

Michael Scholz

Intralogistics Execution System mit integrierten autonomen,
servicebasierten Transportentitäten

FAU Studien aus dem Maschinenbau

Band ...

Herausgeber der Reihe:

Prof. Dr.-Ing. Jörg Franke

Prof. Dr.-Ing. Nico Hanenkamp

Prof. Dr.-Ing. habil. Marion Merklein

Prof. Dr.-Ing. Michael Schmidt

Prof. Dr.-Ing. Sandro Wartzack

Michael Scholz

**Intralogistics
Execution
System mit
integrierten autonomen,
servicebasierten
Transportentitäten**

Dissertation aus dem Lehrstuhl für
Fertigungsautomatisierung und Produktionssystematik (FAPS)
Prof. Dr.-Ing. Jörg Franke

Erlangen
FAU University Press
2019

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im
Internet über <http://dnb.d-nb.de> abrufbar.

Bitte zitieren als

Scholz, Michael. 2019. *Intralogistics Execution System mit integrierten autonomen, servicebasierten Transportentitäten*. Reihe Band.

Erlangen: FAU University Press. DOI: 10.25593/978-3-96147-XXX-X.

Das Werk, einschließlich seiner Teile, ist urheberrechtlich geschützt.
Die Rechte an allen Inhalten liegen bei ihren jeweiligen Autoren.
Sie sind nutzbar unter der Creative Commons Lizenz BY-NC.

Der vollständige Inhalt des Buchs ist als PDF über den OPUS Server
der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:
<https://opus4.kobv.de/opus4-fau/home>

Verlag und Auslieferung:

FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck:

ISBN: 978-3-96147-XXX-X (Druckausgabe)

eISBN: 978-3-96147-XXX-X (Online-Ausgabe)

ISSN: 2625-9974

DOI: 10.25593/978-3-96147-XXX-X

Fakultätstitelblatt

**Intralogistics Execution System mit integrierten
autonomen, servicebasierten Transportentitäten**

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg

zur
Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von

Michael Scholz, M.Eng., Dipl.-Ing. (FH)

aus Bamberg

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen

Prüfung: 19.07.2019

Vorsitzender des

Promotionsorgans: Prof. Dr.-Ing. Reinhard Lerch

Gutachter:

Prof. Dr.-Ing. Jörg Franke
Prof. Dr.-Ing. Johannes Fottner,
TU München
Prof. Dr. rer. pol. Peter Schuderer,
TH Ingolstadt

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik (FAPS) an der Friedrich-Alexander-Universität Erlangen-Nürnberg.

Mein besonderer Dank gilt dem Lehrstuhlinhaber Prof. Dr. Jörg Franke für das in mich gesetzte Vertrauen und die Möglichkeit zur Promotion. Die stetige Unterstützung meiner Forschungstätigkeiten sowie die zahlreichen wertvollen Diskussionen haben die Ergebnisse meiner Arbeit sehr bereichert und entscheidend geprägt. Weiterhin möchte ich mich für die gewährten Freiheiten bedanken, durch die ich mein Interesse und meine Neugierde rund um die Intralogistik und Ressourceneffizienz in jeder Hinsicht entfalten konnte.

Des Weiteren geht mein Dank an den Gründer des Lehrstuhls, Prof. Dr. Klaus Feldmann, für die Übernahme des Vorsitzes des Promotionsverfahrens, sowie Prof. Dr. Johannes Fottner für die Übernahme des zweiten Gutachtens und Prof. Dr. Evi Hartmann als weiteres Mitglied des Prüfungsausschusses. Ebenfalls bedanken möchte ich mich bei Herrn Prof. Dr. Peter Schuderer, der mir die Möglichkeit gegeben hat, mich in zahlreiche Industrieprojekte einzubringen. Durch das entgegengebrachte Vertrauen und die eigenverantwortliche Projektdurchführung konnte ich mich auch persönlich weiterentwickeln.

Ein großer Dank gilt zudem allen Mitarbeiterinnen und Mitarbeitern am Lehrstuhl FAPS für den regen fachlichen Austausch und die kollegiale Arbeitsatmosphäre. Besonders bedanken möchte ich mich bei den Kolleginnen und Kollegen der Biomechatronik, der GFB und natürlich der Technik und Verwaltung. Ferner bedanke ich mich bei allen hochmotivierten und engagierten studentischen Mitarbeitern für ihr Mitwirken an dieser Arbeit.

Ein herzlicher Dank gilt auch meiner gesamten Familie, die mich auf meinem Bildungsweg stets gefördert und bei der Umsetzung meiner Interessen unterstützt hat. Besonders bedanken möchte ich mich bei meiner Frau und meinem Sohn für das Verständnis, wenn die Arbeitstage etwas länger wurden. Lieber Noah, liebe Kathrin, schön, dass es euch gibt!

Güglingen, im Juli 2019

Michael Scholz

Inhaltsverzeichnis

Abkürzungsverzeichnis	x
1 Einleitung	1
1.1 Motivation und Hintergrund	1
1.2 Zielsetzung	5
1.3 Vorgehensweise und Aufbau der Arbeit	7
2 Gestaltungsformen innerbetrieblicher Material- und Informationsflüsse	11
2.1 Etablierte Formen des Material- und Informationsflusses	12
2.1.1 Verrichtungsorientierte Produktionsorganisation	12
2.1.2 Prozessfolgeorientierte Produktionsorganisation	13
2.2 Ausprägungen wissenschaftlicher Organisationsstrukturen	14
2.2.1 Agile Fabrikstrukturen	14
2.2.2 Fraktale Fabrikstrukturen	15
2.2.3 Holone Fabrikstrukturen	17
2.2.4 Auswirkungen auf den Material- und Informationsfluss	18
2.3 Zusammenfassung und Schlussfolgerung	19
3 Grundlagen autonomer, flexibler intralogistischer Systeme und mobiler Transportroboter	21
3.1 Fahrerlose Transportsysteme als flexible Einheiten der Intralogistik	22
3.1.1 Einteilung von unstetigen innerbetrieblichen Fördersystemen	22
3.1.2 Fahrwerkskinematik etablierter fahrerloser Transportfahrzeuge	24
3.2 Auftragsallokation und -zuordnung innerhalb Fahrerloser Transportsysteme	26
3.2.1 Auftragsgenerierung und -einlastung	26
3.2.2 Auftragsverwaltung und -zuordnung	27
3.2.3 Auftragsbearbeitung und Routenführung	28
3.3 Ortung, Digitalisierung des Arbeitsraums und Pfadplanung	28
3.3.1 Spurführung, Standortbestimmung und Lageerfassung Fahrerloser Transportsysteme	29
3.3.2 Weltmodellerstellung intralogistischer mobiler Transportroboter	31

3.3.3	Bahnplanung intralogistischer mobiler Transportroboter	35
3.4	Ausführungsformen aktueller Fahrerloser Transportsysteme und mobiler Transportroboter	39
3.4.1	Kommerziell verfügbare Systeme	39
3.4.2	Systeme in der Forschung und Entwicklung	43
3.5	Zusammenfassung und Handlungsbedarfe	46
4	Methode eines Schichtenmodells zur intralogistischen Auftragsallokation und -bearbeitung	49
4.1	Architektur des Schichtenmodells und der Auftragsallokationsebene	50
4.1.1	Aufbau und Funktionsweise	50
4.1.2	Beschreibung der Auftragsallokation	53
4.1.3	Beschreibung der Auftragsschätzung und -durchführung	57
4.2	Methode zur Beschreibung allgemeingültiger Hauptfähigkeiten von Transportfahrzeugen	59
4.2.1	Umsetzung des hybriden Steuerungsparadigmas	59
4.2.2	Zustandsautomat der Hauptfähigkeit Forecast	60
4.2.3	Zustandsautomat der Hauptfähigkeit Acting	62
4.3	Einbindung der Fahrzeugagenten in ein Multi-Agenten-System zur Auftragsallokation	66
4.3.1	Allgemeine Beschreibung von Softwareagenten	66
4.3.2	Ableitung des spezifischen Agentenverhaltens innerhalb des Allokationsalgorithmus	69
4.3.3	Ableitung des spezifischen Fahrzeugagentenverhaltens durch Kommunikation mit den Zustandsautomaten	74
4.4	Zusammenfassung und Beurteilung	78
5	Verteilte Servicearchitektur zur Digitalisierung des Arbeitsumfelds, der Routen- und Pfadplanung	81
5.1	Architektur des verteilten Systems	82
5.1.1	Aufbau und Funktionsweise der Dienste	82
5.1.2	Beschreibung der systemübergreifenden Routenplanung	85
5.1.3	Ableitung eines auftragsbezogenen globalen Pfads aus der fahrzeugtypspezifischen Adjazenzmatrix	87
5.1.4	Beschreibung der auftragsbezogenen Pfadplanung	90
5.2	Erfassung des Arbeitsraums durch dezentrale Infrastruktursensoren	91
5.2.1	Methode zur Separation von dynamischen und statischen Hindernissen	91
5.2.2	Methode zur Verfolgung dynamischer Hindernisse	93

5.2.3	Übermittlung der dynamischen Objektdaten.....	96
5.2.4	Fusionierung der adaptiven Hintergrundbilder zu einem statischen Arbeitsraum.....	97
5.3	Bereitstellung einer fahrzeugtypspezifischen Routenplanung	98
5.3.1	Komplettierung des Weltmodells	99
5.3.2	Anreicherung des Weltmodells mit Zusatzinformationen	102
5.3.3	Durchführung einer fahrzeugtypspezifischen Routenplanung.....	104
5.4	Fahrzeuginterne Planung des Pfads.....	105
5.4.1	Glättung des globalen Pfads durch Interpolation.....	106
5.4.2	Einbindung lokaler Informationen bei der Pfadausführung...	107
5.5	Zusammenfassung und Beurteilung	109
6	Transfer der intralogistischen Auftragsallokation in eine ereignisdiskrete Simulation	113
6.1	Architektur zur Kopplung eines MAS mit einer DES	114
6.1.1	Einsatzfelder ereignisdiskreter Materialflusssimulationen	114
6.1.2	Lösungsansatz zur Integration des iLES in ereignisdiskrete Materialflusssimulationen.....	115
6.2	Methode zur Einbindung des Allokationsalgorithmus in der Planungsphase.....	116
6.2.1	Einbindung der intralogistischen Ontologie in eine Simulationsumgebung.....	116
6.2.2	Optimierung des Layouts und der Auftragsdefinition anhand von KPIs während der Planungsphase.....	118
6.3	Überführung des Allokationsalgorithmus aus der Planungsphase in die Betriebsphase	118
6.3.1	Transfer der Ontologie auf die physischen Transporteinheiten	119
6.3.2	Kontinuierliche Verbesserung des Simulationsmodells durch die Rückkopplung der operativen Entitäten	120
6.4	Validierung des Simulationsmodells	121
6.4.1	Beschreibung des Simulationsszenarios.....	121
6.4.2	Produktionsszenarien und Bewertung der Simulationsergebnisse	122
6.5	Zusammenfassung und Beurteilung	124
7	Beispielhafte Umsetzung und Validierung des Systemansatzes	127
7.1	Beschreibung der prototypischen Gesamtarchitektur.....	128

