das simulierte Modell zu realisieren.

Voraussetzung dafür, die Geschwindigkeit des simulierten Gazebo-Modells mit den in Simulink errechneten Werten zu repräsentieren ist zunächst, dass die Physics Engine in Gazebo nicht zusätzlich Einflüsse wie z.B. Reibung berücksichtigt. Diese werden im Simulink-Modell bereits mit in die Dynamik aufgenommen. Dazu gibt es die Möglichkeit, das Modell des FTF in der Physics Engine als sogenannten kinematischen Körper zu modellieren.

Mehrere Physics Engines unterscheiden drei Arten von Festkörpern: Zum einen solche, die alle dynamischen Eigenschaften berechnet werden. Zusätzlich dazu gibt es statische Körper, die sich nie bewegen können. Andere Körper können aber mit ihnen kollidieren, die reaktive Kraft wirkt sich auch entsprechend aus. Und als drittes gibt es „Kinematische Körper“ [15]. Diese können sich explizit bewegen, allerdings geschieht dies nicht unter dem Einfluss der Physics Engine. Wenn sie mit dynamischen Körpern kollidieren, verdrängen sie diese, allerdings wirkt auf den kinematischen Körper keine Kraft aus der Kollision, die seine Bewegung beeinflusst. Kollisionen zwischen statischen und kinematischen Körpern werden erkannt, haben aber keinen Einfluss auf die beteiligten Körper [15].

Die Erstellung des Plugins ist in Abbildung 7 skizziert. Das Simulink-Modell ganz links kann für sich selbst auf Plausibilität und Genauigkeit getestet werden. Dabei ist wichtig, dass ein fixed-Step Solver eingesetzt wird. Ein variable-Step Solver kann nicht in den Simulationstakt von Gazebo eingebunden werden. Aus dem Simulink-Modell wird C oder C++-Code generiert. Dieser Code ist ohne MATLAB allein stehend lauffähig. Die Eingänge des Modells können gesetzt werden und Simulationsschritte können einzeln durchgeführt werden. Auf die Ausgänge des Simulationsmodells kann zugriffen werden.

in C++ implementiert wird. Es besitzt einige Methoden, die zu bestimmten Zeiten in der Simulation aufgerufen werden, z.B. beim Laden eines Plugins in Gazebo oder während eines Simulationsschritts. In der Methode, die in jedem Simulationsschritt ausgeführt wird, kann nun ein Schritt im Dynamikmodell mit Inputs aus der Gazebo-Simulation erfolgen. Die Ergebnisse der Simulation können als ROS-Messages für andere Komponenten der FTF-Steuerung weitergegeben werden. Innerhalb des Plugins wird die Geschwindigkeit des Körpers in Gazebo und dessen n Physiksimulation auf den Wert der Dynamiksimulation errechneten Wert gesetzt. So bewegt sich der Körper auf die Weise, wie die Dynamiksimulation es vorgibt.

3.4.2 Vergleich der Position in den beiden Simulationen

In der Simulation des FTF wird die Bewegung des FTF mit dem Simulink-Modell errechnet und in Gazebo angezeigt. Weitere simulierte Aspekte wie die Datenerfassung durch Sensoren oder die Kollisionserkennung finden in Gazebo statt. Nun soll überprüft werden, ob eine solche Kopplung der beiden Simulationstools möglich ist und korrekte Ergebnisse liefert. Konkret soll ausgeschlossen werden, dass die Physics Engine in Gazebo die Bewegung des FTF-Modells zusätzlich zur Vorgabe aus Simulink beeinflusst.

Die Kopplung der Simulationen findet in jedem Simulationsschritt statt. Voraussetzung dafür ist, dass die Schrittweite beider Simulationen fest und gleich groß ist. Nun wird in jedem gekoppelten Simulationsschritt ein Schritt im Simulink-Modell mit den aktuellen Soll-Winkelgeschwindigkeiten für das FTF durchgeführt. Eine der Ausgaben des Simulink-Modells ist die Geschwindigkeit des FTF für die entsprechende Vorgabe. Diese Geschwindigkeit wird auf das FTF-Modell in Gazebo gegeben. Gazebo berechnet nun mit dieser Geschwindigkeit die neuen Positionsdaten des Fahrzeugs und überprüft, ob Kollisionen mit anderen Körpern in der Simulation vorliegen.

Nun soll gezeigt werden, dass die Simulationsergebnisse aus Gazebo und Simulink übereinstimmen. Hierbei ist zu beachten, dass die Simulink-Simulation basierend auf den Geschwindigkeitsvorgaben für die Räder des FTF dessen Geschwindigkeit, Position und Beschleunigung berechnet. Gazebo berechnet auf Basis dieser Geschwindigkeit die Position des Fahrzeugs in der simulierten Szene. Position und Beschleunigung aus dem Simulink-Modell werden für die gekoppelte Simulation nicht verwendet. Um auszuschließen, dass in der Berechnung von Gazebo physikalische Effekte wie Reibung Einfluss auf die berechnete Position nehmen, werden die Positionsdaten in beiden Simulationen in jedem Simulationsschritt aufgezeichnet und verglichen.

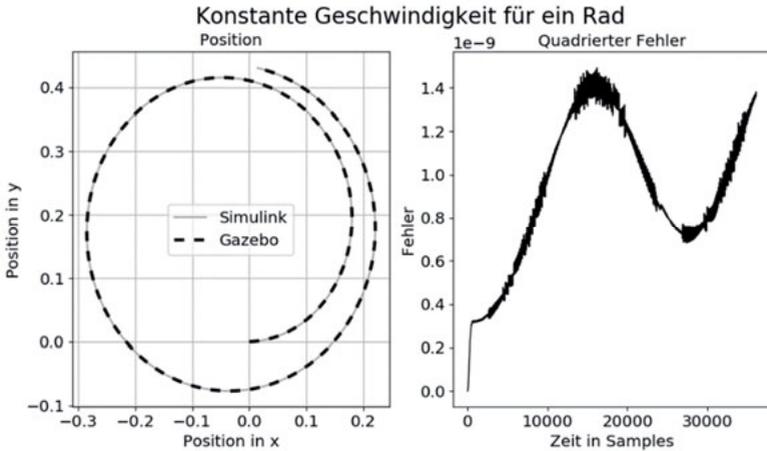


Abbildung 8: Position und quadrierter Fehler für ein angetriebenes Rad mit gleichbleibender Geschwindigkeit.

Abbildung 8 und Abbildung 9 zeigen jeweils im linken Plot die in beiden Simulationen gemessenen Werte. Dabei ist die Position des FTF in beiden Ebenen dargestellt. Die durchgezogene graue Linie zeigt die Daten der Simulink-Simulation, die gestrichelte schwarze Linie die Daten der Gazebo-Simulation. Die beiden Abbildungen unterscheiden sich durch die Geschwindigkeitsvorgaben an das FTF. Im Versuch zu Abbildung 8 wurde einem Rad eine konstante Geschwindigkeit vorgegeben, im Versuch zu Abbildung 9 wurden die Geschwindigkeiten für beide Räder während der Aufnahme geändert. Es ist zu beobachten, dass die gemessenen Positionen sich nicht maßgeblich unterscheiden, die beiden Linien liegen nahezu exakt aufeinander.

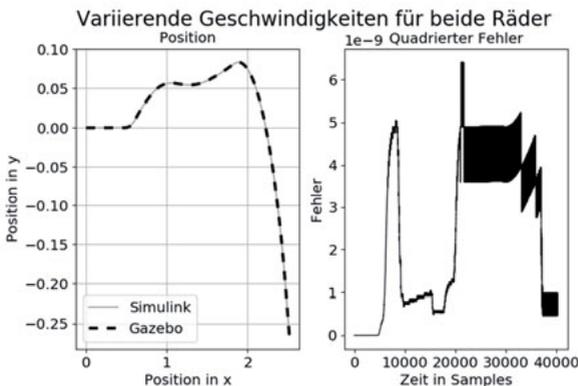


Abbildung 9: Position und quadrierter Fehler für eine Fahrt, in der beide Räder mit wechselnden und verschiedenen Geschwindigkeiten angesteuert wurden.

Um die Unterschiede zwischen den Simulationsergebnissen zu zeigen ist im rechten Plot in Abbildung 8 und Abbildung 9 der quadrierte Fehler (8) in der Positionsmessung abgebildet.

$$e = (x_{Simulink} - x_{Gazebo})^2 + (y_{Simulink} - y_{Gazebo})^2 \quad (8)$$

Hier ist zu beachten, dass der Fehler in einer Größenordnung von 10^{-9} liegt. Dies entspricht der numerischen Genauigkeit der verwendeten Werte. Daher wird davon ausgegangen, dass diese Unterschiede vernachlässigbar sind. Damit konnte gezeigt werden, dass eine Kopplung der Simulation der FTF-Dynamik in Simulink und eine Simulation der Bewegung, der Kollisionserkennung und der Sensorinformationen in Simulink wie im vorigen Kapitel beschrieben möglich ist.

3.5 Ausblick

In diesem Paper wurde das Projekt SmartFTF vorgestellt, in dem fahrerlose Transportfahrzeuge in einer bekannten Umgebung unbekanntem und zum Teil dynamischen Hindernissen ausweichen sollen. Es wurden drei Algorithmen zur lokalen Bahnplanung vorgestellt: der Elastic Band Algorithmus, der Dynamic Window Approach und das Vector Field Histogram-Verfahren. Alle drei verfolgen verschiedene Ansätze, um ihr Ziel zu realisieren. Der Elastic Band Algorithmus betrachtet und verändert einen vor der Fahrt geplanten Pfad. Der Dynamic Window Approach setzt auf die Integration der dynamischen Grenzen des Fahrzeugs in den Algorithmus. Der Vector Field Histogram Algorithmus bezieht direkt rohe Sensorsignale ein und komprimiert diese. Für das Projekt soll einer der Algorithmen ausgewählt werden. In jedem Fall sollte auf eine Erweiterung zur Erzeugung einer glatten Trajektorie Wert gelegt werden. Ebenso muss überprüft werden, inwiefern die Strategie des ausgewählten Algorithmus in Bezug auf dynamische Hindernisse angepasst werden muss.

Im Projekt SmartFTF wurde bisher ein kinematisches FTF entwickelt. Dieses Modell kann zur Schätzung der Beladung des FTF verwendet werden. Ebenfalls wurde eine Softwaresimulation des FTF entwickelt und validiert.

Mit diesem Modell werden die Algorithmen für die lokale Bahnplanung implementiert und optimiert. Dazu gehört eine Anpassung an den Umgang mit dynamischen Hindernissen und die Generierung glatter, ruckoptimierter Verfahrwege.

Um die Algorithmen untereinander und verschiedene Änderungen am selben Algorithmus objektiv bewerten zu können werden Testkriterien festgelegt und eine Messung dieser Kriterien in der Simulation realisiert.

Ein Kameramodul wird in die Fahrzeuge integriert. Die von ihm aufgenommenen Daten werden sowohl zur Verbesserung der Ortung innerhalb des bekannten Bereiches und der Erkennung von Landmarken und Hindernissen verwendet, als auch in einem weiteren Schritt zur Objekterkennung.

Es wird außerdem eine Strategie entwickelt, wann und in welcher Form die Fahrzeuge der FTF-Flotte erkannte und nicht kartographierte Hindernisse an den Leitstand melden.

3.6 Literatur

- [1] Ullrich, G.: Fahrerlose Transportsysteme. Wiesbaden: Springer Fachmedien Wiesbaden 2014
- [2] Kirks, T., Stenzel, J., Kamagaew, A. u. Hompel, M. ten: Zellulare Transportfahrzeuge für flexible und wandelbare Intralogistiksysteme. Logistics Journal Proceedings (2012)
- [3] Ottmann, T. u. Widmayer, P.: Algorithmen und Datenstrukturen. Berlin, Heidelberg: Springer Berlin Heidelberg 2017
- [4] Quinlan, S. u. Khatib, O.: Elastic bands: connecting path planning and control. [1993] Proceedings IEEE International Conference on Robotics and Automation. IEEE Comput. Soc. Press 1993, S. 802–807
- [5] Fox, D., Burgard, W. u. Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine 4 (1997) 1, S. 23–33
- [6] Borenstein, J. u. Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Transactions on Robotics and Automation 7 (1991) 3, S. 278–288
- [7] OSRF: Gazebo. <http://gazebo.org/>, abgerufen am: 10.10.2018

- [8] Open Source Robotics Foundation: About ROS. <http://www.ros.org/about-ros/>, abgerufen am: 10.10.2018
- [9] Hart, P., Nilsson, N. u. Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4 (1968) 2, S. 100–107
- [10] Thomas H. Cormen, Charles Leiserson, Ronald L. Rivest, .: Kapitel 24 und 25. In: Cormen, T. H. (Hrsg.): *Algorithmen - eine Einführung*. München: Oldenbourg 2007
- [11] Marder-Eppstein, E.: ROS dwa_local_planner, 2018. https://wiki.ros.org/dwa_local_planner, abgerufen am: 09.10.2018
- [12] Ivanjko, E., Petrini, T. u. Petrovic, I.: *Modeling of Mobile Robot Dynamics* (2010)
- [13] Hibbeler, R. C., Wauer, J. u. Seemann, W.: *Technische Mechanik. Always learning*. München: Pearson Deutschland 2012
- [14] Robotics System Toolbox. <https://www.mathworks.com/products/robotics.html>, abgerufen am: 19.10.2018
- [15] BulletPhysics 2.83 User Manual

4 Rapid Process Prototyping – Konzept zur drahtlosen Inbetriebnahme und Prozessentwicklung auf Basis von Bluetooth Low Energy

C. Pallasch, S. Storms, W. Herfs, C. Brecher

4.1 Einleitung

Die moderne Produktionstechnik ist charakterisiert durch eine immer fortschreitende Verzahnung von Informations- und Kommunikationstechnologien mit dem Ziel einer ressourcentechnischen Flexibilisierung und zügigen Adaption von Produktionsprozessen. Obwohl die Digitalisierung zugenommen hat, steht die Industrieautomation vor der Herausforderung, diese Flexibilität zu beherrschen und die ständig wachsende Prozesskomplexität in kürzerer Zeit zu bewältigen. [1] In der industriellen Automatisierung werden kabelgebundene Übertragungstechnologien noch immer hauptsächlich in Kombination mit proprietären Feldprotokollen oder gängigen Ethernet-basierten Protokollen eingesetzt. Während die Anforderungen an Latenz und Zuverlässigkeit erfüllt sind, erfordert die Konfiguration und Integration von Automatisierungsgeräten in eine bestehende Kommunikationsinfrastruktur über den Feldbus nach heutigen Maßstäben einen höheren Aufwand. Neben der mechanischen und elektrischen Integration müssen für jedes Automatisierungsgerät Initialisierungsschritte wie Adressvergabe oder die Definition / Aufteilung des Speichers über eine Geräteinformationsdatei etc. durchgeführt werden, bevor es Informationen von einer übergeordneten Steuerung empfangen und als gültiges Gerät im Bus arbeiten kann. [2] Wird ein Automatisierungsgerät durch ein anderes mit ähnlicher Funktionalität ersetzt, müssen diese Schritte beim Abschalten des Systems erneut durchgeführt werden. Darüber hinaus ist der eigentliche Informationsaustausch aufgrund nicht standardisierter Datenformate und der Darstellung verfügbarer Mess- oder Stellgrößen bei der Codeanpassung von speicherprogrammierbaren Steuerungen (SPS) im Falle eines Gerätetauschs mit erheblichen Risiken behaftet. Häufig muss bestehender Programmcode angepasst oder neu geschrieben werden, um die bestehende Prozessfunktionalität zu gewährleisten. [3] Drahtlose Kommunikationstechnologien bieten räumliche Invarianz und damit Mobilität sowie ubiquitäre Verfügbarkeit ohne vorherige Infrastrukturplanung des Übertragungsmediums (wie bei kabelgebundenen Ansätzen etc.). Der Einsatz von drahtlosen Kommunikationstechnologien in Kombination mit semantischen Informationsmodellen birgt ein großes Potenzial für die drahtlose Inbetriebnahme und Konfiguration. Informationsmodelle ermöglichen eine generische und umfassende Beschreibung der ver-

fügbaren Mess- und Regelgrößen sowie der vorhandene n Gerätefunktionen. Die daraus resultierenden Vorteile sind flexible Rekonfiguration, reduzierter Konfigurationsaufwand und direkte Kommunikation mit Feldgeräten.

Im Gegenzug dazu bestehen heutzutage noch Herausforderungen in der Integration und allgemeinen Anwendung drahtloser Übertragungstechnologien in der industriellen Automatisierung. Zum einen sind kabelgebundene Übertragungstechnologien als sogenannte Feldbusse über Jahre entwickelt und an die Anforderungen automatisierungstechnischer Prozesse angepasst worden. Zum anderen nutzt eine Vielzahl von Geräten noch kabelgebundene Übertragungstechnologie n, sodass eine Integration von drahtlosen Technologien nicht ohne Anpassungen erfolgen kann.

Dieser Beitrag stellt ein Konzept zur drahtlosen Inbetriebnahme und Prozessentwicklung auf Basis von Bluetooth Low Energy vor. Hierfür wird auf die allgemeine Integration von drahtlosen Übertragungstechnologien in der industriellen Automatisierung eingegangen, indem sowohl Konzepte und Methoden zur Inbetriebnahme, Konfiguration und Steuerung von Geräten als auch Konzepte zur Adaption von Geräten an drahtlose Übertragungstechnologien behandelt werden.

4.2 Stand der Technik

Im Folgenden werden die für den Beitrag relevanten Themen der Einbindung und Selbstbeschreibung von Geräten als auch der drahtlosen Übertragungstechnologien behandelt.

4.2.1 Dynamisches Einbinden von Geräten

Aufgrund der hohen Wandelbarkeit von Prozessen gewinnen durch eine hohe Individualisierung von Produkten steigt die Komplexität von Automatisierungslösungen. Das Resultat sind häufige Rekonfigurationen und Umrüstungen von Anlagen mit einhergehenden Produktionsstillstandzeiten. Software als auch Treiber der einzelnen Komponenten in der industriellen Automatisierung sind heterogen gestaltet, da es an technologie- und herstellerübergreifenden Standards fehlt. Eine Lösung ist deshalb die Schaffung einer einheitlichen, flexiblen und dynamischen Vorgehensweise zur Maschinen- und Anlagenrekonfiguration. [4]

Plug and Produce: Eine konzeptuelle Herangehensweise zur Lösung der genannten Herausforderung stellt „Plug and Produce“ dar. In Anlehnung an das in der Consumerbranche eingesetzte „Plug and Play“ Prinzip, soll „Plug and Produce“ die Integration und Konfiguration vollständig automatisieren. Dabei können Geräte vollkommen automatisch eingebunden, Treiber installiert und sofort eingesetzt werden. Heutzutage

können jegliche Geräte. Innerhalb eines Computersystems oder über externe Schnittstellen wie USB oder TCP/IP mittels Universal PnP unkompliziert eingebunden und verwendet werden. [5]

Aus der genannten Technologie können Anforderungen an ein industrielles „Plug and Produce“ System abgeleitet werden. Es können fünf grobe Schritte definiert werden:

- Discovery
- Description
- Control
- Eventing
- Presentation

Discovery beschreibt das Verhalten von Geräten oder Kontrollpunkten bei einer initialen oder erneuten Integration in eine Netzwerkstruktur, bei welchem neue Geräte gesucht oder dem Kontrollpunkt bekannt gemacht werden. **Description** beschreibt die Bekanntmachung der vollständigen Funktionalität eines erkannten Gerätes am Kontrollpunkt, damit eine Steuerung des Gerätes erfolgen kann. Nach erfolgtem Informationsaustausch kann der Kontrollpunkt im **Control** Schritt dem Gerät Anweisungen erteilen. Das Gerät reagiert im **Eventing** Schritt auf Anweisungen oder schreibt Statusvariablen bzw. schickt Statusveränderungen zurück zum Kontrollpunkt. **Presentation** Schritt greift der Kontrollpunkt auf eine Informationsdarstellung des Gerätes zurück, um Parameter oder andere Informationen dem Benutzer anzeigen zu können. [6] Ein effektiver Einsatz des genannten Konzeptes unterliegt der Voraussetzung, dass Geräte desselben Typs vorwiegend die gleichen Funktionalitäten und Parameter aufweisen, damit herstellerübergreifende Profile für Gerätetypen erstellt werden können. Für die Erstellung der Profile ist die Definition eines semantischen Informationsmodells unabdingbar. Datengrundlage zur Definition eines Profils stellen Standards, Richtlinien, Spezifikationen, Kataloge oder Datenblätter dar. [5] Ein weiterer Schritt stellt die Modularisierung einer Anlage in Betracht, bei dem Gerätefunktionalitäten als Services angeboten werden. Dies erhöht zwar den Modularisierungsaufwand, allerdings wird eine Reintegration von Geräten in weiteren Anlagen deutlich reduziert. [7] Die Interessengemeinschaft Automatisierungstechnik der Prozessindustrie NAMUR definiert Anforderungen an solche Module. Hierbei müssen verschiedene Versionen von Leitsystemen und Geräten untereinander auf- und abwärtskompatibel sein. Schnittstellen müssen eindeutig, offen, standardisiert und stabil sein, Abhängigkeiten zu Standard IT Systemen (Betriebssystem, Treiber.....) sind zu minimieren und Betriebssysteme, Geräte und Diagnosetools sollen versionsübergreifend kompatibel sein. [8]

Multiagentensysteme: Multiagentensysteme (MAS) vereinen Konzepte des verteilten Rechnens, künstlicher Intelligenz und Semantik zu einem System und ermöglichen dadurch ebenfalls einen dezentralisierten Ansatz zur Entscheidungsfindung. Dies steht im Gegensatz zur bisherigen zentralisierten und inflexiblen Architektur von Produktionssystemen auf Basis der IEC 61131 Norm und dem Einsatz Speicherprogrammierbaren Steuerungen. [9] MAS bestehen aus mehreren Softwareagenten, welche interagieren, um individuelle und kollektive Ziele zu erreichen. Agenten stellen flexible, autonom handelnde Systeme dar, die über Sensorik und Aktorik mit ihrer Umwelt in Kontakt treten. Dabei handeln Agenten proaktiv als auch reaktiv. Beim reaktiven Handeln reagieren diese auf Veränderungen und passen sich dynamisch an ihre Umgebung an. Beim proaktiven Handeln hingegen agieren Agenten zielorientiert und definieren neue Zwischenziele. Da Agenten innerhalb einer großen und komplexen Systemstruktur arbeiten, interagieren diese mit verschiedenen Kommunikationsmöglichkeiten mit anderen Agenten. [10]

Es wurden bis dato verschiedene industrielle Lösungen auf Basis von Multiagentensystemen konzipiert. Dabei hat die „Foundation of Intelligent Physical Agents“ (FIPA) mit der „Agent Communication Language“ (ACL) einen Standard geschaffen, um die Kommunikationsschnittstelle zwischen Agenten zu spezifizieren. Der Standard ist zwar weit verbreitet, bietet allerdings wenig Flexibilität zur Einbindung in bestehende Systeme. Es ist jedoch möglich, Multiagentensysteme auf Basis von OPC UA zu implementieren und so eine „Service Oriented Architecture“ (SOA) zu schaffen. Aufgrund der immer zunehmenden Verbreitung von OPC UA in den heutigen Systemen, kann die Integration von MAS zur adaptiven Produktion stark vereinfacht werden. [11]

Holare Systeme: Holare Systeme kann als eine Weiterentwicklung zu Multiagentensystemen gesehen werden, dessen Basisbaustein das sogenannte Holon ist. [12] Ein Holon ist ein autonomer und kooperativer Baustein eines Produktionssystems zum Transformieren, Lagern und / oder Validieren von Informationen und physikalischen Objekten. Es besteht aus einem informationsverarbeitendem und häufig auch aus einem physikalischen Teil, und kann wiederum Teil eines anderen Holons sein. Dieser Umstand macht holare Systeme beliebig skalierbar. [13] Die Kernkomponenten der Architektur holarer Systeme bilden Autonomie, Kooperation, Selbstorganisation und Rekonfiguration, womit Aufgaben gemeinsam geplant, ausgeführt und bei Bedarf komplette Systemstrukturen rekonfiguriert werden können. [14]

Holonen besitzen unterschiedliche Schnittstellen und Bereiche verschiedener Funktionalität. Abbildung 1 stellt den generischen Aufbau eines Holons dar, bei dem die Inter-Holon-Schnittstelle die Kooperativität und der Kontrollblock die Autonomie realisiert.

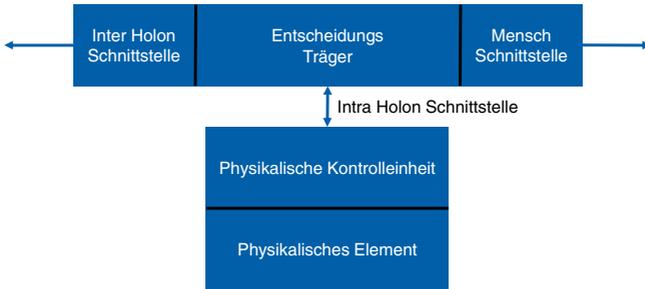


Abbildung 1: Generischer Aufbau eines Holons [14]

Die Referenzarchitektur für holare Produktionssysteme (HMS, Hologonic Manufacturing Systems), dargestellt in Abbildung 2, definiert drei grundlegende Holontypen: ein Ressourcen-, ein Produkt- und ein Auftragsholon. Der Ressourcenholon beinhaltet die Produktionsressource und einen informationsverarbeitenden Teil, welcher die Ressource steuert. Der Produktholon enthält das notwendige Wissen zu Produkt und Prozess, sodass die korrekte Fertigung mit adäquater Qualität gewährleistet werden kann. Der Auftragsholon stellt eine Aufgabe im Produktionssystem dar und ist für eine zeitgemäße Ausführung der gegebenen Aufgabe verantwortlich. [15]

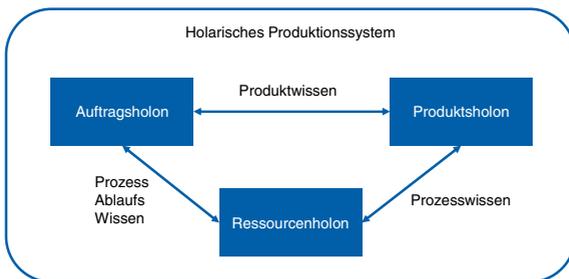


Abbildung 2: Wissensaustausch in einem holaren System [15]

Aufgrund des Informationsaustausches zwischen den einzelnen Holontypen und der Fähigkeit einzelner Holonen, Entscheidungen zu treffen, ist ein HMS ein intelligentes Produktionssystem. Es ermöglicht dynamischere und gerechtere Ressourcenallokationen, sowie simple Integrationen von lokalen und globalen Informations- und Entscheidungsmechanismen durchzuführen als bei zentralisierten Architekturen. [16]

4.2.2 Selbstbeschreibung von Geräten

Um eine herstellerübergreifende Nutzung von Geräten nahtlos zu erreichen, bilden Standards zur Beschreibung von Geräteinformationen einen maßgeblichen Baustein

dar, damit im Rahmen von Industrie 4.0 eine vertikale und horizontale Integration zu bewerkstelligen. Im Laufe der Jahre sind verschiedene Standards unterschiedlicher Arbeitsgruppen konzeptioniert und im Feld umgesetzt worden. Im Folgenden wird im kurzen auf die verschiedenen eingesetzten Standards eingegangen.

le und horizontale Integration zu ne Standards unterschiedlicher worden. Im Folgenden wird im eingegangen.

Field Device Tool und Device Type Manager: Field Device Tool (FDT) ist ein Konzept mit dem die Parametrierung von Produktionsanlagen mittels gerätespezifischen Schnittstellendefinitionen zwischen Komponenten und Engineering Tools vereinheitlicht wird. Somit dient FDT in erster Linie zur Konfiguration und Inbetriebnahme von kompletten Systemen. [17] Die Architektur von FDT wird hersteller- und kommunikationsprotokollinvariant standardisiert, sodass gerätespezifische Daten global abgerufen werden können. [18] In der DIN EN 62435 sind alle Standards und Beschreibungen des FDT/DTM Frameworks spezifiziert.

gen mittels gerätespezifischen Engineering Tools vereinheitlichtur und Inbetriebnahme von urde hersteller- und kommunikationsprotokollinvariant standardisiert, sodass gerätespezifische Daten global abgerufen werden können. [18] In der DIN EN 62435 sind alle Standards und Beschreibungen des FDT/DTM Frameworks spezifiziert.

Zur Nutzung von FDT werden sogenannte Device Type Managers (DTM) benötigt. Diese sind Treiber, die zum einen mit dem Gerät kommunizieren können und zum anderen nach außen hin über eine standardisierte Schnittstelle mit anderen Anwendungen in Kontakt treten können. So können Informationen wie Parameter oder anderweitige Daten mittels grafischer Benutzeroberfläche aus einem Gerät ausgelesen werden. [19]

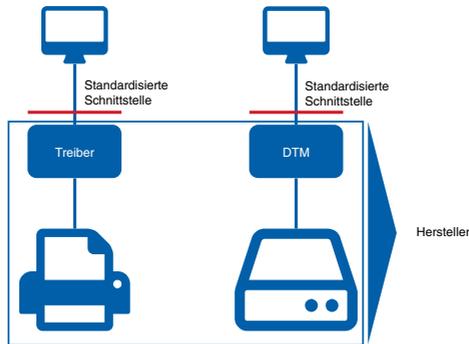


Abbildung 3: Analogie von DTM und Gerätetreiber [20]

Electronic Device Description Language: Die Electronic Device Description Language (EDDL) ist ein Sprachdefinitionsstandard zur Beschreibung von intelligenten Geräten in einem maschinenlesbaren Format. Dabei werden Informationen in sogenannten „Electronic Device Descriptions“ (EDD) abgelegt, die als textbasierte Dokumente auf dem Gerät selber liegen. Syntax und Umfang von EDDL sind in der DIN EN 61804-3 Norm näher beschrieben. Neben der Angabe von Daten und Parametern des Gerä-

tes, beinhaltet EDDL auch die Möglichkeit zur Anzeige verschiedenster grafischer Bedienelemente. Hierfür werden die entsprechenden grafischen Bestandteile von einem Hostgerät, welches die EDD Datei ausliefert, auf Basis von EDD Elementen generiert. [21]

Der wichtigste Aspekt von EDDL ist die rein textbasierte Darstellung von Informationen. Somit können gerätespezifische Informationenplattformunabhängig ausgelesen und bereitgestellt werden. Da EDDL ebenfalls keinen ausführbaren Programmcode enthält, ist eine Installation von Malware unmöglich. [22] Weiterhin ist die Kommunikation mit Geräten ebenfalls spezifiziert mittels sogenannter EDDL Profiles. [23] Ein Auszug unterstützter Kommunikationsprofile ist in Tabelle 4-1 dargestellt.

Tabelle 4-1: Kommunikationsprofile für EDDL [23]

Attribute ID	Kommunikationsprotokoll-profile
0x01	HART Communication Foundation
0x02	Fieldbus Foundation
0x03	PROFIBUS
0x04	PROFINET

Field Device Integration: Das Ziel von Field Device Integration (FDI) ist eine weitgehende Integration von FDI/DTM und EDDL zu einem einheitlichen Format. Das FDI Package ist Mittelpunkt der Technologie und setzt sich aus mehreren Dateien zusammen, wie in Abbildung 4 dargestellt: Das EDD Dokument beschreibt das Gerät, die Logik und die Benutzerschnittstelle. Das optionale benutzerschnittstellen Plug-In basiert auf der Windows Presentation Foundation und bietet frei programmierbare GUI Elemente, wie aus FDT bekannt. Schließlich existieren Anhänge, die verschiedene Zusatzinformationen bieten. Die Logik ist verantwortlich dafür, dass Gerätedaten konsistent bleiben und zum richtigen Zeitpunkt aktualisiert werden.

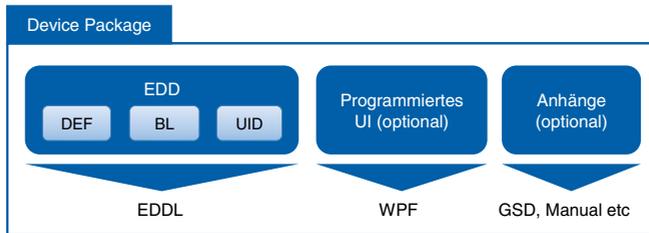


Abbildung 4: Generischer Aufbau eines Device Packages nach [24]

OPC Unified Architecture: OPC Unified Architecture (OPC UA) ist ein Standard zur Vereinheitlichung von Treiberschnittstellen und somit zum generischen Zugriff auf Geräte. Das Kernkonzept von OPC UA stellt einen Adressraum dar, welcher durch verschiedene Knotentypen aufgespannt wird. Alle Knotentypen sowie vorhandene Daten- und Relationstypen basieren auf einer vordefinierten Ontologie [25]. Dabei besitzt ein Knoten eingehende als auch ausgehende Kanten, womit in der Gesamtheit ein Netzwerk aus Relationen aufgebaut wird. Einzelne Knoten sind dabei Informationsträger und stellen z.B. Messwerte, Parametrierinformationen, Methoden oder weitere Typinformationen dar. Auf Basis dieses Adressraums sind OPC UA Services definiert, welche z.B. den allgemeinen Zugriff, die Aufbereitung historischer Daten oder die Kommunikationssicherheit regeln. Durch Kombination der Ontologie und der Services werden in OPC UA Informationsmodelle definiert, welche die Modellierung verschiedener Geräte aber auch anderweitiger größerer Dienste ermöglichen je nach Anwendung und Domäne. Der OPC UA Standard definiert ein Basisinformationsmodell, auf dem weitere anwendungsspezifische Modelle entwickelt werden können. Die Definition und Spezifikation von Informationsmodellen geschieht entweder pro Hersteller auf proprietärer Basis oder wird in einem Gremium definiert.