7.2	Implementierung des generischen Fahrzeugschichtenmodells auf Zielplattformen	133
7.2.1	Umsetzung des typspezifischen Parameterservers in Kombination mit einer allgemeingültigen Fähigkeitsbeschreibung	135
7.2.2	Analyse der Kommunikation zwischen Zustandsautomaten und Fahrzeugagent	137
7.2.3	Analyse der Auftragsdisposition im Multi-Agenten-System	138
7.3	Umsetzung der verteilten Architektur und der intralogistischen Serviceebene	141
7.3.1	Analyse der sensornahen Informationsseparation	142
7.3.2	Analyse der binären Darstellung des Gesamtarbeitsraums	145
7.3.3	Analyse der globalen Routengenerierung	147
7.3.4	Analyse der fahrzeugindividuellen Pfadausführung	151
7.4	Einbindung der Fahrzeugagenten in die ereignisdiskrete Simulationsumgebung	152
7.4.1	Umsetzung der Schnittstelle zwischen dem Multi-Agenten-System und der ereignisdiskreten Simulation	153
7.4.2	Analyse der Simulationsergebnisse	155
7.5	Zusammenfassung und Beurteilung	156
8	Zusammenfassung und Ausblick	159
9	Summary and Outlook	163
	Literaturverzeichnis	167

Abkürzungsverzeichnis

ACL	Agent Communication Language
AKL	Automatisches Kleinteilelager
AMCL	Agent Management System
autoBod	autonomer Bodenroller
BG	BackGround
BSD	Berkeley Software Distribution
CFP	Call For Proposal
CIE	Commision Internationale de l'Éclairage
CKD	Completely Knocked Down
DES	Discrete Event Simulation
DF	Directory Facilitator
DLL	Dynamic Link Library
DOF	Degree Of Freedom
E Flow	Forschungsprojekt: Energieeffiziente, vielseitige und autonome Transportfahrzeuge für den innerbetrieblichen Materialfluss
ERP	Enterprise Resource Planning
FG	ForeGround
FIPA	Foundation for Intelligent Physical Agents
FMC	Fundamental Modelling Concepts
FPGA	Field Programmable Gate Array
FrMS	Fractal Manufacturing System
FTS	Fahrerloses Transportsystem
FTF	Fahrerloses Transportfahrzeug
GMM	Gaussian Mixture Model
GPL	GNU General Public License
GPS	Global Positioning System
HiL	Hardware-in-the-Loop
HMI	Human Machine Interface
HMS	Holonic Manufacturing System
HSV	Farbraum mit Hue H, Saturation S, Value V
ID	Identifikationsnummer
iLES	Intralogistics Execution System
JADE	Java Agent Development Framework

KARIS Pro	Forschungsprojekt: Kleinskaliges Redundantes Intralogistik-System
KATE	Kleine Automatische Transporteinheit
KLT	Kleinladungsträger
KPI	Key Performance Indicator
LGPL	Lesser General Public License
LiDAR	Light Detection And Ranging
LPR	Local Positioning Radar
MAS	Multi-Agent-System
MES	Manufacturing Execution System
MONAMOVE	Forschungsprojekt: Monitoring and Navigation System for Mobile Vehicles
MQR	Multi-Query-Roadmap
MQT	Multi-Query-Tree
MSL	Major Service Layer
MTS	Message Transport System
NMS	Non-Maximum Supression
PPS	Produktionsplanung und -steuerung
PRM	Probabilistic RoadMap
REST	REpresentational State Transfer
RGB	Farbraum mit Rotanteil R, Grünanteil G, Blauanteil B
RPC	Remate Procedure Call
RRT	Rapidly-exploring Random Tree
SBC	Singel-Board Computer
SIFT	Scale-Invariant Feature Transform
SiL	Software-in-the-Loop
SLAM	Simultaneous Localization and Mapping
SOA	Service Oriented Architecture
SPOF	Singel Point Of Failure
STR	Smart Transport Robot
SURF	Speeded Up Robust Feature
VAV	Versatile Autonomous Vehicle
VTF	Vielseitiges Transportfahrzeug
VTs	Vielseitiges Transportsystem
WSIG	Web Service Integration Gateway

1 Einleitung

Der übergeordnete Forschungsinhalt der vorliegenden Dissertationsschrift ist die Realisierung einer intralogistischen Serviceschicht, die eine Allokation von Transportaufträgen auf einzelne Entitäten sowie eine auftragsindividuelle Routenplanung und autonome Pfadausführung ermöglicht. Dadurch wird der etablierte Prozess der innerbetrieblichen Materialflussplanung sowohl in der Planungs- als auch in der Betriebsphase durch eine Regelung substituiert. Den ersten Forschungsschwerpunkt stellt dabei die Konzeption eines Schichtenmodells der Software der Transportfahrzeuge dar, welches durch ein hybrides Steuerungsparadigma die Basisfähigkeiten der Entitäten zu Hauptfähigkeiten aggregiert und die Grundlage der dezentralen Auftragsallokation darstellt. Den zweiten wesentlichen Forschungsinhalt der Arbeit adressiert die serviceorientierte Systemarchitektur (*service oriented architecture* - SOA) zur Digitalisierung des Arbeitsumfelds, der Routenplanung und Pfadausführung, die einen weiteren elementaren Bestandteil der Softwareebene für intralogistischen Dienste bildet. Der dritte Teil der Forschung umfasst die Integration der intralogistischen Auftragsallokationsmethode in eine ereignisdiskrete Simulation (*discrete event simulation* - DES), die ein virtuelles Abbild der Intralogistik während der Planungs- und Betriebsphase ermöglicht. Diese drei Forschungsschwerpunkte bauen auf einer detaillierten Untersuchung der aktuellen Systemlösungen zur Verteilung von Transportaufträgen auf einzelne Einheiten eines Materialflusssystems, der Digitalisierung des Arbeitsbereichs und den etablierten Verfahren der Systemauslegung in der Planungs- und Betriebsphase auf.

1.1 Motivation und Hintergrund

Methoden zur Optimierung des innerbetrieblichen Material- und Informationsflusses sind heutzutage aus etablierten und ganzheitlichen Produktionssystemen nicht mehr wegzudenken. Zwei Grundpfeiler zur Vermeidung von Verschwendung in einer Produktion stellen gemäß den Gestaltungsprinzipien nach VDI 2870 das Fließprinzip und das Pull-Prinzip dar. Diese beziehen sich unmittelbar auf die intralogistischen Prozesse des Material- und Informationsflusses. Das Fließprinzip zielt darauf ab, den Material- und Informationsfluss innerhalb der Fertigung so zu realisieren, dass er schnell, durchgängig und turbulenzarm ausgeführt ist. Das Pull-Prinzip beschreibt den Ansatz, einen Kundenauftrag gegen den Materialfluss durch den Produktionsprozess zu ziehen und orientiert sich somit an konkreten

Aufträgen. Die Kernbotschaft einer schlanken Produktionsgestaltung lässt sich daher so zusammenfassen, dass das Material vom Produktionsbeginn bis zur Fertigstellung des Produktes stets weiterbearbeitet wird, was bedeutet, dass nach Möglichkeit keine Puffermengen zwischen den einzelnen Prozessschritten auftreten. Dies führt unmittelbar zu einer kürzest möglichen Durchlaufzeit mit positiven Effekten auf die Liefertreue und Kapitalbindung im Wertstrom. [1]

Der Grundsatz der schlanken Produktion, das Material während des gesamten Wertschöpfungsprozesses stets im Fluss zu halten, führt bei physischer Realisierung des Materialflusses, vor allem bei Massenprodukten, zu prozessfolgeorientierten Fertigungsstrukturen. Hierbei sind die Betriebsmittel gemäß der Abfolge der notwendigen Prozessschritte von gleichen oder ähnlich zu bearbeitenden Bauteilen, Baugruppen oder Erzeugnissen zusammengefasst. Bei klassischen Transferstraßen und der Fließfertigung ist der Arbeitsfortschritt zeitlich gebunden und eine Änderung der Prozessfolge zwischen den unterschiedlichen Erzeugnissen nicht möglich. Alle prozessfolgeorientierten Strukturen haben gemein, dass ein gerichteter, übersichtlicher Materialfluss entsteht und sie sich durch eine hohe Produktivität auszeichnen. Im Vergleich zu verrichtungsorientierten Fertigungsstrukturen ist jedoch aufgrund der spezialisierten Fertigungseinrichtungen eine geringere Flexibilität des Gesamtsystems einhergehend mit hohen Umstellkosten bei Produktwechseln in Kauf zu nehmen. [2–4]

Als aktuelle Entwicklungstreiber der Logistik im Allgemeinen und der Intralogistik im Speziellen gilt die Globalisierung des Wettbewerbs mit der zunehmenden Marktdynamik, sowie der Wandel von Verkäufer- zu Käufermärkten mit den einhergehenden steigenden Kundenerwartungen. Die Reaktion der Unternehmen auf diese anhaltenden Entwicklungen ist die Konzentration auf Kernkompetenzen mit einer reduzierten Fertigungs- und Wertschöpfungstiefe. Dies erfordert ein ganzheitliches Management entlang der gesamten Wertschöpfungskette, das als Ziel die Optimierung des Gesamtsystems aus Material- und Informationsfluss hat, sowie eine maximale Flexibilität und Dynamik der intralogistischen Prozesse gewährleistet. [5, 6]

Dieser Megatrend zu kundenindividuellen Massenprodukten (Mass Customization) stellt die Synthese zwischen einer Massenproduktion und der prozessfolgeorientierten Fertigung in Frage. Vor allem dann, wenn der kundenindividuelle Nutzen nicht durch eine reine Softwareanpassung erreicht wird, sondern durch Eigenschaften definiert ist, die den Arbeitsplan des Produkts und die Auswahl der Betriebsmittel beeinflussen. Zum einen

stellen diese Produkte weiterhin Massenprodukte dar, wodurch die Vorteile der prozessfolgeorientierten Fertigungssysteme, vor allem die geringe Durchlaufzeit und dadurch schnelle Verfügbarkeit für den Kunden sowie die hohe Produktivität und die dadurch erreichbaren Zielkosten, zum Tragen kommen. Andererseits erfordern kundenindividuelle Produkte mit spezifischen Arbeitsplänen eine hohe Wandlungsfähigkeit des gesamten Produktionssystems und einen individuell angepassten Materialfluss. Die inflexible, starre Verkettung der Betriebsmittel prozessfolgeorientierter, hochproduktiver Fertigungssysteme muss daher durch ein wandelbares Materialflusssystem abgelöst werden. Die notwendige Prozessfolge der einzelnen Betriebsmittel kann dadurch produktspezifisch ausgebildet werden. Durch die flexible Vernetzung wirken sich Störungen einzelner Anlagen nicht auf das Gesamtsystem aus, da der geplante Materialfluss auf Basis der zur Verfügung stehenden Ressourcen während der Durchführung angepasst werden kann. Vorteilhaft ist zudem, dass sich eine bessere Auslastung der Fertigungsmittel gegenüber der starren Verkettung ergibt, da sich die Verfügbarkeit der Anlagen, im Gegensatz zu prozessfolgeorientierten Systemen, nicht auf die gesamte Produktionslinie auswirken. [7, 8]