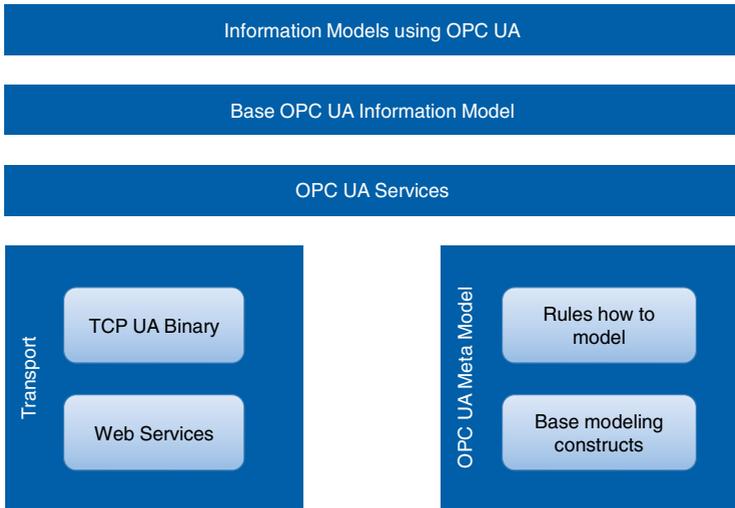


Abbildung 5 Schichtenmodell des OPC UA Stacks

4.2.3 Drahtlose Übertragungstechnologien

Drahtlose Übertragungstechnologien nutzen einen omnipräsenten Übertragungskanal, der mit anderen Kommunikationsteilnehmern geteilt werden muss. Funktechnologien unterscheiden sich grundsätzlich in genutztem Frequenzband, Modulationsart und Protokollstack. Je nach definiertem Anwendungsbereich variieren neben technologische Parameter auch Koexistenzmechanismen, um einen reibungslosen Ablauf der Datenübertragung verschiedener Technologien in gleichen Frequenzbändern zu gewährleisten. Im Folgenden werden gängige Funktechnologien vorgestellt.

Wireless Local Area Network: Das Wireless Local Area Network (WLAN) wurde durch den 802.11 Standard des Institute of Electrical and Electronics Engineers (IEEE) auf PHY und MAC Ebene definiert. Mit diesem Standard wird die kabellose Verbindung von fixierten, portablen und mobilen Stationen in einem lokalen Netzwerk definiert. [26] Es existieren heutzutage verschiedene Umsetzungen des 802.11 Standards, mit verschiedensten möglichen Datenraten und in verschiedenen Frequenzbändern.

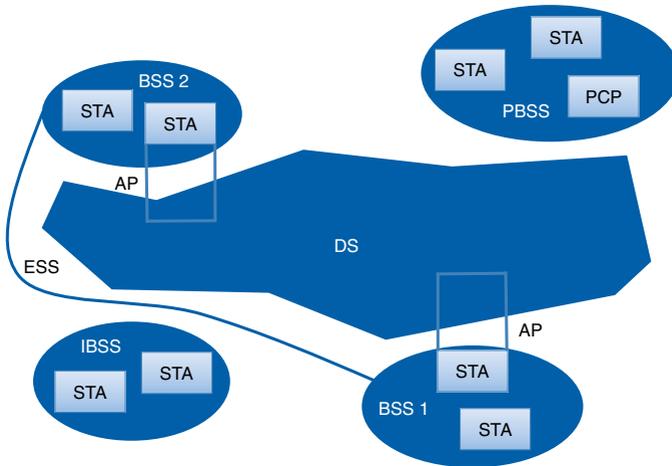


Abbildung 6 Mögliche WLAN 802.11 Netzwerkarchitekturen [26]

Bluetooth Low Energy: Bluetooth Low Energy (BLE) ist aus der Bluetooth Spezifikation als Erweiterung zum herkömmlichen BT Classic entstanden [27]. BLE zielt auf kleine, energieeffiziente Geräte ab, welche nur unregelmäßig kleine Datenpakete verschicken. Kern der neuen Spezifikation ist das Absenden sogenannter Advertisement Pakete zum verbindungslosen Auffinden von Geräten in der Umgebung, welche im Generic Access Profile standardisiert ist. Weiterhin stellen Geräte über das Generic Attribute Profile, strukturell dargestellt in Abbildung 7, Informationen mittels Services, Charakteristiken und Deskriptoren zu Verfügung, wovon eine Reihe von Informationen in der Bluetooth Spezifikation standardisiert ist. In der Version 5.0 von Bluetooth LE wurde die Länge der Advertising Pakete, die Reichweite und die Datenrate erhöht. Bluetooth LE stellt eine Kerntechnologie im Bereich der IoT Technologien dar [28, 29].

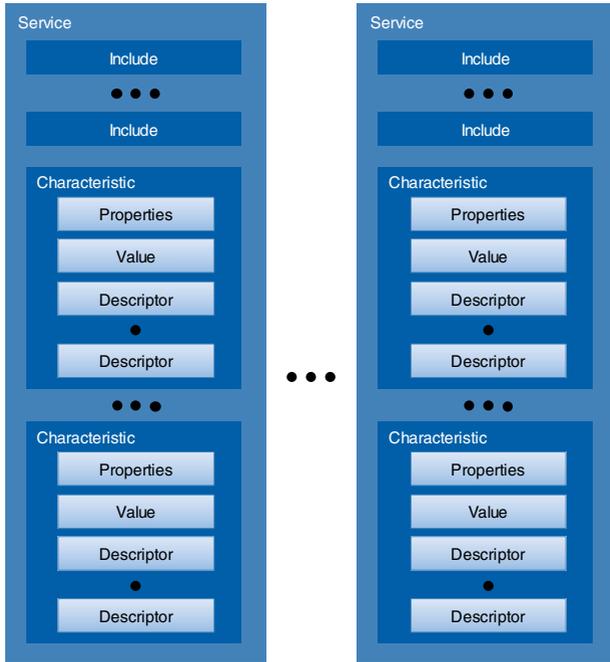


Abbildung 7 Struktureller Aufbau der des Bluetooth LE Generic Attribute Profiles

5G Mobilfunk: Die Nachfolgeneration des Mobilfunkstandards Long Term Evolution (LTE) stellt 5G dar. Verglichen zu LTE, soll das 5G Netzwerk ein 1000-fach höheres Datenvolumen pro Flächeneinheit, die 10 bis 100-fache Anzahl an verbundenen Geräten und Nutzerdatenrate, die 10-fache Akkulaufzeit und eine um das 5-fach geringere Latenz aufweisen. Die Netzwerke der 5ten Generation sollen dichter sein und viele kleine Zellen aufweisen, welche eine große Anzahl an Geräten bei erhöhter Energieeffizienz unterstützen [30]. Dabei soll 5G eine Paketfehlerrate zwischen 10^{-5} bis 10^{-9} besitzen bei einer Ende-zu-Ende Latenz im Bereich weniger Millisekunden. Die vier wesentlichen technologischen Erweiterungen beinhaltet MIMO Beamforming, OFDM mit flexiblen Sub-Trägern, Mini-Slot Übertragung und On-demand Referenzsignale [30–33]. Weiterhin wird das Frequenzspektrum Anteil von unter 1 Ghz bis zu 100 Ghz beinhalten mit Frequenzbändern von bis zu 1 Ghz breite.

NarrowBand IoT: Zur Unterstützung von MTC im zellularen Netz der vierten Generation, LTE-Advanced, hat das Third Generation Partnership Project sich zur Aufgabe gemacht, einen Standard für low-cost Geräte mit niedrigen Datenraten und hoher

Energieeffizienz zu definieren und entwickelte Narrow Band IoT (NB-IoT) [34]. Narrow Band IoT zählt zu den Low Power Wide Area Networks (LPWAN). Damit können über große Distanzen, mehrere Kilometer, Daten bei geringer Leistung übertragen werden, so dass Energieeffizienz gewährleistet ist. Jedoch ist diese Technologie in der möglichen Datenrate stark eingeschränkt. Durch die Anforderung an Leistung und Übertragungsdistanz muss die Modulation der Symbole sehr langsam geschehen [35]. NB-IoT hat eine maximale Datenrate von 160 Bit/S im Uplink und Downlink bei einer Akkulaufzeit von 10 Jahren und einer maximalen Latenz von 10 Sekunden. NB-IoT eignet sich somit für Sensorik mit niedrigem Energieverbrauch und geringen Übertragungskosten.

LoRa / LoRaWAN / SigFox: Low-Power-Wireless Netzprotokolle sind für eine vergleichsweise lange Akkulaufzeit bei geringer Datenrate und hoher Übertragungreichweite optimiert. Long Range (LoRa) ist eine auf der Chirp Spread Spectrum abgeleitete Übertragungstechnologie, welche von der Firma Semtechs erstmalig 2012 in Form von LoRa RF Chips entwickelt wurde. LoRa operiert im 433, 868 oder 915 Mhz Band und hat eine Reichweite im Bereich von 10 bis 20 km bei einem verfügbaren Linkbudget zwischen 155 und 170 dbm und einer Datenrate von bis zu 50 kBit/s. Die Signalübertragung ist unempfindlich gegenüber Störungen, Mehrwegeausbreitung und Fading-Effekten [36]. LoRa wird als Basistechnologie für das Long Range Wide Area Network (LoRaWAN) genutzt. LoRaWAN erweitert LoRa um ein kontrolliertes Medienzugriffsverfahren für mehrere Teilnehmer sowie um Protokolle zum Zugriff auf LoRaWAN Gateways, welche eine Konnektivität zum Internet bereitstellen. Die Infrastruktur des LoRaWAN Netzwerks wird je nach Land und Region entweder durch öffentlich-private oder kommerzielle Gateways bereitgestellt. Der Zugang zum Netzwerk ist somit frei oder mit zu entrichtenden Netzgebühren verbunden. SigFox als weitere Low-Power-Wireless Technologie hat Analogien zu LoRa bzw. LoRaWAN, nutzt allerdings Ultra-Narrow-Band Modulation im 868 bzw. 905 Mhz Band. Sigfox ist darüber hinaus ein französisches Telekommunikationsunternehmen, welches als Provider den Ausbau des eigenen Netzes weltweit vorantreibt [37].

4.3 Drahtlose Geräteintegration

Im Folgenden wird ein Konzept zur drahtlosen Geräteintegration auf Basis von Bluetooth LE vorgestellt. Dabei wird zunächst auf die verschiedenen Stufen drahtloser Geräteintegration und deren Möglichkeiten bzw. Potenziale bzgl. Engineering eingegangen. Anschließend wird das Konzept stufenweise vorgestellt.

4.3.1 Stufen der drahtlosen Geräteintegration

Die Integration drahtloser Übertragungstechnologien hängt stark von vorliegenden Gegebenheiten, gewünschten Anforderungen und Absichten zur Nutzung drahtloser Features ab. Insgesamt lassen sich die drei Stufen Cable Replacement, Wireless Adapted und Wireless Integrated feststellen:

Cable Replacement: Das Cable Replacement zielt darauf ab eine vorhandene kabelgebundene Übertragungslösung durch eine funkbasierte zu ersetzen, wie in Abbildung 8 dargestellt. Dabei bleibt der Übertragungskanal transparent für beide Teilnehmer der Kommunikation erhalten, sodass keine „spürbaren“ Veränderungen für die darüber liegenden Übertragungsschichten auftreten.

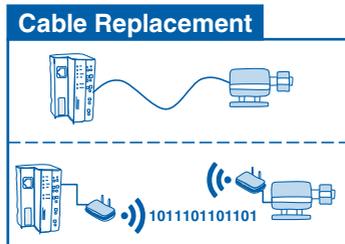


Abbildung 8 Cable Replacement / Niedrigste Integrationsstufe

Bei einem Cable Replacement werden so viele Daten entnommen wie notwendig, um Informationsverluste zu vermeiden und das Protokollverhalten auf beiden Seiten weiterhin aufrecht zu erhalten. Je nach geforderter Übertragungsrate und Zuverlässigkeit muss die verfügbare Kanalbandbreite des Funkkanals soweit und so oft wie notwendig eingespart werden, um Latenzen bzw. Jitter gering zu halten und Verbindungsabbrüche zu verhindern.

Dies kann z.B. dadurch geschehen, dass Nachrichten, die das erste Mal auftreten, in einen Speicher zwischengepuffert und indiziert werden zwischen Sender und Empfänger des Funkkanals. Tritt die Nachricht ein weiteres Mal auf wird nur der jeweilige Index übertragen, um so Kanalbandbreite einzusparen. Eine weitere Möglichkeit zur Einsparung der Bandbreite ist die Übertragung von Unterschieden zwischen bisher aufgetretenen Nachrichtenpaketen.

Eine Cable Replacement lässt sich mit zwei Funkmodulen bewerkstelligen, welche jeweils an einem Ende der Übertragung mit dem entsprechenden Gerät per Kabel verbunden sind. Da es sich hierbei um eine Punkt-zu-Punkt-Verbindung handelt, müssen beide Geräte miteinander gepaired werden. Es werden hierbei keine weiteren Potentiale drahtloser Übertragung genutzt, da es sich um eine reine Protokollemlulation über Funk handelt.

Wireless Adapted: Die nächste Stufe, dargestellt in Abbildung 9, stellt Wireless Adapted dar. Dieser Ansatz ist vom technischen Aufbau her analog zum Cable Replacement. Es werden allerdings Punkt-zu-Multipunkt Verbindungen verwendet und angebundene Geräte melden sich über eine Funkschnittstelle an andere Teilnehmer aktiv oder passiv. Weiterhin können Informationen über einzelne Geräte ausgelesen werden, Parameter gesetzt oder Kommandos abgesetzt werden.

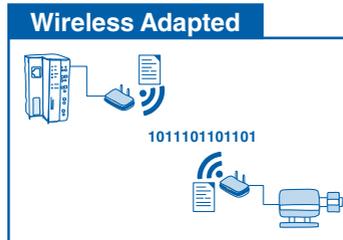


Abbildung 9 Wireless Adapted / Erweiterte Integration

Für die Realisierung von Wireless Adapted muss zunächst die kabelbasierte Kommunikation zum jeweiligen Gerät abgedeckt sein. Hierfür muss im Funkmodul selber oder einem Erweiterungsboard der jeweilige Treiber für das Gerät vorliegen und die Kommunikation mittels definierter Protokoll abgewickelt werden. Weiterhin muss eine Nutzdatenextraktion stattfinden, sodass nur relevante Informationen über die Funkschnittstelle übertragen werden. Somit wird die kabelgebundene Kommunikation adaptiert an die Funkschnittstelle.

Über die Funkschnittstelle werden Geräteinformationen, Parameter, verfügbare Methoden oder Kommandos und weitere Eigenschaften und Fähigkeiten bereitgestellt. Diese Informationen können einzelne Kommunikationsteilnehmer nutzen, um während der Prozessausführung adaptiv auf sich ändernde Randbedingungen zu reagieren.

Aufgrund der Selbstbeschreibungsmöglichkeiten und dem omnipräsenten Zugriff über eine Funkschnittstelle, können ebenfalls einzelne Methoden oder Kommandos in einem Rapid Engineering Ansatz zur Unterstützung der Prozessentwicklung herangezogen werden. Prozessabläufe können so schnell zusammengestellt und getestet werden ohne vorherige Installationsplanung.

Aufgrund der am Markt verfügbaren kabelgebundenen Geräte, stellt diese Integrationsstufe einen unumgänglichen Schritt zur Einführung drahtloser Übertragungstechnologien dar, welcher hier auch genutzt wird, um die im späteren Verlauf vorgestellten Konzepte anwenden zu können.

Wireless Integrated: Diese Integrationsstufe bildet im Prinzip die höchste Stufe dar. Die Konzepte zur Funkschnittstelle der Integrationsstufe Wireless Adapted werden hierbei originalgetreu übernommen. Da in dieser Stufe keine kabelgebundene Verbindung auftritt, müssen allerdings somit keine Geräte treiber und Protokollimplementierung vorliegen, da Informationen direkt über das Funkmodul übertragen werden.

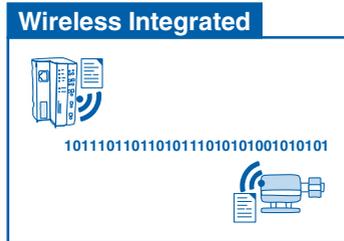


Abbildung 10 Wireless Integrated / Höchste Integrationsstufe

4.3.2 Übersicht des Konzepts

Abbildung 11 stellt das Zielbild und die Methodologie des Beitrags dar. Das Konzept besteht aus einer Kombination von drahtloser Kommunikation, Geräteselbstbeschreibung, generischen Schnittstellen für den Zugang und Unterstützung bei der Prozessentwicklung. Automatisierungsgeräte stellen ihre Fähigkeiten und ihren aktuellen Zustand über die drahtlose Kommunikation zur Verfügung. Mit diesen Informationen in Kombination mit einer geeigneten Basisstation ist es in der Lage, Automatisierungsprozesse zu prototypisieren, indem er Fähigkeiten von Geräten in der umliegenden Umgebung zu entsprechenden Prozessschritten zuordnet.

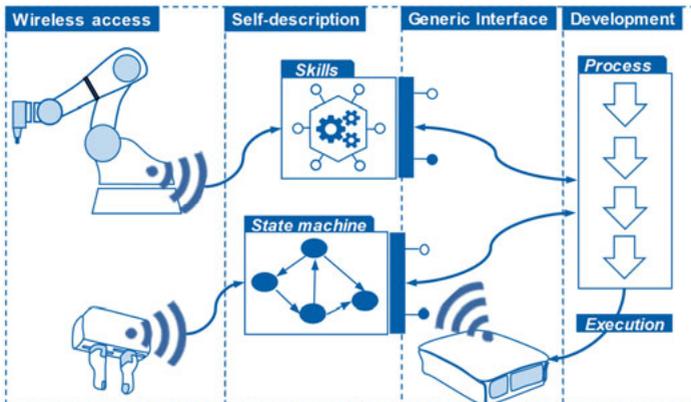


Abbildung 11 Zielbild des Konzepts

4.3.3 Enumeration und Fähigkeiten von Geräten

Die Realisierung von "Plug and Produce" für Geräte erfordert eine automatische Geräteenumeration und die Verwendung einer fähigkeitsbasierten Gerätebeschreibung. Die Geräteerkennung wird durch die Übertragung spezifischer Advertisement Pakete erreicht. Die Struktur des Werbepaketes ist in Abbildung 12 dargestellt und besteht aus einem Flag, einer Sendeleistungspegelangabe, einer „Erscheinungsinformation“ und einem kurzen lokalen Namensfeld. Innerhalb des Flagfeldes sind die ersten niederwertigsten Bits bereits durch die BLE-Spezifikation reserviert, so dass das sechste Bit als Hinweis darauf verwendet wird, dass das jeweilige Feldgerät ein integrierbares Automatisierungsgerät ist. Die anderen Felder werden als Zusatzinformationen verwendet (z.B. kann das Erscheinungsbild wiederverwendet werden, um anzugeben, welcher Gerätetyp vorliegt).

Advertising Packet			
Flags	TX Power Level	Appearance	Short Local Name
0b001xxxx	-21dB	0x1234	KUKA KR16

Abbildung 12 Aufbau eines BLE Advertisement Pakets zur Identifizierung von Geräten

Die fähigkeitsbasierte Gerätebeschreibung muss herstellerunabhängig und äquivalent zwischen Geräten mit gleicher Funktionalität sein. Fähigkeiten werden thematisch klassifiziert, so dass ähnliche Fähigkeiten in Gruppen zusammengefasst werden können. Diese Skillgruppen geben einen groben Überblick über die Funktionalität und können für einen ersten Vergleich herangezogen werden. Zur Darstellung dieser werden GATT-Dienste verwendet. Für bestimmte Gerätetypen können Dienste entweder als erforderlich oder als optional definiert werden. Wenn Dienste erforderlich sind, muss ein Gerätetyp diesen Dienst implementieren, um als "Plug and Produce"-kompatibles Gerät seines Typs zu gelten. Die eindeutige Identifizierung von Diensten durch die Verwendung von UUIDs stellt sicher, dass Dienste geräteübergreifend definiert werden.

Individuelle Fähigkeiten innerhalb der Dienste / Skillgruppen werden durch GATT-Merkmale repräsentiert. Merkmale werden auch durch UUIDs identifiziert. Die semantische Repräsentation und Abstraktion einer Fähigkeit durch ein Merkmal ermöglicht die Trennung einer Funktionalität und einer expliziten Implementierung eines Gerätetreibers. Die Merkmale können als notwendig oder optional definiert werden. Geräte eines bestimmten Typs müssen unterschiedliche Grundfunktionen haben. So müssen beispielsweise Industriegreifer in der Lage sein, Objekte zu greifen und freizugeben. Das Bewegen der Greiffinger in eine definierte Position kann jedoch nur optional sein,

da Greifer ohne bewegliche Finger wie z.B. Vakuumpgreifer existieren. Durch den Vergleich der vorhandenen Eigenschaften kann ein Gerätemodell aufgebaut und mit den erforderlichen Geräteprofilen verglichen werden.

Um die Fähigkeiten der Geräte in Bezug auf ihre möglichen Parameter weiter zu spezifizieren, werden BLE-Deskriptoren verwendet. Die Fähigkeiten werden in drei verschiedene Typen klassifiziert:

- Ausführbare Methoden bzw. Kommandos
- Geräteparameter
- Weitere Fähigkeiten

Um diese zu unterscheiden, wurden eigene Deskriptoren definiert und den Charakteristiken hinzugefügt.

4.3.4 Methoden, Parameter und Eigenschaften

Methoden werden als Charakteristiken mit Hilfe von Deskriptoren dargestellt, um ihre Parameter zu spezifizieren. Jede Methode hat mindestens einen Rückgabeparameter, gefolgt von weiteren Eingabeparametern. Um eine Charakteristik als ausführbare Methode zu identifizieren, wurde der benutzerdefinierte Deskriptor 0x2918 definiert. Er stellt den Rückgabewert der Methode dar und besteht aus einem Feld, das den Datentyp, den sogenannten Formattyp, des Rückgabewertes definiert, und einem zweiten Feld, das den zuletzt berechneten Rückgabewert enthält. Dieser Deskriptor spezifiziert die einfachste Form einer Methode, die keine Eingabeparameter erwartet und einen Rückgabeparameter zurückgibt.

Darüber hinaus ist es zur Beschreibung von Methodenparametern und Gerätefähigkeiten notwendig, die Art des Parameters oder der Fähigkeit genau zu identifizieren. Zu diesem Zweck werden Messniveaus für Skalentypen eingeführt. Die Skalentypen sind in verschiedene Arten unterteilt. Zunächst ist zwischen qualitativen und quantitativen Skalentypen zu unterscheiden. Qualitative haben einen geringeren Informationsgehalt, da sie keine Rangfolge haben. Quantitative können miteinander verglichen werden. Somit kann neben dem direkten Vergleich auch eine Größenbeziehung hergestellt werden. Darüber hinaus können Merkmale in kontinuierlicher und diskreter Form zur Verfügung gestellt werden. Diskret bedeutet, dass sie nur einen endlichen Satz von Werten annehmen können. Kontinuierliche Skalentypen können innerhalb eines oder mehrerer definierter Intervalle unendlich viele Werte annehmen. Insgesamt wurden vier verschiedene Skalentypen definiert, um die Art des Parameters oder der Fähigkeit zu identifizieren: metrisch kontinuierlich, metrisch diskret, ordinal skaliert und nominal skaliert.

Jeder Methodenparameter oder jede Gerätefähigkeit verwendet einen Deskriptor, der angibt, welcher Skalierungstyp verwendet wird, um die zugrundeliegenden Daten auszu- drücken. Die Deskriptorstruktur für jeden Skalentyp ist in Abbildung 13 dargestellt.

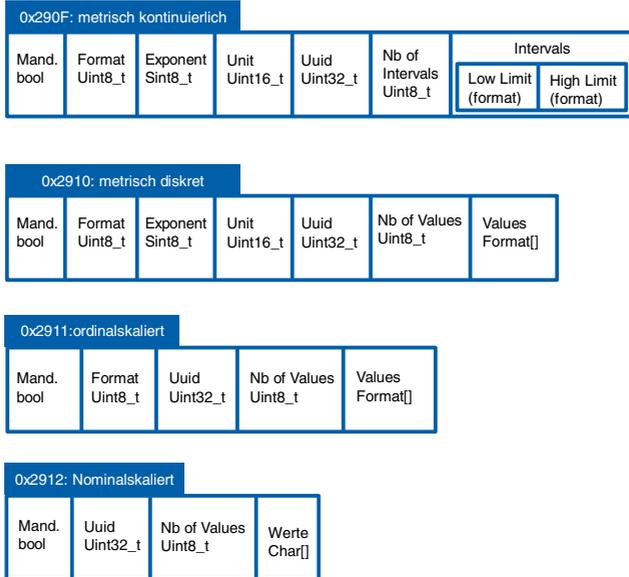


Abbildung 13 Deskriptorstrukturen zur Angabe des Skalentyps

Dieser Ansatz ermöglicht eine retrospektive Beschreibung der Gerätefähigkeiten (Selbstbeschreibung) und kann verwendet werden, um die Verwendung der richtigen Datentypen abzuleiten und sicherzustellen.

4.3.5 Gerätezustand

Während der Prozessausführung muss der Prozesscontroller jederzeit in der Lage sein, das System vollständig abzubilden. Zu diesem Zweck muss jedes Gerät in der , error, done usw.) an den Prozesscontroller zu übertragen, um für einen anderen Zustandsautomaten ausgestattungsmodell als notwendiger Dienst hat daher eine einzige Charakteristik, die beim Lesen den aktuellen Zustand zu geben, daher werden Notifications verwendet, um ein Ereignis dieser Änderungen zu erreichen. Um ein Bild aller möglichen Zustände zu erhalten, werden diese als Deskriptoren der aktiven Zustandscharakteristik hinzugefügt. Zustände werden global durch UUIDs identifiziert. Es

sind einige generische Zustände, wie z.B. uninitialized, idle oder error, definiert, in denen sich alle Geräte befinden können.

Abbildung 14 zeigt beispielhaft den Zustandsautomaten eines servoelektrischen Greifers.

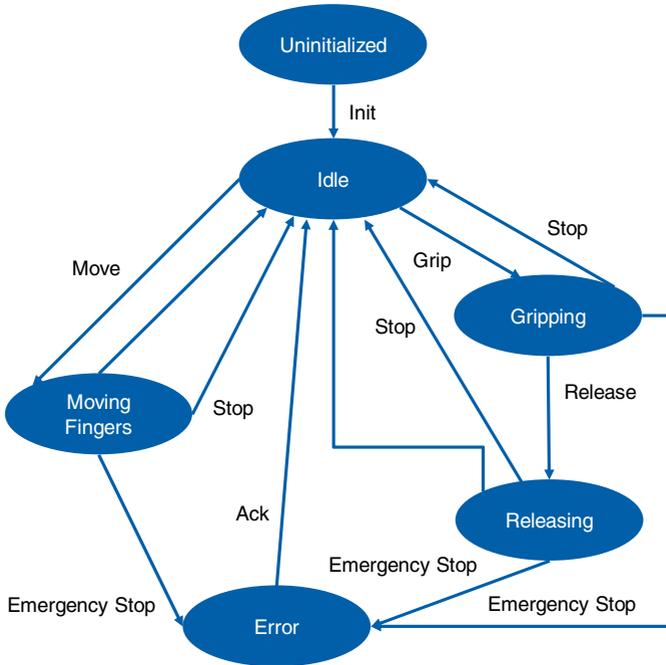


Abbildung 14 Zustandsautomat eines servoelektrischen Greifers

4.3.6 Online und offline Prozessentwicklung

Mit dem oben erläuterten Selbstbeschreibungskonzept ist es möglich, nach nahegelegenen Geräten zu suchen und die Gerätefähigkeiten mit den entwickelten Prozessen abzugleichen. Darüber hinaus können Methoden und Parameter von Geräten erfasst und genutzt werden, um Ingenieure in ihrer Prozessentwicklungsphase zu unterstützen.

Basierend auf den bisher vorgestellten Konzepten stellt sich ein Ansatz zur funktionsbasierten Unterstützung in der Entwicklung von automatisierungstechnischen Prozessen wie folgt dar:

- Drahtlose Feldgerätenumeration

- Scan von Methoden, Parametern und Fähigkeiten
- Definition von Prozessschritten
- Filterung von Prozessschritten nach Gerätefähigkeiten
- Zuordnung von Schritten zu gescannten Feldgeräten
- Drahtlose Ausführung des Prozesses

Bevor auf die Prozessentwicklung eingegangen wird, wird auf die Rolle der Basisstation bzw. des Prozesscontrollers eingegangen.

Basisstation: Die Basisstation ist der zentrale Knotenpunkt der Netzwerkarchitektur. Diese verbindet sich mit den Geräten und setzt Kommunikationsab. Grundvoraussetzung für die Basisstation ist die Fähigkeit, die Rolle eines BLE-Masters zu übernehmen. Die Station enthält eine Liste aller Prozessschritte und übernimmt die Geräteverwaltung. Die Basisstation ordnet den Geräten einzelne Prozessschritte zu und prüft, ob die Geräte für den zugeordneten Schritt geeignet sind. Darüber hinaus wird kontinuierlich nach neuen Geräten gesucht, während die alten Geräte auf ihre Verfügbarkeit überwacht werden. Vor der Prozessausführung initialisiert die Basisstation die Geräte, indem sie einen Satz von Initialisierungsmethoden aufruft und Geräteparameter einstellt. Während der Ausführung wird der Prozess auf Fehler überwacht. Darüber hinaus bietet die Basisstation dem menschlichen Bediener Diagnosefunktionen, die anzeigen, welches Gerät sich in Reichweite befindet und welche Prozessschritte welchem Gerät zugeordnet sind. Die Konfiguration und Zuordnung ist vom Bediener manuell einstellbar, wenn die Station nicht automatisch die optimale Anpassung von Prozessschritt zu Gerät findet. Schließlich bietet die Station dem Bediener oder Ingenieur die Möglichkeit, Prozesse zu definieren und auszuführen.

Prozessentwicklung: Da Informationen mit dem vorgestellten Ansatz gesammelt werden können, ist es möglich, den Ingenieur oder Betreiber in der Prozessentwicklungsphase zu unterstützen und sogar neue Möglichkeiten für die Prozessentwicklung zu eröffnen. Offline- und Online-Prozessentwicklung als zwei mögliche Methoden der Prozessentwicklung werden bereitgestellt.

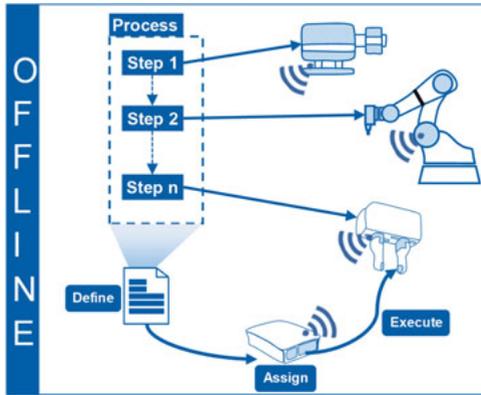


Abbildung 15 Offline Entwicklung von Automatisierungsprozessen

In der Offline-Prozessentwicklung, wie in Abbildung 15 dargestellt, definiert der Ingenieur nachfolgende Prozessschritte und vergibt in jedem Schritt. In dieser Phase ist eine direkte Interaktion mit den Geräten nicht zwingend erforderlich. In einer zweiten Phase wird ein Gerät jedem Schritt zugeordnet. Basierend auf den Prozessschritten werden Geräte vorgeschlagen. Diese müssen den Prozessschrittenanforderungen übereinstimmen (z. B. wenn ein kraftgeregeltes Greifen erforderlich ist). Nachdem die Schritte definiert sind, kann der Gesamtprozess von der Basisstation automatisch durchgeführt werden. Sobald sich das alte Gerät in der Basisstation befindet, wird es als zuzuordnendes Gerät für die Prozessschritte vorgeschlagen, wenn die Anforderungen erfüllt sind. Wenn in der Basisstation Standardgeräteparameter gespeichert sind (z.B. Greifkraft, Beschleunigung, Sicherheitsoffset etc.), ist es möglich, diese Werte für das neue Gerät automatisch zu überschreiben.

15 dargestellt, definiert der Ingenieur in seiner ersten Phase Namen für jeden Schritt. In dieser Phase ist eine direkte Interaktion mit den Geräten nicht zwingend erforderlich. In einer zweiten Phase wird ein Gerät jedem Schritt zugeordnet. Basierend auf den Prozessschritten werden Geräte vorgeschlagen. Diese müssen den Prozessschrittenanforderungen übereinstimmen (z. B. wenn ein kraftgeregeltes Greifen erforderlich ist). Nachdem die Schritte definiert sind, kann der Gesamtprozess von der Basisstation automatisch durchgeführt werden. Sobald sich das alte Gerät in der Basisstation befindet, wird es als zuzuordnendes Gerät für die Prozessschritte vorgeschlagen, wenn die Anforderungen erfüllt sind. Wenn in der Basisstation Standardgeräteparameter gespeichert sind (z.B. Greifkraft, Beschleunigung, Sicherheitsoffset etc.), ist es möglich, diese Werte für das neue Gerät automatisch zu überschreiben.

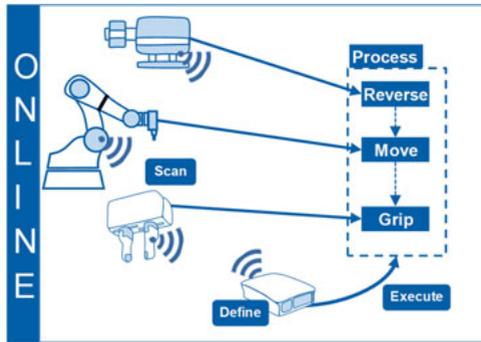


Abbildung 16 Online Entwicklung von Automatisierung sprozessen

In der Online-Prozessentwicklung, wie in Abbildung 16 dargestellt, wird zunächst ein Skill-Scan für jedes Gerät in Reichweite durchgeführt. Wählend ein Ingenieur nun Methoden aus Geräten aus und definiert nachfolgende Prozessschritte. Da vor der Prozessentwicklungsphase ein Geräte-Scan durchgeführt wurde, kann der Ingenieur direkt konkrete Parameter für Gerätemethodenaufrufe angeben. Zusätzlich können die Standardparameter des Geräts live auf dem Gerät ein gestellt werden. Nachdem alle Prozessschritte definiert sind, wird der Gesamtprozess ausgeführt. Der Fall eines Gerätetauschs wird ähnlich wie bei der Offline-Entwicklung behandelt.

Der Abgleich von Gerätefähigkeiten mit den Anforderungen an die Prozessschritte verringert unbeabsichtigte Fehler bei der Prozessentwicklung (z.B. bietet der Greifer kein kraftgeregeltes Greifen). Die Integrationszeiten der Geräte können verkürzt werden, da das System in der Lage ist, Standardparameterwerte automatisch zu überschreiben und den Bediener oder Ingenieur bei der Auswahl geeigneter Geräte zu unterstützen. Darüber hinaus kann der a priori Geräte-Scan, der in der Online-Prozessentwicklung verwendet wird, verwendet werden, um ein aktuelles Bild der eingebauten Geräte und ihrer Eigenschaften zu erhalten. Daher ist es nicht notwendig, die korrekte Version der Gerätespezifikationen oder Beschreibungen in speziellen Dateien auszuchecken und zu suchen. Schließlich kann mit der Online-Prozessentwicklung ein Rapid Process Prototyping bzw. Rapid Engineering durchgeführt werden, da der Ingenieur den Prozess direkt an der Anlage testen und Fehler im Prozess frühzeitig erkennen kann.

4.3.7 Demonstrator

Um die in den letzten Abschnitten vorgestellten Konzepte und Methoden zu bewerten, wurden ein Demonstrator sowie ein Beispielszenario entwickelt.

Die Umsetzung der Feldgerätekonzepte erfolgte mit einem ESP32 und einem LAN8720 ETH-Board, wie in Abbildung 17 dargestellt.



Abbildung 17 LAN8720 mit ESP32 Platine

Die medienunabhängige Schnittstelle (MII) des ESP32 Schnittstelle (MDI) des LAN8720 über eine speziell angefertigte Leiterplatte (PCB) verbunden, um eine elektrische Modulation der Ethernet-Signale zu ermöglichen. Der ESP32 beinhaltet sowohl den konkreten Gerätetreiber als auch die BLE-Service-Implementierung, so dass das Automatisierungsgerät ein Upgrade der drahtlosen Kommunikation erhalten kann. Die Umsetzung des Basissternem Raspberry Pi 3 Model B und einer in Node.js entwickelten Anwendung. Der Demonstrator besteht aus insgesamt drei Geräten .

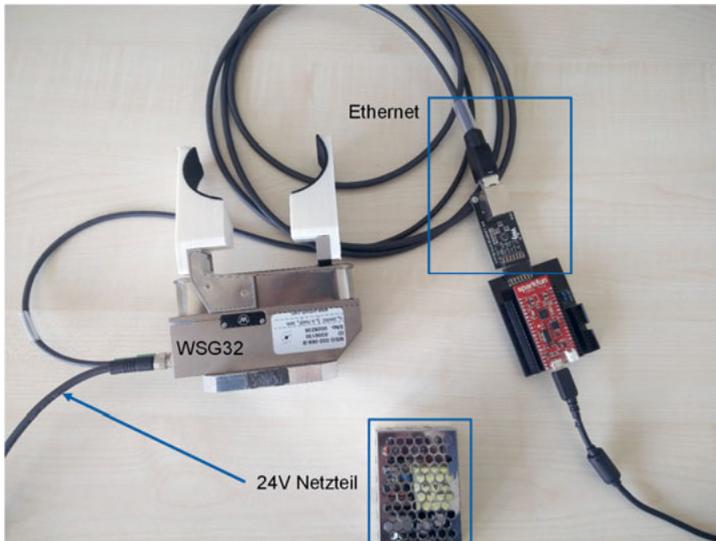


Abbildung 18 Servo-elektrischer Greifer der Firma Weiss Robotics

Abbildung 18 als erstes Gerät einen servo-elektrischen Greifer, der mit einem Ethernet-Anschluss für den Steuerungs- und Statusdatenaustausch ausgestattet ist, der in der Lage ist, kraftgeregelte Greifbefehle auszuführen. Ein zweites Gerät ist ein weiterer servo-elektrischer Greifer, der mit einer UART-Schnittstelle ausgestattet ist, die in der Lage ist, Greifbefehle auszuführen, jedoch ohne kraftgesteuerte Rückmeldung.

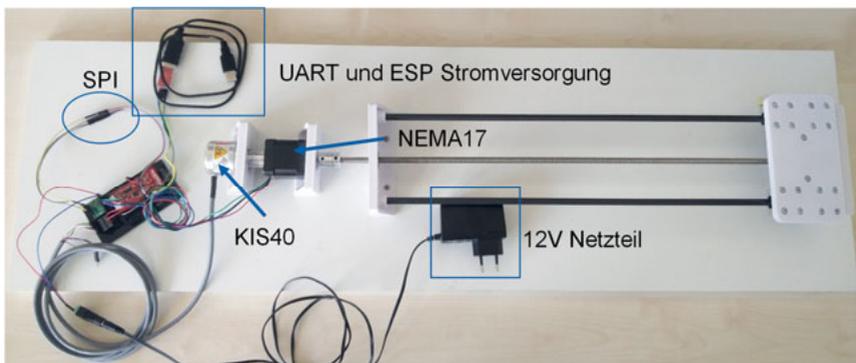


Abbildung 19 Linearachse mit NEMA 17 Schrittmotor und Inkrementalgeber

Schließlich zeigt Abbildung 19 eine Linearachse, die aus einem NEMA 17-Schrittmotor, einer Schrittmotorsteuerung und einem Inkrementalgeber mit 4096 Schritten pro Umdrehung besteht.

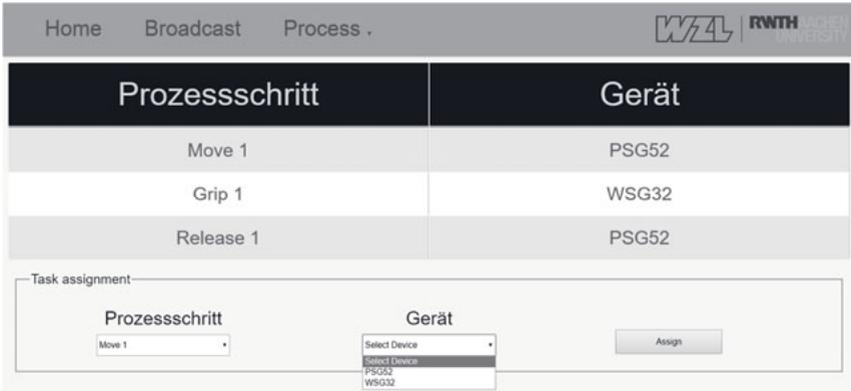


Abbildung 20 GUI für die offline Prozessentwicklung

Der Beispielprozess besteht aus drei Prozessschritten. Zuerst wird ein Fahrbefehl ausgeführt, bei dem sich die Linearachse oder ein Greifer auf eine bestimmte Position bewegt. Anschließend wird ein kraftkontrolliertes Greifen durchgeführt. Abschließend wird ein Freigabekommando ausgeführt.

Abbildung 20 zeigt einen Ausschnitt aus der Offline-Prozessentwicklungsübersicht, in der die Gerätezuordnung bereits erfolgt ist. Die Benutzeroberfläche der Offline-Prozessentwicklung ermöglicht das Hinzufügen, Entfernen und Konfigurieren von Prozessschritten. Prozessschritte werden einfach benannt und können gescannten verfügbaren Geräten zugeordnet werden. Abbildung 21 zeigt einen Ausschnitt aus der Online-Prozessentwicklung, in dem ein gescannter servoelektrischer Greifer mit seiner Benutzeroberfläche dargestellt ist. Diese Benutzeroberfläche ermöglicht die Anzeige aller Methoden aller gescannten Geräte sowie die Anzeige von Methoden, die sich auf jedes Gerät beziehen. Darüber hinaus ist es analog zur Offline-Prozessentwicklung möglich, Prozessschritte hinzuzufügen, zu entfernen und zu konfigurieren.

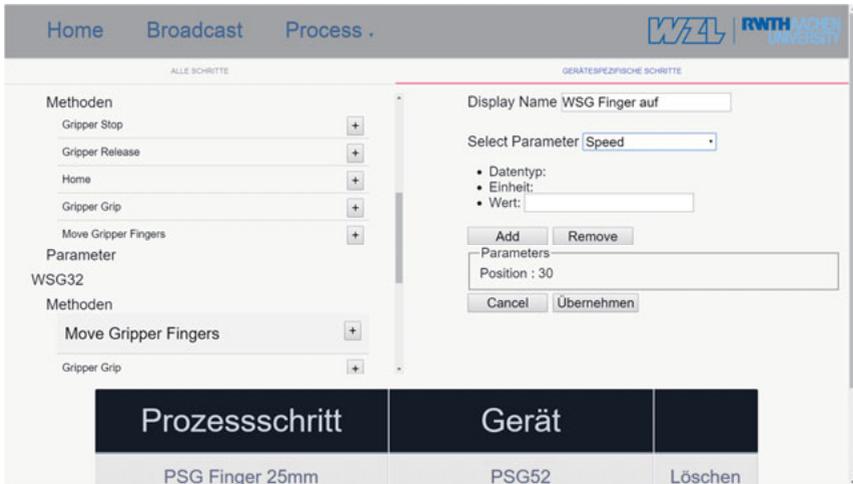


Abbildung 21 GUI für die online Prozessentwicklung

4.4 Zusammenfassung und Ausblick

In diesem Beitrag wurden Konzepte und Methoden zur Inbetriebnahme von Automatisierungsgeräten und zur Unterstützung der Entwicklungsphase auf Basis von 802.15.1 (Bluetooth Low Energy) und einem Informationsmodell zur Selbstbeschreibung von Automatisierungsgeräten vorgestellt.

Der Ansatz zeigte neue Potenziale für die industrielle Geräteenumeration, Inbetriebnahme, Konfiguration, Unterstützung bei der Prozessentwicklung. Die Geräte auf der Grundlage gesannter Gerätefunktionen und Integrationsprozesse und Entwicklungsphasen beschleunigen den Einsatz von BLE, das in fast allen heute verwendeten Smart Devices verfügbar ist, eine direkte Interaktion mit Automatisierungsgeräten und damit eine schnelle Konfiguration möglich. Durch das retrospektive / selbstbeschreibende Informationsmodell jedes Automatisierungsgerätes wird ein schneller Gerätetausch ermöglicht, da die Basisstation in der Lage ist, Methoden und Parameter für neu integrierte Geräte abzubilden / zu überschreiben.

Da sich das Konzept auf die Inbetriebnahme und Konfiguration von Automatisierungsgeräten konzentriert, wurde der eigentliche Echtzeit-Datenaustausch nicht berücksichtigt. Es gibt daher weitere Konzepte, bei denen die Gesamtkonfiguration in Inbetriebnahme und Echtzeitbetrieb unterteilt ist. In der Hochlauf- und Konfigurationsphase nutzen alle Geräte das oben dargestellte Konzept zur Inbetriebnahme. Sobald ein Prozessdatenaustausch stattfinden soll, wechseln alle Funkverbindungen in den Echtzeitmodus, um Echtzeitanforderungen zu erfüllen. Darüber hinaus wird ein automatisiertes

Vereinfachung der Inbetriebnahme während der Prozessentwicklung (Bluetooth Low Energy) und einem Informationsmodell vorgestellt.

Die Automatisierung im Bereich der Ramp-up-Prozesse sowie der Auswahl auf Basis von Fähigkeiten und Prozessanforderungen kann integrieren. Darüber hinaus ist durch den Einsatz von Smart Devices verfügbar

den Smart Devices verfügbar ist, eine direkte Interaktion mit Automatisierungsgeräten und damit eine schnelle Konfiguration möglich. Durch das retrospektive / selbstbeschreibende Informationsmodell jedes Automatisierungsgerätes wird ein schneller Gerätetausch ermöglicht, da die Basisstation in der Lage ist, Methoden und Parameter für neu integrierte Geräte abzubilden / zu überschreiben.

Training eines Zustandsmodells eines Automatisierungssystems auf Basis der Konzepte und maschinellen Lernalgorithmen untersucht. Beide Ansätze werden in zukünftigen Arbeiten untersucht.

Zu allerletz, da sich das Konzept auf die Integrationsstufe „Wireless Adapted“ bezieht, muss eine Möglichkeit geschaffen werden, spezifische Gerätetreiber mittels generischer Treiber zu ersetzen. Ein erster Ansatz, wie in Abbildung 22, zeigt dabei die Grundstruktur einer Runtime Engine bzw. generischen Treibers, der die Schnittstelle, den Payload und die Applikation aus den Stufen voneinander entkoppelt. Dadurch soll eine Parametrierbarkeit geschaffen werden, sodass der Zugriff auf gerätespezifische Informationen von der Applikation aus immer auf gleichem Wege erfolgen soll und keine Änderungen in der Software bzw. Firmware des jeweiligen Funkmoduls vorgenommen werden müssen. Dieser Ansatz wird ebenfalls in zukünftigen Arbeiten weiter untersucht.

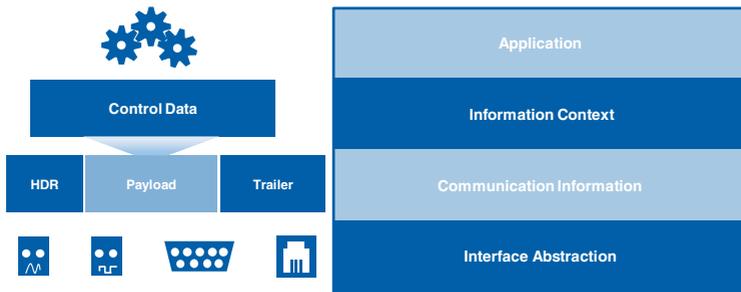


Abbildung 22 Generischer Treiber mittels schrittweiser Protokollentkopplung

4.5 Literatur

- [1] D. Trotta and P. Garengo, "Industry 4.0 key research topics: A bibliometric review," in 2018 7th International Conference on Industrial Technology and Management (ICITM), 2018, pp. 113–117.
- [2] L. Dürkop, L. Wisniewski, S. Heymann, B. Lücke, and J. Jasperneite, "Analyzing the engineering effort for the commissioning of industrial automation systems," in 2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA), 2015, pp. 1–4.
- [3] M. Wienke, O. Niggemann, S. Faltinski, and J. Jasperneite, "mINA-DL: A novel description language enabling dynamic reconfiguration in industrial automation," in ETFA2011, 2011, pp. 1–4.
- [4] V. Hammerstingl and G. Reinhart, "Unified Plug&Produce architecture for automatic integration of field devices in industrial environments," in 2015 IEEE International Conference on Industrial Technology (ICIT 2015): Seville, Spain, 17 - 19 March 2015 ; [proceedings], 2015, pp. 1956–1963.
- [5] S. Hodek and J. Schlick, "Ad hoc field device integration using device profiles, concepts for automated configuration and web service technologies: Plug&Play field device integration concepts for industrial production processes," in 9th International Multi-Conference on Systems, Signals and Devices (SSD), 2012: 20 - 23 March 2012, Chemnitz, Germany ; [including 4 conferences], 2012, pp. 1–6.
- [6] Informationstechnik - UPnP Gerätearchitektur - Teil 1: UPnP Device Architecture Version 1.0, ISO/IEC 29341-1, 2011.
- [7] S. Wassilew, L. Urbas, J. Ladiges, A. Fay, and T. Holm, "Transformation of the NAMUR MTP to OPC UA to allow plug and produce for modular process automation," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 2016, pp. 1–9.
- [8] NAMUR Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, "NE 105 Anforderungen an die Integration von Feldbus-Geräten in Engineering-Tools für Feldgeräte," NAMUR Interessengemeinschaft Automatisierungstechnik der Prozessindustrie, Feb. 2016. [Online] Available : <http://www.namur.net/de/empfehlungen-u-arbeitsblaetter/aktuelle-nea.html>. Accessed on: Aug. 25 2017.
- [9] P. Leitão and P. Vrba, "Recent Developments and Future Trends of Industrial Agents," in Lecture Notes in Computer Science, Holonic and Multi-Agent Systems for Manufacturing, D. Hutchison et al., Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 15–28.
- [10] S. Glückselig, "Holonische Multiagentensimulation," Diplomarbeit, Institut für Informatik, Bayerische Julius-Maximilians-Universität , Würzburg, 2005.

- [11] M. Hoffmann et al., "Semantic integration of multi-agent systems using an OPC UA information modeling approach," in Proceedings, 2016 IEEE 14th International Conference on Industrial Informatics (INDIN): Palais des Congrès du Futuroscope, Futuroscope - Poitiers, France, 19-21 July, 2016, pp. 744–747.
- [12] L. Ferariu and D. Panescu, "Pattern recognition for holonic manufacturing systems," in 2009 European Control Conference (ECC), Budapest, Ungarn, 2009, pp. 1257–1262.
- [13] P. Ordóñez de Pablos and M. D. Lytras, Eds., Knowledge management and drivers of innovation in services. Hershey, Pa.: Information Science Reference, 2012.
- [14] D. C. McFarlane and S. Bussmann, "Holonic Manufacturing Control: Rationales, Developments and Open Issues," in Advanced Information Processing, Agent-Based Manufacturing: Advances in the Holonic Approach, M. Deen, Ed., Berlin and Heidelberg: Springer, 2003, pp. 303–326.
- [15] H. van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," Computers in Industry, vol. 37, no. 3, pp. 255–274, 1998.
- [16] A. Giret, D. Trentesaux, M. A. Salido, E. Garcia, and E. Adam, "A holonic multi-agent methodology to design sustainable intelligent manufacturing control systems," Journal of Cleaner Production, 2017.
- [17] K. Lipinski and H. Lackner, FDT (field device tool). [Online] Available: <http://www.itwissen.info/FDT-field-device-tool.html>. Accessed on: Jul. 17 2017.
- [18] FDTGroup, What Is A Device Type Manager (DTM)? Available: <https://fdt-group.org/device-type-manager-dtm/>. Accessed on: Jul. 17 2017.
- [19] FDTGroup, Overview. [Online] Available: <https://fdtgroup.org/technology/overview/>. Accessed on: Mai. 23 2017.
- [20] T. Takeuchi, "FDT/DTM Framework for new field device tools," Yokogawa, 2007. [Online] Available: <https://web-material3.yokogawa.com/rd-tr-r00044-004.pdf>. Accessed on: Jul. 18 2017.
- [21] EDDL, "Device Integration Made Easy," Apr. 2014. [Online] Available: http://www.eddl.org/SiteCollectionDocuments/TechPapers/wp_EDDL%20device%20integration.pdf. Accessed on: Jul. 20 2017.
- [22] Emerson Process Management, "FDT/DTM, and Enhanced EDDL, what's best for the user," 2008. [Online] Available: <http://www.emersonprocessperts.com/papers/FDTDTMandEnhancedEDDLWhatsBestForTheUser.pdf>. Accessed on: Jul. 21 2017.
- [23] Funktionsbausteine für die Prozessautomation und elektronische Gerätebeschreibungssprache – Teil 4: Interpretation von Gerätebeschreibung EN 61804-4, 2016.

- [24] FDI Cooperation, "FDI Field Device Integration Technology," FDI Cooperation, Jan. 2012. [Online] Available: http://www.fdi-cooperation.com/tl_files/images/content/Publications/FDI-White_Paper.pdf. Accessed on: Jul. 28 2017.
- [25] OPC Unified Architecture - Teil 4: Dienste (IEC 62541-4:2015) EN 62541-4, 2015.
- [26] G. Anastasi, E. Borgia, M. Conti, and E. Gregori, "IEEE 802.11b Ad Hoc Networks: Performance Measurements," *Cluster Comput*, vol. 8, no. 2-3, pp. 135–145, 2005.
- [27] S. Raza, P. Misra, Z. He, and T. Voigt, "Bluetooth smart: An enabling technology for the Internet of Things," in 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob): 19–21 Oct. 2015, Abu Dhabi, United Arab Emirates ; including workshops, pp. 155–162.
- [28] K. Townsend, *Getting started with Bluetooth low energy*. Sebastopol, CA: O'Reilly, 2014.
- [29] K.-H. Chang, "Bluetooth: A viable solution for IoT? [Industry Perspective]," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 6–7, 2014.
- [30] A. Muthanna et al., "Analytical Evaluation of D2D Connectivity Potential in 5G Wireless Systems," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems: 16th International Conference, NEW2AN 2016, and 9th Conference, ruSMART 2016, St. Petersburg, Russia, September 26-28, 2016*, Proceedings, O. Galinina, S. Balandin, and Y. Koucheryavy, Eds., Cham: Springer International Publishing, 2016, pp. 395–403.
- [31] Ericsson AB, "5G Radio Access," Ericsson AB, 2016. [Online] Available: <https://www.ericsson.com/assets/local/publications/white-papers/wp-5g.pdf>. Accessed on: Aug. 16 2017.
- [32] T. Wirth, M. Mehlhose, J. Pilz, B. Holfeld, and D. Wieruch, "5G new radio and ultra low latency applications: A PHY implementation perspective," in 2016 50th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2016, pp. 1409–1413.
- [33] B. Holfeld et al., "Radio channel characterization at 5.85 GHz for wireless M2M communication of industrial robots," in *2016 IEEE Wireless Communications and Networking Conference: 3-6 April 2016*, 2016, pp. 1–7.
- [34] A. E. Mostafa, Y. Zhou, and V. W.S. Wong, "Connectivity maximization for narrowband IoT systems with NOMA," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [35] Link Labs, "Selecting a Wireless technology for new Industrial Internet of Things Products: A guide for Engineers and Decision Makers," Link Labs, Annapolis, 2016. [Online] Available: <https://www.link-labs.com/selecting-a-wireless-technology-for-industrial-iiot>. Accessed on: Aug. 14 2017.

- [36] O. Khutsoane, B. Isong, and A. M. Abu-Mahfouz, "IoT devices and applications based on LoRa/LoRaWAN," in IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, 2017, pp. 6107–6112.
- [37] Y. Chung, J. Y. Ahn, and J. Du Huh, "Experiments of A LPWAN Tracking(TR) Platform Based on Sigfox Test Network," in 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, 2018, pp. 1373–1376.

5 Dienstbasierte Architekturen für Robotersysteme

M. Rudorfer, C. Krause, A. Vick, J. Krüger

5.1 Einleitung

Die Idee für verteilte, dienstbasierte Architekturen stammt zwar aus der Softwareentwicklung, gewann im Laufe der letzten zehn Jahre aber immer mehr auch in der Automatisierungstechnik an Bedeutung. So wurde bereits 2008 vorgeschlagen, Service-Oriented Architectures (SOA) nicht nur zur Unterstützung der Geschäftsprozesse sondern auch auf den unteren Ebenen der Automatisierungspyramide zu verwenden [1]. Die im Zusammenhang mit dem echtzeitkritischen Produktionsumfeld entstehenden zusätzlichen Anforderungen an SOA werden gelegentlich unter dem Begriff Manufacturing 2.0 zusammengefasst. Seitdem gab es viele Arbeiten auf dem Gebiet so dass virtualisierte Steuerungen und dienstbasierte Architekturen mittlerweile zum Stand der Technik gehören [2].

Dieser Trend zeigt sich auch in der Robotik. Es wurden Architekturen geschaffen, um Roboter sowohl miteinander (M2M) als auch mit der Cloud (M2C) zu verbinden [3]. Das Projekt RoboEarth hat zum Ziel, eine Art World Wide Web für Roboter zu entwickeln, das es den Robotern nicht nur erlaubt Informationen auszutauschen sondern auch Verhaltensweisen voneinander zu lernen [4]. Neben diesen hauptsächlich auf mobile Service-Roboter bezogenen Projekten gibt es jedoch auch verschiedene Ansätze, dienstbasierte Architekturen in industriellen Robotersystemen anzuwenden, um beispielsweise energieeffizientere Bahnplanung in der Cloud zu ermöglichen [5].

In diesem Artikel haben wir es uns zur Aufgabe gemacht, die verschiedenen Ansätze der dienstbasierten Steuerungsarchitekturen zu vergleichen und deren Vor- und Nachteile gegenüber zu stellen. Insbesondere stellen wir dabei auch eigene Ansätze vor, die im Laufe der Zeit am Fachgebiet Industrielle Automatisierungstechnik der TU Berlin sowie in der Abteilung Prozessautomatisierung und Robotik des Fraunhofer IPK entwickelt wurden. Ziel ist es dabei, geeignete Schlüsseltechnologien sowie Best Practices zur Gestaltung solcher verteilten Architekturen zu identifizieren.

Nach der Definition von Zielen und Gestaltungsmöglichkeiten in Abschnitt 5.2 folgt in Abschnitt 5.3 der Stand der Technik und Methodenübersicht dienstbasierter Robotersteuerungen. Anschließend werden im Abschnitt 5.4 einige am PTZ entwickelte Ansätze vorgestellt und im Abschnitt 5.5 hinsichtlich der Gestaltungskriterien diskutiert und Best Practices abgeleitet. Zum Abschluss geben wir in Abschnitt 5.6 eine Zusammenfassung unserer Erkenntnisse und einen Ausblick auf weitere Schritte.

5.2 Ziele und Gestaltungsmöglichkeiten

Dienstbasierte Architekturen werden immer mit dem Robotersystem einen bestimmten Nutzen zu generieren wir uns daher zunächst an, welche Vorteile dies kon zeigen wir die Gestaltungsmöglichkeiten auf und def Aspekte wir bei der Analyse von dienstbasierten Ar werden.

Ziel eingesetzt, für das jeweilige . In diesem Abschnitt schauen kret sein können. Anschließend inieren damit gleichzeitig welche chitekturen genauer betrachten

5.2.1 Angestrebte Vorteile

Durch die Verbindung der Roboter mit der Cloud erge ten, von der Auslagerung von Berechnungen bis hin z bzw. Kooperation der Roboter untereinander. In [6] ten strukturiert und grob in die in Abbildung darg reiche unterscheiden sich insbesondere im Grad der

ben sich zahlreiche Möglichkei ur Absprache und Kollaboration wurden die vielfältigen Möglichkei- estellten Stufen eingeteilt. Diese Be- Vernetzung.



Abbildung 1: Vier Stufen der Vernetzung mit ihren jeweiligen Möglichkeiten für dienstbasierte Architekturen.

So beinhaltet die erste Stufe zwar eine Verbindung des Roboters mit der Cloud, jedoch noch keine Vernetzung der Roboter untereinander. Da bei geht es um die Auslagerung komplexer Berechnungen wie beispielsweise Bahnplanu ng oder Bildverarbeitung. Kosten können eingespart werden, da die Anforderung en an die lokale Hardware hinsichtlich Rechenleistung sinken. Eine erste Form de r Vernetzung der Roboter unterei- nander ist das geteilte Wissen in Stufe 2. Die Robo ter können auf eine (möglicher- weise manuell angelegte) zentrale Datenbasis zugrei fen, die beispielsweise Karten- material oder eine Objektdatenbank enthält. In der

des Roboters mit der Cloud, jedoch noch keine Vernetzung der Roboter untereinander. Da bei geht es um die Auslagerung komplexer Berechnungen wie beispielsweise Bahnplanu ng oder Bildverarbeitung. Kosten können eingespart werden, da die Anforderung en an die lokale Hardware hinsichtlich Rechenleistung sinken. Eine erste Form de r Vernetzung der Roboter unterei- nander ist das geteilte Wissen in Stufe 2. Die Robo ter können auf eine (möglicher- weise manuell angelegte) zentrale Datenbasis zugrei fen, die beispielsweise Karten- material oder eine Objektdatenbank enthält. In der

durch die einzelnen Robotersysteme und deren Sensoren weiter angereichert und durch Data Mining oder Maschinelles Lernen kann aus den Daten und Informationen neues Wissen generiert werden. In der höchsten Stufe werden die Roboter tatsächlich auch direkt miteinander vernetzt, sodass sie Aufgaben wie beispielsweise die Kartierung ihres Umfelds oder das Tragen besonders unhandlicher Gegenstände kollaborativ bewältigen können.

Welche der Stufen erreicht werden kann ist dabei abhängig von der konkreten Ausgestaltung der dienstebasierten Architektur. Die große Mehrheit der aktuellen Ansätze im Bereich Cloud Robotics nutzt die Anbindung an die Cloud in Stufe 1 und 2 zur Steigerung der Rechenleistung sowie für zentrales Daten-/ Wissensmanagement [5].

5.2.2 Aspekte dienstebasierter Architekturen

Bei der Auslagerung von Funktionalitäten in die Cloud stellt sich an einem Punkt zwangsläufig die Frage nach der Grenze. In [7] und [8] wurde gezeigt, dass grundsätzlich, mit bestimmten Anforderungen an das Netzwerk sowie modellprädiktiver Regelung, alle Elemente der numerischen Bewegungssteuerung virtualisiert und ausgelagert werden können. Solche Systeme werden in [5] als web-basierte Robotersysteme bezeichnet und von cloud-basierten Systemen abgegrenzt. Diese begriffliche Unterscheidung ist jedoch schwer nachvollziehbar, weshalb wir von Roboter aus der Cloud bzw. Anwendung aus der Cloud sprechen (siehe Abbildung 2). Bei letzterem verbleibt das Roboter-Betriebssystem lokal, und nur die konkrete Anwendung wird durch Dienste aus der Cloud realisiert. Dieser Weg wird im Vergleich häufiger verfolgt, da die echtzeitkritische Kommunikation zwischen Roboter Betriebssystem und Hardware Komponenten nicht durch Netzwerkprobleme beeinträchtigt werden kann.

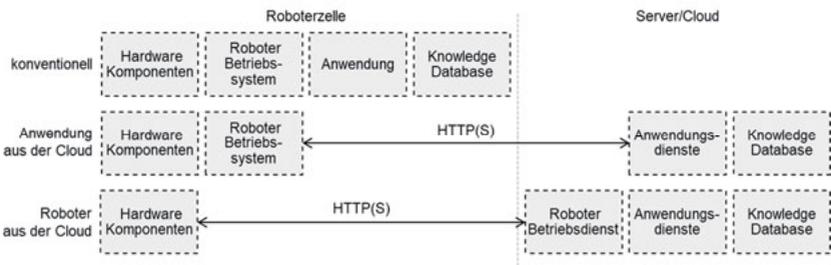


Abbildung 2: Unterscheidung der dienstebasierten Architekturen und konventionellen Methoden nach Partitionierung der ausgelagerten Komponenten (Vgl. [5]).

Das führt direkt zur Fragestellung nach der Partitionierung und Granularität der Anwendungsdienste. Die Partitionierung beschreibt allgemein die Verteilung der Funktionalitäten auf die einzelnen Dienste, während die Granularität den funktionalen Umfang beziehungsweise die „Größe“ eines Dienstes umschreibt. Die optimale Partitio-

nierung und Granularität zu finden ist auch in der dienstebasierten Automatisierung gezielt untersucht, obwohl sie einen großen Einfluss auf Rechenzeit, Netzwerklast und Wiederverwendbarkeit der Dienste an

Sind die Funktionalitäten auf die unterschiedlichen Dienste zur eigentlichen Anwendung zusammengesetzt im Allgemeinen als Komposition der Dienste. Konkret und zwar die Orchestrierung und die Choreographie [10]. Bei der Orchestrierung gibt es einen zentralen Knoten, der nach einem Request-Response Schema die Dienste anspricht und so die Anwendung „dirigiert“. Die einzelnen Informationen über den Prozess speichern, im Gegensatz dazu ist die Information über den Programm auf die Knoten verteilt. Jeder einzelne Dienst weiß nach dem Publish-Subscribe Prinzip, von welchem Dienst er seine Informationen erhält und welcher Dienst dann aufgerufen werden muss. Um die jeweilige Komposition der Dienste zu ändern muss bei der Orchestrierung nur der zentrale Knoten aktualisiert werden, um anschließend selbstständig den neuen Ablauf zu realisieren. Die Choreographie erfordert dagegen die Aktualisierung aller an der Veränderung beteiligten Dienste, um die jeweiligen Datenquellen auszutauschen.

Um tatsächlich mit den Diensten zu kommunizieren gibt es eine Vielfalt an Kommunikationsprotokollen. In [11] wurden im Kontext des Internet of Things einige populäre Protokolle, darunter REST/http, MQTT, AMQP und DDS, hinsichtlich Aspekten wie Ausnutzung der Bandbreite, Energieverbrauch und Sicherheitsumfeld ist OPC UA weit verbreitet und wird als 4.0 gehandhabt [12]. Erste Implementierungen einer dienstebasierten Robotersteuerung mit OPC UA zeigen die Flexibilität der Anwendung zur Laufzeit [13]. Darüber hinaus sind auch moderne Protokolle wie gRPC und GraphQL interessant, die von Google beziehungsweise Facebook entwickelt wurden. Die in einem System verwendeten Dienste müssen nicht zwangsläufig dasselbe Kommunikationsprotokoll implementieren, so können die entsprechenden Adaptern überbrückt werden. Diese Arbeit wird durch zum Teil vorhandene, vorgefertigte Gateways unterstützt. Beispielsweise kann ein gRPC Service über ein solches Gateway auch als RESTful API angesprochen werden [14]. Ein Manufacturing Service Bus [15] kann bei der Integration von Diensten helfen.

Softwareentwicklung ein viel diskutiertes Thema. In der dienstebasierten Automatisierung werden diese Aspekte nur selten gezielte untersucht, obwohl sie einen großen Einfluss auf Rechenzeit, Netzwerklast und Wiederverwendbarkeit der Dienste an

Sind die Funktionalitäten auf die unterschiedlichen Dienste verteilt, so müssen diese Dienste zur eigentlichen Anwendung zusammengesetzt werden. Dies beschreibt man im Allgemeinen als Komposition der Dienste. Konkret und zwar die Orchestrierung und die Choreographie [10]. Bei der Orchestrierung gibt es einen zentralen Knoten, der nach einem Request-Response Schema die Dienste anspricht und so die Anwendung „dirigiert“. Die einzelnen Informationen über den Prozess speichern, im Gegensatz dazu ist die Information über den Programm auf die Knoten verteilt. Jeder einzelne Dienst weiß nach dem Publish-Subscribe Prinzip, von welchem Dienst er seine Informationen erhält und welcher Dienst dann aufgerufen werden muss. Um die jeweilige Komposition der Dienste zu ändern muss bei der Orchestrierung nur der zentrale Knoten aktualisiert werden, um anschließend selbstständig den neuen Ablauf zu realisieren. Die Choreographie erfordert dagegen die Aktualisierung aller an der Veränderung beteiligten Dienste, um die jeweiligen Datenquellen auszutauschen.