Die technischen Befähiger zur flexiblen Verknüpfung einzelner Betriebsmittel und der Ausbildung eines intralogistischen Produktionsnetzwerks stellen Fahrerlose Transportsysteme (FTS) dar, die eine Kombination mehrerer Fahrerloser Transportfahrzeuge (FTF) und einer zentralen Leitsteuerung darstellen. Die Entwicklung der FTS lässt sich in vier Epochen unterteilen [9]. Bereits in den 1950er Jahren wurden die Vorteile der FTS erkannt. Jedoch basierten diese frühen Systeme auf einfachsten Spurfolgetechniken mit Schienensystemen und taktilen Sensoren zur Kollisionsvermeidung. Das Einsatzgebiet erstreckte sich daher lediglich auf die unflexible Überbrückung großer Strecken. Während der Automatisierungseuphorie in den 1970er und 80er Jahren erfolgte die Einführung der induktiven Spurführung über einen im Boden eingelassenen Draht. Die Umsetzung von Datenübertragungssystemen zwischen der Leitsteuerung und einzelner Fahrzeuge ermöglichte es, Abzweige- und Haltevorgänge an weiterhin starr verlegten Leitlinie zu realisieren. Dadurch konnten innerhalb eines Materialflusssystems verschiedene Quellen und Senken miteinander verbunden werden und es bildeten sich große FTS aus. Die darauffolgende dritte Epoche adressierte die Restriktion des physischen Leitliniensystems. Im Gegensatz zu den induktiven oder optischen Spurführungssystemen kommen seitdem auch flexiblere virtuelle Leitlinien zum Einsatz. Diese Leitlinien sind lediglich im Leitsystem hinterlegt, wodurch diese ohne physische Eingriffe an sich ändernde Rahmenbedingungen angepasst werden können. Hierzu ist

eine exakte Positionsbestimmung der FTF an jedem Ort des Arbeitsraums notwendig. Durchgesetzt haben sich insbesondere die Lasernavigation und die Magnetrasternavigation. Nach [9] ist die aktuelle, vierte Epoche davon geprägt, dass intelligente Sensorsysteme bei FTS Einzug halten und damit ein autonomer Betrieb der Fahrzeuge ermöglicht wird, da diese auf Hindernisse und Veränderungen ihrer Umgebung reagieren können. Eine manuelle Erstellung sämtlicher Verknüpfungen zwischen Materialquellen und Senken und die Vordefinition der möglichen Fahrrouten am Leitsystem ist somit nicht mehr notwendig. Zudem reagieren die Fahrzeuge dieser Systeme auf nicht planbare Hindernisse innerhalb der errechneten Fahrstrecke und die Pfadplanung wird entsprechend angepasst. [10, 9]

Dieser hohe Freiheitsgrad befähigt FTF nicht nur dazu, einzelne Betriebsmittel zu einem Produktionsnetzwerk zu verbinden, sondern auch die Bereitstellung und Zuführung von Verbrauchs- und Verbaumaterial zu übernehmen. Vor allem im Montagebereich von Massenprodukten erfolgt der Wiederbeschaffungsprozess bei etablierten prozessfolgeorientierten Produktionssystemen überwiegend mit zyklisch verkehrenden Routenzügen. Diese verbinden die Materialquellen wie Vormontage- und Vorfertigungsbereiche oder das Wareneingangslager mit fertigungsnahen Lagerstufen. Die Versorgung dieser Lagerstufen erfolgt bei schlanken Produktionssystemen zwar bedarfsorientiert über Kanban-Systeme, die Fahrzyklen der Routenzüge sind allerdings einem vorher definierten Fahrplan unterworfen. Die Wiederbeschaffung ist somit nicht mit dem direkten Bedarf der Fertigungseinrichtungen synchronisiert. Dies macht sich vor allem durch eine stark schwankende Auslastung der eingesetzten Routenzüge und die Größe der Zwischenlagerstufen bemerkbar. [11]

Vielseitig einsetzbare und autonome FTF (*versatile autonomous vehicle* - VAV) versprechen diese Prozesse effizienter und vor allem ressourcenschonender abzubilden. Die höhere Anzahl an notwendigen Transporteinheiten im Vergleich zum zyklischen Routenzug und der damit verbundene Energiebedarf werden durch die Steigerung des Auslastungsgrads der Fahrzeuge und der Reduzierung der zurückgelegten Wege überkompensiert. Da die geringeren Abmessungen der Transportfahrzeuge eine direkte Belieferung der Produktionsanlage ermöglichen, entstehen weitere positive Effekte wie beispielsweise der Entfall der fertigungsnahen Lagerstufen. Dadurch werden manuelle Handlungsschritte eingespart und der Flächenbedarf der Logistik in der Produktion reduziert, was mit einer Erhöhung der Flächenproduktivität und Verringerung des Energieeinsatzes pro Produkt einhergeht. [11]

Moderne Applikationen von VAV adressieren diese Anwendungsfelder bereits, sind aber aus dem Automatisierung- und Autonomieansatz getrieben, welcher das einzelne Fahrzeug oder das spezifische System befähigt, ein Modell seiner Umwelt zu generieren. Die Fahrzeuge werden mit einer Vielzahl an Sensoren ausgestattet, um selbstständig oder im Verbund das Weltmodell zu erstellen. Dieses weist jedoch aufgrund der beschränkten Reichweite der Sensoren Schwächen auf und stellt ein Abbild von historischen Umgebungsbedingungen dar. Die autonome Routenplanung der Fahrzeuge erfolgt somit auf Basis eines veralteten Weltmodells. Ein weiterer Lösungsansatz verwendet kostengünstig Fahrzeuge, die mit zentral bereitgestellten Umgebungsdaten versorgt werden. Dadurch ergibt sich eine maximale Flexibilität im Hinblick auf Veränderungen im Betriebsumfeld, da bei einem geeigneten Sensorkonzept den Fahrzeugen zur Pfadausführung stets ein aktuelles Weltmodell zur Verfügung steht. [12]

Aus dem Ansatz der Autonomie und Automatisierung des Einzelsystems ergibt sich zudem eine konzeptionelle Schwäche der aktuellen Lösungsansätze. Die Fahrzeuge sind auf spezifische, im Vorfeld definierte Anwendungsfälle konzipiert und damit auch die Fähigkeiten des Systems an diese Restriktionen geknüpft. Die Interpretation der Sensordaten zur Digitalisierung des Betriebsumfelds und Generierung des Weltmodells sind nicht auf andere Anwendungsfelder oder Fahrzeugtypen übertragbar. Weiterhin erfolgt die Vergabe der Transportaufträge auf die einzelnen Entitäten entweder über eine Leitsteuerung oder innerhalb eines proprietären Verteilmechanismus. Eine dezentrale systemübergreifende Auftragseinlastung ist nicht gegeben. Die Ansätze stellen somit Insellösungen dar, die eine ganzheitliche intralogistische Wertstromlösung nicht adressieren und die Anforderung an zukünftige Produktionssysteme, ein flexibles und wandelbares intralogistisches Produktionsnetzwerk auszubilden, nicht erfüllen.

1.2 Zielsetzung

Die Zielsetzung dieser Arbeit ist daher die elementaren Softwaremodule zur Auftragsverteilung und Weltmodellerstellung von intralogistischen, hochflexiblen und autonomen Materialflusssystemen von der Konstruktion und Auslegung der physischen Transportentitäten zu entkoppeln. Der Fokus liegt auf der Definition und Etablierung einer intralogistischen Serviceebene, die dem Produktionsnetzwerk ein *Intralogistics Execution System* (iLES) zur Verfügung stellt.

Ein zentrales Ziel ist die Erforschung einer Methode zur Allokation und Abarbeitung von Transportaufträgen innerhalb eines intralogistischen

Transportnetzwerks. Hierfür ist eine Methode zu erforschen und zu entwickeln, die die Grundfähigkeiten von Transportentitäten auf Basis der verbauten Sensoren und Aktoren generisch beschreibt. Die Applikation auf jegliches spezifische Zielsystem ist daher über eine standardisierte Parametrisierung gewährleistet, ohne dass die elementaren Grundfunktionen wie die Routenplanung oder die Ortung im Arbeitsraum adaptiert werden müssen. Somit kann auch eine übergeordnete Aggregation dieser Basisfähigkeiten in einem Zustandsautomaten erfolgen, um komplexe Fähigkeiten einer Transporteinheit allgemeingültig zu beschreiben. Eine weitere Forschungsaufgabe ist die Modellierung von systemunabhängigen Hauptfähigkeiten, die dem Softwareagenten der Transportentität ermöglichen, diese in einem Multi-Agenten-System (MAS) bereitzustellen. Im Kontext der dezentralen Auftragsvergabe stellt die Beschreibung des auf logistischen *Key Performance Indicator* (KPI) basierenden Auftragsallokationsalgorithmus des iLES den abschließenden Forschungsbedarf dar (vgl. Bild 1). Durch diese mehrschichtige und generische Methode zur Beschreibung von autonomen und flexiblen Transportsystemen ist eine von der Zielplattform unabhängige und somit systemübergreifende Auftragsvergabe möglich.

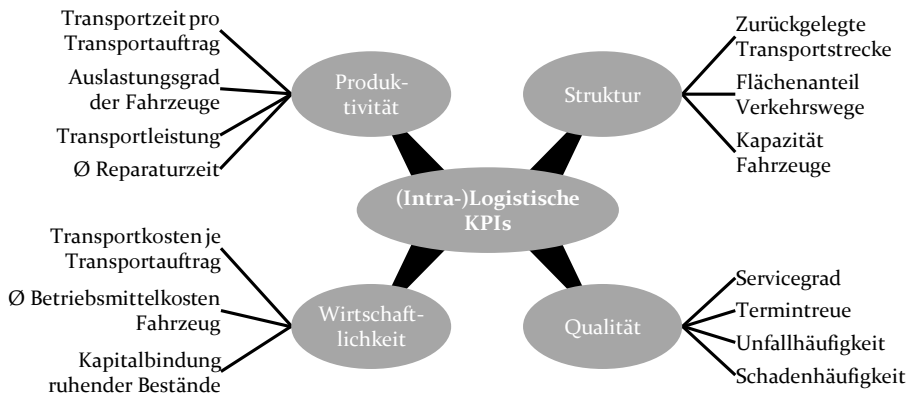


Bild 1: KPI für Materialfluss und Transport nach dem Logistik-Kennzahlensystem von Schulte [13]

Die Funktion der Auftragsvergabe bildet jedoch nur einen elementaren Baustein einer servicebasierten Intralogistikschicht ab. Die Digitalisierung des Arbeitsumfelds zur Routenplanung und Pfadausführung sowie die Selbstortung der Entitäten ist seit dem Entfall der physischen Leitlinien ein fundamentaler Bestandteil aller Systeme. Bei der angestrebten servicebasierten Weltmodellerstellung ist die Aufteilung einzelner Funktionen in einer verteilten Architektur zielführend. Lütteke [14] beschreibt hierzu ein

System, in der Infrastruktursensoren genutzt werden, um für eine Flotte, bestehend aus artgleichen VAV, das Weltmodell zentral zu erzeugen. Die in dieser Arbeit adressierte Forschung führt diesen Ansatz konsequent fort und überführt die elementare Aufgabe der Arbeitsraumdigitalisierung in fahrzeugtypunabhängige Dienste. Der Kernaspekt ist die Entwicklung einer Methode, die die Funktion der notwendigen Dienste beschreibt, ihre Verteilung in der Systemarchitektur adressiert und eine fahrzeugtypübergreifende Verwendung erlaubt.