Um tatsächlich mit den Diensten zu kommunizieren gibt es eine Vielfalt an Kommunikationsprotokollen. In [11] wurden im Kontext des Internet of Things einige populäre Protokolle, darunter REST/http, MQTT, AMQP und DDS, hinsichtlich Aspekten wie Ausnutzung der Bandbreite, Energieverbrauch und Sicherheitsumfeld ist OPC UA weit verbreitet und wird als 4.0 gehandhabt [12]. Erste Implementierungen einer dienstebasierten Robotersteuerung mit OPC UA zeigen die Flexibilität der Anwendung zur Laufzeit [13]. Darüber hinaus sind auch moderne Protokolle wie gRPC und GraphQL interessant, die von Google beziehungsweise Facebook entwickelt wurden. Die in einem System verwendeten Dienste müssen nicht zwangsläufig dasselbe Kommunikationsprotokoll implementieren, so können die entsprechenden Adaptern überbrückt werden. Diese Arbeit wird durch zum Teil vorhandene, vorgefertigte Gateways unterstützt. Beispielsweise kann ein gRPC Service über ein solches Gateway auch als RESTful API angesprochen werden [14]. Ein Manufacturing Service Bus [15] kann bei der Integration von Diensten helfen.

5.3 Stand der Technik

5.3.1 Cloud Computing und Robotik

Cloud Computing kann als eine Reihe von Diensten definiert werden, die den Zugriff auf verteilte Algorithmen ermöglicht und gemeinsam genutzte Rechenressourcen über

Softwareentwicklung ein viel diskutiertes Thema. In der dienstebasierten Automatisierung werden diese Aspekte nur selten gezielte untersucht, obwohl sie einen großen Einfluss auf Rechenzeit, Netzwerklast und Wiederverwendbarkeit der Dienste an

das Internet, einschließlich Informationsverarbeitung, Speicher und riesiger Datensätze verwendet. Es gibt derzeit mehrere kommerzielle Cloud-Computing-Diensteanbieter wie Amazon Web Services, Google Compute Engine und Microsoft Azure. Daneben ist eine Vielzahl von regionalen Anbietern verfügbar und einige große und mittlere Unternehmen folgen bereits dem Trend eine eigene Cloud in ihren Rechenzentren zu betreiben.

Übertragung von Roboteranwendungen in die Cloud erfordert spezielle Frameworks oder Servicemodelle, die diese Übermittlung entsprechend unterstützen. In Cloud-Computing-Angeboten werden im Allgemeinen drei Arten von Servicemodellen angeboten [16], Infrastruktur as a Service (IaaS), Plattform as a Service (PaaS) und Software as a Service (SaaS). Die Servicemodelle unterscheiden sich in der Komplexität der angebotenen Funktionen von reinem (virtuellem) Betriebssystem bis hin zu vollständiger Software mit Internetschnittstelle.

In der jüngsten Vergangenheit wurde auch "Robot as a Service" (RaaS) als Cloud Recheneinheit vorgeschlagen, die die nahtlose Integration von Robotern und eingebetteten Geräten im Internet und Cloud-Computing-Umgebungen ermöglicht [17][18].

Im folgenden Unterabschnitt werden die üblicherweise in der Literatur beschriebenen Techniken der Cloud-Robotik Architektur erläutert.

5.3.2 Cloud Robotik System Architekturen

Die folgenden drei Werke von Arumugam et al. [19], Mohanarajah et al. [20] und Riazuelo et al. [21] können als bedeutende Werke betrachtet werden, die das Projekt von „Softwarearchitektur zur Integration von Cloud-Diensten mit Roboterhardware“ vorgeschlagen und umgesetzt haben.

DAVinCi [19] ist ein Software Framework, das verschiedene Vorteile des Cloud Computing wie Skalierbarkeit und Parallelität für Serviceroboter nutzt. Das DAVinCi Framework basiert auf dem Hadoop-Cluster in Kombination mit dem Robot Operating System (ROS) welches als Messaging-Framework für Roboter-Ökosysteme sehr bekannt ist.

Das RoboEarth-Projekt [4] wird von einer Open-Source-Cloud-Robotikplattform namens Rapyuta [20] gesteuert. Die Systemarchitektur wird in Abbildung 3 aufgezeigt. Das besondere an der Systemarchitektur ist, dass das System den Robotern das einheitliche und sichere Delegieren von rechenintensiven Aufgaben in die Cloud ermöglicht. Die Computerumgebungen in Rapyuta bieten Zugriff auf die RoboEarth Wissensspeicher, in dem Roboter Karten, Objektmodelle und Aktionsrezepte herunterladen können. Die Umgebung ist kompakt miteinander verknüpft und für den einfachen Einsatz von ganzen Roboterteams geeignet.

C2TAM [21] ist ein Cloud-Robotik-Framework, das speziell für kooperatives Tracking entwickelt wurde. C2TAM bietet ein visuelles SLAM System, das auf einem verteilten Framework basiert. Die Clients können auf die Kartendatenbank zugreifen und selbst

auch neue Abschnitte erkunden. Dabei werden die Spe-
 rechnungen (wie z.B. das Bundle Adjustment) in der
 mehrere mobile Roboter problemlos eine größere Regi-
 kalisieren.

icherung der Daten sowie Be-
 Cloud durchgeführt. So können
 on kartieren und sich darin lo-

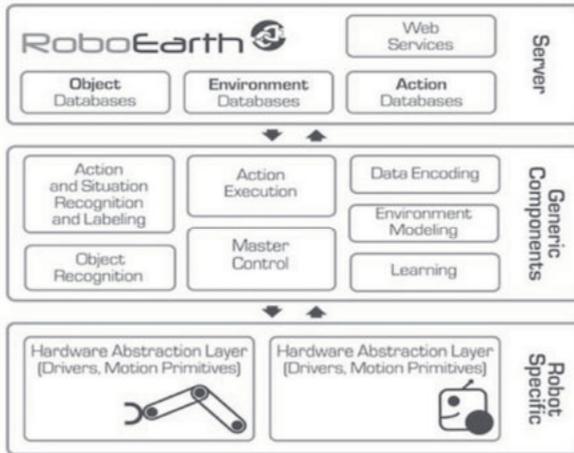


Abbildung 3: Die RoboEarth Systemarchitektur aus [4]

Rosbridge [22][23] wurde als Middleware-Abstraktion
 logie bietet Anwendungsprogrammierern, die selbst k
 Zugriff auf ein standardisiertes, minimalistisches
 work. Rosbridge bietet einen einfachen, Socket-basi
 stellen und Algorithmen mit bspw. JavaScript. Lee n
 webbasierter Lösungen für Roboter Anwendungen in ei
 umgebung namens ROSProcessingjs [24], die es ermögl
 icht die Entwicklung von Ro-
 boteranwendungen in einem Webbrowser durchzuführen.
 in [25] kombinierten Rosbridge und Ros.js, eine Ope
 n-Source-Bibliothek in JavaScript, die die Kommunikati
 on zwischen dem Browser und der Rosbridge verwaltet.
 Toris et al. [26] benutzte Roboter-Web-Tools dann zur
 Weiterentwicklung der Cloud Ro-
 botik durch die Entwicklung effizienterer Methoden
 zur Übertragung von sogenannten
 „Topics“ mit hoher Bandbreite.

sschicht entwickelt. Die Techno-
 eine Robotiker sein müssen, den
 Anwendungsentwicklungs-Framer
 erten Zugriff auf Roboterschnitt-
 nutzte Rosbridge und die Vorteile
 einer Anwendungsentwicklungs-
 icht die Entwicklung von Ro-
 boter-Web-Tools wie
 n-Source-Bibliothek in JavaScript,
 der Rosbridge verwaltet. Tor
 is et al. [26] benutzte Roboter-Web-Tools dann zur
 Weiterentwicklung der Cloud Ro-
 botik durch die Entwicklung effizienterer Methoden
 zur Übertragung von sogenannten

Die Ubiquitous Network Robot Platform (UNR-PF) [27]
 von Roboterdiensten zu vereinfachen. Die Autoren in
 gierten Methoden aus dem RoboEarth-Projekt eine ve
 Diensten zur Entwicklung einer vernetzten Roboterar
 chitektur gelingt. Hiermit soll die
 Entwicklung, Verwaltung und Anpassung von verteilte

zielt darauf ab, das Entwickeln
 [28] beschreiben wie mit inte-
 rteilte Ausführung von UNR-PF
 chitektur gelingt. Hiermit soll die
 n Roboteranwendungen verein-

facht werden. In [29] wird beschrieben wie integrierte Cloud-Robotik mit Software-Produktlinien kombiniert werden. Auf Basis von Software-Produktlinien können Endanwender komplexe Robotikanwendungen entwickeln, ohne dass sie sich mit den „Low-Level“-Details, wie das Konfigurieren von Algorithmen und das Entwerfen von Architekturen beschäftigen müssen.

Die RSi-Research-Cloud (RSi-Cloud) [30] ermöglicht die Integration von Roboterdiensten mit Internetdiensten. Die RSi-Cloud verfügt über ein standardisiertes Kommunikationsprotokoll, das als Robot-Service-Netzwerkprotokoll (RSNP) [31][32] bezeichnet wird. Verschiedene Arten von neuartigen Diensten werden hier angeboten, indem unterschiedliche Servicekomponenten unter Verwendung der Roboterservicespezifikation miteinander kombiniert werden. Das Robot Service Network Protocol (RSNP) wurde zur Entwicklung eines verteilten Protokolls für ein Dienste-Framework verwendet, das die Koordination verschiedener Geräte, Roboter und Servicefunktionen ermöglicht. Narita et al. [33] schlagen ein Verfahren vor, um zuverlässige Cloud-basierte Roboterdienste für RSNP-basierte Kommunikation im Zusammenhang mit Problemen wie WLAN-Trennung, Roboterservice-Problemen und Systemfehlern bei Robotern zu ermöglichen.

Um Cloud-basierte Dienste für den Alltag des Menschen zu entwickeln, wurde das Konzept der "Robot Cloud" vorgeschlagen [35]. Diese wurde unter Verwendung der beliebten Google App Engine entwickelt und soll durch das Konzept der serviceorientierten Architektur (SOA) ermöglichen, die Funktionsmodule in der Robot Cloud dementsprechend so zu erweitern, dass diese schließlich flexibler, ausbaufähiger und wiederverwendbarer werden. Um das Problem der Aufgabenzuweisung in der Cloud anzugehen, schlugen die Autoren einen "Location based Packing Scheduling-Algorithmus" (LPS-Algorithmus) vor, der mehrere Anforderungen zusammenfasst und diese dann entsprechend so ordnet, dass man sie einem einzigen Roboter zuweisen kann.

5.4 Konzepte und Umsetzungen von dienstebasierten Szenarien der industriellen Robotik und Bildverarbeitung

Dieser Abschnitt beschreibt die am Institut entwickelten Ansätze für dienstebasierte Architekturen in der Robotik. Dabei werden unterschiedliche Kommunikationsmechanismen und Ausführungsumgebungen untersucht und hinsichtlich Granularität und Partitionierung verglichen.

5.4.1 Integration eines modularen Industrieroboters mit ROS am Beispiel einer Pick-and-Place Aufgabe mit Kollisionsvermeidung

Der modulare Industrieroboter besteht aus sechs PowerCube Modulen, die einzeln auf Positions-, Geschwindigkeits- und Stromebene über einen CAN-Bus angesteuert werden können. Für diesen CAN-Bus wurde ein Adapter entwickelt, der Steuerungsbedenke aus dem Netzwerk empfängt und an die Robotermodule weiterleitet. Mit diesem

Hardwaredienst ist schon die Integration des Roboters in eine dienstbasierte Gesamtarchitektur möglich. Der Hardwaredienst muss in der Nähe des Roboters laufen, da hier die elektrische Verbindung (über den CAN-Bus) realisiert ist. ROS erfolgt durch einen weiteren Softwareknoten, dem der Hardwaredienst angeschlossen ist. Dieser Knoten kann nun schon auf einem entfernten Rechner laufen, da die Kommunikation nun rein IP-basiert ist. Weitere Dienste wie Inverse Kinematik, Bahnplanung, Umfelderkennung, Objekterkennung, Kollisionserkennung sind als (teilweise open-source erhältliche) ROS Knoten implementiert und im Netzwerk verteilt. Die Knoten können sich nun in geeigneter Reihenfolge abonnieren (publish-subscribe Verfahren) und so eine Komposition der Dienste erstellen. Diese Verkettung erfolgt zur Choreographie der einzelnen Knoten.

rs in eine dienstbasierte Gesamt-Nähe des Roboters laufen, da die Anbindung an das ROS-Framework implementierten Rechner laufen, da die ste wie Inverse Kinematik, Bahnplanung, Visualisierung usw. Knoten implementiert und im Netzwerk abonneren (publish-subscribe Verfahren) durch implizite der Anwendung durch implizite ur Kompilierzeit der einzelnen Knoten.

In unserem Anwendungsbeispiel wurden vier Hardwarekomponenten (2D Kamera, 3D Kamera, Roboter und Monitor) durch sechs Dienstknoten (siehe Abbildung 4, links) verkettet (siehe Abbildung 4, rechts). Alle Knoten befinden sich außerhalb der eigentlichen Roboteransteuerung und jeder Knoten implementiert nur eine einzige Aufgabe. Es handelt sich daher um eine „Roboter-aus-der-Cloud“-Partitionierung mit feiner Granularität. Der Benutzer kann mit der so komponierten Anwendung die geplante Trajektorie des Roboters im Kamerabild sehen (Augmented Reality, siehe Abbildung 4, rechts erkannten Hindernissen (siehe Abbildung 4, rechts, grün) in Echtzeit neu geplant.

omponenten (2D Kamera, 3D Kamera, Roboter und Monitor) durch sechs Dienstknoten (siehe Abbildung 4, links) verkettet (siehe Abbildung 4, rechts). Alle Knoten befinden sich außerhalb der eigentlichen Roboteransteuerung und jeder Knoten implementiert nur eine einzige Aufgabe. Es handelt sich daher um eine „Roboter-aus-der-Cloud“-Partitionierung mit feiner Granularität. Der Benutzer kann mit der so komponierten Anwendung die geplante Trajektorie des Roboters im Kamerabild sehen (Augmented Reality, siehe Abbildung 4, rechts erkannten Hindernissen (siehe Abbildung 4, rechts, grün) in Echtzeit neu geplant.

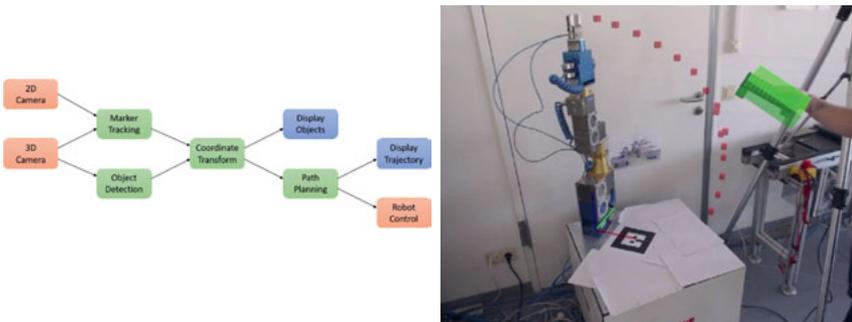


Abbildung 4: Systemarchitektur der verteilten Dienste und Umsetzung im Mensch-Roboter Interaktionsszenario

Durch den Zugriff auf eine große Modulbibliothek von Robotersystemen und –anwendungen vereinfacht nur unter bestimmten Bedingungen getestet und die führt zu erheblichem Aufwand in der Fehlersuche. Au Anwendungen nicht zur Laufzeit verändert oder skali

n ROS ist die initiale Integration Allerdings sind viele Module reie Verknüpfung von Knoten Berdem können die so erstellten ert werden.

5.4.2 Intuitiver Montagearbeitsplatz mit Dual-Arm Roboter und cloudbasierten OPC UA Diensten

Der verstärkte Bedarf aus Industrie und Wirtschaft an schnell wandelbare Produktionsmittel führt im Fall der Montageprozesse zu einer hybriden Lösung aus teilautomatisierter Handhabung und manueller Feinmontage und Qualitätskontrolle. Hier wird der komplette Ablauf einer Montageaufgabe zerlegt und je nach Bauteil und Qualifikationen des Werkers eine Verteilung der Teilaufgaben zwischen Roboter und Mensch vorgenommen. Dazu werden die Einzelkomponenten des Roboters, wie Kameras, Greifer, und Roboterarm sowie deren (An)Steuerung ebenfalls modularisiert und autark lauffähig und mit der Integration eines OPC UA Servers web- bzw. cloud-fähig gemacht. Diese Hardwaredienste werden dann im Netzwerk veröffentlicht und können durch einen zentralen Orchestringsdienst also OPC UA Client abonniert und so einer Anwendung zur Verfügung gestellt werden. Die Logik der Anwendung wird dann durch weitere Dienste ergänzt, die für Bildverarbeitung, Aufgabenplanung und Handhabung zuständig sind (siehe Abbildung 5).

Ein weiterer Anwendungsfall stellt eine Objektverfolgung dar. Hierbei werden die Hardwaredienste von Robotertreiber und Kameratreiber wiederverwendet und andere Dienste einfach ausgetauscht. Das heißt der Mehrwertdienst Pick&Place wird durch einen Visual-Servoing-Dienst ersetzt. Der Bilderkennungsdienst bekommt ein Update, um unnötige Klassifizierungsfunktionen herauszunehmen und dafür ortsabhängige Parameter wie x und y zurück zu geben, womit die Granularität des Dienstes also abgenommen hat. Ein Belichtungssteuerungsdienst kann wahlweise ebenfalls aktiviert werden, um bei wechselnden Lichtverhältnissen die zuverlässige Funktionalität des Bilderkennungsdienstes zu gewährleisten. Die gesamte Dienstarchitektur hat sich nun nur teilweise geändert um mit der gleichen Hardware einen ganz anderen Anwendungsfall zu bearbeiten. Hierbei werden die einfache Wiederverwendbarkeit und die damit verbundenen Vorteile der Zeitersparnis und Flexibilität deutlich.

Durch die Verwendung von OPC UA als industrienahe Kommunikationsstandard ist die Verbreitung in industriellen Anwendungen und Anbindung von weiteren Geräten und Diensten vereinfacht. Die Zusammenstellung der Dienste zur Laufzeit ist mit OPC UA ebenfalls möglich, was eine schnelle Reaktion auf veränderte Bedingungen erlaubt. Beispielsweise können Greifer und Kameras und deren Hardwaredienste ausgetauscht werden, ohne die übrigen Dienste des Systems zu berühren. Auch die Bildverarbeitung kann ein Update bekommen um neue zu erkennende Objekte einzubinden.

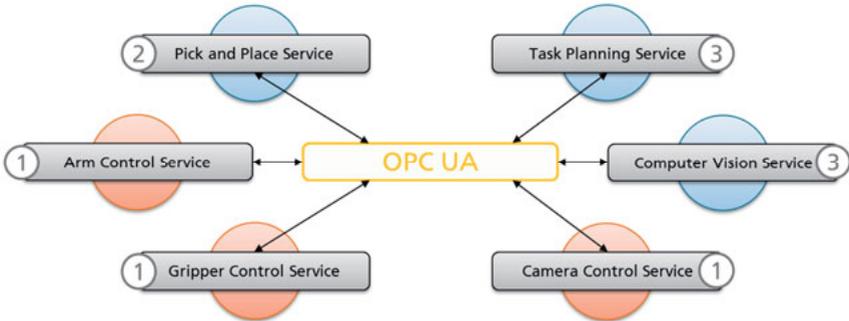


Abbildung 5: Komposition von Steuerungsdiensten für eine robotergestützte Handhabungsaufgabe.

5.4.3 Zustandslose Roboterdienste für eine Orchestrierung mit HTTP und JSON

Im Rahmen von Studierendenprojekten wurde der Tischdienstbasierte Roboteranwendungen auf Basis von REST/HTTP zu entwickeln. Eine besondere Herausforderung dabei war die Anwendung dieser Prinzipien einer ressourcenorientierten Architektur. Die einzelnen Dienste stellen Ressourcen mit einer URI zur Verfügung, mit denen über einfache CRUD-Operationen (Create, Read, Update, Delete) interagiert werden kann. Dazu werden die Standard-Verben von HTTP Requests genutzt (GET, POST, etc.), die Antworten werden über den MIME Type definiert und können bspw. Bilder oder beliebige in JSON

roboter 7bot [36] benutzt um REST/http zu entwickeln. Eine besondere Herausforderung dabei war die Anwendung dieser Prinzipien einer ressourcenorientierten Architektur. Die einzelnen Dienste stellen Ressourcen mit einer URI zur Verfügung, mit denen über einfache CRUD-Operationen (Create, Read, Update, Delete) interagiert werden kann. Dazu werden die Standard-Verben von HTTP Requests genutzt (GET, POST, etc.), die Antworten werden über den MIME Type definiert und können bspw. Bilder oder beliebige in JSON

Im konkreten Use-Case wurden zunächst unsortierte, farbig Holzbausteine über eine Kamera detektiert, und vom Roboter mittels eines Sauggreifers manipuliert. Das Ziel war es, bestimmte Muster oder Buchstabenkombinationen aus den Bausteinen zu legen, wie beispielhaft in Abbildung 6 dargestellt.

farbig Holzbausteine über eine Kamera detektiert, und vom Roboter mittels eines Sauggreifers manipuliert. Das Ziel war es, bestimmte Muster oder Buchstabenkombinationen aus den Bausteinen zu legen, wie beispielhaft in Abbildung 6 dargestellt.

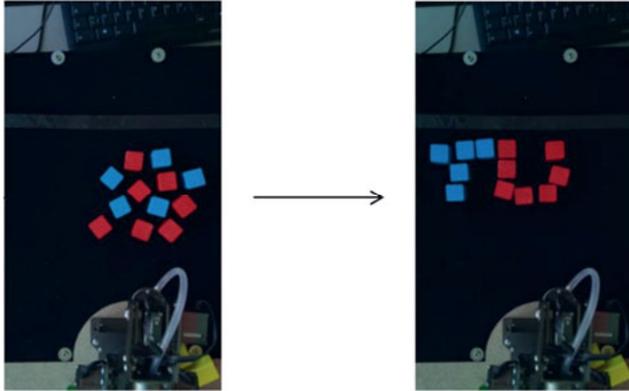


Abbildung 6: Das Anwendungsbeispiel für den 7bot. Links: unsortierte Bausteine. Rechts: Der Roboter hat die Steine nach Farben sortiert und „TU“ geschrieben. Die Positioniergenauigkeit ist nicht im Fokus des Projekts und wird maßgeblich durch den Ti schroboter beeinträchtigt.

Zur Umsetzung dieses Szenarios wurden zunächst eine Reihe technischer Dienste entwickelt. Diese umfassen die einzelnen Hardware-Dienste sowie Funktionalitäten zur Bildverarbeitung und Objekterkennung. Anschließend wurden die Anwendungsdienste programmiert, welche die Prozesslogik enthalten und die technischen Dienste konsumieren. Es handelt sich also um eine Orchestrierung, wobei die Anwendungsdienste die zentralen Clients sind. Die Unterscheidung der Dienste ist in Abbildung 7 zu sehen.

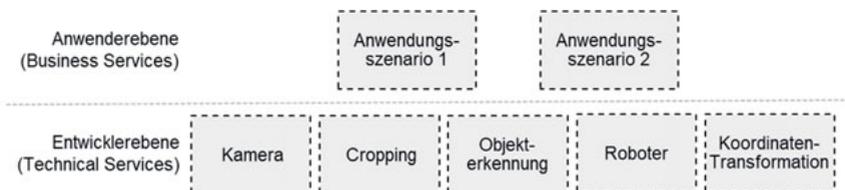


Abbildung 7: Die Dienste wurden in Anwenderebene und Entwicklerebene unterteilt. Neue Anwendungen können in diversen Hochsprachen (z.B. Python) niedrigschwellig entwickelt werden.

Bei diesem Use-Case handelt es sich um den Fall Anwendung aus der Cloud, der Roboter selbst wird lokal über ROS angesteuert und bietet eine Schnittstelle über HTTP nach außen. Die Dienste wurden relativ feingranular erstellt, insbesondere am Beispiel des Cropping Dienstes zu erkennen, der lediglich die Funktionalität hat ein Bild zuzuschneiden. Der Objekterkennungsdienst ist etwas komplexer und kann bzgl. Farbe und Form der gesuchten Objekte konfiguriert werden.

Vorteilhaft bei der Entwicklung hat sich erwiesen, dass die Dienste sehr einfach browserbasiert getestet werden können. Die Kodierung der Daten in JSON ist sehr einfach und durch die Lesbarkeit gut nachvollziehbar. Es entsteht ein Overhead bei der Übertragung binärer Daten durch das base64-Encoding. Das Protokoll ist zustandslos, d.h. jede Anfrage enthält alle Informationen die zur Beantwortung notwendig ist, es werden keinerlei Sitzungsinformationen gespeichert. Dies ermöglicht eine Skalierbarkeit der Dienste, ausgenommen der Hardwaredienste, da diese auf physische Ressourcen zugreifen.

Die Dienste REST-konform zu implementieren und somit den Prinzipien der ressourcen-orientierten Architektur zu folgen bietet Vor- und Nachteile. Ein Vorteil ist der optimierte Datenfluss. Obwohl die Dienste der Anwendungsebene als zentraler Orchestrationsclient fungieren, so müssen nicht sämtliche Daten auch über diesen Knoten übertragen werden. So wird beispielsweise beim Aufruf der Adresse der neu erstellten Bildressource zurückgegeben, anstatt das vollständige Bild zu übertragen. Den Bildverarbeitungsdiensten werden dann lediglich diese Adressen als Input geschickt, und sie holen sich dann über einen weiteren Request selbst die Bilddaten beim entsprechenden Dienst. Diese Vorgehensweise reduziert zwar die Menge an Daten, die übertragen werden muss, erhöht jedoch die Anzahl der Anfragen, die insgesamt getätigt werden muss. Im REST Konzept ist die Schnittstelle selbstbeschreibend und man kann sich vom Entrypoint (der Endpunkts-URI des Dienstes) über Links zu den einzelnen Ressourcen verbinden. Damit ist der Client robust gegenüber Änderungen der einzelnen Adressen, benötigt aber mehrere Anfragen um beim Ziel zu landen. Sofern Daten aus mehreren Ressourcen abgefragt werden müssen, sind auch mehrere Anfragen notwendig. Dies ist beispielsweise der Fall, wenn vom Kamerarserver ein neues Bild gemacht und ausgelesen werden soll. Zunächst wird über die POST Methode eine neue Bildressource erstellt, als Antwort kommt der Link zu dem Bild zurück. Um dieses Bild tatsächlich abzurufen, muss es nun mit einem GET Befehl abgefragt werden. Sollen zusätzlich die Kameraparameter abgefragt werden, ist ein weiterer GET Befehl notwendig.

Problematisch gestalten sich bei HTTP/JSON die Hardware-Dienste, wie z.B. der Dienst für den Roboter. In unserem Fall haben wir die Fahrbefehle blockierend gestaltet, so dass die HTTP Response erst kam, wenn der Roboter auch tatsächlich sein Ziel erreicht hat. Dies führt allerdings bei längeren Fahrwegen zu einem Timeout der Verbindung – transparenter wäre es die Antwort vom Server zu bekommen, dass der Fahrbefehl angenommen wurde und ausgeführt wird. Das erhöht jedoch die Komplexität im Client, da dieser nun aktiv beim Server nachfragen muss, wann die Fahrbewegung abgeschlossen wurde. Aufgrund der Zustandslosigkeit des Protokolls gibt es keinerlei Möglichkeit, dem Client diese Information vom Server aus zukommen zu lassen.

5.4.4 Plattformunabhängiges Kommunikationssystem für die Steuerung des UR5 mit gRPC

Eine Gruppe von Mitarbeitern des Instituts nahm an der Industrial Category der World Robot Challenge teil. Die Aufgaben reichten von einfachen Montageoperationen über Bin Picking / Kitting bis hin zur Montage einer kompletten Getriebeeinheit. Die detaillierte Aufgabenstellung des Wettbewerbs sowie der gesamte Aufbau hinsichtlich Hardwaren umfassend in [38] beschrieben. An dieser Stelle werden die Aspekte der verwendeten dienstebasierten Architektur

er dem Teamarbeit BerlinAUTs 2018 [37] teil. Die Aufgaben Picking / Kitting bis hin zur Montage einer kompletten Getriebeeinheit. Die detaillierte Aufgabenstellung des Wettbewerbs sowie der gesamte Aufbau hinsichtlich Hardwaren umfassend in [38] beschrieben. An dieser Stelle werden die Aspekte der verwendeten dienstebasierten Architektur

Das System besteht aus einem Universal Robot UR5 und insgesamt drei Kameras für Objekterkennung, Markerdetektion sowie Visual Servoing. Die Aufteilung der Dienste ist relativ grob granular, sodass insgesamt nur lediglich drei komplexe Dienste genutzt wurden (siehe Abbildung 8). Dies ist zum einen der Roboter Controller, der die Bewegung des Roboters sowie des Greifers vollständig steuert. Der Vision Server, der die komplette Kette von der Bildaufnahme bis hin zur Bildverarbeitung für den jeweiligen Anwendungsfall realisiert. Darüber hinaus gibt es noch einen Konfigurationsserver, der Informationen über Bauteile und deren Greifpunkte enthält, die Konfiguration des Arbeitsplatzes (bspw. welche Teile in welchen Bins liegen), und die Beschreibung der Aufgabe (welche Teile zu einem Kit zusammengestellt werden sollen). Da diese Informationen dynamisch von den einzelnen Modulen erfragt werden können, ist eine flexible Anpassung des Arbeitsplatzes möglich, ohne an existierendem Code Veränderungen durchführen zu müssen. Der Prozess Controller ist schlussendlich die Prozesslogik und nutzt die zuvor beschriebenen

er dem Teamarbeit BerlinAUTs 2018 [37] teil. Die Aufgaben Picking / Kitting bis hin zur Montage einer kompletten Getriebeeinheit. Die detaillierte Aufgabenstellung des Wettbewerbs sowie der gesamte Aufbau hinsichtlich Hardwaren umfassend in [38] beschrieben. An dieser Stelle werden die Aspekte der verwendeten dienstebasierten Architektur

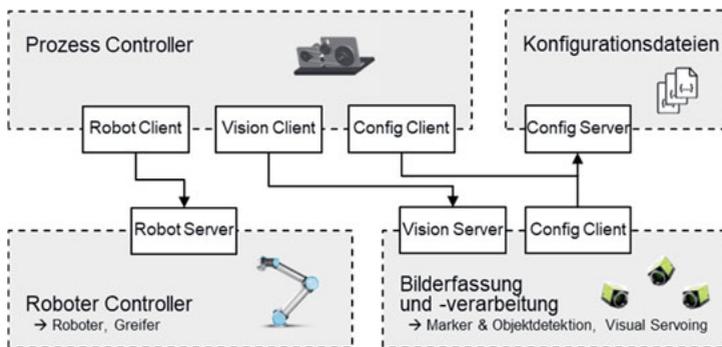


Abbildung 8: Die verschiedenen Dienste im Aufbau der BerlinAUTs (Vgl. [38]).

Diese grobe Partitionierung erlaubt eine große Flexibilität und macht die Anwendungsentwicklung auf seitens des Prozess Controllers einfacher. Darüber hinaus gibt es einen geringen Kommunikationsoverhead, was die gesamte Performance steigert. An einigen Stellen bringt die Partitionierung aber auch Probleme mit sich. Insbesondere

die kombinierte Bilderfassung und –verarbeitung kann in der Entwicklung nachteilig sein: Sobald ein größerer Funktionsumfang in der Bildverarbeitung benötigt wird, muss dieser Dienst verändert werden – es ist nicht möglich, einen zusätzlichen Dienst zu entwickeln, der lediglich die neue Funktionalität enthält. Eine Trennung von Bilderfassung und Bildverarbeitung erlaubt hier eine deutlich höhere Flexibilität.

Darüber hinaus ist die Integration des Greifers in den Roboter Controller zwar für diesen Use-Case sehr nützlich. Sobald jedoch ein anderer Greifer verwendet wird (bspw. Sauggreifer statt 2-Finger-Greifer), ist eine Änderung der Schnittstellen für den gesamten Roboterdienst erforderlich. Die Trennung von Greifer und Roboter wäre hier sinnvoll.

Zur Kommunikation zwischen den Diensten wird gRPC genutzt, welches auf den Protocol Buffers basiert und sich durch strenge Typisierung und hohe Effizienz (u.a. Dank HTTP2.0) auszeichnet. Im Gegensatz zum vorher beschriebenen REST Use-Case sind die Dienste nicht Ressourcen-orientiert, es handelt sich vielmehr um Remote Procedure Calls.

5.5 Diskussion und Best Practices

Die Vorstellung der verschiedenen Use-Cases zeigt auf, dass sich die Architekturen für dienste-basierte Robotersysteme in ihrer Umsetzung sehr stark unterscheiden können. Um Best Practices für die Gestaltung ableiten zu können, werden wir die in Abschnitt 5.2.2 vorgestellten Aspekte von dienstebasierten Architekturen für Robotersysteme im Detail für die vorgestellten Use-Cases diskutieren.

5.5.1 Grad der Auslagerung in die Cloud

Ein entscheidender Punkt ist die Frage, wann eine Aufgabe in die Cloud ausgelagert werden sollte und wann diese besser mit lokalen Ressourcen zu lösen sei. Wir haben dazu insbesondere zwei Typen unterschieden: Anwendung aus der Cloud und Roboter aus der Cloud. Bei letzterem erfolgt die Robotersteuerung aus der Cloud, d.h. Interpolation findet in der Cloud statt und es werden bspw. Achsgeschwindigkeiten an den Roboter gesendet. Dies ist insbesondere dann interessant, wenn ein Low-Level Zugriff auf den Roboter notwendig ist, bspw. für Aufgaben wie Direct Visual Servoing oder Kraftregelung. Dies stellt allerdings weiterhin eine große Herausforderung dar, da der Regelgakt dabei von dem Zusammenspiel verteilter Dienste abhängig ist. Zusätzlich müssen Sicherheitsfeatures implementiert werden, damit bei Abbruch der Verbindung der Roboter nicht endlos mit der zuletzt gesetzten Achsgeschwindigkeit weiterfährt, sondern autonom durch einen lokalen Controller in einen sicheren Zustand übergeht.

Für die meisten Aufgaben ist das Modell Anwendung aus der Cloud ausreichend. Die Robotersteuerung verbleibt dabei lokal und stellt eine High-Level Schnittstelle zur Verfügung, die gewöhnliche Bewegungsbefehle oder sogar bereits zusammengesetzte Bewegungen (Pick-and-Place) annehmen kann. Sicherheitsfeatures verbleiben ebenfalls lokal und erfordern daher im Allgemeinen keine zusätzliche Berücksichtigung in der Architektur. Die Anforderungen an das Netzwerk und die Effizienz der Kommunikation sinken. Unserer Meinung nach ist dieses Modell zunächst zu bevorzugen, und nur sofern erforderlich sollte das Modell Roboter aus der Cloud verwendet werden.

5.5.2 Partitionierung und Granularität

Das Aufteilen der Funktionalitäten und das Finden der korrekten Größe eines Dienstes ist häufig ein iterativer Prozess. Es ist schwer im Voraus bereits zu erahnen, auf welche Arten und Weisen die Dienste wiederverwendet werden könnten. So ergibt sich insbesondere bei der Umsetzung neuer Use-Cases mit existierenden Diensten, dass ggf. eine andere Partitionierung besser geeignet wäre.

Grundlegende Richtlinien lassen sich dennoch formulieren. So ist es beispielsweise ratsam, hardwarebezogene Funktionalitäten nicht mit reinen softwarebasierten Funktionalitäten zu vermischen, wie bei dem Use_Case in Abschnitt 5.4.4 für den Dienst zur Bilderfassung und -verarbeitung geschehen. Damit wird ermöglicht die Anwendung mit anderer Hardware (z.B. neuen Kameras) auszuführen, indem lediglich ein Dienst (z.B. Bilderfassungsdienst) ausgetauscht oder neu konfiguriert werden muss.

Etwas entschärft wird die Problematik der korrekten Granularität durch die Möglichkeit zusammengesetzter Dienste. So könnte ein Teilprozess aus den Operationen A, B, und C bestehen. Sind die Operationen als einzelne Dienste umgesetzt, so können diese Funktionalitäten einzeln genutzt werden. Der sich daraus zusammensetzende Teilprozess kann aber ebenfalls als Dienst angeboten werden und stellt damit eine größere Granularitätsstufe dar. Für dieses Muster können die einzelnen Dienste also auch relativ fein granular sein. Solche Verschachtelungen stellen jedoch kein Allheilmittel dar, da zusätzliche Kommunikationswege entstehen. Die Netzwerklast steigt und die Performance der Anwendung kann sinken. Insbesondere bei Bildverarbeitungsdiensten werden große Datenmengen übertragen.

5.5.3 Komposition der Dienste

Die vorgestellten Use-Cases sind fast immer als Orchestrierung zusammen gesetzt, das bedeutet dass ein Client, der die Prozesslogik kennt, die Dienste aufruft und in der richtigen Reihenfolge benutzt. Der Vorteil daran ist, dass die Instanzen der Dienste ohne weiteres in mehreren Prozessen gleichzeitig genutzt werden können und so Skalierungseffekte besser wirken. Durch den zentralen Knoten ist der Prozessablauf besser nachvollziehbar und möglicherweise einfacher zu Debuggen, jedoch werden auch zusätzliche Kommunikationswege in Kauf genommen.

Das Prinzip der Choreographie setzt genau da an. Je nisse direkt an den nächsten Dienst, der diese verale Client und die Kommunikationswege sind so kurz das auch, dass die Prozesslogik in die einzelnen Kn Teilweise sind auch Mischformen möglich. Bei Nutzun gesprochenen zusammengesetzten Dienste ist vorstell tionen nach dem Client/Server Prinzip aufgerufen we einer Choreographie verankert ist.

der Dienst sendet seine Ergeb- rebeiten soll. Damit entfällt der zent- wie möglich. Jedoch bedeutet oten integriert wird. g der im vorigen Abschnitt an- bar, dass die einzelnen Opera- rden, der Dienst selbst aber in

5.5.4 Kommunikationsprotokolle

Auch wenn die vorgestellten Use-Cases immer nur auf koll basierten, so muss das nicht zwangsläufig so s demischen Beispielen geschuldet, dass sämtliche gen wickelt wurden. Sobald beispielsweise Dienste von D den sollen, muss mit der angebotenen Schnittstelle dene Kommunikationsprotokolle zu unterstützen könne nutzt werden.

einem Kommunikationsproto- ein. Tatsächlich ist es nur den aka- utzten Dienste auch selbst ent- rittanbietern eingebunden wer- gearbeitet werden. Um verschie- n Adapter oder Gateways be-

Bei uns wurden ROS, OPC UA, REST und gRPC als Schni ttstellen bzw. Frameworks verwendet. ROS ist insbesondere bei der Entwicklung von Robotersystemen unange- fochten, im Kontext der Produktion geht es jedoch n icht darum ständig neue Kompo- nenten zu entwickeln, sondern aus einem Katalog von Diensten die eigene Anwen- dung schnellstmöglich ohne großen Einarbeitungsaufw and zu komponieren. Dafür sind andere Lösungen zum Teil besser geeignet.

ttstellen bzw. Frameworks von Robotersystemen unange- icht darum ständig neue Kompo- Diensten die eigene Anwen- and zu komponieren. Dafür

Dabei eignet sich REST insbesondere für ressourcen- basierte Strukturen wie zum Bei- spiel Wissensdatenbanken. Zur Kapselung von Prozess logik ist gRPC ideal. OPC UA ist für beides weitestgehend gut zu nutzen. Insbeso ndere bei zeitkritischen Systemen ist auf effiziente Kommunikation zu achten, so wie sie bei gRPC und OPC UA umge- setzt wird.

logik ist gRPC ideal. OPC UA ndere bei zeitkritischen Systemen sie bei gRPC und OPC UA umge-

ROS ist nur für Unix Systeme verfügbar und man hat als Entwickler die Wahl zwischen Python und C++. Bei OPC UA und gRPC werden deutlich mehr Sprachen unterstützt, bei REST werden schließlich ganz normale Web-Techno logien benutzt und die Pro- grammierer sind völlig frei in ihrer Wahl.

als Entwickler die Wahl zwischen mehr Sprachen unterstützt, logien benutzt und die Pro-

5.6 Zusammenfassung und Ausblick

In dieser Arbeit konnten wir Einblicke in verschied ene Robotersysteme geben, die mit dienstebasierten Architekturen umgesetzt wurden. Wi r haben die einzelnen Systeme

ene Robotersysteme geben, die mit r haben die einzelnen Systeme

hinsichtlich verschiedener Aspekte im Detail untersucht und ihre Stärken und Schwächen identifiziert. So konnten wir Best Practices für die Gestaltung weiterer dienstebasierter Robotersysteme ableiten.

Darüber hinaus gibt es natürlich noch weitere Aspekte, die im Zusammenhang mit dienstebasierten Robotersystemen früher oder später aufkommen werden. Eines dieser Themen ist die automatische Koordinierung und Ressourcenzuteilung. Jüngste Arbeiten haben Koordinierungsmechanismen vorgeschlagen. Darunter sind eine Art Auktion [39], sequentielle Spiele [40] und "belief propagation"-basierende Methoden [41] für Echtzeit-Ressourcenzuteilung und deren geeigneter Abruf in cloudbasierten mobilen Robotersystemen.

Im Zusammenhang damit stehen auch die Anforderungen an das Netzwerk und die auftretenden Latenzen, was insbesondere zum Tragen kommt wenn echtzeitkritische Systeme aus der Cloud realisiert werden (vgl. Roboter aus der Cloud). Als Perspektiven sind hier 5G und Time Sensitive Networking (TSN) zu nennen. Das Verhalten und die Laufzeit der Dienste selbst muss dabei jedoch auch einer bestimmten Quality of Service (QoS) entsprechen und ggf. deterministische werden. In einer kürzlich erschienenen Arbeit [42] schlugen die Autoren vor, eine "RobotCloud" zu benutzen, der ganz spezifische Aufgaben mit jeweils immer absehbarem Zeitverhalten zugeteilt werden.

Ein letzter Aspekt den wir an dieser Stelle nennen möchten ist die Sicherheit und der Schutz der Daten in solchen Systemen. Das Manipulieren der Kommunikation kann nicht nur zum Ausfall der Systeme führen, sondern tatsächlich auch physische Schäden verursachen, wenn die Gewalt über die Steuerung des Roboters von Hackern oder böswilligen Nutzern übernommen wird. Geeignete Authentifizierungsverfahren und mehrschichtige Verschlüsselungsmechanismen werden derzeit für einen unzweifelhaften Zugriff auf Cloud-basierende Dienstleistungen erforscht [34]. In diesem Zusammenhang wurde von Forschern auch der Begriff der Kryptorobotik als Vereinheitlichung von Internetsicherheit und Robotik verwendet [43]. Andererseits dürfen sich die integrierten Sicherheitsmechanismen für Verschlüsselung und Datenvalidierung natürlich nicht negativ auf das Echtzeitverhalten der Roboter-Cloud-Systeme auswirken.

5.7 Danksagung

Wir bedanken uns ganz herzlich bei allen, die an den vorgestellten Use-Cases mitgewirkt haben und damit die Grundlage für diesen Beitrag geschaffen haben. Dazu gehören: Jan Guhl, Moritz Chemnitz, Oliver Heimann, Christian Niebuhr, Ondřej Vonásek, Jonas Wassermann, Julian Deprez, Abdulrahman Hussain, Karuna Koch, Ahmad Mohammad, Michael Steudter, Vu Ninh Le, Jan Hackbart, David Romero, Yongbao Tian, Yunfan Zhao.

5.8 Literatur

- [1] MESA International, Service-oriented Architecture in Manufacturing Guidebook: Whitepaper 27. Chandler, AZ: MESA International, IBM Corporation and Capgemini, 2008.
- [2] J. Krüger et al., “Innovative control of assembly systems and lines,” *CIRP Annals*, vol. 66, no. 2, pp. 707–730, 2017.
- [3] G. Hu, W. P. Tay, and Y. Wen, “Cloud robotics: architecture, challenges and applications,” *IEEE network*, vol. 26, no. 3, 2012.
- [4] M. Waibel et al., “Roboearth,” *IEEE Robotics & Automation Magazine*, vol. 18, no. 2, pp. 69–82, 2011.
- [5] X. V. Wang, L. Wang, A. Mohammed, and M. Givehchi, “Ubiquitous manufacturing system based on Cloud: A robotics application,” *Robotics and Computer-Integrated Manufacturing*, vol. 45, pp. 116–125, 2017.
- [6] B. Qureshi and A. Koubâa, “Five traits of performance enhancement using cloud robotics: A survey,” *Procedia Computer Science*, vol. 37, pp. 220–227, 2014.
- [7] A. Vick, V. Vonásek, R. Pěnička, and J. Krüger, “Robot control as a service—towards cloud-based motion planning and control for industrial robots,” *Robot Motion and Control (RoMoCo)*, 2015 10th International Workshop on 2015, pp. 33–39.
- [8] A. Vick, J. Guhl, and J. Krüger, “Model predictive control as a service—Concept and architecture for use in cloud-based robot control,” *Methods and Models in Automation and Robotics (MMAR)*, 2016 21st International Conference on, 2016, pp. 607–612.
- [9] M. Rudorfer, T. J. Pannen, and J. Krüger, “A Case Study on Granularity of Industrial Vision Services,” in *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, 2018, p. 59.
- [10] C. Peltz, “Web services orchestration and choreography,” *Computer*, vol. 36, no. 10, pp. 46–52, 2003.
- [11] J. Dizdarevic, F. Carpio, A. Jukan, and X. Masip-Bruin, “Survey of Communication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration,” *arXiv preprint arXiv:1804.01747*, 2018.
- [12] T. Bauernhansl, J. Krüger, G. Reinhart, and G. Schuh, “WGP-Standpunkt Industrie 4.0,” *Wissenschaftliche Gesellschaft für Produktionstechnik WGP e.V.* 2016.

- [13] A. Vick and J. Krueger, "Using OPC UA for Distributed Industrial Robot Control," *ISR 2018; 50th International Symposium on Robotics*, Munich, Germany, 2018, pp. 1-6.
- [14] gRPC Ecosystem, *grpc-gateway*, <https://github.com/grpc-ecosystem/grpc-gateway>, aufgerufen am 04.01.2019.
- [15] J. Minguez, "Der Manufacturing Service Bus," in *Digitale Produktion*: Springer, 2013, pp. 271–289.
- [16] Mell, P.; Grance, T. *The Nist Definition of Cloud Computing*; Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2011; S. 1–3.
- [17] Chen, Y.; Du, Z.; García-Acosta, M. Robot as a service in cloud computing. In *Proceedings of the 2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE)*, Nanjing, China, 4. – 5. Juni 2010; S. 151–158.
- [18] Chen, Y.; Hu, H. Internet of intelligent things and robot as a service. *Simul. Model. Pract. Theory* 2013, S. 34, 159–171.
- [19] Arumugam, R.; Enti, V.R.; Liu, B.; Wu, X.; Basakaran, K.; Kong, F.F.; Kumar, A.S.; Meng, K.D.; Kit, G.W. *DAVinCi: A cloud computing framework for service robots*. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 3.–7. Mai 2010; S. 3084–3089.
- [20] Mohanarajah, G.; Hunziker, D.; D'Andrea, R.; Waißel, M. Rapyuta: A cloud robotics platform. *IEEE Trans. Autom. Sci. Eng.* 2015, S. 12, 481–493.
- [21] Riazuelo, L.; Civera, J.; Montiel, J. C2tam: A cloud framework for cooperative tracking and mapping. *Robot. Auton. Syst.* 2014, 62, 401–413.
- [22] Crick, C.; Jay, G.; Osentoski, S.; Jenkins, O. C. ROS and rosbridge: Roboticians out of the loop. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human–Robot Interaction*, Boston, MA, USA, 5.–8. März 2012; ACM: New York, NY, USA, 2012; S. 493–494.
- [23] Crick, C.; Jay, G.; Osentoski, S.; Pitzer, B.; Jenkins, O.C. *Rosbridge: Ros for nonros users*. In *Robotics Research*; Springer: Cham, Schweiz, 2017; S. 493–504.
- [24] Lee, J. *Web Applications for Robots Using Rosbridge*; Brown University: Providence, RI, USA, 2012.
- [25] Alexander, B.; Hsiao, K.; Jenkins, C.; Suay, B.; Toris, R. Robot web tools [ros topics]. *IEEE Robot. Autom. Mag.* 2012, S. 19, 20–23.
- [26] Toris, R.; Kammerl, J.; Lu, D.V.; Lee, J.; Jenkins, O.C.; Osentoski, S.; Wills, M.; Chernova, S. Robot web tools: Efficient messaging for cloud robotics. In

- Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Deutschland, 28. September–2 Oktober 2015; S. 4530–4537.
- [27] Sato, M.; Kamei, K.; Nishio, S.; Hagita, N. The ubiquitous network robot platform: Common platform for continuous daily robotic services. In Proceedings of the 2011 IEEE/SICE International Symposium on System Integration (SII), Kyoto, Japan, 20.–22. Dezember 2011; S. 318–323.
- [28] Tenorth, M.; Kamei, K.; Satake, S.; Miyashita, T.; Hagita, N. Building knowledgeenabled cloud robotics applications using the ubiquitous network robot platform. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; S. 5716–5721.
- [29] Gherardi, L.; Hunziker, D.; Mohanarajah, G. A software product line approach for configuring cloud robotics applications. In Proceedings of the 2014 IEEE 7th International Conference on Cloud Computing (CLOUD), Anchorage, AK, USA, 27. Juni–2. Juli 2014; S. 745–752.
- [30] Kato, Y.; Izui, T.; Tsuchiya, Y.; Narita, M.; Ueki, M.; Murakawa, Y.; Okabayashi, K. Rsi-cloud for integrating robot services with internet services. In Proceedings of the IEEE IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society, Melbourne, Australien, 7.–10. November 2011; S. 2158–2163.
- [31] Nakagawa, S.; Ohyama, N.; Sakaguchi, K.; Nakayama, H.; Igarashi, N.; Tsunoda, R.; Shimizu, S.; Narita, M.; Kato, Y. A distributed service framework for integrating robots with internet services. In Proceedings of the 2012 IEEEth26 International Conference on Advanced Information Networking and Applications (AINA), Fukuoka, Japan, 26.–29. März 2012; S. 31–37.
- [32] Nakagawa, S.; Igarashi, N.; Tsuchiya, Y.; Narita, M.; Kato, Y. An implementation of a distributed service framework for cloud-based robot services. In Proceedings of the IEEE IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Kanada, 25.–28. Oktober 2012; S. 4148–4153.
- [33] Narita, M.; Okabe, S.; Kato, Y.; Murakawa, Y.; Okabayashi, K.; Kanda, S. Reliable cloud-based robot services. In Proceedings of the IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10.–13. November 2013; S. 8317–8322.
- [34] Nandhini, C.; Doriya, R. Towards secured cloud-based robotic services. In Proceedings of the 2017 International Conference on IEEE Signal Processing

- and Communication (ICSPC), Coimbatore, Indien, 28.– 19. Juli 2017; S. 165–170.
- [35] Du, Z.; He, L.; Chen, Y.; Xiao, Y.; Gao, P.; Wang, T. Robot Cloud: Bridging the power of robotics and cloud computing. *Future Generation Comput. Syst.* 2017, S. 74, 337–348.
- [36] Kickstarter Projekt 7bot, <https://www.kickstarter.com/projects/1128055363/7bot-a-powerful-desktop-robot-arm-for-future-invention>, aufgerufen am 28.01.2019.
- [37] World Robot Challenge 2018, <http://worldrobotsummit.org/en/wrc2018/>, aufgerufen am 29.01.2019.
- [38] O. Kröger, J. Guhl, O. Heimann, M. Katanacho, C. Niebuhr, M. Rudorfer, T. Özkaya, J. Wassermann, J. Hügler, T. Pannen, J. Krüger, „A Service-oriented Robotic Manufacturing System: Lessons Learned from Participating in the World Robot Challenge 2018“, zur Veröffentlichung angenommen, 2019.
- [39] Kehoe, B.; Patil, S.; Abbeel, P.; Goldberg, K. A survey of research on cloud robotics and automation. *IEEE Trans. Autom. Sci. Eng.* 2015, S. 12, 398–409.
- [40] Wang, L.; Liu, M.; Meng, M.Q.H. A pricing mechanism for task oriented resource allocation in cloud robotics. In *Robots and Sensor Clouds*; Springer: Cham, Schweiz, 2016; S. 3–31.
- [41] Kong, Y.; Zhang, M.; Ye, D. A belief propagation-based method for task allocation in open and dynamic cloud environments. *Knowledge-Based Syst.* 2017, S. 123-132.
- [42] Li, Y.; Wang, H.; Ding, B.; Shi, P.; Liu, X. Toward QoS-aware cloud robotic applications: A hybrid architecture and its implementation. In *Proceedings of the 2016 Intl IEEE Conferences Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, Toulouse, Frankreich, 18. – 21. Juli 2016; S. 33–40.
- [43] Morante, S.; Victores, J.G.; Balaguer, C. Cryptobotics: Why robots need cyber safety. *Front. Robot. AI* 2015, S. 2, 23.

6 Smart Centering – Methodik zur intelligenten Zentrierung rotationssymmetrischer Körper für die Null-Fehler-Fertigung in mehrstufigen Produktionssystemen

C. Reiff, F. Eger und Prof. A. Verl, ISW, Uni Stuttgart

6.1 Einleitung

Im Zuge der Globalisierung und des damit verbundene n zunehmenden Wettbewerbs zwischen Niedriglohnländern und der europäischen Fertigungsindustrie spielen die Produktqualität, die daraus resultierende Kundenzufriedenheit sowie die Wirtschaftlichkeit und Ressourceneffizienz der Produktion eine entscheidende Rolle für den Erfolg der Unternehmen. Damit die europäische Fertigungsindustrie diese enormen Herausforderungen bewältigen kann und auch in Zukunft konkurrenzfähig bleibt, müssen die bisherigen Ansätze zur Produktionsplanung und die bestehenden Fertigungsabläufe, insbesondere hochqualitativer Bauteile grundlegend überdacht und umfassend weiterentwickelt werden. Der gemeinsame Wunsch der Unternehmen von Mittelstand bis Großkonzern vereint, ist eine kostengünstige und flexible Gestaltung der Fertigung unter dem Einsatz modernster Methoden und Technologien, mit dem Ziel eine Produktion zu realisieren, die möglichst keine zeitintensive Nacharbeit erfordert oder gar Ausschuss produziert – eine Null-Fehler-Fertigung.

In der Vergangenheit lag der Schwerpunkt auf der Betrachtung einzelner und getrennter Prozesse unter dem Einsatz statischer Prozessleitsysteme. Aber auch nach der Analyse und Optimierung einzelner Produktionsschritte ist die Möglichkeit einer Fehlerentstehung in Form von sich über mehrere Prozessfortpflanzenden und überlagernden Fehlerbildern gegeben. Dazu zählen beispielsweise Maß- und Formabweichungen durch Werkzeugverschleiß oder Inhomogenität des verarbeiteten Rohmaterials. Da das fehlerhafte Bauteil bisher weiterhin seinen vorgesehenen Produktionsprozess durchläuft, werden solche Fehler kumuliert, was in vielen Fällen zu Ausschuss oder hohen Kosten durch zeitintensive Nacharbeit führt [1]. Diese Abweichungen sind oft nicht vorhersehbar und ihre Ursachen können meistens nicht ad hoc erkannt werden, insbesondere dann, wenn nur eine begrenzte Anzahl von Sensorsystemen und Analyse-Methoden eingesetzt werden. Diese Defekte werden daher oft erst am Ende eines Fertigungsprozesses, während der Qualitätskontrolle festgestellt.

In dieser Arbeit wird am Beispiel der Fertigung von rotationssymmetrischen Radsatzwellen für Schienenfahrzeuge, der Einsatz von Null-Fehler-Fertigungsansätzen unter Berücksichtigung des gesamten Produktionssystems exemplarisch demonstriert.

Dazu wurde eine browserbasierte Anwendung zur bauteilabhängigen Ausrichtung der Wellen entwickelt, welche Fehler die in einer frühen Phase des Fertigungsprozesses entstanden sind, identifiziert und innerhalb der eigentlichen Prozesskette und somit in Richtung des Materialflusses durch Anpassung späterer Prozessschritte, in Abhängigkeit der Ausprägung des Fehlers, kompensiert.

Dieser Beitrag ist wie folgt aufgebaut: Abschnitt 6.2 beschreibt die Grundlagen der Null-Fehler-Fertigung in mehrstufigen Produktionssystemen. Anschließend wird in Kapitel 6.3 die Fertigung der Radsatzwellen erläutert, um die dabei resultierende Problematik darzustellen. Abschnitt 6.4 stellt auf Basis des industriellen Anwendungsfalls die Methodik zur Optimierung des Gesamtsystems unter Verwendung einer adaptiven Fertigungsstrategie vor. Daraufhin wird in Abschnitt 6.5 die entwickelte browserbasierte Anwendung vorgestellt, die die beschriebene Methodik den Anwendern zugänglich macht sowie die relevanten Prozessschritte verknüpft, um eine automatisierte Ausführung der Optimierung zu ermöglichen. Abschließend werden die Ergebnisse in Kapitel 6.6 zusammengefasst und kritisch beleuchtet sowie zukünftige Arbeiten und Potenziale des Systems im Hinblick auf die gesamte Prozesskette zur Fertigung der Radsatzwellen beschrieben.

6.2 Grundlagen der Null-Fehler-Fertigung in mehrstufigen Produktionssystemen

Die Idee der Null-Fehler-Fertigung zielt darauf ab, Ausschuss und Nacharbeit zu eliminieren, indem mehrstufige Produktionssysteme durch datengesteuerte und lernbasierte Ansätze analysiert und optimiert werden. Das von der Europäischen Union, im Rahmen des Clusters Horizon 2020 geförderte Projekt ForZDM [2], befasst sich mit genau dieser Thematik. Das internationale Projekt hat es sich zum Ziel gesetzt, eine generische Architektur und Vorgehensweise zu erarbeiten, die von der Sensorentwicklung, über die zentralisierte Datenerfassung, die Datenanalyse unter Einsatz statistischer Methoden und lernender Algorithmen, bis hin zu proaktiven Eingriffen in den eigentlichen Fertigungsprozess, das gesamte Spektrum einer globalen Null-Fehler-Fertigung abbildet.

Entscheidend für die erfolgreiche Realisierung einer Null-Fehler-Fertigung ist das Erreichen einer möglichst transparenten Produktion [3]. Neben der Erfassung von Prozessvariablen und maschinenbezogenen Messwerten, müssen auch Qualitätsdaten der Bauteile aufgezeichnet werden, um die Zustände der Produktion sowie des Bauteils so lückenlos wie möglich erfassen zu können. Dabei ist es von enormer Bedeutung, die Messwerte den entsprechenden Bauteilen zuzuordnen zu können. Dies wird beispielsweise durch die Vergabe von QR-Codes realisiert, mit welchen jedes Bauteil eindeutig identifiziert werden kann. Die umfassende Datenerfassung, -Verarbeitung und -Auswertung bildet den Grundstein für alle weiteren übergeordneten Systeme, da

die frühzeitige Identifikation möglicher Abweichungen innerhalb der Produktionslinie in Richtung des Materialflusses kompensieren zu können. Damit eine zentralisierte Auswertung ermöglicht wird, werden alle Messwerte und Informationen in einer Datenbank gespeichert. Um die vorhandenen Potenziale der zunehmenden Vernetzung und zentralen Datenerfassung im Zuge der Digitalisierung der Produktion zu nutzen, ist es wichtig den ganzheitlichen Charakter des mehrstufigen Produktionssystems zu berücksichtigen [4].

Um Wissen aus den in der zentralen Datenbank gesammelten Daten extrahieren zu können, wird die Wissens-Extraktions-Plattform (Knowledge Capturing Platform – KCP) eingeführt [5]. Die Plattform beinhaltet verschiedene statistische und analytische Werkzeuge zur Verarbeitung der Daten. Die über das Multisensornetzwerk und die Datenmanagement-Plattform gesammelten Daten stellen die wichtigste Wissensquelle zur Identifikation der Ursachen der Fehlerentstehung und deren Ausbreitungsmechanismen entlang der Produktionslinie dar. Die KCP bildet somit die Schnittstelle zwischen den in der Datenbank verfügbaren Daten und der Steuerungsplattform, die auf Grundlage der ausgewerteten Daten und gewonnenen Erkenntnisse, Optimierungsstrategien definiert und ausführt. Dazu gehören zum einen Strategien zur Fehlervermeidung und zum anderen Maßnahmen zur Kompensation bereits aufgetretener Fehler.

Um die Fehlerentstehung von Grund auf zu vermeiden, müssen Wertänderungen von Variablen, die die Bildung von Fehlern implizieren, überwacht werden. Daher müssen prozessübergreifende Korrelationen auf Grundlage statistischer Analysen berücksichtigt werden, um die Zusammenhänge zwischen diesen Veränderungen und der Entstehung von Fehlern zu identifizieren. Dies ermöglicht eine vorausschauende Anpassung der beeinflussenden Prozess- oder Maschinenparameter an die transienten Bedingungen, die zum Beispiel durch Werkzeugverschleiß zu Abweichungen von der Soll-Geometrie, beziehungsweise zu späteren Defekten führen. Die Berücksichtigung stochastischer Defekte oder nicht beeinflussbarer Abweichungen kann jedoch nicht immer garantiert werden. Daher sind nachgeschaltete Kompensationsstrategien notwendig, um Fehler zu beheben, deren Entstehung nicht verhindert werden konnte. Dabei ist der erste wichtige Schritt die frühestmögliche Identifizierung des Fehlers und damit die Ermittlung möglicher Maßnahmen zur Kompensation oder zumindest Reduzierung der Auswirkungen der Abweichung. Ziel ist in erster Linie die Anpassung der Parameter der geplanten nachfolgenden Prozessschritte, die sich auf die Eigenschaften des Fehlers auswirken. Darüber hinaus kann es auch notwendig sein, die gesamte Prozesskette neu zu definieren und die Reihenfolge der einzelnen Prozessschritte zu ändern. Darüber hinaus kann auch die Anpassung der geplanten Nenngeometrie eines anderen Bauteils mit beibehaltenen Fehlern eventuell vollständige Fehler eventuell vollständige

dig beseitigt werden können. Dadurch wird die Produktion von Bauteilen für einen anderen Auftrag ermöglicht und damit Ausschuss reduziert sowie kosten- und zeitintensive Nacharbeit eingespart [6].

Abbildung zeigt die schematische Darstellung eines mehrstufigen Produktionssystems. Der obere Teil stellt die klassische Architektur mit der bisherigen rückwärtsgerichteten Fehlerkorrektur dar. Der untere Teil veranschaulicht die Methodik zur vorbeugenden Fehlervermeidung und Fehlerkompensation durch die Anpassung relevanter Maschinenparameter in Richtung des Materialflusses.

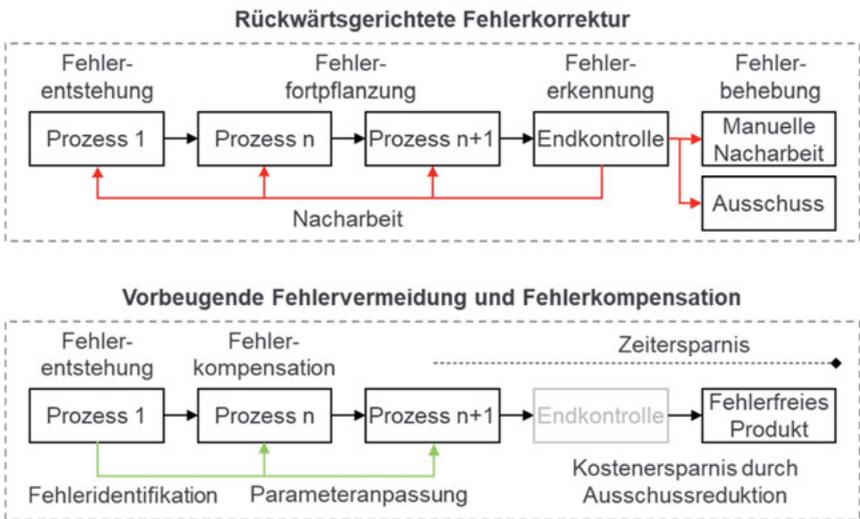


Abbildung 1: Schematische Darstellung eines mehrstufigen Produktionssystems: (oben) klassische Architektur, (unten) Ansatz der Null-Fehler-Fertigung.

6.3 Die Produktion von Radsatzwellen und die auftretende Problematik

Rotationssymmetrische Bauteile werden in einer Vielzahl von Anwendungen in den unterschiedlichsten Branchen eingesetzt. Im Rahmen des EU-Projekts ForZDM stehen zwei industrielle Anwendungsfälle zur Verfügung, um die Ansätze der Null-Fehler-Fertigung anhand rotationssymmetrischer Bauteile zu demonstrieren. Einer dieser Anwendungsfälle bezieht sich auf die Produktion von Turbinenwellen für den Einsatz in der Luft- und Raumfahrtindustrie. Der zweite Anwendungsfall behandelt die Produktion

von Radsatzwellen für den Schienenverkehr. Da diese Komponenten sicherheitsrelevant sind und während der Nutzung starken Belastungen unterliegen, müssen hohe Qualitätsanforderungen erfüllt werden. In diesem Beitrag wird der Herstellungsprozess von rotationssymmetrischen Radsatzwellen verwendet, um die Möglichkeiten von Nullfehler-Fertigungsansätzen in mehrstufigen Produktionssystemen zu veranschaulichen. Der relevante Teil der Fertigung für die vorgestellte Methodik bezieht sich auf die in Abbildung 2 dargestellten Prozessschritte.

Komponenten sicherheitsrelevanten unterliegen, müssen hohe Qualitätsanforderungen erfüllt werden. In diesem Beitrag wird der Herstellungsprozess von rotationssymmetrischen Radsatzwellen verwendet, um die Möglichkeiten von Nullfehler-Fertigungsansätzen in mehrstufigen Produktionssystemen zu veranschaulichen. Der relevante Teil der Fertigung für die vorgestellte Methodik bezieht sich auf die in Abbildung 2 dargestellten Prozessschritte.

Zu Beginn des Herstellungsprozesses wird ein Rohling erhitzt und in mehreren Schiedeprozessen in eine zylindrische Form gebracht. Anschließend werden weitere Wärmebehandlungen durchgeführt, um die durch den Schiedeprozess eingebrachten Eigenspannungen zu reduzieren und die Werkstoffeigenschaften zu verbessern. Danach wird das Bauteil spanend bearbeitet. Dabei werden die Enden des Bauteils abgeschnitten und das Werkstück so auf die gewünschte Länge gebracht. Die Endflächen dienen daraufhin als Referenzebene für den nachfolgenden Drehprozess. Zu diesem Zweck werden in einem vorgelagerten Bearbeitungsschritt (Ausrichten) auf beiden Stirnseiten der Welle Referenzbohrungen eingebracht. Diese Bohrungen helfen bei der Identifizierung der Rotationsachse des Bauteils für den nachfolgenden Drehprozess. Innerhalb des Drehprozesses wird abschließend die gewünschte Kontur der Welle erzeugt.

g erhitzt und in mehreren Schiedeprozessen in eine zylindrische Form gebracht. Anschließend werden weitere Wärmebehandlungen durchgeführt, um die durch den Schiedeprozess eingebrachten Eigenspannungen zu reduzieren und die Werkstoffeigenschaften zu verbessern. Danach wird das Bauteil spanend bearbeitet. Dabei werden die Enden des Bauteils abgeschnitten und das Werkstück so auf die gewünschte Länge gebracht. Die Endflächen dienen daraufhin als Referenzebene für den nachfolgenden Drehprozess. Zu diesem Zweck werden in einem vorgelagerten Bearbeitungsschritt (Ausrichten) auf beiden Stirnseiten der Welle Referenzbohrungen eingebracht. Diese Bohrungen helfen bei der Identifizierung der Rotationsachse des Bauteils für den nachfolgenden Drehprozess. Innerhalb des Drehprozesses wird abschließend die gewünschte Kontur der Welle erzeugt.