Insbesondere im Hinblick auf das Forschungsziel einer dezentralen Auftragsallokation durch die Verhandlungsmethoden eines MAS ergibt sich eine dritte, maßgebliche Forschungsaufgabe. Im Gegensatz zur klassischen Fertigungsplanung ist die Auswirkung von diskreten Störeinflüssen auf das MAS bei steigender Systemkomplexität nicht vorhersagbar. Die Reaktion jeder einzelnen Entität der Auftragsvergabe ist zwar deterministisch, jedoch steigt mit zunehmender Anzahl an Verhandlungspartnern die Anzahl der Einzelentscheidungen und deren Wechselwirkungen. Das Ziel ist daher eine Methode zu erforschen und umzusetzen, die die Ontologie der beteiligten Entitäten des intralogistischen Auftragsallokationsmechanismus in eine DES integriert. Hindurch entsteht ein virtuelles Abbild der Fertigungs-umgebung, mit dem während der Planungsphase die Strategie der Verhandlungsteilnehmer hinsichtlich des globalen Produktionsziels optimiert werden kann. Zudem erlaubt dieser digitale Zwilling in der anschließenden Betriebsphase die Reaktion des Materialflusssystems auf diskrete Störeinflüsse und dessen Auswirkungen vorherzusagen. Dadurch kann die Strategie einzelner Teilnehmer im laufenden Betrieb gezielt beeinflusst werden.

1.3 Vorgehensweise und Aufbau der Arbeit

Nach einer kurzen Einführung in die Grundsätze etablierter Fertigungssysteme und der Gegenüberstellung der Eigenschaften gängiger Gestaltungsformen der Prozessanordnung, werden im Kapitel 2 die wissenschaftlichen Arbeiten und Ansätze zur Flexibilisierung des intralogistischen Wertstroms unter Verwendung neuer Organisationsstrukturen beschrieben.

Ein übergreifender Bestandteil dieser neuartigen Organisationsstrukturen ist die Umsetzung eines frei konfigurierbaren Produktionsnetzwerkes, bei dem die vorhandenen Betriebsmittel in jeglicher Reihenfolge im Arbeitsplan eines Produkts enthalten sein können. Die wesentliche Voraussetzung hierfür ist die Verwendung von flexiblen und autonomen Einheiten für den innerbetrieblichen Materialtransport zwischen den Betriebsmitteln. Kapi-

tel 3 gibt daher einen Überblick über die bereits durchgeführten Forschungen, Entwicklungen und Applikationen von FTS. Im Fokus stehen hier die elementaren Funktionen der Auftragszuordnung und -bearbeitung sowie die Methoden zur Erfassung des Arbeitsraums und der Fahrzeugpose. Zudem gibt dieses Kapitel einen Einblick auf aktuelle kommerzielle und wissenschaftliche Ausführungsformen von FTS und bildet somit die Basis für die Formulierung der adressierten Forschungsfragen.

Ein wesentlicher Forschungsbereich ist eine systemübergreifende und damit generische Beschreibung der Fähigkeiten intralogistischer, mobiler Transportroboter, die die Basis für die Umsetzung eines dezentralen Auftragsallokationsmechanismus bildet. Als Lösungsansatz wird in Kapitel 4 die erforschte Methode eines allgemeingültigen Softwareschichtenmodells der Transportroboter thematisiert. Durch diese werden die Grundfähigkeiten der Fahrzeuge unter Verwendung des hybriden Steuerungsparadigmas in einem Zustandsautomaten zu Hauptfähigkeiten aggregiert. Das Schichtenmodell ist dabei auf kein spezifisches System abgestimmt, sondern zielt darauf ab, auf unterschiedlichste Entitäten appliziert werden zu können. Die Methode erfüllt somit die zusätzliche Forderung, unterschiedliche Fahrzeugtypen über eine einheitliche Softwarearchitektur in ein gemeinsames MAS einzubinden und dadurch die Allokation von Transportaufträgen über ein iLES abzubilden.

Kapitel 5 adressiert den Forschungsbedarf, die Digitalisierung des Arbeitsraums sowie die Routenplanung und Pfadausführung als fundamentale Bestandteile eines hochflexiblen intralogistischen Transportsystems in eine dienstbasierte Serviceebene zu überführen. Die entwickelte Methode zielt darauf ab, die notwendigen Algorithmen in einer SOA abzubilden, um die Skalierbarkeit des Gesamtsystems zu gewährleisten. Ein Schwerpunkt ist, die Hindernisse im erfassten Arbeitsraum der verteilten Infrastruktursensoren in dynamische und statische Objekte zu differenzieren. Der Sensor wird durch eine integrierte Auswerteeinheit befähigt, Dienste bereitzustellen, die diese Informationen zur weiteren Verarbeitung anbieten. Die übergeordnete Serviceebene selbst verfügt schließlich über Dienste, die die bereitgestellten und vorverarbeiteten Informationen der Infrastruktursensoren zusammenführen, mit Restriktionen wie Hauptverkehrswegen und Sperrflächen anreichern, um schließlich einen Streckenkartenservice für die Transportroboter anzubieten. Auf den Fahrzeugen erfolgt, unter Einbindung von lokalen Arbeitsrauminformationen, anhand dieser Streckenkarten die auftragsspezifische Pfadplanung und -ausführung. Im Fo-

kus der Forschungsarbeit stehen neben der Skalierbarkeit des Arbeitsraums auch die Skalierbarkeit der Größe der Fahrzeugflotte sowie eine Variation der Fahrzeugtypen abbilden zu können.

Ein vielversprechender Ansatz um Gesamtsystemverhalten bei der Auftragsallokation eines iLES auf Agentenbasis transparenter und vorhersagbar zu gestalten, ist die Erstellung eines digitalen Abbilds des Systems in einer ereignisdiskreten Simulationsumgebung. Kapitel 6 adressiert daher eine Methode zur Integration und Optimierung der Ontologie zur Auftragsvergabe. Zudem beschreibt der Ansatz die kontinuierliche Verbesserung des bestehenden Modells während der Betriebsphase durch die Integration der Betriebskennwerte der Transportentitäten.

Die erforschten Methoden zur Auftragsallokation, Arbeitsraumdigitalisierung und der Schaffung eines digitalen Systemabbilds werden exemplarisch in einem Versuchsszenario umgesetzt. In Kapitel 7 wird die Implementierung der einzelnen Dienste in vorhandene Softwareframeworks und -tools beschrieben und die Applikation auf Prototypen der Systementitäten dargestellt. Im Anschluss erfolgt eine Validierung der Methoden und eine Evaluation des exemplarischen Gesamtsystems.

Kapitel 8 fasst die Arbeit abschließend zusammen und gibt ausblickende Hinweise auf mögliche Folgearbeiten.

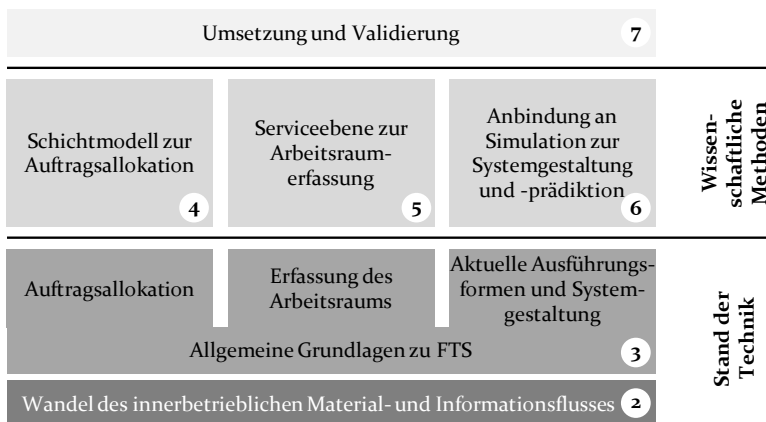


Bild 2: Aufbau der Arbeit mit Kapitelnummern

2 **Gestaltungsformen innerbetrieblicher Material- und Informationsflüsse**

Die stetig steigende Nachfrage nach individualisierten Produkten führt dazu, dass die Anzahl der Varianten bei Massenprodukten sukzessive zunimmt, wie das Beispiel der Automobilindustrie zeigt. Hier konnte die produzierte Stückzahl mit der Einführung des Fließbands signifikant gesteigert werden, was zu einer Reduktion der relativen Herstellkosten führte und das Automobil vom Luxusgut zum Massenprodukt entwickelte [15]. Der Trend, dass die Stückzahl je Variante steigt, ist jedoch rückläufig [16], was auf die Variantensteigerung der Hersteller [17] zurückzuführen ist. Im Anschluss an die Deckung des Grundbedürfnisses nach autonomer Mobilität verlangte der Markt eine Individualisierung der Produkte. Analog zur Automobilindustrie ist dieser Trend bei allen Endkonsumentenprodukten der entwickelten Länder zu beobachten und wird in der Literatur als *Mass Customization* [18] bezeichnet. Es beschreibt die Herstellung individueller Produkte, die den Anforderungen und Wünschen des Kunden folgen. Die Fertigung erfolgt nach den Grundsätzen der Massenproduktion, um deren Vorteile wie Skaleneffekte, Qualität und Automatisierung zu nutzen. Das Ziel ist folglich die Produktion von Unikaten zu den Kosten eines Massenprodukts. [19]

Neben der *Soft Customization*, bei der die Individualisierung nach der Fertigung erfolgt, besteht ein zunehmender Trend zur *Hard Customization*, bei der auch der Herstellungsprozess involviert ist [18]. Folglich stoßen die etablierten Ansätze der Organisations- und Informationsstruktur der Produktionswirtschaft an ihre Grenzen. [20]

Abschnitt 2.1 zeigt die spezifischen Vor- und Nachteile der etablierten Formen des Material- und Informationsflusses auf. Der darauffolgende Abschnitt 2.2 adressiert die Ausprägungen aktueller wissenschaftlicher Organisationsstrukturen, die eine Antwort auf den Trend der *Mass Customization* liefern sollen. Abgerundet wird das Kapitel durch Abschnitt 2.3, der eine Schlussfolgerung aus den Erkenntnissen in Bezug auf die intralogistischen Materialflusssysteme zieht.

Teile der nachfolgend dargestellten Ergebnisse sind angelehnt an die am Lehrstuhl FAPS entstandenen und vom Autor angestoßenen studentischen

Arbeiten [S₁–S₁₀]. Bestandteile der Erkenntnisse wurden der internationalen Fachwelt auch in den vom Autor erstellten Veröffentlichungen [P₁, P₂] zugänglich gemacht.

2.1 Etablierte Formen des Material- und Informationsflusses

Die etablierten Ausprägungsformen innerbetrieblicher Organisationsstrukturen lassen sich durch zwei diametrale Produktionstypen unterscheiden: die verrichtungsorientierte und die prozessfolgeorientierte Struktur. Beide Organisationsformen haben spezifische Vor- und Nachteile, wodurch ihre Ausprägung in bestimmten Einsatzbereichen erkennbar sind.