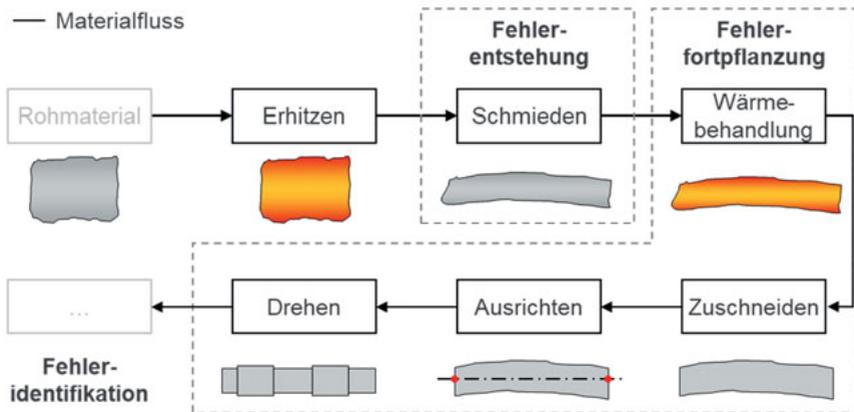


Abbildung 2: Fehlerentstehung in einem mehrstufigen Produktionssystem zur Fertigung rotationssymmetrischer Radsatzwellen.

Durch das Schmieden, die Wärmebehandlung sowie die Lagerung des Teils im heißen Zustand weist die Welle oftmals eine Bogenform auf, wodurch die nachgelagerten spanenden Prozesse beeinträchtigt werden. Die Bogenform der Welle führt dazu, dass

durch das manuelle Festlegen der Rotationsachse während des Ausrichtens, die spätere Kontur der Welle nicht gefertigt werden kann, stimmt den Bereichen der Welle fehlt. Abbildung 3 zeigt die daraus resultierenden Defekte. Dieser Umstand wird oft erst gegen Ende der Produktion erkannt und kann nicht nachbearbeitet werden. Darüber hinaus kommt es zu einem ungleichmäßigen Materialabtrag während des Drehprozesses, der zu einem starken Verschleiß der eingesetzten Werkzeuge, durch die schwankenden Prozesskräfte führt. Da es sich bei den Wellen um sehr große Bauteile handelt (mit einer Länge von bis zu 2400 mm), ist es im bisherigen Zustand für die Maschinenbediener äußerst schwierig, die Referenzbohrungen unter Berücksichtigung der vorliegenden sowie der geplanten Geometrie festzulegen. Infolgedessen werden bisher sehr hohe Toleranzen verwendet, um möglichst viel Material für die Positionierung der Referenzbohrungen zur Verfügung zu haben. Dieses überschüssige Material muss jedoch anschließend wieder entfernt werden. Dieses Vorgehen ist zeitintensiv, verschleißt die Werkzeuge und führt zu unnötigen Materialkosten.

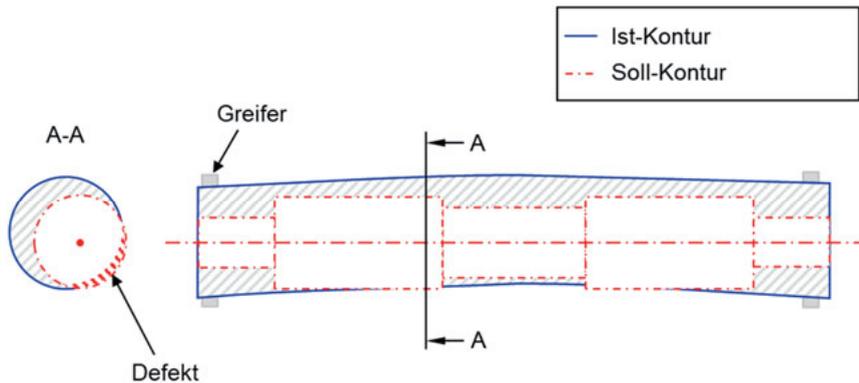


Abbildung 3: Derzeitige Positionierung der Referenzbohrungen und resultierende Problematik in Form von fehlendem Material aufgrund der Bogenform des Ausgangbauteils.

6.4 Smart Centering

Um der beschriebenen Problematik proaktiv entgegenzuwirken und das Ziel einer Null-Fehler-Fertigung zu erreichen, muss die optimale Positionierung der Referenzbohrung basierend auf der tatsächlich vorliegenden Geometrie unter Berücksichtigung der gewünschten Soll-Geometrie ermittelt werden. Das entwickelte System kann dadurch bereits in einem frühen Stadium des Herstellungsprozesses möglichen Ausschuss

identifizieren und diesem entgegenwirken (siehe Abbildung 4). Somit kann vorhergesagt werden, ob ein Bauteil grundsätzlich fehlerfrei herstellbar ist und folglich wertvolle Ressourcen eingespart werden.

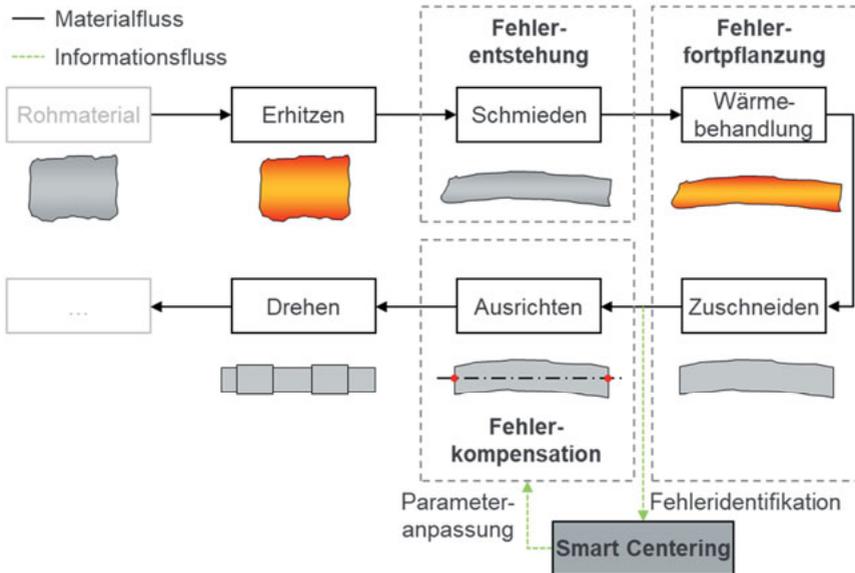


Abbildung 4: Fehlerkompensation in einem mehrstufigen Produktionssystem zur Fertigung rotationssymmetrischer Radsatzwellen.

Abbildung 5 zeigt schematisch die optimierte Positionierung der Referenzbohrung (grün) in Abhängigkeit der Soll-Kontur und der tatsächlichen Bauteilgeometrie (blau).

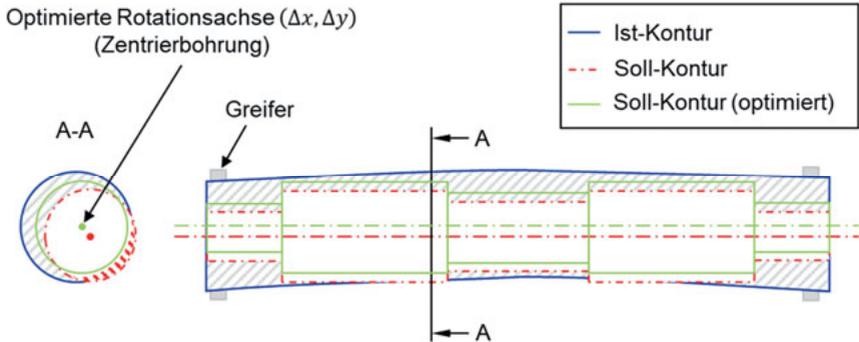


Abbildung 5: Optimierte Positionierung der Referenzbohrung unter Berücksichtigung der Soll-Kontur und der tatsächlichen Bauteilgeometrie (Ist-Kontur).

Um die Eingangsgrößen (Soll-Kontur und Messwerte der Außenkontur) zentralisiert verarbeiten zu können und die Ergebnisse der Optimierung direkt an den nachfolgenden Prozessschritt weiterleiten zu können, ist eine Architektur notwendig, die die vorhandenen Komponenten miteinander verknüpft. Abbildung 6 zeigt das cyber-physische System zur Realisierung des Smart Centering Ansatzes. Zur Messung der Ist-Geometrie wird ein Messsystem eingesetzt. Die Messwerte werden über OPC UA an einen PC übertragen. Die Soll-Geometrie wird aus technischen Zeichnungen und CAD-Modellen abgeleitet und in ein parametrisches Modell in Form einer Tabelle übertragen. Diese bauteilbeschreibenden Tabellen werden auf dem PC gespeichert. Die browserbasierte Anwendung, in welcher die entwickelten Algorithmen ausgeführt werden, liest beide Eingangsgrößen ein und verarbeitet diese. Im Anschluss an die Optimierung, kann das Ergebnis über OPC UA an die CNC-Maschine zur Anbringung der Referenzbohrungen übertragen werden. Zusätzlich werden alle Ergebnisse in einer zentralen Datenbank gespeichert.

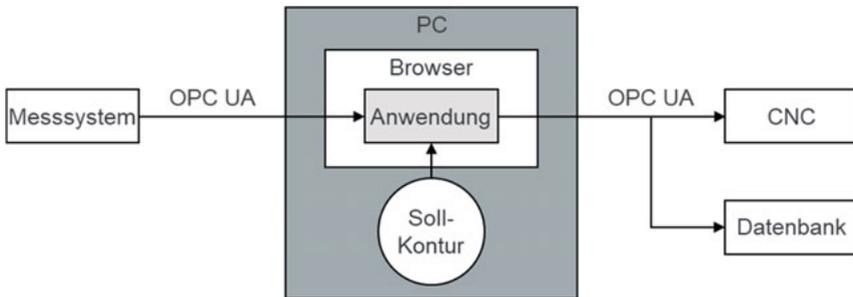


Abbildung 6: Schematische Darstellung des entwickelten cyber-physischen Systems zur Auswertung und Optimierung der Positionierung der Referenzbohrung.

Neben den eigentlichen Messwerten, wird in der Messstation jedes Bauteil mit Hilfe eines individuellen QR-Codes eindeutig identifiziert und diese Information ebenfalls an das System übermittelt. Diese Identifizierung ist notwendig, um spätere Analysen durchführen zu können und Messwerte, welche in anderen Prozessen erfasst wurden, diesem Bauteil zuordnen zu können. Dadurch können unter anderem Korrelation relevanter Parameter prozessübergreifend ermittelt werden. Darüber hinaus wird jedem Bauteil eine weitere Identifikationsnummer zugeordnet, die die geplante Soll-Geometrie für dieses Bauteil definiert. Diese Information wird ebenfalls in der Messstation erfasst und dem System bereitgestellt. Mit Hilfe dieser Nummer kann, die gewünschte Soll-Geometrie aus den zur Verfügung stehenden parametrischen Modellen automatisch ausgewählt werden. Des Weiteren bietet sich die Möglichkeit, im Falle eines nicht zu kompensierenden Defekts, eine andere Soll-Geometrie zu prüfen, die beispielsweise in ihren Abmessungen kleiner ist und stattdessen gefertigt werden könnte. Durch diese Maßnahme wird potenzieller Ausschuss weiter reduziert.

Im nachfolgenden Kapitel wird das zur Erfassung der Ist-Geometrie verwendete Messsystem vorgestellt sowie die zur Modellierung der Kontur eingesetzten Algorithmen erläutert. Kapitel 6.4.2 beinhaltet die Grundlagen zur Beschreibung der Soll-Geometrie in Form eines parametrischen Modells. In Kapitel 6.4.3 wird die Vorgehensweise zur Analyse und Optimierung der Referenzbohrung eingeführt und anschließend in Kapitel 6.4.4 validiert.

6.4.1 Messsystem und Modellierung der Ist-Geometrie

Das Messsystem dient dazu die tatsächliche Geometrie und somit auch die oftmals vorliegende Bogenform der Welle zu erfassen. Abbildung 7 zeigt den schematischen Aufbau des Messsystems. Die Messeinrichtung besteht aus einem Rahmen in Form eines U-Profiles, an dem vier Messköpfe in einem Winkel von 90 Grad montiert sind.

Bei den Messköpfen handelt es sich um laserbasierte Abstandsmesssensoren. Für die Modellierung der Ist-Kontur wird davon ausgegangen, dass die Sensoren exakt ausgerichtet und zentriert sind. Dies kann angenommen werden, da etwaige Abweichungen auf Grund der Positionierung bereits innerhalb des Messsystems erfasst und kompensiert werden. Für die Messung wird das Bauteil auf Stützbacken positioniert und der Rahmen entlang der Längsachse der Welle verfahren. Die Position in Längsrichtung der Messstation wird aus dem Encodersignal des Motors zur Bewegungserzeugung berechnet. Der gewählte Aufbau ermöglicht eine flexible Anzahl von Messungen entlang der Längsachse, um die Kontur des Bauteils zu erfassen. Mit zunehmender Anzahl an Messungen kann die tatsächliche Geometrie des Bauteils entsprechend exakter reproduziert werden. Für die vorliegende Arbeit wurden alle 10 mm Messungen durchgeführt, was bei einer Länge der Welle von 2400 mm in Summe 240 Messungen ergibt. In einem Messsystem muss die Anzahl an Messungen auf Basis der erzielten Ergebnisse bewertet und gegebenenfalls die Auflösung der Messung erhöht oder falls möglich auch reduziert werden.

Die Messwerte werden in einer zentralen Datenbank gesammelt, um zusätzliche Analysen zu ermöglichen. Des Weiteren werden die Messwerte via OPC UA dem beschriebenen cyber-physischen System bereitgestellt. Aufgrund der begrenzten Anzahl von Messpunkten je Messebene ist eine zusätzliche Modellierung erforderlich, um die Ist-Geometrie detailgetreu für die anschließende Analyse und Optimierung darzustellen. Deshalb wurden im Rahmen dieser Arbeit drei verschiedene Ansätze zur Modellierung der Querschnittskontur der Welle entwickelt. Das Ziel der Modellierung ist es, die tatsächliche Kontur des Bauteils so exakt wie möglich abzubilden, um eine möglichst effiziente Optimierung zu ermöglichen sowie die Positionierung innerhalb der Ist-Geometrie sicherzustellen.

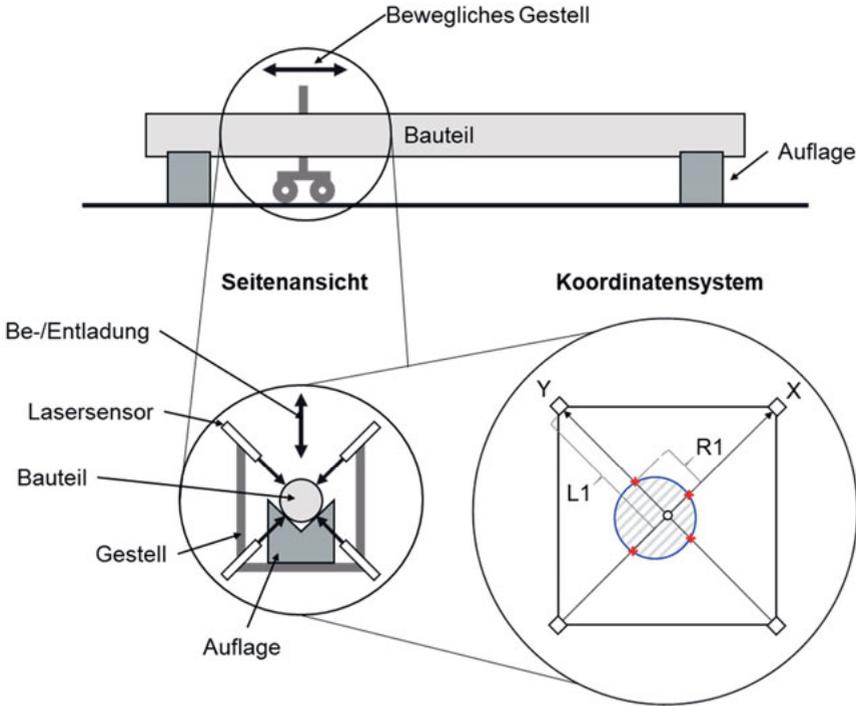


Abbildung 7: Messsystem auf Basis von vier Abstandsmessensoren zur Erfassung der Ist-Geometrie einer Welle.

Der erste Ansatz zur Modellierung der Ist-Kontur auf Grundlage der vier erfassten Messpunkte basiert auf der Annahme, dass der Querschnitt der Welle durch einen Kreis beschrieben werden kann. Unter Verwendung eines Least Squares Algorithmus [7] wird ein Kreis an die Messpunkte angenähert. Das Verfahren minimiert dabei die Summe der Quadrate der geometrischen (euklidischen) Abstände zu einem gegebenen Satz von $n \geq 3$ Punkten in \mathbb{R}^2 , mit $\{(x_i, y_i) \mid 0 \leq i \leq n\}$. Die zu minimierende Zielfunktion \mathcal{F} ist folglich gegeben durch

$$\mathcal{F} = \sum_{i=1}^n d_i^2. \tag{1}$$

Mit der allgemeinen Kreisfunktion, mit dem Mittelpunkt (x_k, y_k) und dem Radius R nach

$$(x - a)^2 + (y - b)^2 = R^2, \tag{2}$$

ergibt sich der geometrische Abstand d_i zu

$$d_i = \sqrt{(x_i - a)^2 + (y_i - b)^2} - R. \tag{3}$$

Dieser Ansatz führt zu einem nichtlinearen Least Squares Problem, das auf Basis von Newton-Verfahren, wie beispielsweise dem Trust-Region-Verfahren oder dem Levenberg-Marquardt-Algorithmus [8] gelöst werden kann. Die Grundlagen und Ansätze zur Lösung nichtlinearer Optimierungsprobleme unter Ein- und zwei Variablen werden hier nicht näher erläutert, können jedoch den entsprechenden Quellen entnommen werden. In der Literatur können darüber hinaus eine Reihe weiterer mathematischer Ansätze [9–13] zur Approximation von Kreisen oder Ellipsen an Datenpunkte nachgelesen werden.

Für den Fall, dass die Querschnittskontur der Welle jedoch keinen exakten Kreis bildet, führt dieser Ansatz dazu, dass die Modellpunkte sowohl innerhalb, als auch außerhalb des berechneten Kreises befinden. Dies führt dazu, dass in der späteren Analyse und Optimierung Bereiche der Ist-Geometrie als vorhanden beziehungsweise nicht vorhanden angenommen werden, obwohl das exakte Gegenteil der Fall ist. Dadurch wird die Genauigkeit, Zuverlässigkeit und Effizienz der Lösung deutlich erniedrigt. Um zu vermeiden, dass Material in Bereichen erwartet wird, in welchen es kein Material gibt, wird bei diesem Ansatz ein Kreis mit dem minimalen Radius, sprich der Distanz von Mittelpunkt des optimierten Kreises zum nächsten liegenden Messpunkt bestimmt. Dadurch wird sichergestellt, dass alle Punkte auf oder außerhalb des neu ermittelten Radius liegen.

Der zweite Ansatz berechnet den geometrischen Schwerpunkt der erfassten Messpunkte. Die arithmetische Mittelwertposition in Form (\bar{x}, \bar{y}) wird berechnet durch

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i \right). \tag{4}$$

Der damit berechnete Mittelpunkt wird anschließend verwendet, um ebenfalls einen Kreis mit minimalem Radius aufzuspannen und ähnelt daher dem ersten Ansatz, ist jedoch weniger rechenintensiv. Mit diesem Verfahren kann der geometrische Mittelpunkt unabhängig von der Anzahl der Messpunkte bestimmt werden. Während der erste Ansatz mindestens drei Messpunkte benötigt, um einen Kreis einzupassen. Dennoch werden auch hier Bereiche in denen, auf Grund der Messpunkte zwar mit Gewissheit Material vorhanden ist, nicht berücksichtigt.

Der dritte Ansatz ist insbesondere für den Fall geeignet, wenn die Querschnittskontur stark von einer Kreisform abweicht. Bei diesem Ansatz wird eine kubische Interpolation verwendet, um die Messpunkte miteinander zu verbinden. Durch den Einsatz der Interpolation und der zusätzlichen Bedingungen wird sichergestellt, dass alle Messpunkte, auf der modellierten Kontur liegen. Darüber hinaus können damit auch Formen angenähert werden, die keiner symmetrischen Form entsprechen und somit auch starke Verformungen berücksichtigt werden. Grundsätzlich können verschiedene Interpolationsverfahren zur Modellierung der Kontur zwischen den Datenpunkten eingesetzt werden. Dazu zählen beispielsweise die lineare oder kubische Interpolation [14–17]. Im Folgenden wird eine kubische Interpolation, in Form eines Polynoms dritter Ordnung, verwendet [18]. Das Polynom hat die Bedingung, dass es zweimal kontinuierlich differenzierbar ist. Die kubische Interpolation wird aufgrund ihrer Eigenschaft verwendet, „glatte“ Ergebnisse zu erzielen [19]. Um eine geschlossene Modellierung und die Stetigkeit des Polynoms, unter Verwendung der kubischen Interpolation zu erreichen, werden die Messpunkte von kartesischen Koordinaten in Polarkoordinaten transformiert. Nach der Interpolation wird der Mittelpunkt ausgehend von den berechneten Stützpunkten mit Gl. (4) bestimmt.

Abbildung 8 veranschaulicht die berechneten Konturen des Querschnitts der Welle auf Basis der drei vorgestellten Modellierungsansätze. Es zeigt sich, dass sich die jeweiligen Mittelpunkte der eingeführten Methoden leicht unterscheiden. Bei den ersten beiden Ansätzen unter Verwendung des Minimalkreises ist zu erkennen, dass verfügbares Material nicht berücksichtigt wird. Für den, mit Hilfe des Least Squares-Algorithmus ermittelten Kreis werden hingegen Bereiche miteinbezogen, in denen kein Material vorhanden ist. Die vorgestellte kubische Interpolation hat diesen Nachteil nicht, insbesondere bei einer elliptischen Form des Querschnitts, kann diese exakter angenähert werden, ohne dabei gesicherte Materialbereiche zu vernachlässigen. Es ist daher zu erwarten, dass die Kontur der Welle durch die Interpolation am besten beschrieben werden kann. Dadurch entsteht mehr Freiraum für die Positionierung des Sollteils innerhalb der Ist-Geometrie. Diese Annahmen müssen noch während des tatsächlichen Betriebs auf Basis mehrerer Auswertungen validiert werden.

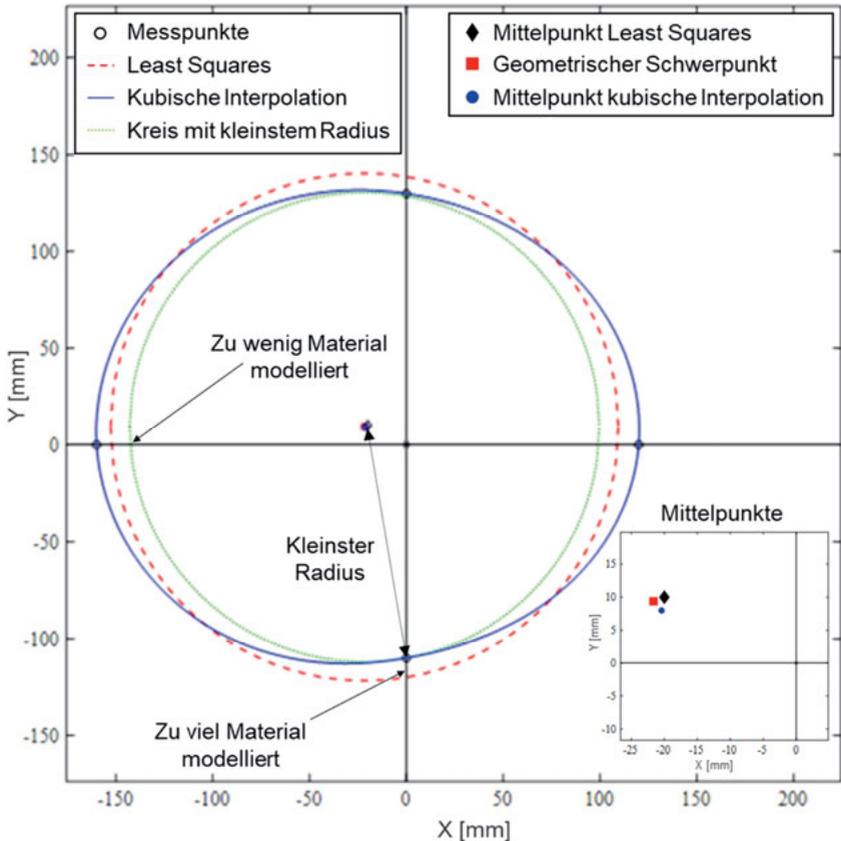


Abbildung 8: Modellierungsansätze zur Abbildung der Ist-Kontur auf Basis von vier Messpunkten.

6.4.2 Parametrisches Modell der Soll-Geometrie

Das parametrische Modell dient dazu die Soll-Geometrie der Welle abzubilden und den nachfolgenden Analyse- und Optimierungsmethoden zur Verfügung zu stellen. Das parametrische Modell wird auf Basis verfügbarer technischer Zeichnungen und CAD-Modelle erstellt. Dazu wurde eine Struktur entwickelt und in einer Tabelle abgefragt, welche die für die spätere Auswertung relevanten Informationen enthält. Die Struktur zur Erstellung des parametrischen Modells ist in Tabelle 6-1 anhand von Beispieldaten dargestellt. In den ersten beiden Spalten wird der Bereich auf der Längsachse durch einen Start- und Endpunkt definiert. In den nächsten Spalten wird der Start- und Endradius im dazugehörigen Bereich festgelegt. Zwischen Start- und Endradius

wird bei der Verarbeitung der Daten linear interpoliert. Die nächsten beiden Spalten enthalten die obere und untere Toleranzgrenze für die angegebenen Radien.

Tabelle 6-1: Struktur zur Abbildung der Soll-Geometrie eines rotationssymmetrischen Bauteils.

Z_Start	Z_End	R_Start	R_End	Upper_TOL	Lower_TOL
0	100	76	76	0,5	-0,5
100	200	76	89	0,5	-0,5
⋮	⋮	⋮	⋮	⋮	⋮

Da es sich bei den Wellen um rotationssymmetrische Bauteil handelt, wird die beschriebene Kontur um die Mittelachse (*Radius* = 0) rotiert, um die vollständige Geometrie abzubilden. Neben der erwähnten linearen Interpolation ist es denkbar, weitere Interpolationsarten, wie beispielsweise eine kubische Interpolation zu implementieren, um Radien oder andere Übergänge abbilden zu können. Dazu ist es jedoch notwendig, weitere Parameter zu erfassen, welche beispielsweise den gewünschten Radius definieren würden.

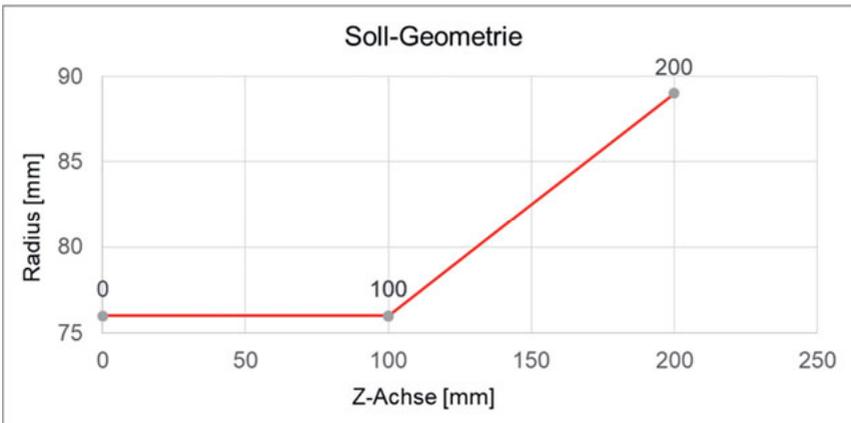


Abbildung 9: Nomenklatur für die Gleichungen der Optimierung.

6.4.3 Analyse und Optimierung der Referenzbohrung

Nachdem sowohl die Ist-Geometrie als auch die Soll-Geometrie der zu fertigenden Welle erfasst sind, können diese nun eingelesen, verarbeitet und mit einander verglichen werden. Dazu wird zu Beginn des Vorgangs die Soll-Geometrie in Bezug zum Koordinatensystem der Messstation positioniert. Anschließend werden die den Messebenen entsprechenden Soll-Werte ausgewertet und überprüft, ob diese außerhalb der Ist-Kontur des jeweiligen Querschnitts liegen. Ist dies der Fall, kann davon ausgegangen werden, dass ein späterer Defekt in Form von fehlendem Material entstehen wird.

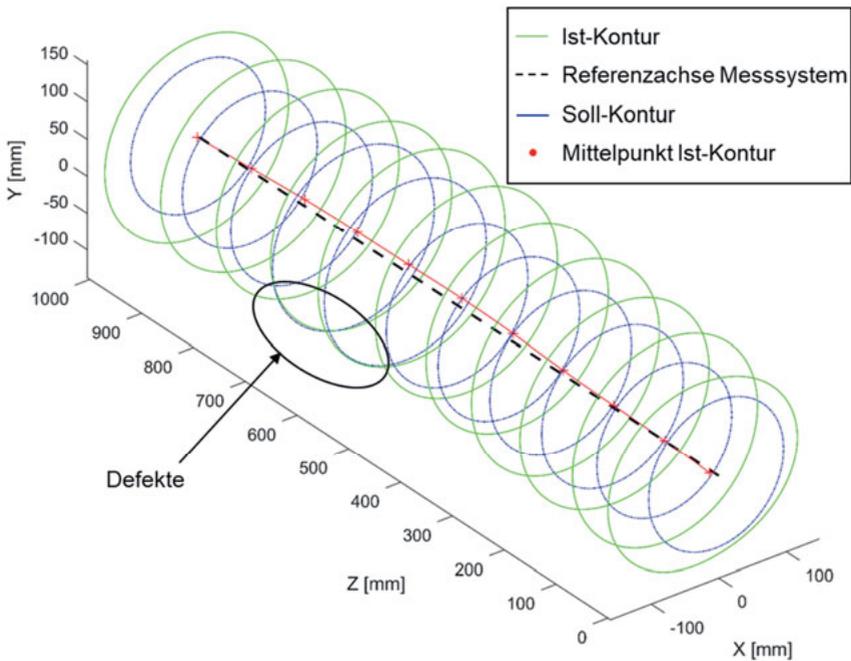


Abbildung 10: Initiale Positionierung und Auswertung der Ist-Kontur bezüglich der Soll-Kontur.

Diese Tatsache kann durch die Ausrichtung der Sollgehörigen Rotationsachse, welche durch die Position wird, kompensiert werden. Da das Bauteil bereits an verbleiben für die Optimierung und Ausrichtung der Freiheitsgrade in X- und Y-Richtung. Das bedeutet, parallel zur Referenzachse des Messsystems verschob

Geometrie und somit der dazuer Referenzbohrung bestimmt nähernd seine finale Länge hat, Rotationsachse ausschließlich dass die Achse der Welle stets en wird. Würde man das Soll-

Bauteil ebenfalls um die X- oder Y-Achse rotieren, würde die Länge des Ist-Bauteils nicht ausreichen. Folglich muss die Verschiebung $(\Delta x, \Delta y)$ unter Berücksichtigung aller Messebenen optimiert werden. Dazu wurden drei Zielfunktionen entwickelt, welche im Nachfolgenden vorgestellt werden. Abbildung 11 zeigt die dargestellten Gleichungen.

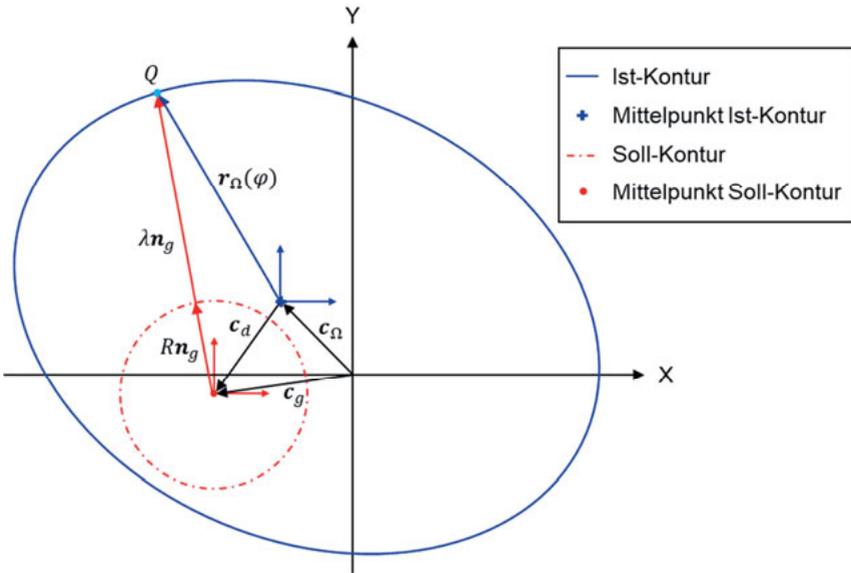


Abbildung 11: Nomenklatur für die Gleichungen der Optimierung.