2.1.1 Verrichtungsorientierte Produktionsorganisation

In Betrieben, bei denen kundenorientierte Einzel- und Kleinserienfertigungen abgebildet werden müssen, dominieren überwiegend verrichtungsorientierte Organisationsstrukturen mit der Werkstattfertigung als ihre wichtigste Erscheinungsform. Hierbei sind die Betriebsmittel nach dem technologischen Verfahren gruppiert und die wechselnden Produktsortimente durchlaufen zur Bearbeitung diese Werkstätten anhand individueller Materialflüsse (vgl. Bild 3).

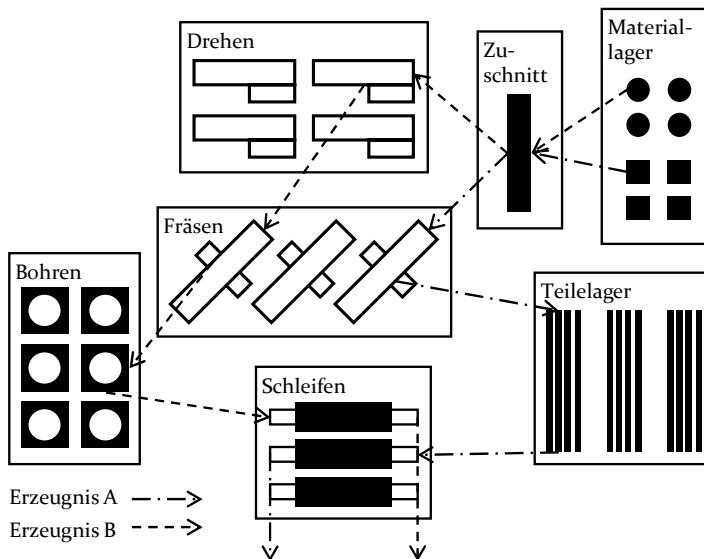


Bild 3: Verrichtungsorientierte Organisationsform am Beispiel einer Werkstattfertigung (in Anlehnung an [2])

Der wesentliche Vorteil dieser Organisationform besteht in ihrer hohen Flexibilität hinsichtlich wechselnder Produktvarianten und -typen sowie Produktionsmengen. Unter den technologisch ähnlichen Betriebsmitteln ist zudem eine nivellierte Auslastung gut steuerbar und hohe Transparenz hinsichtlich Gesamtauslastung des Personals und der Anlagen gewährleistet. Weiterhin stellen die Bewältigung von Auftragsspitzen und die erleichterte Beseitigung von Störungen durch Kapazitätsreserven sowie das hochqualifizierte, disponibel einsetzbare Personal einen Vorteil dieser Produktionstypen dar. [21, 22]

Nachteilig wirken sich allerdings die weit streuenden und langen Durchlaufzeiten der Produkte in Verbindung mit hohen Beständen zwischen den Werkstätten aus. Dies beruht auf dem hohen Anteil intralogistischer Transport- und vor allem Warte- und Liegezeiten die ca. 80 - 90% der Durchlaufzeit [23] betragen und in einer hohen Kapitalbindung resultieren. [24]

2.1.2 Prozessfolgeorientierte Produktionsorganisation

Prozessfolgeorientierte Produktionstypen haben ihr Hauptanwendungsgebiet in der Herstellung von Massenprodukten. Die Betriebsmittel sind dabei entsprechend der Reihenfolge der Bearbeitungsschritte angeordnet und mitunter stark auf die spezifische Aufgabe spezialisiert. Gemäß der Bild 4 gibt es zwei Produktionskonzepte, die Fließfertigung und die Gruppenfertigung. Während bei ersterer nur gleiche Prozessfolgen zugelassen sind, können bei der Gruppenfertigung eine begrenzt variable Prozessfolge realisiert werden. [25]

Der wesentliche Vorteil dieser Organisationsform liegt vor allem in der hohen Produktivität durch den Einsatz spezialisierter Betriebsmittel, was geringe Produktionsstückkosten zur Folge hat. Die geringen Transportwege und die signifikante Reduktion der Zwischenlagerstufen senken die Durchlaufzeit und verringern die Kapitalbindung. Zudem führt der linearisierte Materialfluss zu einer hohen Übersichtlichkeit des Produktionsgeschehens und vereinfacht die Steuerung. [26]

Nachteilig wirken sich jedoch die hohen Kosten der Produktionsvorbereitung und die relativ geringe Flexibilität des Systems, einhergehend mit hohen Umstellkosten, aus. Bei einer starren Verkettung der Produktionsanlagen entstehen zudem eine Reduzierung der Gesamtverfügbarkeit des Systems und hohe Fixkosten aufgrund der spezialisierten Produktionskapazitäten. Weiterhin besteht der Zwang, eine vorgegebene Taktzeit auch über unterschiedliche Varianten hinweg einzuhalten [27]

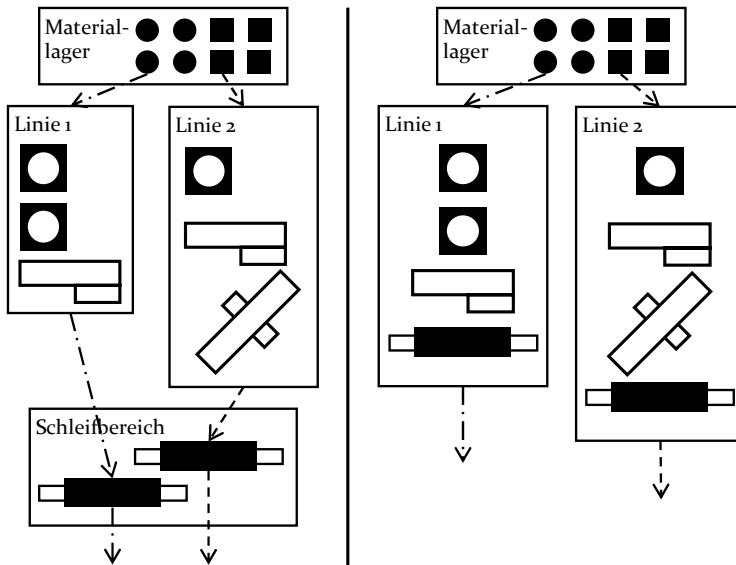


Bild 4: Prozessfolgeorientierte Organisationsform (in Anlehnung an [2]), links: Gruppenfertigung, rechts: Fließfertigung

2.2 Ausprägungen wissenschaftlicher Organisationsstrukturen

Der Fokus aktueller wissenschaftlicher Bemühungen liegt daher darauf, eine Antwort auf die Anforderungen aus der kundenindividuellen Massenproduktion mithilfe neuartiger Organisationsstrukturen zu geben. Übergreifend kristallisiert sich heraus, dass der Ansatz zur Verschmelzung der spezifischen Vorteile der prozessfolgeorientierten und der verrichtungsorientierten Struktur zielführend ist. In der wissenschaftlichen Gemeinschaft haben sich die drei grundlegenden Nomenklaturen, die

- agile
- holone
- und fraktale

Organisationsform etabliert, die im Folgenden näher beschrieben sind. [28]

2.2.1 Agile Fabrikstrukturen

Die agile Struktur beschreibt im Allgemeinen ein flexibles und wandlungsfähiges System, das keinem starren und vorgefertigten Konzept folgt, sondern jeden Einzelfall differenziert betrachtet [29]. Dennoch existieren für Unternehmensprozesse Richtlinien, die sich in den vier strategischen, branchenübergreifenden Dimensionen

- Schaffen von Kundenmehrwert
- Kooperation, zum Ausbau der Wettbewerbsfähigkeit
- Organisation, zur Bewältigung von Wandel und Unbeständigkeit
- Nutzung des Einflusses des Menschen und der Informationen als Hebelkraft

ausprägen [30].

Der Produktionsprozess einer agilen Struktur gliedert sich in Entscheidungspunkte, an denen, auf Basis einer Wissens- und Erfahrungsgrundlage, der weitere Ablauf bestimmt wird [31]. Einige Entscheidungspunkte werden bereits im Voraus festgelegt und aggregieren die innerhalb des Prozesses gesammelten Informationen, um über das weitere Vorgehen zu entscheiden. Andere werden durch ungeplante Ereignisse während des Prozesses ausgelöst, auf die reagiert werden muss. Auf diese Weise bildet sich anhand der Einzelentscheidungen eine differenzierte Durchführung des geplanten Prozesses aus.

Im Vergleich zur schlanken Produktion zeigt der Ansatz der agilen Fabrikstruktur das gemeinsame zentrale Ziel der Kundenzufriedenheitssteigerung. Während die schlanke Produktion jedoch durch eine hohe Standardisierung der Prozesse einen Kostenvorteil erzielt, steht die Variantenvielfalt und individuelle Produktgestaltung im Fokus der agilen Produktion. Dies äußert sich vor allem in den Triebfedern beider Systeme. Die schlanke Produktion setzt auf eine kontinuierliche Reduktion von Verschwendung, die agile Produktion auf eine stetige Erhöhung der Systemflexibilität. Die wissenschaftlichen Beschreibungen zur Umsetzung einer agilen Produktion sind eher generischer Natur und im Detail wenig konkret. Die Methoden einer fraktalen oder holonischen Organisationsform können daher als konkrete Ausführungsform eines agilen Ansatzes interpretiert werden.

2.2.2 Fraktale Fabrikstrukturen

Ein Fraktal beschreibt gebrochene Strukturen, die sich wiederholen und dem Ganzen oder dem kleinsten Teil stets ähnlich sind [32]. Die Selbstähnlichkeit bedeutet, dass alle Objekte eine Menge ähnlicher Komponenten enthalten und sich damit eine Reihe von Zielen und Visionen teilen [33]. Die Form des Ganzen und die darin beschriebenen Strukturen organisieren sich autonom und sind veränderlich [34].