Betrachten wir einen Messabschnitt, lässt sich wie in Abbildung 11 dargestellt, die Vektorschleife für einen Punkt Q auf der modellierten Ist-Kontur Ω einer Ebene ableiten zu

$$c_g + Rn_g + \lambda n_g = c_\Omega + r_\Omega. \tag{5}$$

In der Annahme dass $\|n_g\|_2 = 1$ ist, können wir Gl. (5) umgestellt nach λ schreiben als

$$\lambda = \|c_\Omega + r_\Omega - c_g\| - R \tag{6}$$

und die Gleichung somit lösen. Wobei diese Gleichung für jeden Punkt der Kontur berechnet werden muss. Alle drei Optimierungsprobleme sind so formuliert, dass der Mindestabstand $\lambda_{\min,k}$ zwischen der Soll- und Ist-Kontur des jeweiligen Messabschnitts $k \in \mathbb{N} (k = 1, \dots, m)$ größer oder gleich Null sein muss. Daraus folgt

$$\lambda_{\min,k} = \min_{r_{\Omega,k}} (\|c_{\Omega,k} + r_{\Omega,k} - c_g\|) - R \geq 0. \quad (7)$$

Wir können die Zielfunktion unseres ersten Optimierungsproblems so beschreiben, dass der Abstand zwischen dem geometrischen Mittelpunkt der Ist-Kontur und dem Mittelpunkt der Soll-Kontur minimiert wird:

$$\min_{c_g} \left(\sqrt{\sum_{k=1}^m \|c_{d,k} = c_g - c_{\Omega,k}\|^2} \right). \quad (8)$$

Dieser Ansatz führt dazu, dass die Mittelachse der Soll-Geometrie so nah wie möglich an die Mittelpunkte der Ist-Kontur gelegt wird, sprich die resultierende Unwucht wird minimiert.

Die zweite Zielfunktion maximiert den kleinsten Abstand $\lambda_{\min,k}$ aller Messabschnitte m nach

$$\min_{c_g} \left(- \min_{k=1 \dots m} (\lambda_{\min,k}) \right) \quad (9)$$

und erhöht dadurch die zu erwartende Prozesssicherheit.

Die dritte Zielfunktion minimiert die Differenz zwischen dem kleinsten und größten Abstand der Soll- und Ist-Kontur aller Messebenen nach

$$\min_{c_g} \left(\sqrt{\sum_{k=1}^m (\lambda_{\max,k} - \lambda_{\min,k})^2} \right), \quad (10)$$

wobei $\lambda_{\max,k}$ angelehnt an Gl. (7) berechnet wird. Ziel dieser Optimierung ist ein möglichst homogener Materialabtrag. Heißt, dass das abzutragende Material in jedem Abschnitt möglichst gleichmäßig verteilt ist.

Die nichtlinearen multivariablen Optimierungsprobleme können unter Zuhilfenahme von Methoden wie dem Innere-Punkte-Verfahren (engl. Interior-point method) oder dem Trust-Region-Verfahren [20–22] gelöst werden.

6.4.4 Validierung der Optimierungsergebnisse

Die Auswertung der Optimierungsergebnisse erfolgt durch die Überprüfung der jeweiligen, beeinflussten Parameter. Dabei zeigt sich, dass die erwarteten Ziele erreicht werden (siehe Tabelle 6-2).

Tabelle 6-2: Ergebnisse der Optimierungsmethoden anhand eines Versuchsaufbaus.

Parameter

		Minimaler Abstand	Unwucht	Differenz des minimalen und maximalen Abstands
Optimierung	Hohe Sicherheit	6,182 mm	458 kg * mm	63,443 mm
	Geringe Unwucht	5,074 mm	13 kg * mm	67,967 mm
	Homogene Bearbeitung	5,975 mm	365 kg * mm	16,647 mm

6.5 Browserbasierte Anwendung zur automatisierten Korrektur der Referenzbohrung

Die in Abbildung 12 dargestellte Oberfläche der in browserbasierten Anwendung zeigt die Ergebnisse eines Analyse- und Optimierungsdurchgangs. Die für die Modellierung und Optimierung verwendeten Verfahren, können durch den Anwender selbstständig festgelegt werden. Nach der Optimierung wird sowohl die Soll- (blau) als auch Ist-Kontur (grün) der Welle visualisiert. In der Visualisierung werden darüber hinaus die Ebenen hervorgehoben, in welchen ein Defekt zu erwarten ist. Jede Ebene kann zusätzlich in einer weiteren Darstellung im Detail betrachtet werden. Neben den Visualisierungen werden eine Reihe weiterer Werte erhoben (siehe Abbildung 12). Auf Basis dieser Informationen kann der Maschinenbediener abschätzen, inwieweit eine Fertigung möglich ist und welche Randbedingungen während des Drehprozesses auftreten werden (beispielsweise Unwucht). Darüber hinaus können die Ergebnisse dazu verwendet werden, vorhergehende Prozesse zu verbessern. Durch die Optimierung können so beispielsweise bisher notwendigen Toleranzen reduziert und folglich Kosten, durch Verringerung des Materialeinsatzes und der Bearbeitungsdauer, eingespart werden.

Python/Django entwickelten browserbasierten Analyse- und Optimierungsdurchgangs, können durch den Anwender selbstständig festgelegt werden. Nach der Optimierung wird sowohl die Soll- (blau) als auch Ist-Kontur (grün) der Welle visualisiert. In der Visualisierung werden darüber hinaus die Ebenen hervorgehoben, in welchen ein Defekt zu erwarten ist. Jede Ebene kann zusätzlich in einer weiteren Darstellung im Detail betrachtet werden. Neben den Visualisierungen werden eine Reihe weiterer Werte erhoben (siehe Abbildung 12). Auf Basis dieser Informationen kann der Maschinenbediener abschätzen, inwieweit eine Fertigung möglich ist und welche Randbedingungen während des Drehprozesses auftreten werden (beispielsweise Unwucht). Darüber hinaus können die Ergebnisse dazu verwendet werden, vorhergehende Prozesse zu verbessern. Durch die Optimierung können so beispielsweise bisher notwendigen Toleranzen reduziert und folglich Kosten, durch Verringerung des Materialeinsatzes und der Bearbeitungsdauer, eingespart werden.

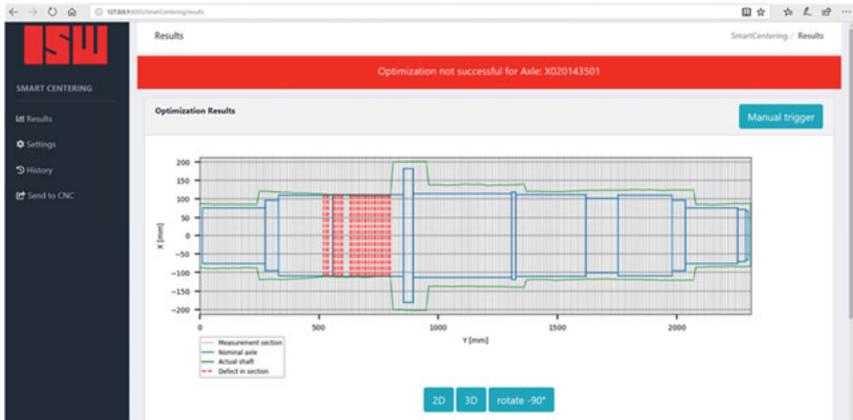


Abbildung 12: Visualisierung der Ebenen in denen ein Defekt zu erwarten ist.

Wie bereits in Kapitel 6.4 erläutert wird in Abbildung 13 die, für den Fall, dass eine ursprünglich geplante Soll-Geometrie nicht fertigbar ist, möglichen Alternativen aufgelistet. Die optimierte Position der Referenzbohrung kann für jede erfolgreich getestete Welle an die CNC-Maschine via OPC UA gesendet werden. Zusätzlich muss auf der CNC-Maschine das, der Soll-Geometrie entsprechende NC-Programm geladen werden.

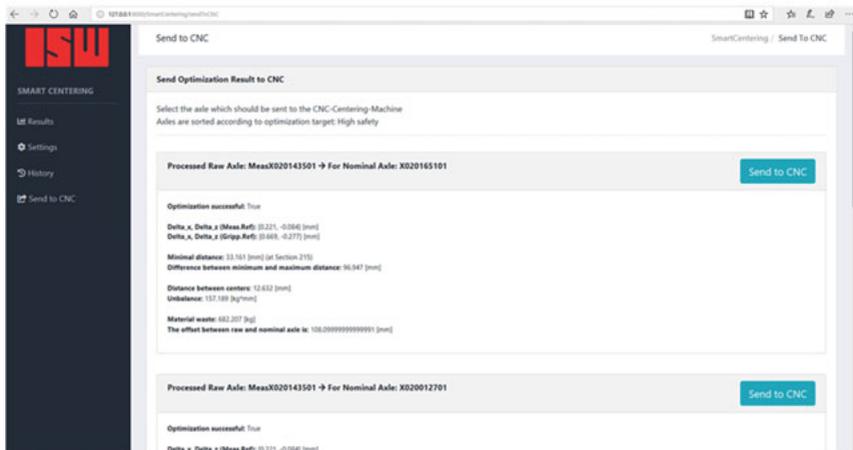


Abbildung 13: Auswahl der möglichen fertigen Wellen für den Fall, dass die ursprünglich geplante Geometrie nicht ohne Defekte herstellbar ist.

6.6 Zusammenfassung und Ausblick

Die vorgestellte Smart Centering Strategie zielt darauf ab, eine wirtschaftlichere Fertigung von rotationssymmetrischen Bauteilen durch eine intelligente Ausrichtung der Rotationsachse zu erreichen. Dazu wurden Ansätze der Null-Fehler-Fertigung in einem industriellen Anwendungsfall realisiert und in Form eines cyber-physischen Systems in die Prozesskette integriert. Um Fehler, bedingt durch Verformungen des Ausgangsmaterials zu kompensieren, werden sowohl die Soll-Geometrie als auch die tatsächliche Geometrie des Bauteils berücksichtigt. Für einen Vergleich beider Geometrien wurden verschiedene Modellierungsmethoden vorgestellt und diskutiert. Darauf aufbauend wurden Algorithmen entwickelt, welche die Position der Referenzbohrung hinsichtlich verschiedener Kriterien optimieren. Darüber hinaus werden für den Fall, dass eine Welle voraussichtlich nicht gefertigt werden kann, weitere Soll-Geometrien überprüft und so alternative Vorschläge zur Fertigung gemacht. Das System berechnet eine Vielzahl von Parametern, wie Unwucht und minimaler Abstand der Soll-Geometrie zur Ist-Geometrie und kann dadurch zur weiteren Verbesserung der gesamten Prozesskette verwendet werden. Folglich können auf Basis der Auswertung dieser Daten, zukünftig Toleranzen gesenkt und dadurch Material eingespart und der Werkzeugverschleiß sowie die benötigte Bearbeitungsdauer reduziert werden.

In zukünftigen Arbeiten müssen die entwickelten Algorithmen im realen Prozess validiert werden. Dabei liegt besonderes Augenmerk auf der Genauigkeit der vorgestellten Modellierungsansätze und der Effizienz der verschiedenen Optimierungsfunktionen.

6.7 Danksagung

Dieses Projekt wird im Rahmen des Forschungs- und Innovationsprogramms Horizon 2020 der Europäischen Union im Rahmen der Finanzhilfsvereinbarung Nr. 723698 gefördert. Dieses Dokument gibt nur die Meinung des Autors wieder und die Kommission ist nicht verantwortlich für die Verwendung der darin enthaltenen Informationen.

6.8 Literatur

- [1] M. Colledani et al., “Design and management of manufacturing systems for production quality,” *CIRP Annals - Manufacturing Technology*, vol. 63, no. 2, pp. 773–796, 2014.
- [2] F. Eger et al., “Zero Defect Manufacturing Strategies for Reduction of Scrap and Inspection Effort in Multi-stage Production Systems,” *Procedia CIRP*, vol. 67, pp. 368–373, 2018.
- [3] F. Eger, C. Reiff, B. Brantl, M. Colledani, and A. Verl, “Correlation analysis methods in multi-stage production systems for reaching zero-defect manufacturing,” *Procedia CIRP*, vol. 72, pp. 635–640, 2018.
- [4] K.-S. Wang, “Towards zero-defect manufacturing (ZDM)—a data mining approach,” *Adv. Manuf.*, vol. 1, no. 1, pp. 62–74, 2013.
- [5] F. Eger, C. Reiff, M. Colledani, and A. Verl, “Knowledge Capturing Platform in Multi-Stage Production Systems for Zero-Defect Manufacturing: Submitted to 25th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), 2018.
- [6] R. Isermann, *Fault-Diagnosis Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.
- [7] M. Zatarain, A. Mendikute, and I. Inziarte, “Raw part characterisation and automated alignment by means of a photogrammetric approach,” *CIRP Annals*, vol. 61, no. 1, pp. 383–386, 2012.
- [8] N. Chernov and C. Lesort, “Least Squares Fitting of Circles,” *Journal of Mathematical Imaging and Vision*, vol. 23, no. 3, pp. 239–252, 2005.
- [9] D. C. Sorensen, “Newton’s Method with a Model Trust Region Modification,” *SIAM J. Numer. Anal.*, vol. 19, no. 2, pp. 409–426, 1982.
- [10] L. D. Cooper, “Circle fitting by linear and non-linear least squares,” *Journal of Optimization Theory and Application*, vol. 69, 1992.
- [11] V. Pratt, “Direct Least-Squares Fitting of Algebraic Surfaces,” *Computer Graphics*, vol. 21, no. 4, pp. 145–152, 1987.
- [12] A. Al-Sharadqah and N. Chernov, “Error analysis for circle fitting algorithms,” *Electron. J. Statist.*, vol. 3, no. 0, pp. 886–911, 2009.
- [13] D. Umbach and K. Jones N., “A Few Methods for Fitting Circles to Data,” *Transactions on Instrumentation and Measurement*, 2000.
- [14] W. Gander, G. H. Golub, and R. Strebel, “Least-squares fitting of circles and ellipses,” *BIT*, vol. 34, no. 4, pp. 558–578, 1994.
- [15] E. Maeland, “On the comparison of interpolation methods,” *IEEE transactions on medical imaging*, vol. 7, no. 3, pp. 213–217, 1988.
- [16] T. B. Sprague, “Shape Preserving Piecewise Cubic Interpolation,” 1990.

- [17] H. S. Hou and H. C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 6, 1978.
- [18] F. N. Fritsch and R. E. Carlson, "Monotone Piecewise Cubic Interpolation," *Society for Industrial and Applied Mathematics*, vol. 17, no. 2, 1980.
- [19] C. H. Lee, "Restoring Spline Interpolation of CT Images," *IEEE transactions on medical imaging*, vol. 2, no. 3, pp. 142–149, 1983.
- [20] W. A. Halang, R. Langlais, and E. Kugler, "Cubic spline interpolation for the calculation of retention indices in temperature-programmed gas-liquid chromatography," *Anal. Chem.*, vol. 50, no. 13, pp. 1829–1832, 2002.
- [21] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An Interior Algorithm for Nonlinear Optimization That Combines Line Search and Trust Region Steps," 2004.
- [22] T. F. Coleman and Y. Li, "An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds," *Society for Industrial and Applied Mathematics*, vol. 6, no. 2, pp. 418–445, 1996.

7 Ein Konzept zur In-Betrieb-Rekonfiguration von Seilrobotern

F. Trautwein, A. Verl, A. Pott

7.1 Einleitung und Literaturübersicht

Bei parallelen Seilroboter erfolgt die Steuerung der Plattformbewegung über das Verstellen der Seillängen mittels Winden. Die Seile werden strukturell parallel von den Antriebsmotoren über Winden und Umlenkrollen zu der Plattform geführt. Seilroboter gehören somit zur Klasse der parallelen Roboter. Allgemeine Beispiele für Seilroboter sind der Nist-Robocrane [1], Falcon [2], CoGiRo [3] sowie die Familie der IPAnema-Roboter vom Fraunhofer Institut für Produktion und Automatisierung (IPA) [4]. Gegenüber herkömmlichen parallelen Roboter wie einer Gough-Steward-Plattform [5] haben Seilroboter den Vorteil, dass die Seile als Antriebselemente sehr leicht und flexibel sind. Die bewegten Massen sind somit deutlich geringer, was dynamischere Bewegungen im Betrieb ermöglicht. Durch die Flexibilität können Seile auf Winden aufgespult werden, was eine einfache und große Längenänderung der Antriebselemente ermöglicht. Bedingt durch die genannten Eigenschaften kann bei Seilrobotern der Arbeitsraum sehr groß und flexibel ausgeführt werden. Verschiedene Anwendungen wo die großen Arbeitsräume ausgenutzt werden sind beispielsweise das Drucken von Häuserfassaden aus Beton [6] oder Wartungsarbeiten an Flugzeugen [7, 8]. Pott et al. präsentieren in [9] die Anwendung eines hängenden Seilroboters für Ausstellungszwecke in einem Expo-Pavillon.

Für die Funktionsbeschreibung eines Seilroboters ist die Formulierung eines nutzbaren Arbeitsraums ein wichtiger Teil. Der Grundgedanke für die Beschreibung eines Arbeitsraums ist immer sämtliche Posen zusammenzufassen, die ein entsprechendes Kriterium erfüllen. Für das Konzept des „Wrench-Closure-Workspace“ (WCW) ist das Kriterium, dass sich eine beliebige externe Last mit strikt positiven Seilkräften balancieren lässt [10]. Da prinzipiell beliebig große Lasten zugelassen sind, ist das Konzept des WCW eine rein geometrische Größe. Gouttefarde zeigt in [11], dass die Grenzen des WCW durch Polynome beschrieben werden können, die stückweise zusammengesetzt werden. Bei ebenen Systemen besteht die Hülle aus quadratischen Funktionen, bei räumlichen Systemen aus kubischen Polynomflächen. Das beschriebene Konzept eignet sich nur sehr bedingt für die praktische Anwendung, da zulässige Seilkräfte immer durch die ertragbare Maximalbelastung limitiert sind. Ein Konzept, das die Seilkräfte auf ein definiertes Intervall beschränkt ist der „Wrench-Feasible-Workspace“ (WFW). Bei diesem Konzept wird eine äußere Last festgelegt und alle Posen sind Teil des Arbeitsraums, die eine Abstützung mit Seilkräften aus dem spezifizierten Intervall gewährleisten. Merlet zeigt in [12], dass die Grenzen des WFW für hängende Seilroboter ebenfalls durch Polynome beschrieben werden können. Jedoch bestehen

die Hüllensegmente bei ebenen Systemen aus Polynome n 6-ten Grades und für räumliche Systeme aus Polynomen 8-ten Grades. Bei beiden beschriebenen Konzepten müssen für die geschlossene Darstellung des Arbeitsraums die Schnittpunkte, bzw. entsprechend zusammengesetzt Schnittkurven der einzelnen Polynome gefunden und eine effiziente numerische Annäherung der geschlossenen Arbeitsraumhülle ermöglicht werden. Pott stellt in [13, 14] zwei Verfahren vor, die einen Arbeitsraum für unterschiedlichste Anwendungen zu formulieren. Ansätze für die Berücksichtigung von dynamischen Eigenschaften werden beispielsweise von Barrette et. al. [15] und Gagliardini et. al. [16] vorgestellt.

Eine weitere wichtige Eigenschaft eines Seilroboters ist die mechanische Steifigkeit an einer Pose. Die Steifigkeit ist ein Maß für den Zusammenhang zwischen einer Plattenformverschiebung und der daraus resultierenden Reaktionskraft in den Seilen [17, 18]. Die gesamte Steifigkeit eines Seilroboters setzt sich hauptsächlich aus zwei Anteilen zusammen. Der erste Anteil wird maßgeblich durch die materielle Steifigkeit der Seile beeinflusst. Der zweite Anteil ist von der Geometrie abhängig und wird hauptsächlich durch die Wirkrichtung der Seilkraft bestimmt. Über eine Eigenwertanalyse des richtungsabhängigen Anteils können Singularitäten innerhalb des Arbeitsraums identifiziert werden [19].

Der Arbeitsraum und die Steifigkeit eines Seilroboters sind Teil der Roboteranalyse und beschreiben das Verhalten des Systems hinsichtlich eines Kriteriums. Das Gegenstück dazu ist die Auslegung oder Synthese des Roboters, wobei eine Anforderung definiert wird und eine passende Robotergeometrie gesucht wird. Merlet unterteilt in [20] den Prozess der Robotersynthese in zwei Phasen. Bei der strukturellen Synthese werden die grundlegendsten Eigenschaften wie Bewegungsprofil, Anzahl der Seile oder der Roboterklasse festgelegt. In der Phase der Parametersynthese werden dann die geometrischen Parameter des Roboters abgestimmt. Von Hao et. al. [21], Merlet [22] und Bruckmann [23] wird das Problem des Roboterentwurfs mit Techniken der Intervallarithmetik gelöst. Von Ottaviano und Ceccarelli wird in [24] der Entwurf als Optimierungsproblem aufgefasst und numerisch gelöst. In diesem Beitrag wird ein Konzept zur Vorgehensweise für die Rekonfiguration eines Seilroboters vorgestellt. Dabei bedeutet Rekonfiguration die Struktur des Seilroboters an eine wechselnde Anforderung anzupassen. Die Rekonfiguration eines Roboter ist somit eng mit der Synthese verwandt. Da bei einer Rekonfiguration der Roboter jedoch bereits besteht ist die Menge an freien Parametern für gewöhnlich geringer. Gagliardini beschreibt in [25, 26] einen Ansatz für die Rekonfiguration von Seilrobotern. Dabei werden für die Variationsmöglichkeiten des Roboters diskrete Mengen angenommen. Die diskreten Zustände werden zu einem Zustandsgraph zusammengefasst. Anschließend wird die Strategie identifiziert, indem ein optimaler Pfad durch den Zustandsgraph gesucht wird.

Der folgende Beitrag gliedert sich wie folgt. Im ersten Kapitel werden die gängigen Klassifizierungen sowie die Grundlagen der theoretischen Roboteranalyse vorgestellt. Im zweiten Kapitel wird der Begriff der Rekonfiguration sowie die Motive und Ziele genauer eingeführt. Aus den Zielen einer Rekonfiguration wird in Kapitel drei ein Ansatz für die Rekonfiguration abgeleitet. In Kapitel vier wird mit einem generischen Datenmodell, eine zentrale Größe der Rekonfiguration eingeführt. Im letzten Kapitel werden die Inhalte zusammengefasst und einen Ausblick auf folgende Arbeiten gegeben.

7.2 Theoretische Grundlagen der Seilroboter

7.2.1 Klassifizierung von Seilrobotern

Wie beschrieben werden bei Seilrobotern die Antriebsstränge strukturell parallel angeordnet. Die Anzahl der verwendeten Antriebsstränge hat einen zentralen Einfluss auf die Eigenschaften des Gesamtsystems. Ein Maß für das Verhältnis zwischen den Seilen als kinematische Bindungen und den Freiheitsgraden ist der Redundanzgrad

$$r = m - n. \tag{2}$$

Wobei die Anzahl der Seile mit m und die Anzahl der möglichen Bewegungsrichtungen mit n bezeichnet wird. In Abbildung sind für die betrachteten Klassen schematische Beispiele dargestellt.

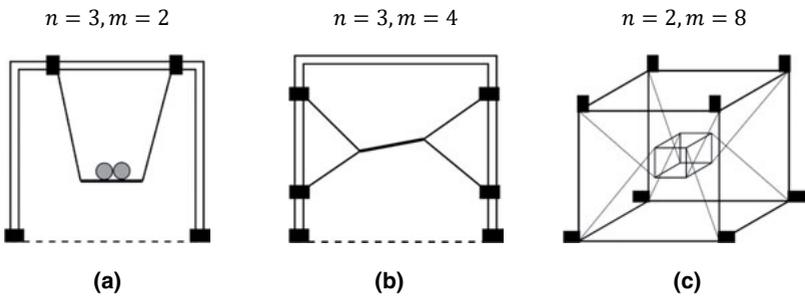


Abbildung 1: (a) Unterbestimmter Seilroboter (IRPM) ; (b): Voll-bestimmter Seilroboter (CRPM); (c): Redundant verspannter Seilroboter (RRPM)

Nach Ming und Higuchi [27] lassen sich Seilroboter in die folgenden vier Kategorien unterteilen.

- $m < n \leq 6$: Der Seilroboter verfügt über weniger Seile als Freiheitsgrade. Da weniger Seile als Bewegungsfreiheitsgrade installiert sind, können nicht alle Bewegungsrichtungen kontrolliert werden. Systeme der Klasse werden üblicherweise als „incompletely retrained positioning mechanism (IRPM)“ bezeichnet.

- $n = m$: Der Seilroboter ist kinematisch voll-bestimmt. Da Seile einseitige mechanische Bindungen darstellen, sind die abstützbaren externen Lasten begrenzt. Seilroboter mit Redundanzgrad $r = 0$ werden üblicherweise zur Klasse der unterbestimmten Systeme gezählt.
- $n + 1 = m$: Wird der Seilroboter mit einem zusätzlichen Seil verspannt, werden die abstützbaren Lasten ausschließlich durch Seilkraftgrenzen bestimmt und alle Bewegungsrichtungen kontrolliert werden. Die Klasse wird mit „completely restraint position mechanism (CRPM)“ bezeichnet.
- $n + 1 < m$. Die Arbeitsplattform des Seilroboters ist redundant verspannt. Durch die Redundanz ist die Kraftverteilung zwischen den Seilen nicht mehr eindeutig. Die Klasse der redundanten Seilroboter wird mit „redundantly restraint position mechanism (RRPM)“ bezeichnet.

Neben dem Grad der Redundanz ist es möglich einen Seilroboter über die möglichen Bewegungsrichtungen zu beschreiben. Das Bewegungsmuster wird von der Anwendung bestimmt und wird während des Entwurfs umgesetzt. In der folgenden Auflistung sind die wichtigsten Varianten zusammengefasst.

- 1T: Eindimensionale Translation
- 2T: Translation in der Ebene
- 3T: Räumliche Translation
- 1R2T: Ebene Starrkörperbewegung
- 2R3T: Räumliche Bewegung mit Rotation um zwei Achsen
- 3R3T: Räumliche Starrkörperbewegung

In Abbildung 2 sind zwei verschiedene Bewegungsprofile eines Seilroboters abgebildet. In Abbildung 2-(a) besteht die Roboterplattform aus einer Punktmasse, die über drei translatorische Freiheitsgrade verfügt. In Abbildung 2-(b) ist ein räumlicher Roboter dargestellt, der mit acht Antriebsseilen verspannt ist und sechs Bewegungsfreiheitsgrade besitzt.

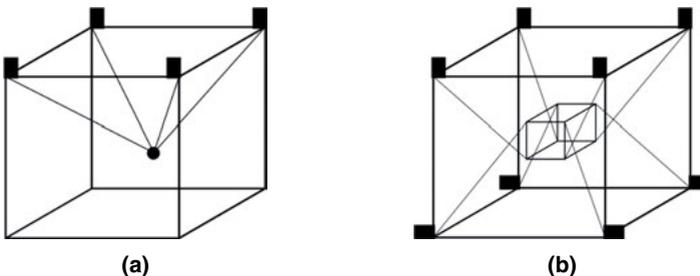


Abbildung 2: (a) 3T-Bewegungsmuster; (b) 3R3T-Bewegungsmuster

Im folgenden Unterpunkt werden die gängigen Definitionen der theoretischen Betrachtungsweise von Seilrobotern eingeführt.

7.2.2 Theoretische Grundlagen

Wie bereits beschrieben, erfolgt die Steuerung der Bewegung über das Verstellen der insgesamt i Antriebsseile. Da Seile nur einseitige mechanische Bindungen darstellen, werden für die volle Kontrollierbarkeit lt. Kawamura [2] mindestens 1 Seile benötigt. Für den allgemeinsten Fall eines räumlichen Seilroboters mit 6 Freiheitsgraden wird der Seilroboter üblicherweise mit $m = 8$ Seilen verspannt. In Abbildung 3 ist eine vereinfachte Darstellung eines allgemeinen räumlichen Seilroboters abgebildet.

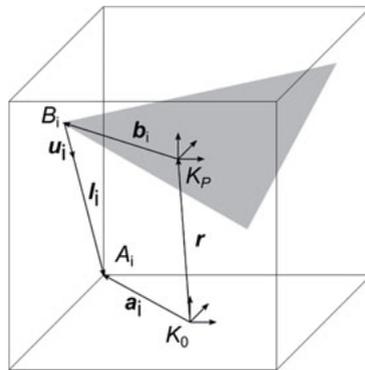


Abbildung 3: Schematische Darstellung eines allgemeinen räumlichen Seilroboters

Die Position der Roboterplattform wird über den Ortsvektor im globalen Koordinatensystem K_0 gemessen. Die Anschlagpunkte am Rahmen A_i werden über die Vektoren a_i im globalen Koordinatensystem adressiert. Die Seilbefestigungen auf der Plattform werden mit B_i bezeichnet und im lokalen Plattformkoordinatensystem K_P über den Ortsvektor b_i gemessen. Die Seile l_i werden als Vektoren über die kinematische Schleifenschließbedingung formuliert als,

$$l_i = a_i - r - Rb_i \quad \text{mit} \quad i = 1, \dots, m. \tag{3}$$

Wobei die orthogonale Rotationsmatrix R die Verdrehung des lokalen Koordinatensystems gegenüber der globalen Referenz beschreibt. Die Wirkrichtung der Seilkraft an der Plattform wird über den Einheitsvektor in Seilrichtung u_i dargestellt. Mit den kinematischen Definitionen kann die Strukturgleichung und damit das statische Gleichgewicht der Roboterplattform dargestellt werden als,

$$A^T$$

$$\begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \\ \mathbf{Rb}_1 \times \mathbf{u}_1 & \dots & \mathbf{Rb}_m \times \mathbf{u}_m \end{bmatrix} \begin{bmatrix} f_1 \\ \vdots \\ f_m \end{bmatrix} + \begin{bmatrix} \mathbf{f}_P \\ \boldsymbol{\tau}_P \\ \mathbf{w}_P \end{bmatrix} = \mathbf{0}. \quad (4)$$

Die Strukturmatrix A^T verteilt somit den Vektor der Seilkräfte \mathbf{f} auf die Systemfreiheitsgrade. Im Vektor \mathbf{w}_P werden die Kräfte und Drehmomente zusammengefasst, die als externe Last auf die Plattform einwirken. Die Strukturgleichung ist für die Betrachtung und Analyse des Seilroboters eine der zentralen Größen. Ausgehend von dieser Gleichung können nahezu sämtliche Eigenschaften abgeleitet werden. Im folgenden Kapitel wird der Begriff der Rekonfiguration sowie die Motive und Ziele eines entsprechenden Vorgangs eingeführt.

7.3 Definition der Rekonfiguration

Im Rahmen eines DFG-Forschungsprojektes wird die Möglichkeit einer In-Betrieb-Rekonfiguration von Seilrobotern im Hinblick auf die Anwendung im Bereich der Handhabungstechnik untersucht. In diesem Beitrag wird unter dem Begriff der Rekonfiguration verstanden, dass die Struktur des Seilroboters verändert wird um das System auf veränderte Anforderungen einzustellen. Die Rekonfiguration eines Seilroboters ist eng mit dem Entwurf oder der Synthese verwandt. Die Gemeinsamkeit ist, dass eine Anforderung gegeben ist und ein zugehöriger Roboter gesucht wird, der diese Anwendung möglichst gut erfüllt. Der grundlegende Unterschied ist, dass prinzipiell bei einem initialen Entwurf sämtliche Parameter frei wählbar sind. Bei einer Rekonfiguration besteht die Struktur bereits und nicht alle Parameter sind variabel. In der folgenden Auflistung sind Eigenschaften aufgelistet, die im Rahmen einer Rekonfiguration angepasst und verändert werden können:

- Bewegungsprofil des Seilroboters
- Nutzlast des Seilroboters
- Optimierung der maximal möglichen Plattformrotation
- Größe und Topologie des Arbeitsraums
- Steifigkeit des Roboters

In der folgenden Auflistung sind die Stellschrauben zur Anpassung des Roboters zusammengefasst. Die tatsächlichen rekonfigurierbaren Parameter ergeben sich jedoch aus den Gegebenheiten am bestehenden Roboter.

- An- und Abkoppeln von Antriebssträngen
- Veränderung der Seilbefestigungen

In Abbildung 4 sind die betrachteten Rekonfigurationen anhand eines ebenen Beispiels dargestellt. In der mittleren Abbildung ist die initiale Konfiguration als Ausgangsbasis dargestellt. Durch Verschieben der Anschlagpunkte auf der Plattform und dem Rahmen können beispielsweise Modifikationen des Arbeitsraums oder der Steifigkeit erreicht werden. Im rechten Teil der Abbildung erfolgt eine Erhöhung der Nutzlast durch Zuschalten von weiteren Winden.

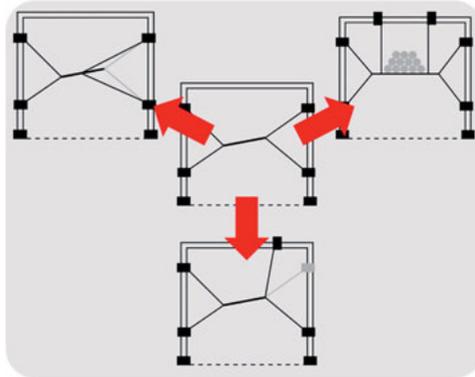


Abbildung 4: Darstellung der variablen Parameter

Die Maßnahmen zur Rekonfiguration lassen sich prinzipiell in zwei Gruppen unterteilen. Das An- und Abkoppeln von Antriebssträngen stellt einen tiefen Eingriff in das Systemverhalten des Roboters dar. Für die Strukturmatrix bedeutet beispielsweise das Entfernen eines Seils, dass eine Spalte entfällt. Bei einer Verschiebung von Anschlagpunkten erfolgt keine Beeinflussung der mathematischen Struktur und damit ist es mehr eine Änderung der Parameter. So wird im Folgenden das An- und Abkoppeln von Antriebssträngen als eine strukturelle Rekonfiguration bezeichnet. Das Verschieben von Anschlagpunkten verändert die Geometrie des Roboters und wird daher als geometrische Rekonfiguration bezeichnet. Wie beschrieben, bezieht sich die Rekonfiguration auf ein bestehendes Robotersystem. In Abbildung 5 ist der institutseigene Demonstrator abgebildet, der für sämtliche weiteren Analysen und Betrachtungen verwendet wird.