Im Vordergrund einer fraktalen Fabrik steht die Schaffung von sich selbst regelnden Einheiten, die jedoch als ein Ganzes funktionieren. Hierzu ist das *Fractal Manufacturing System* (FrMS) in einer hierarchischen Struktur aufgebaut, in der sich die einzelnen Fraktale jeder Ebene wiederum aus

Fraktalen zusammensetzen (vgl. Bild 5) [35]. Die Fraktale sind dabei immer von unten nach oben strukturiert, wobei die höheren Ebenen diejenigen Aufgaben übernehmen, die von Fraktalen niedrigerer Ordnung aufgrund fehlender Informationen nicht erfüllt werden können. Die Fraktale stehen in einer dynamischen Dienstleistungsbeziehung, bei dem jedes seine eigenen konkreten Ziele und Aufgabenstellungen besitzt und diese autonom verfolgt [36]. Durch die Eigenschaft dieser Selbstorganisation ist es den Fraktalen möglich, eigene Methoden zur Problemlösung zu wählen und durchzuführen. Ihre Ziele sowie die internen und externen Beziehungen können sie in einem dynamischen Prozess selbst erkennen und formulieren. Die ablaufenden Prozesse befinden sich dabei in ständiger Verbesserung. Das bedeutet, dass sich auch die Gestaltung des Systems an die aktuell vorherrschenden Anforderungen anpasst. [32]

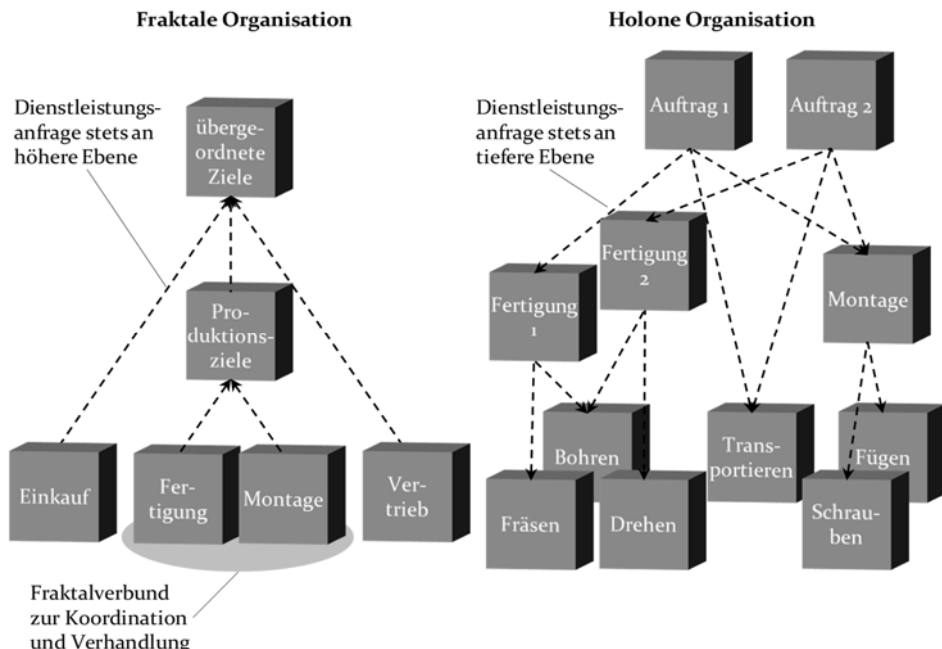


Bild 5: Prinzip eines Fractal Manufacturing Systems (in Anlehnung an [35]) links, Prinzip eines Holonic Manufacturing Systems (in Anlehnung an [36]) rechts

Im Unterschied zur schlanken Produktion, welche die Fokussierung des gesamten Unternehmens direkt auf die Gesamtziele adressiert, unterteilt das fraktale Konzept die Unternehmensziele in Teilziele der Fraktale, um diese besser beherrschen und differenziert erreichen zu können [34]. In der Praxis wird dieser Ansatz für Unternehmensbereiche und Abteilungen in Unternehmen eingesetzt, bei der die einzelnen Strukturen eigene

Profitcenter darstellen. Gemäß Bild 5 zielen diese auf einen eigenen betriebswirtschaftlich Erfolg ab. Höhere Instanzen werden lediglich hinzugezogen, falls ein Informationsdefizit bei den Fraktalen niedrigerer Ordnung herrscht. Ein Transfer dieses Ansatzes auf Werkstatt oder Maschinenebene ist bislang nicht erfolgt.

2.2.3 Holone Fabrikstrukturen

Ein Holon beschreibt halbautonome Einheiten, die in einer Hierarchie miteinander verbunden sind, um ein Ganzes zu bilden [37]. In dieser Holarchie arbeiten die einzelnen Holone zusammen, indem sie ihre individuellen Fähigkeiten und ihr Wissen kombinieren, um die Systemziele zu erreichen. Jedes einzelne Holon kann dabei dynamisch und parallel zu mehreren Holarchien gehören und damit seine Autonomie und Individualität bewahren. Diese Systematik (vgl. Bild 5) wird als *Holonic Manufacturing System* (HMS) beschrieben [38].

Die Koordination und die selbstständige Organisation der Holone wird durch dynamische virtuelle Cluster realisiert. Ein Cluster existiert während die Kooperationsaufgabe aktiv ist und beendet sich, sobald diese Aufgabe abgeschlossen ist [39]. Besonders charakteristisch für die Ausbildung von Holarchien ist ihre Rekursivität. Das heißt, ein Holon kann einerseits selbst aus mehreren Holonen zusammengesetzt sein und damit eine Holarchie darstellen und andererseits selbst Teil einer Holarchie sein, innerhalb der es eine autonome und kooperative Einheit bildet. [40]

Im Gegensatz zur fraktalen Struktur, die eine gleichzeitige und iterative Zielkoordination der unterschiedlichen Ebenen vorsieht, erfolgt die Koordination innerhalb einer Holarchie *top-down* anhand unvollständiger Pläne und *bottom-up* als Entscheidungen und Leistungen. Zudem können Fraktale ihre Funktionen neu definieren, während Holone sich unter Beibehaltung ihrer Funktionen aufgabenabhängig zusammenschließen. Weiterhin steht eine flexible Strategie zur eigenständigen Funktionserfüllung unter Berücksichtigung der vorgegebenen Richtlinien im Vordergrund, während Fraktale die Erfüllung der individuellen Aufgabe priorisieren.

Gemäß Bild 5 ist bei einem HMS der Transfer auf Werkstatt und Maschinenebene möglich. Die untergeordneten Ebenen stellen ihre Dienste zur Verfügung, die von höheren Ebenen angefragt werden können. Die Kommunikation und Entscheidungsfindung erfolgt technisch z.B. über ein MAS.

2.2.4 Auswirkungen auf den Material- und Informationsfluss

Alle drei Ausprägungsformen zur Flexibilisierung der Organisationsstruktur intralogistischer Wertströme haben gemein, dass der dritten Bedeutung der Logistik nach Klaus [41], dem Management von Flüssen und Prozessen, ein höherer Stellenwert eingeräumt wird. Die dritte Bedeutung der Logistik beschreibt die (Intra-)Logistik als ein Ressourcennetz mit einem Gewebe von Flüssen und Prozessen, die über das Netz gelenkt werden und in denen materielle (Flüsse) und immaterielle Objekte (Prozesse) fließen. Diese gehen über Kanten ein, werden in den Knoten verarbeitet und fließen über die Kanten wieder aus dem System. Dadurch wird eine zeitliche Harmonisierung aller Aktivitäten gemäß der von Kunden und Märkten vorgegebenen Takte erreicht. Zudem ist das Logistiksystem mengenmäßig atmungsfähig und kann sich den Anforderungen der (End-)Kunden anpassen. [41]

Weiterhin greift aufgrund der stochastischen Prozesse in einem flexiblen und rekonfigurierbaren Fertigungsnetz der Ansatz der Warteschlangentheorie zur Berechnung der Bestände im System [42, 43]. Sie bedient sich zur Beschreibung von Fertigungsnetzwerken eines Grundmodells, das aus Bearbeitungsstätten, die über ein oder mehrere parallel arbeitende gleichartige Maschinen bzw. Arbeitsplätze verfügen, und einem Puffer besteht. Die Aufträge treffen einzeln und zu zufälligen Zeitpunkten vor den Bearbeitungsstätten ein. Ein neu ankommender Auftrag wird bearbeitet, sofern mindestens eine Maschine frei ist, andernfalls muss er sich in die Warteschlange einreihen. [42]

Zur expliziten Herleitung der Warteschlangentheorie wird auf die entsprechende Fachliteratur [44] verwiesen. Hervorzuheben ist, dass die Warteschlangentheorie eine zeitdiskrete Betrachtung bedingt und die einzelnen Ereignisse, wie die Initialisierung der Fertigungsaufträge, voneinander unabhängig sind. Für die Wahrscheinlichkeit P_n , dass im eingeschungen Zustand n Bauteile vor einer Fertigungsanlage zur Bearbeitung bereitstehen gilt in Abhängigkeit der Auftragseingangsrate λ und der Bearbeitungsrate μ demnach:

$$P_n = \left(\frac{\lambda}{\mu}\right)^n \cdot \left(1 - \frac{\lambda}{\mu}\right) \quad (1)$$

Der Auslastungsgrad ϱ der Bearbeitungsstätte ist definiert als:

$$\varrho = \frac{\lambda}{\mu} < 1 \quad (2)$$

Für den Erwartungswert $E[X]$ der mittleren Anzahl an Fertigungsaufträgen im System gilt:

$$E[X] = \sum_{n=0}^{\infty} P_n \cdot n = \frac{\lambda}{\mu - \lambda} \quad (3)$$

Der Erwartungswert $L[X]$ der mittleren Anzahl an Fertigungsaufträgen in der Warteschlange, dem Puffer vor der Anlage, ergibt sich zu:

$$L[X] = E[X] - \varrho = \frac{\lambda^2}{\mu(\mu - \lambda)} \quad (4)$$

Gemäß der oben aufgeführten Beziehungen folgt die in Bild 6 dargestellte Charakteristik einer Warteschlange in Bezug auf die Bestände. Hieraus lassen sich zwei markante Gesetzmäßigkeiten für alle Systeme ableiten, die auf der Warteschlangentheorie beruhen:

- Übersteigt die Auftragseingangsrate λ die Bearbeitungsrate μ , werden die Warteschlangen unendlich lang.
- Ab einem (System-)Auslastungsgrad ϱ von über 80 % steigen die Erwartungswerte für die Aufträge im System und den Warteschlangen exponentiell an.

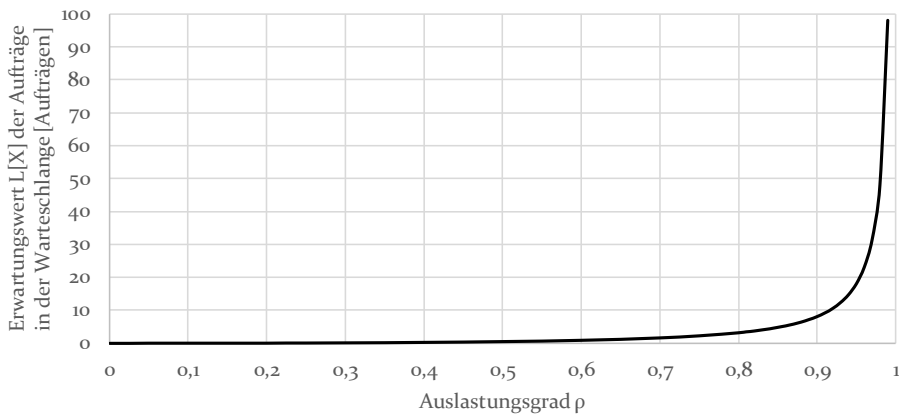


Bild 6: Erwartungswert $L[X]$ der mittleren Anzahl von Fertigungsaufträgen in einer Warteschlange über dem Anlagenauslastungsgrad ρ

2.3 Zusammenfassung und Schlussfolgerung

Durch die ausgereiften Methoden und Gestaltungsprinzipien der schlanken Produktion ist es möglich, hocheffiziente und verschwendungsarme

Fertigungsstrukturen auszulegen und zu betreiben. Bei der Herstellung von Massenprodukten haben sich zudem prozessfolgeorientierte Produktionsstrukturen als zielführend erwiesen. Die systembedingten Nachteile, die mit der Anordnung der verwendeten Betriebsmittel anhand der Abfolge der notwendigen Prozessschritte einhergehen, werden aufgrund der hohen Produktivität eines gerichteten und übersichtlichen Materialflusses in Kauf genommen. Die spezifischen Vorteile einer verrichtungsorientierten Produktionsgestaltung werden aufgrund der hohen Durchlaufzeiten und notwendigen Losgrößen und der damit verbundenen Kapitalbindung im intralogistischen Wertstrom bei der Planung einer Großserienfertigung nicht berücksichtigt.