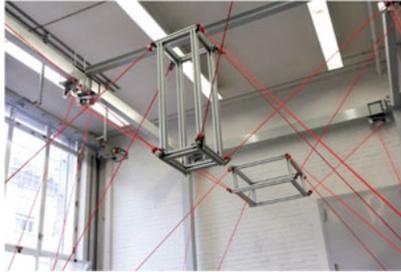


Abbildung 5: Rekonfigurierbare Hardware des COPacabana

Der Seilroboter COPacabana verfügt über insgesamt 16 separate Antriebsstränge, die beliebig miteinander kombiniert werden können. Acht der Umlenkrollen sind oberhalb der Plattform angebracht. Die weiteren acht Umlenkrollen sind unterhalb der Plattform montiert. Mit dieser Installation sind sämtliche Klassen von über- bis unterbestimmt realisierbar. Eine geometrische Rekonfiguration kann umgesetzt werden, indem die Anschlagpunkte auf den Profilmaterialien verschoben werden. Im folgenden Unterpunkt wird ein Konzept eingeführt, das eine ganzheitliche Rekonfiguration des Systems ermöglicht. Bei dem abgebildeten COPacabana-Roboter sind prinzipiell alle möglichen Roboterstrukturen umsetzbar. Um die Vielfalt der möglichen Varianten zu reduzieren, werden bei der Struktursynthese ausschließlich beidseitig verspannte Konfigurationen mit acht Antriebsseilen betrachtet.

7.4 Konzept der Rekonfiguration

Das Konzept für einen rekonfigurierbaren Seilroboter gliedert sich in insgesamt vier Phasen. In der ersten Phase erfolgt die Definition der Anforderung durch den Bediener. In der zweiten Phase die Struktur- und Geometriesynthese gefolgt von einem Ansatz zur Rekonfiguration der Steuerung. Abschließend wird die praktische Umsetzung der neuen Konfiguration betrachtet.

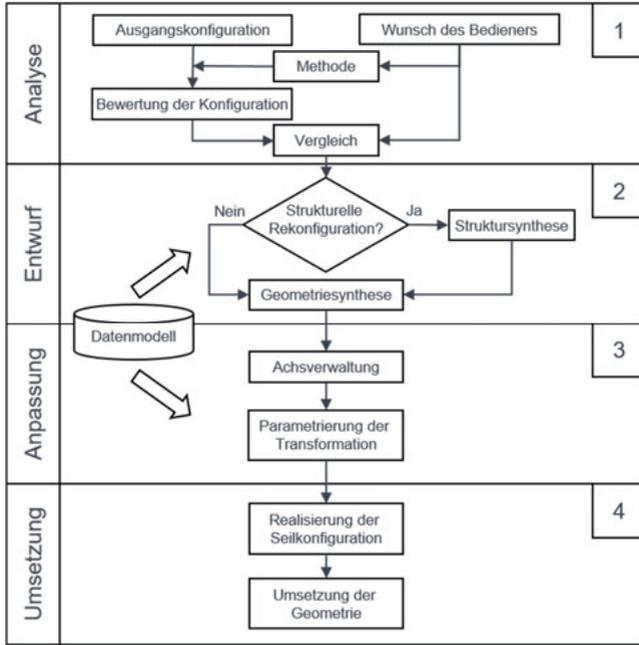


Abbildung 6: Konzept der Rekonfiguration

7.4.1 Analyse der Anforderung

Zu Beginn einer Rekonfiguration steht die veränderte Anforderung an die Eigenschaften des Seilroboters. Die Anforderungen werden durch den Bediener formuliert und können folgende vier Merkmale umfassen:

- Arbeitsraum des Seilroboters
- Nutzlast des Seilroboters
- Steifigkeit des Seilroboters
- Die maximal mögliche Rotation der Plattform

Auf der Seite des Roboters ist die Ausgangslage der installierten Konfiguration. Dazu zählen die aktuell montierten Seile und die zugehörigen Befestigungspunkte. Die Eigenschaften werden repräsentiert durch:

- Indexvektor der verwendeten Seile
- Parametervektor für die Geometrie der Seile

Die Anforderung des Bedieners definiert wie die Charakteristik des angepassten Systems aussehen soll. Da der Wunsch des Bedieners bereits ein Kriterium (Arbeitsraum,

Nutzlast, Steifigkeit) beinhaltet, muss der aktuell e Roboter hinsichtlich des entsprechenden Merkmals analysiert werden.

7.4.2 Entwurf der benötigten Rekonfiguration

Während der zweiten Phase der Rekonfiguration wird ein System identifiziert, das den Anforderungen möglichst genau entspricht. Innerhalb des ersten wichtigen Funktionsblocks muss eine Entscheidung getroffen werden, ob mit der aktuellen Seilkonfiguration die gewünschte Anforderung abgedeckt werden kann. Ist das nicht der Fall, muss eine Struktursynthese durchgeführt werden. Entsprechend der Einschränkung auf acht Antriebsseile muss ein kombinatorisches Problem gelöst werden, das für den Anwendungsfall aus den 16 Winden die acht besten Seile selektiert. Die Ausgangswerte der Struktursynthese sind somit die benötigten Antriebsseile. Wird keine Struktursynthese benötigt, kann direkt die Geometriesynthese erfolgen. Die Aufgabe der Geometriesynthese ist es, die geometrischen Parameter des Seilroboters entsprechend der gewünschten Anwendung festzulegen. Die Ausgangswerte der Entwurfsphase sind der Indexvektor der gewünschten Seile sowie der zugehörige Parametervektor der Geometrie. Bei der Struktur- sowie Geometriesynthese werden die Informationen zu den möglichen Systemkonfigurationen aus dem Datenmodell bezogen. In diesem Datensatz sind zu allen 16 Antriebssträngen die initiale n Koordinaten der Seilangriffspunkte sowie die zulässigen Variationsmöglichkeiten des Roboters abgespeichert. Eine gett 7.5 aufgeführt.

7.4.3 Anpassung der Steuerung

Bei der bestehenden Steuerungsarchitektur sind die Seilwinden als Achsen in einem CNC-Kanal zusammengefasst. Für die strukturelle Rekonfiguration werden die Achsen entsprechend dem Indexvektor angepasst. Bei der installierten Steuerung (Beckhoff TwinCAT 3) ist dies über die Achsverwaltung möglich, indem die nicht mehr benötigten Achsen angefordert werden. Nachdem der Kanal mit den Achsen aus dem Indexvektor ausgestattet ist, müssen die kinematischen Transformationen entsprechend dem Parametervektor angepasst werden. Die kinematischen Transformationen werden benötigt um den beidseitigen Zusammenhang zwischen den Seillängen und der zugehörigen Plattformposition zu beschreiben. Durch die Einschränkung, dass nur verspannte Systeme mit acht Seilen betrachtet werden, bleibt die grundlegende Funktionsweise der Algorithmen für die inverse und direkte Kinematik erhalten. Bei einer Änderung der Geometrie müssen jedoch die Funktionsparameter der Methoden angepasst werden.

7.4.4 Umsetzung der Rekonfiguration

Nachdem die Rekonfiguration der Steuerung erfolgreich durchgeführt wurde, muss am Seilroboter die entsprechende Konfiguration eingestellt werden. Die geometrischen Parameter des Seilroboters müssen für eine ausreichende Genauigkeit sehr exakt kalibriert sein. Diese Kalibrierung müsste prinzipiell nach jeder Rekonfiguration nochmals

durchgeführt werden. Ein solcher Kalibriervorgang würde eine Rekonfiguration wirtschaftlich unattraktiv machen. In diesem Beitrag soll der Fokus eher auf der Vorstellung einer allgemeinen Möglichkeit zur Rekonfiguration liegen als auf der geometrischen Kalibrierung. So wird im Folgenden davon ausgegangen, dass der Seilroboter in der initialen Konfiguration kalibriert ist. Eine genaue Kenntnis über die Koordinaten der Anschlagpunkte nach der Rekonfiguration wird über konstruktive Maßnahmen sichergestellt. Entsprechende Maßnahmen wäre beispielsweise eine diskrete Verschiebung zu ermöglichen, indem die Anschlagpunkte auf einem exakt bekannten Lochraster variiert werden können. Für eine kontinuierliche Verschiebung der Anschlagpunkte wäre es denkbar den Anschlagpunkt auf einer Linearführung mit einer integrierten Positionsmessung zu montieren. Mit diesen Annahmen bedeutet es für die letzte Phase der Rekonfiguration, dass die Winden und Anschlagpunkte entsprechend dem Index- und Parametervektor eingestellt werden. Nachdem die Umrüstung der Hardware durchgeführt wurde kann der Seilroboter wieder In-Betrieb gesetzt werden. Wie in Punkt 7.4.2 und 7.4.3 beschrieben ist, muss für die Auswahl einer geeigneten Roboterkonfiguration sämtliche mögliche Variationen bekannt sein. Im folgenden Unterpunkt wird ein Datenmodell vorgestellt, dass in Bezug auf eine initiale Referenzkonfiguration alle relevanten Strukturvarianten in einem Datenmodell gebündelt zusammenfasst.

7.5 Datenmodell eines rekonfigurierbaren Seilroboters

Im ersten Abschnitt wird die theoretische Interpretation der Rekonfiguration erläutert. Im zweiten Abschnitt werden die mathematischen Definitionen mit den zugehörigen Parametermengen eingeführt. Im letzten Abschnitt werden die beschriebenen Strukturen auf den COPacabana-Roboter angewandt.

7.5.1 Auffassung der Rekonfiguration

Entsprechend der Definitionen aus Abbildung 3 können die Ortsvektoren für die Befestigungspunkte beschrieben werden als,

$$\mathbf{a}^i = \begin{bmatrix} a_x^i \\ a_y^i \\ a_z^i \end{bmatrix}, \mathbf{b}^i = \begin{bmatrix} b_x^i \\ b_y^i \\ b_z^i \end{bmatrix} \in \mathbb{R}^3 \quad \text{mit } i = 1, \dots, m. \quad (5)$$

Die geometrischen Parameter aller kinematischen Ketten werden anschließend zu einem Parametervektor

$$\Phi = [\mathbf{a}_1, \dots, \mathbf{a}_m, \mathbf{b}_1, \dots, \mathbf{b}_m]^T, \quad (6)$$

zusammengefasst. Der Vektor $\Phi \in \mathbb{R}^{2(3 \times m)}$ enthält somit sämtliche Parameter, die einen Roboter für eine entsprechende Konfiguration erzeugen. Die absoluten Koordinaten der Befestigungspunkte werden so aufgefasst, dass die aktuellen Positionen der Befestigungspunkte aus einer Summe eines Referenzwertes und einer relativen Verschiebung bestehen. Die Summe kann formuliert werden als,

$$\Phi = \bar{\Phi} + \Delta\Phi. \quad (7)$$

Der erste Anteil

$$\bar{\Phi} = [\bar{a}_1, \dots, \bar{a}_m, \bar{b}_1, \dots, \bar{b}_m]^T, \quad (8)$$

Beschreibt den Roboter in der Referenzkonfiguration. Die relative Verschiebung dazu wird beschrieben über den Vektor,

$$\Delta\Phi = [\Delta a_1, \dots, \Delta a_m, \Delta b_1, \dots, \Delta b_m]^T. \quad (9)$$

Jede Komponente des Verschiebungsvektors ist Teil einer abgeschlossenen Menge, die einen zulässigen Bereich um die Referenzkoordinaten beschreibt. Da die Verschiebungsmöglichkeiten der Anschlagpunkte immer konstruktiv begrenzt sind, ist die Zahlenmenge immer ein abgeschlossenes Intervall. Somit beschreiben die Intervallgrenzen die Endwerte an die die Befestigung maximal verschoben werden kann. Für die innere Struktur der Menge sind prinzipiell zwei Varianten denkbar. Zum einen ist es möglich, dass mit dem Befestigungspunkt zwischen zwei Endwerten sämtliche Zwischenwerte angenommen werden können. Dies würde bei spielsweise einen Fall beschreiben, wo der Befestigungspunkt über einen Linearantrieb verstellt werden kann. Die zweite Variante wäre, dass zwischen den Grenzwerten eine feste und diskrete Anzahl an Punkten existiert, an die der Seilanschluss angebracht werden kann. Eine denkbare Konstruktion wäre eine Leiste mit entsprechenden Befestigungsbohrungen. Bei einer allgemeinen diskreten Mengenbeschreibung ist nicht von Relevanz, ob die Aufteilung der Zwischenwerte einer Gesetzmäßigkeit folgt. Die Kernaufgabe bei der Implementierung eines Datenmodells ist es somit eine Beschreibung für die Referenzkonfiguration sowie die zugeordneten Variationsbereiche in Form der diskreten oder kontinuierlichen Zahlenmengen zu definieren.

Das Datenmodell des rekonfigurierbaren Seilroboters soll mit dem gängigen Format der „Extensible Markup Language (XML)“ aufgebaut werden. Bei diesem Format werden die Daten in hierarchischer Struktur abgebildet. Für die Darstellung des Seilroboters eignet sich das sehr gut, da sich das System einfach in unterschiedliche Hierarchieebenen gliedern lässt. Die oberste Ebene besteht aus dem gesamten Roboter. Die zweite Ebene gliedert sich in die einzelnen Antriebsstränge. Für den Fall des CO-Pacabana ergibt sich eine zweite Ebene die aus insgesamt 16 Knoten besteht. Jeder Antriebsstrang wiederum wird durch die Befestigungspunkte auf der Plattform bzw. auf dem Rahmen repräsentiert. Die Befestigungspunkte werden wiederum beschrieben durch die kartesischen Koordinaten der Referenzpunkte sowie die zugeordneten Variationsbereiche. In der folgenden Auflistung sind nochmals schrittweise die einzelnen Ebenen des XML-Dokuments dargestellt.

- Roboter
 - Kette1
 - Rahmen

- x_0
- y_0
- z_0
- Zulässige x -Variation
- Zulässige y -Variation
- Zulässige z -Variation
- Plattform
 - ...
- Kette2
- Kette3
- .
- .

In Bezug auf das Konzept der Rekonfiguration aus Abschnitt 7.4 zeigt sich hier auch die Hierarchie zwischen der strukturellen und geometrischen Rekonfiguration. Die Auswahl der benötigten Antriebsseile erfolgt auf der Ebene der kinematischen Ketten. Die Abstimmung der Geometrie erfolgt zwei Ebenen unterhalb. Wie in der Darstellung beschrieben, erfolgt die Speicherung der zulässigen Variationsbereiche auf der Ebene der kartesischen Koordinaten. Im folgenden Unterkapitel wird definiert, wie die betrachteten kontinuierlichen und diskreten Mengen durch entsprechende Attribute beschrieben werden können.

7.5.2 Kontinuierliche Parametermengen

Die Implementierung einer kontinuierlichen Mengendefinition kommt aus der konstruktiven Möglichkeit die Seilbefestigung zwischen zwei Grenzen beliebig positionieren zu können. Die mathematische Definition der Menge erfolgt hier exemplarisch für die Befestigung am Rahmen des i -ten Seils. Als Wirkrichtung wird die kartesische x -Richtung definiert. So kann das zulässige Intervall mit den entsprechenden Grenzwerten formuliert werden als,

$$\Delta a_x^i \in [\check{a}_x^i, \hat{a}_x^i] \quad \text{mit} \quad \check{a}_x^i, \hat{a}_x^i \in \mathbb{R}. \tag{10}$$

Wobei \check{a}_x^i die untere, beziehungsweise \hat{a}_x^i die obere Intervallgrenze beschreiben. Für den initialen Parameter \bar{a}_x^i gilt, dass der Wert innerhalb oder auf den Grenzen des Intervalls liegen muss. Die Relation kann dargestellt werden als,

$$\check{a}_x^i \leq \bar{a}_x^i \leq \hat{a}_x^i. \tag{11}$$

Für die vollständige Beschreibung der Menge sind mit nur die obere und untere Intervallgrenze notwendig. In der folgenden Auflistung sind die zur vollständigen Beschreibung benötigten Attribute zusammengefasst.

- Maximalwert

- Minimalwert

7.5.3 Diskrete Parametermengen

Bei einer allgemeinen diskreten Menge müssen die enthaltenen Komponenten keiner Regelmäßigkeit folgen. Die Mengendefinition erfolgt über die Zusammenfassung der zulässigen Punkte. Der zulässige Bereich kann somit dargestellt werden als,

$$\Delta a_x^i \in \{a_{x,1}^i, \dots, a_{x,d}^i\} \quad \text{mit} \quad d \in \mathbb{N}. \quad (12)$$

In der Definition einer allgemeinen und unstrukturierten Menge sind ebenfalls die diskreten Mengen mit innerer Regelmäßigkeit enthalten. Wenn beispielsweise die Löcher auf der Befestigungsleiste in einem regelmäßigen Abstand gebohrt werden, ist die Aufteilung der Stützstellen zwischen den Grenzwerten linear. So wäre es prinzipiell möglich anstatt der gesamten diskreten Stützstellen ein Gesetzmäßigkeit zu hinterlegen. Für die lineare Skalierung könnte die Anzahl der Stützstellen oder der Abstand zwischen den Stützstellen abgespeichert werden. Die einzelnen Zwischenwerte könnten dann aus den Intervallgrenzen und der Anzahl der Punkte berechnet werden. Da das Datenmodell so allgemein wie möglich gehalten werden soll, wird die Definition auf unstrukturierte Mengen beschränkt. In der folgenden Aufzählung sind die Attribute aufgeführt, die benötigt werden um eine diskrete Menge abzuspeichern.

- Maximalwert
- Minimalwert
- Anzahl der Elemente
- Vektor mit Zahlenwerten

Mit den Festlegungen für die Struktur des XML-Schemas sowie den Definitionen der zulässigen Variationsmengen wird im folgenden Unterkapitel das Datenmodell auf den institutseigenen Demonstrator COPacabana übertragen und angewandt.

7.5.4 Datenmodell des COPacabana

Wie bereits beschrieben, verfügt der Seilroboter in dem vorhandenen Bauraum über insgesamt 16 separate Antriebsseile. Prinzipiell können zwei Seilroboter mit jeweils acht Seilen parallel betrieben werden. Das gesamte Layout eignet sich jedoch ebenfalls dazu, die Antriebsstränge zu einem System zusammenzufassen und je nach Anwendung den passenden Roboter zu generieren. In Abbildung 7 ist das betrachtete System in Referenzkonfiguration abgebildet.

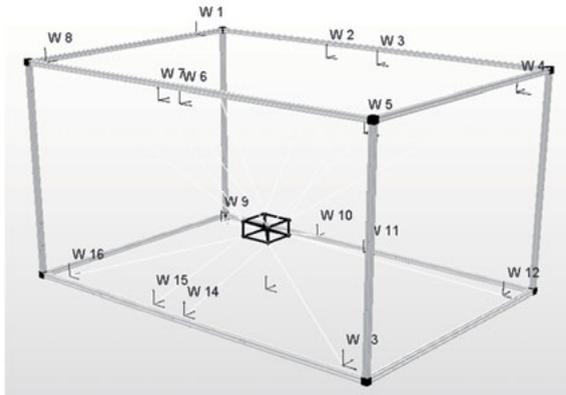


Abbildung 7: Rekonfigurierbarer Seilroboter COPacabana

Die zugehörige XML-Struktur ist in der folgenden Abbildung dargestellt. In der ersten Ebene unterhalb der gesamten Roboterstruktur wird für jede kinematische Kette ein Knoten eingefügt. Zusätzlich zur Benennung der jeweiligen Kette können noch individuelle Informationen wie beispielsweise die verwendete Winde hinterlegt werden.

```
<geometry>
<chain id="1" winctype="IPAnema-Winde">
<chain id="2" winctype="IPAnema-Winde">
<chain id="3" winctype="IPAnema-Winde">
<chain id="4" winctype="IPAnema-Winde">
<chain id="5" winctype="IPAnema-Winde">
<chain id="6" winctype="IPAnema-Winde">
<chain id="7" winctype="IPAnema-Winde">
<chain id="8" winctype="IPAnema-Winde">
<chain id="9" winctype="IPAnema-Winde">
<chain id="10" winctype="IPAnema-Winde">
<chain id="11" winctype="IPAnema-Winde">
<chain id="12" winctype="IPAnema-Winde">
<chain id="13" winctype="IPAnema-Winde">
<chain id="14" winctype="IPAnema-Winde">
<chain id="15" winctype="IPAnema-Winde">
<chain id="16" winctype="IPAnema-Winde">
</geometry>
```

Abbildung 8: XML-Schema der kinematischen Ketten

Für jede der abgebildeten Ketten werden in den unterliegenden Ebenen die Koordinaten der Befestigungspunkte angelegt. In der folgenden Abbildung ist beispielhaft für die erste kinematische Kette ein Schema dargestellt. Die Variationsmöglichkeiten werden zum einen für die Befestigungen am Rahmen und zum anderen an der Plattform

exemplarisch gezeigt. Für die diskrete Menge („Discrete“) werden die Grenzwerte sowie die gesamte Anzahl der Stützstellen abgelegt. Dem gehörigen Koordinaten wird separat hinzugefügt. Zwischen Wertebereichen („Continuous“) werden ausschließlich Neben den Definitionen für die zulässigen Mengen ist eine Verschiebung implementiert. Hiermit kann beispielsweise („Fixed“) gekennzeichnet werden oder aber ein Parameter verschoben werden.

rete“) werden die Grenzwerte sowie der eigentliche Vektor mit den zugehörigen Speicherungen eines kontinuierlichen die Intervallgrenzen benötigt. t als dritte Variante eine konstante eise ein fester Parameter („Fixed“) konstant aus der Referenzlage

```
<base_y Location="base" VariationType="Discrete" dofId="y" id="1">
  <properties NumberOfSamples="6" max="0.3" min="-1" />
  <values s1="-1" s2="-0.5" s3="0" s4="0.1" s5="0.2" s6="0.3" />
</base_y>
<base_z Location="base" VariationType="Continuous" dofId="z" id="1">
  <properties VariationSpan="2" max="1" min="-1" />
  <values />
</base_z>
<platform_x Location="platform" VariationType="Fixed" dofId="x" id="1">
  <properties delta="0" max="0" min="0" />
  <values />
</platform_x>
```

Abbildung 9: XML-Schema der Variationsmengen

In diesem Datenmodell sind somit alle zulässigen Abwandlungen des initialen Roboters gesammelt, die an dem Demonstrator realisiert werden können. Im folgenden Abschnitt werden die Inhalte des Beitrags zusammengefasst und es wird ein Ausblick auf die nächsten Schritte in Richtung eines rekonfigurierbaren Seilroboters gegeben.

wandlungen des initialen Roboters werden können. Im folgenden Abschnitt werden die Inhalte des Beitrags zusammengefasst und es wird ein Ausblick auf die nächsten Schritte in Richtung eines rekonfigurierbaren Seilroboters gegeben.

7.6 Zusammenfassung und Ausblick

In diesem Beitrag wird ein Konzept für die prinzipiell rekonfigurierbaren Seilroboter vorgestellt. Dazu wird in Abschnitt 7.4 eine allgemeine Vorgehensweise skizziert, die sich prinzipiell in vier Phasen gliedert. In Phase eins erfolgen die Auswertung der Vorgabe des Maschinenbedieners sowie eine Bewertung der aktuellen Konfiguration. Nach der Analysephase erfolgt der Entwurf einer Roboterstruktur, die die gegebenen Anforderungen erfüllt. Der Entwurf des Roboters wird dabei in eine Struktur- und Geometriesynthese unterteilt. Nachdem eine passende Roboterstruktur identifiziert wurde, muss das Steuerungsprogramm des Seilroboters angepasst werden. Im Fall einer Rekonfiguration der Struktur müssen die nicht mehr benötigten Antriebe gegen die benötigten ausgetauscht werden. Anschließend werden die kinematischen Transformationen mit der neuen Geometrie parametrisiert. Nachdem die Rekonfiguration auf der Seite der Software stattgefunden hat, kann die neue Roboter-

elle Funktionsweise eines rekonfigurierbaren Seilroboters vorgestellt. Dazu wird in Abschnitt 7.4 eine allgemeine Vorgehensweise skizziert, die sich prinzipiell in vier Phasen gliedert. In Phase eins erfolgen die Auswertung der Vorgabe des Maschinenbedieners sowie eine Bewertung der aktuellen Konfiguration. Nach der Analysephase erfolgt der Entwurf einer Roboterstruktur, die die gegebenen Anforderungen erfüllt. Der Entwurf des Roboters wird dabei in eine Struktur- und Geometriesynthese unterteilt. Nachdem eine passende Roboterstruktur identifiziert wurde, muss das Steuerungsprogramm des Seilroboters angepasst werden. Im Fall einer Rekonfiguration der Struktur müssen die nicht mehr benötigten Antriebe gegen die benötigten ausgetauscht werden. Anschließend werden die kinematischen Transformationen mit der neuen Geometrie parametrisiert. Nachdem die Rekonfiguration auf der Seite der Software stattgefunden hat, kann die neue Roboter-

konfiguration an der Maschine umgesetzt werden. In Abschnitt 7.5 wird ein Datenmodell vorgestellt, das es ermöglicht die strukturellen und geometrischen Rekonfigurationsoptionen in einem XML-Schema zu speichern. Das Datenmodell beinhaltet somit sämtliche Abwandlungen, die mit dem Demonstrator möglich sind und liefert die Randbedingungen für die Synthese und die entsprechende spätere Parametrierung der Steuerung.

Die Arbeiten, die dem beschriebenen Konzept folgen, fokussieren sich auf die Anwendung und Implementierung der beschriebenen Vorgehensweise. Für den Teil der Struktur- und Geometriesynthese stellt sich die Frage, wie Ansätze aus dem allgemeinen Roboterentwurf angewandt bzw. verwendet werden können. Ein weiterer Punkt ist die Implementierung einer flexiblen Umgebung, um den Tausch der CNC-Achsen im Steuerungsprogramm durchzuführen sowie eine anschließende zuverlässige Parametrierung der kinematischen Transformationen.

7.7 Danksagung

Dieser Beitrag entstand im Rahmen einer Projektförderung der Deutschen Forschungsgemeinschaft (DFG-Projektnummer: PO 1570/5-1) an der Universität Stuttgart.

7.8 Literatur

- [1] J. S. Albus, R. V. Bostelman und N. G. Dagalakis, „The NIST ROBOCRANE“ in Bd. 3, Journal of Research of the National Institute of Standards and Technology, 1992, S. 373–385.
- [2] S. Kawamura, W. Choe, S. Tanaka und S. R. Pandian, „Development of an Ultrahigh Speed Robot FALCON using Wire Drive System“ ICIRA 1995 IEEE International Conference on Robotics and Automation, Nagoya, Japan, 1995, S. 215–220.
- [3] J. Lamaury und M. Gouttefarde, „Control of a large redundantly actuated cable-suspended parallel robot“ in 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, Mai. 2013 - Mai. 2013, S. 4659–4664.
- [4] A. Pott et al., „IPAnema: A family of Cable-Driven Parallel Robots for Industrial Applications“ in Mechanisms and Machine Sciences, Bd. 12, Cable-Driven Parallel Robots, T. Bruckmann und A. Pott, Hg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 119–134.
- [5] D. Stewart, „A Platform with Six Degrees of Freedom“, Proceedings of the Institution of Mechanical Engineers, Jg. 180, Nr. 1, S. 371–386, 1965.
- [6] P. Bosscher, R. L. Williams, L. S. Bryson und D. Castro-Lacouture, „Cable-suspended robotic contour crafting system“, Automation in Construction, Jg. 17, Nr. 1, S. 45–55, 2007.
- [7] D. Q. Nguyen, M. Gouttefarde, O. Company und F. Pierrot, „On the analysis of large-dimension reconfigurable suspended cable-driven parallel robots“ 2014

- IEEE International Conference on Robotics and Automation (ICRA) 2014, S. 5728–5735.
- [8] D. Q. Nguyen und M. Gouttefarde, „Study of reconfigurable suspended cable-driven parallel robots for airplane maintenance“ in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, S. 1682–1689.
- [9] P. Tempel, F. Schnelle, A. Pott und P. Eberhard, „Design and Programming for Cable-Driven Parallel Robots in the German Pavilion at the EXPO 2010 Shanghai“, *Chineses*, Jg. 3, Nr. 3, S. 223–241, 2015.
- [10] Richard Verhoeven, „Analysis of the Workspace of Tendon-based Stewart Platforms“. PhD thesis, University of Duisburg-Essen, 2004.
- [11] M. Gouttefarde, J.-P. Merlet und D. Daney, „Determination of the Wrench-closure Workspace of 6-dof Parallel Cable-driven Mechanisms“ in 2006.
- [12] J. P. Merlet, „On the workspace of suspended cable-driven parallel robots“ in 2016 IEEE International Conference on Robotics and Automation (ICRA) 2016, S. 841–846.
- [13] A. Pott, „Forward Kinematics and Workspace Determination of a Wire Robot for Industrial Applications“ in *Advances in Robot Kinematics: Analysis and Design*, J. Lenarčič und P. Wenger, Hg., Dordrecht: Springer Netherlands, 2008, S. 451–458.
- [14] A. Pott und W. Kraus, „Determination of the wrench-closure translational workspace in closed-form for cable-driven parallel robots“ in 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, S. 882–887.
- [15] G. Barrette und C. M. Gosselin, „Determination of the Dynamic Workspace of Cable-Driven Planar Parallel Mechanisms“, *Journal of Mechanical Design*, Jg. 127, Nr. 2, S. 242–248, 2005.
- [16] L. Gagliardini, S. Caro und M. Gouttefarde, „Dimensioning of cable-driven parallel robot actuators, gearboxes and winches according to the twist feasible workspace“ in 2015 IEEE International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden, Aug. 2015 - Aug. 2015, S. 99–105.
- [17] S. Behzadipour und A. Khajepour, „Stiffness of Cable-based Parallel Manipulators With Application to Stability Analysis“, *J. Robotic Syst.*, Jg. 128, Nr. 1, S. 303, 2006.
- [18] V. Schmidt, W. Kraus und A. Pott, „Presentation of Experimental Results on Stability of a 3 DOF 4-Cable-Driven Parallel Robot Without Constraints“, *Mechanisms and Machine Science, Cable-Driven Parallel Robots*, A. Pott und T. Bruckmann, Hg., Cham: Springer International Publishing, 2015, S. 87–99.
- [19] A. Pott, *Cable-Driven Parallel Robots*. Cham: Springer International Publishing, 2018.
- [20] Jean-Pierre Merlet, „Optimal Design of Robots“, S. 311–318.
- [21] F. Hao und J.-P. Merlet, „Multi-criteria optimal design of parallel manipulators based on interval analysis“, *Mechanism and Machine Theory*, Jg. 40, Nr. 2, S. 157–171, 2005.

- [22] J.-P. Merlet, „An improved design algorithm based on interval analysis for spatial parallel manipulator with specified workspace“ in Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164) Seoul, South Korea, Mai. 2001, S. 1289–1294.
- [23] T. Bruckmann Dr.-Ing., „Auslegung und Betrieb redundanter paralleler Seilroboter“.
- [24] E. Ottaviano und M. Ceccarelli, „Optimal design of CaPaMan (Cassino Parallel Manipulator) with a specified orientation workspace“ *Robotica*, Jg. 20, Nr. 02, 2002.
- [25] L. Gagliardini, S. Caro, M. Gouttefarde und A. Girin, „Discrete reconfiguration planning for Cable-Driven Parallel Robots“, *Mechanism and Machine Theory*, Jg. 100, Supplement C, S. 313–337, <http://www.sciencedirect.com/science/article/pii/S0094114X16000513>, 2016.
- [26] L. Gagliardini, „Discrete Reconfigurations of Cable-Driven Parallel Robots“.
- [27] A. Ming und T. Higuchi, „Study on multiple degree-of-freedom positioning mechanism using wires (part 1) - Concept, design and construction“, *International Journal of the Japan Society for Precision Engineering*, Jg. 28, S. 131–138, 1994.



Werden Sie Autor
im VDI Verlag!

Publizieren Sie
in „Fortschritt-
Berichte VDI“

Veröffentlichen Sie die Ergebnisse Ihrer interdisziplinären technikorientierten Spitzenforschung in der renommierten Schriftenreihe **Fortschritt-Berichte VDI**. Ihre Dissertationen, Habilitationen und Forschungsberichte sind hier bestens platziert:

- **Kompetente Beratung und editorische Betreuung**
- **Vergabe einer ISBN-Nr.**
- **Verbreitung der Publikation im Buchhandel**
- **Wissenschaftliches Ansehen der Reihe Fortschritt-Berichte VDI**
- **Veröffentlichung mit Nähe zum VDI**
- **Zitierfähigkeit durch Aufnahme in einschlägige Bibliographien**
- **Präsenz in Fach-, Uni- und Landesbibliotheken**
- **Schnelle, einfache und kostengünstige Abwicklung**

PROFITIEREN SIE VON UNSEREM RENOMMEE!

www.vdi-nachrichten.com/autorwerden

VDI verlag

Die Reihen der Fortschritt-Berichte VDI:

- 1 Konstruktionstechnik/Maschinenelemente
 - 2 Fertigungstechnik
 - 3 Verfahrenstechnik
 - 4 Bauingenieurwesen
- 5 Grund- und Werkstoffe/Kunststoffe
 - 6 Energietechnik
 - 7 Strömungstechnik
- 8 Mess-, Steuerungs- und Regelungstechnik
 - 9 Elektronik/Mikro- und Nanotechnik
 - 10 Informatik/Kommunikation
 - 11 Schwingungstechnik
- 12 Verkehrstechnik/Fahrzeugtechnik
 - 13 Fördertechnik/Logistik
- 14 Landtechnik/Lebensmitteltechnik
 - 15 Umwelttechnik
 - 16 Technik und Wirtschaft
 - 17 Biotechnik/Medizintechnik
 - 18 Mechanik/Bruchmechanik
 - 19 Wärmetechnik/Kältetechnik
- 20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
 - 21 Elektrotechnik
 - 22 Mensch-Maschine-Systeme
- 23 Technische Gebäudeausrüstung

ISBN 978-3-18-369902-5