Die Kombination der etablierten Produktionsformen ist einerseits marktgetrieben, andererseits sind durch die zunehmende Digitalisierung der Fertigung und der Kostendegression im Bereich der Sensorik, Aktorik und SBC (*single board computer*) die technischen Befähiger für eine flexible Gestaltung des Materialflusses ökonomisch verfügbar. Die marktgetriebenen Wandlungsfaktoren fordern eine zunehmende Individualisierung der Produkte und somit einer steigenden Flexibilisierung der hochproduktiven prozessfolgeorientierten Fertigungsstrukturen.

Der Trend zu individualisierten Massenprodukten erfordert daher eine Verschmelzung der spezifischen Vorteile verrichtungs- und prozessfolgeorientierter Systeme in holonen oder fraktalen Fabrikstrukturen, um weiterhin von den Produktivitätsvorteilen einer Massenproduktion profitieren zu können. Die intralogistischen Materialflusssysteme nehmen dabei eine Schlüsselrolle ein. Sollten diese in der Lage sein, Betriebsmittel in einem nach dem Verrichtungsprinzip orientierten Fertigungslayout schnell, kostengünstig und individuell miteinander zu vernetzen, ließen sich die Vorteile beider Organisationsarten kombinieren. In der Wissenschaft werden daher bereits weiterentwickelte Produktionsstrukturen mit dem notwendigen Material- und Informationsfluss erforscht. Die Konzepte und Modellfabriken zu diesen Ansätzen zeigen die Gemeinsamkeit, dass der Materialfluss auf autonome, intelligente und vielseitige Transportmittel fußt und somit einen essentiellen Teil der Realisierung darstellt. [45]

3 Grundlagen autonomer, flexibler intralogistischer Systeme und mobiler Transportroboter

Die Analyse der Hauptanforderung zur Ausbildung hochflexibler und produktiver Fertigungssysteme zeigt, dass eine Verschmelzung der spezifischen Vorteile der verrichtungsorientierten und prozessfolgeorientierten Systeme notwendig ist und dies durch die Verwendung von autonomen, intelligenten und vielseitigen Transportmitteln ermöglicht wird. Am erfolgversprechendsten zeigt sich hierzu der Einsatz von FTS, welche die einzelnen Betriebsmittel im *One-Piece-Flow* verknüpfen und somit ein hochflexibles Produktionsnetzwerk ausbilden. Das folgende Kapitel gibt daher einen Überblick über den aktuellen Stand der Forschung und Technik dieser Befähigertechnologie und zeigt den notwendigen Forschungsbedarf auf.

Abschnitt 3.1 ordnet die FTS im Bereich der unstetigen innerbetrieblichen Fördersysteme ein und beschreibt die typischen Fahrwerksausführungen mit den daraus abgeleiteten charakteristischen Eigenschaften. Der darauffolgende Abschnitt 3.2 adressiert die etablierten Formen der Auftragsallokation und -bearbeitung der am Markt verfügbaren FTS als eine der Schlüsselanforderungen eines flexiblen Materialflusssystems. Die zweite entscheidende Funktion eines autonomen Systems, die Erfassung des Arbeitsraums und der Fahrzeugpose, wird im Abschnitt 3.3 fokussiert. Der Abschnitt 3.4 gibt einen umfassenden Überblick der am Markt verfügbaren und aktuell in der Wissenschaft erforschten Systeme mit FTF und intralogistischen Transportrobotern. Abgerundet wird das Kapitel durch den Abschnitt 3.5, der die aktuellen Technologien zusammenfassend beurteilt und den Handlungsbedarf im Hinblick auf die Anforderungen einer hochflexiblen Fertigungsstruktur formuliert.

Teile der nachfolgend dargestellten Ergebnisse sind angelehnt an die am Lehrstuhl FAPS entstandenen und vom Autor beauftragten studentischen Arbeiten [S11–S15].

3.1 Fahrerlose Transportsysteme als flexible Einheiten der Intralogistik

Die Möglichkeit der freien Verknüpfung der einzelnen Betriebsstätten stellt eine der Hauptanforderungen an ein Materialflusssystem in einem hochflexiblen Produktionsnetzwerk dar. Hierzu gibt es unterschiedliche technologische Ansätze für stetige und unstetige innerbetriebliche Fördersysteme. Weiterhin unterscheiden sich die Bewegungsfreiheitsgrade und daher auch die Einsatzmöglichkeiten etablierter FTF anhand ihrer spezifischen Fahrwerkskinematik.

3.1.1 Einteilung von unstetigen innerbetrieblichen Fördersystemen

Innerbetriebliche Fördersysteme und -mittel sind Maschinen und Anlagen, die zum Transport von Stückgütern oder Schüttgütern verwendet werden. Neben der Unterteilung der Systeme nach dem Fördergut ist es aus logistischer Sicht sinnvoll, eine Unterteilung anhand der Einsatzfelder und dem Kriterium der Beförderungsart vorzunehmen. Es lassen sich demnach folgende Gruppen unterscheiden [46]:

1. Rohrleitungssysteme: Zum Transport von gasförmigen, flüssigen oder festen Massengütern
2. Bandförderanlage: Zum Transport von Massenschüttgütern (Sand, Erz, Kohle) und leichten Schüttgütern (Kisten, Kartons, Kleinladungsträger)
3. Rollen- und Kettenförderer: Zum Transport von schweren Stückgütern (Paletten, Gitterboxen)
4. Fahrzeugsysteme: Beförderung des Transportguts in Transporteinheiten mit eigenem Antrieb auf einem antriebslosen Transportnetz
5. Fördersysteme: Beförderung des Transportguts mit oder ohne Ladungsträger auf einem angetriebenen Transportnetz

Die Systeme von 1 bis 3 zählen zur Art der Stetigförderer, die das Transportgut kontinuierlich bewegen. Unstetigförderer (4 und 5) arbeiten hingegen diskontinuierlich, da auf einen Fördervorgang jeweils ein Vorgang ohne Last folgt. Neben der Unterteilung in Förder- und Fahrzeugsysteme kann eine explizitere Differenzierung nach der Eigenschaft eines flurfreien und flurgebundenen Transportvorgangs vorgenommen werden. Eine zusätzliche Unterscheidung stellt die Einteilung in manuell und automatisch betriebene Lastaufnahmesysteme dar.

Anhand dieser spezifischen Charakteristika können die etablierten Ausführungsformen innerbetrieblicher unstetiger Materialflusssysteme in einer Matrix (Tabelle 1) klassiert werden. Hochflexible Produktionsnetzwerke zur Verknüpfung der spezifischen Vorteile von verrichtungs- und prozessfolgeorientierten Fertigungsstrukturen erfordern vielseitige automatisch betriebene Lastfördermittel. Aufgrund der geforderten Skalierbarkeit und der geringeren Gesamtkosten ist zudem die Verwendung von Systemen auf Basis von Einzelfahrzeugen zielführend. Vor allem die flurgebundenen FTS stellen unter Berücksichtigung dieses Anforderungsprofils das höchste Potenzial als Materialflusssysteme in einem hochflexiblen Produktionsnetzwerk dar.

Nach VDI 2510 [10] sind FTS innerbetriebliche, flurgebundene Fördersysteme mit automatisch gesteuerten Fahrzeugen, deren primäre Aufgabe der Materialtransport, nicht aber der Personentransport ist. Sie werden innerhalb und außerhalb von Gebäuden eingesetzt und bestehen im Wesentlichen aus:

- einem oder mehreren FTF,
- einer Leitsteuerung,
- Einrichtungen zur Standortbestimmung und Lageerfassung,
- Einrichtungen zur Datenübertragung,
- Infrastruktur und peripheren Einrichtungen.

Tabelle 1: Unterteilung innerbetrieblicher unstetiger Materialflusssysteme (in Anlehnung an [47])

		Fahrzeugsysteme		Fördersysteme	
		flurfrei	flurgebunden	flurfrei	flurgebunden
Manuell betriebene Lastmittel	Deckenkran		Handwagen	Hebezeuge mit oder ohne Seil und Kette	Röllchenbahn
	Portalkran		Gabelhubwagen	Winden	
			Frontgabelstapler		
			Schubmaststapler		
			Seitenstapler		
			Schlepper		
			Flurförderer		
Automatisch betriebene Lastmittel	Aufzug		Elektropalettenbahn	Hängebahn und Elektrohängebahn	Rollenbahn
	Flugroboter		Kanalfahrzeug	Schienenförderanlage	Gurtbandförderer
			Shuttle		Kettenförderer
			FTS	Trolleybahn	Unterflurschleppkettenförderer

Ein FTF ist definiert als ein flurgebundenes Fördermittel mit eigenem Fahr-antrieb, das automatisch gesteuert und berührungslos geführt wird. FTF

dienen dem Materialtransport und zwar zum Ziehen oder Tragen von Fördergut mit aktiven oder passiven Lastaufnahmemitteln. [10]

3.1.2 Fahrwerkskinematik etablierter fahrerloser Transportfahrzeuge

Um den FTF eine maximale Beweglichkeit in der Ebene zu gewährleisten, ist anzustreben, alle drei Freiheitsgrade (*Degree of Freedom* – DOF) die sich für ein flurgebundenes System ergeben, zu ermöglichen. Die Translation in zwei lineare unabhängige Bewegungsrichtungen und die Rotation um die Hochachse sollen dabei unabhängig voneinander erzeugt und superpositioniert werden können. Der Ansatz zur Klassierung unterschiedlicher Kinematiken von mobilen Robotern stellt aufgrund der allgemeingültigen Beschreibung ebenfalls eine Methode zur Definition von Fahrwerken etablierter FTF dar [47]. Hierzu wird der Grad der Mobilität δ_m beschrieben, der angibt, in wie viele unabhängige DOF eine ebene Bewegung bei abschließlicher Geschwindigkeitsänderung der Traktionsantriebe ausgeführt werden kann. Die Anzahl der Antriebe ist hierbei beliebig, jedoch berechnet sich deren jeweilige Geschwindigkeit aus den Parametern der vorgegebenen Gesamtbewegung. Zudem wird der Grad der Lenkbarkeit δ_s eingeführt, der sich aus den Lenkwinkeln jedes lenkbaren Rades ergibt und Werte von 0 bis 2 betragen kann. Für alle mobilen Roboter und FTF wird der Grad der Manövrierbarkeit δ_M als die Summe aus δ_m und δ_s gebildet und lässt sich mathematisch wie folgt beschreiben [47]:

$$2 \leq \delta_m + \delta_s \leq 3 \quad (5)$$



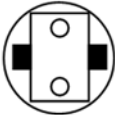







mit

$$1 \leq \delta_m \leq 3; 0 \leq \delta_s \leq 2$$

Aus diesen Beziehungen ergeben sich lediglich fünf unterschiedliche Kombinationsmöglichkeiten der Fahrwerke, die mit ihren charakteristischen Eigenschaften und einer Prinzipskizze in Tabelle 2 aufgeführt sind.

Die Kinematiken mit $\delta_M = 3$ sind allesamt holonom, da sich die Pose des Fahrzeugs in allen drei DOF unabhängig voneinander verändern und superpositionieren lässt. Diese Fahrwerkstypen werden als flächenbeweglich beschrieben [10]. Bei den Fahrwerken der Typen (2,0) und (1,1) hingegen müssen bei der Berechnung der Bewegungsplanung zum Erreichen der Zielpose Rangiervorgänge eingeplant werden. Bei den flächenbeweglichen Kinematiken unterscheidet sich zudem die omnidirektionale Ausführung des Typs (3,0) von den quasi-omnidirektionalen der Typen (2,1) und (1,2).

Tabelle 2: Einteilung unterschiedlicher Fahrwerkskinematiken (in Anlehnung an [48])

	Typ (3,0)	Typ (2,0)	Typ (2,1)	Typ (1,1)	Typ (1,2)
δ_m	3	2	2	1	1
δ_s	0	0	1	1	2
δ_M	3	2	3	2	3
Prinzip- skizze					
Legende	 Mecanum Rad	 festes Antriebsrad	 festes Stützrad	 drehbares Stützrad	 drehbares Antriebsrad

In kommerziellen mobilen Robotern und FTF kommt aufgrund der hohen Robustheit gegenüber unterschiedlichen Bodenbeschaffenheiten und unebenen Untergründen überwiegend der Fahrwerkstyp (1,2) zum Einsatz [48]. Die technische Realisierung ist dabei nicht auf zwei angetriebene Räder beschränkt. Die Systeme besitzen im Allgemeinen vier lenkbare Antriebseinheiten. Aufgrund ihres hohen technischen Aufwands ist bei diesen Systemen jedoch mit höheren Anschaffungskosten zu rechnen [14].

Bei omnidirektionalen Fahrwerken werden zur technischen Realisierung Mecanumräder [49] eingesetzt. Sie verfügen über den höchsten Grad an Mobilität, besitzen allerdings aufgrund ihrer komplexen konstruktiven Umsetzung unterschiedliche Nachteile im praktischen Einsatz. Der Aufstandspunkt je Rolle wandert während einer Umdrehung quer über die Breite des Rads, was zu oszillierenden Biegemomenten und folglich zu Vibrationen bei höheren Geschwindigkeiten führt. Außerdem resultiert die geeignete Anordnung der Rollen auf den Rädern in einer Kraft senkrecht zur Fahrtrichtung. Dies erfordert eine steife Lagerung der Räder und einen hohen Haftreibungskoeffizienten zwischen den Rollen und dem Untergrund, um Schlupf zu vermeiden. Omnidirektionale Fahrwerke sind daher eher für Anwendungsfelder mit geringen Transportgeschwindigkeiten und guten Untergrundbedingungen geeignet. [49, 50]

Für den Einsatz von kostengünstigen FTF empfiehlt sich die Verwendung von Fahrwerken des Typs (2,1). Eine typische Ausführungsform setzt dabei auf ein Differenzialfahrwerk des Typs (2,0) auf und erweitert diesen um eine vertikal drehbare Plattform. Der Vorteil dieser Fahrwerksvariante liegt vor allem in der geringen mechanischen Komplexität. Durch den Einsatz von zwei Fahrantrieben und einem zusätzlichen Antrieb zur Rotation der

Plattform benötigt diese Kinematik lediglich drei Antriebseinheiten für eine holonome Ausführung. [14]

3.2 Auftragsallokation und -zuordnung innerhalb Fahrerloser Transportsysteme

Zur Verteilung von Transportaufträgen in einem FTS kommt in den aktuellen Ausführungsformen eine zentrale Leitsteuerung zum Einsatz. Die Einlastung und Generierung der Aufträge in das System erfolgt über die Anbindung an übergeordnete Instanzen, wie einem *Enterprise Resource Planning* (ERP) oder in Kooperation mit einem *Manufacturing Execution System* (MES). Die Verwaltung der Fahraufträge sowie die Disposition und Abwicklung erfolgt durch die zentrale Leitsteuerung [10].

3.2.1 Auftragsgenerierung und -einlastung

Nach VDI 4451-7 [51] besteht eine Leitsteuerung aus Hard- und Software, deren Hauptbestandteil ein Computerprogramm gemäß Bild 7 ist und auf einem oder mehreren Rechnern läuft. Neben der Koordination der FTF stellt es die Integration des FTS in das innerbetriebliche Materialflusssystem dar.

Die Leitsteuerung ist über die Benutzerschnittstelle mit den Informationssystemen des Anwenders verbunden. Sie verfügt über verschiedene Schnittstellen, um die Aufträge aus dem ERP oder MES entgegenzunehmen oder über ein HMI (*Human Machine Interface*) zur manuellen Auftragseinlastung. Der Auftrag besteht aus Informationen zu mindestens einer Quelle, einer Senke und einer eindeutigen Auftragsnummer. Zudem können weitere Parameter wie die Priorität oder ein notwendiger Endzeitpunkt übermittelt werden. Die interne Materialflussteuerung ist optional und wird nur verwendet, wenn übergeordnete Informationssysteme die Aufträge mit unvollständigen Informationen übermitteln und diese von der Leitsteuerung ergänzt werden müssen. [52]

Die Transportaufträge werden anschließend in drei aufeinanderfolgenden Schritten bearbeitet:

- Transportauftragsverwaltung
- Fahrzeugdisposition
- Fahrauftragsabwicklung

Am Ende dieses Prozesses stehen die spezifischen Fahraufträge, die an die Fahrzeuge übermittelt, von diesen durchgeführt und schließlich an die

Leitsteuerung als erledigt gemeldet werden. Hauptaufgabe der Leitsteuerung ist somit die Lösung des Optimierungsproblems aus minimaler Auftragsbearbeitungszeit bei minimalem Fahrzeugeinsatz, dem klassischen *Traveling-Salesman Problem*, während die FTF lediglich als Automaten zur physischen Durchführung dienen.

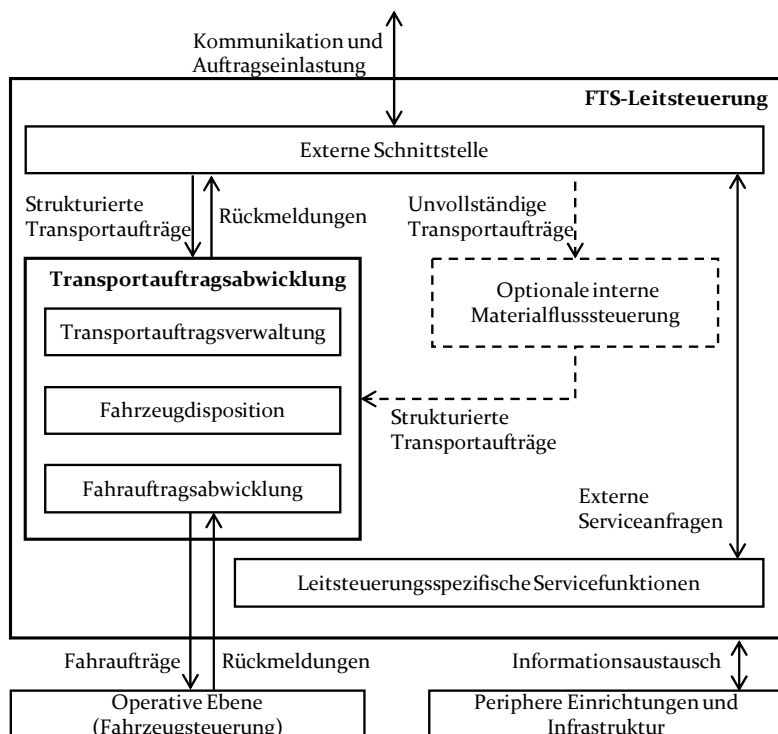


Bild 7: Funktionaler Aufbau einer FTS-Leitsteuerung (in Anlehnung an [52])

3.2.2 Auftragsverwaltung und -zuordnung

Die Abarbeitung der Transportaufträge erfolgt bei FTS in der Reihenfolge ihrer Entstehung. Die Transportauftragsverwaltung stuft die einzelnen Aufträge ein, vergibt eine zeitliche Priorität und prüft fortlaufend ihre Ausführbarkeit. Abschließend wird die Freigabe zur Zuordnung der einzelnen Aufträge auf die FTF erteilt und optional eine Rückmeldung des Status an das übergeordnete einlastende System gegeben. [53]

Der Programmbereich der Fahrzeugdisposition verteilt die generierten und freigegebenen Transportaufträge auf die jeweiligen FTF. In kleinen oder sehr einfachen Strukturen wird hierzu lediglich ein freies Fahrzeug mit der

Bearbeitung des Transports beauftragt. Komplexe Systeme optimieren die Disposition anhand mehrerer Einflussfaktoren und Kennzahlen wie [54]:

- Weg- oder Zeitoptimierung der Anfahrtswege zur Quelle
- Vermeidung von Blockaden und Staus auf dem Wegstreckennetz
- Optimierung des Routenplans
- Reduzierung von Leerfahrten
- Optimierung der Batterieladestrategie und des Energiemanagements

3.2.3 Auftragsbearbeitung und Routenführung

Die Fahrauftragsabwicklung stellt die letzte Instanz der Transportauftragsabwicklung dar und ist direkt mit der Fahrzeugsteuerung verbunden. Es werden auftragsbezogene Sequenzen erzeugt, die Fahrzeugbewegungen und Aktionen beschreiben. Die Abwicklung der Fahraufträge und die Routenführung übernimmt die Verkehrsleitsteuerung. Diese basiert auf etablierten Verfahren aus dem Eisenbahnverkehr und arbeitet mit Blockungsbereichen bzw. Blockungsstrecken. Hierbei wird ein bestimmter Streckenbereich nach der Anforderung und Zuteilung für ein Fahrzeug reserviert, so dass kein weiterer Teilnehmer diesen Bereich befahren kann. Dieses Vorgehen sorgt für eine störungsfreie Fahrt der FTF und verhindert Kollisionen vor allem in Kreuzungsbereichen und Ballungsknoten. [55]

3.3 Ortung, Digitalisierung des Arbeitsraums und Pfadplanung

Neben der Auftragsallokation und -zuordnung auf Systemebene, stellt die Erfassung des Arbeitsraums und der Fahrzeugpose auf Fahrzeugebene eine weitere entscheidende Funktion eines FTS dar. Hierbei gilt es zwei grundlegende Systemansätze zu unterscheiden. Die in Kapitel 3.2 beschriebene, etablierte Infrastruktur von FTS führt dazu, dass sich die Funktion der FTF auf die Durchführung der von der Leitsteuerung verteilten Aufträge reduziert. In Systemen mit einem zentralen Verkehrsleitsystem stellt das einzelne Fahrzeug lediglich einen ausführenden Automaten dar, der die auftragsbezogenen Sequenzen abarbeitet. Hierbei ist es unerheblich, ob die Sequenzen direkt von der Leitsteuerung erzeugt und an die Fahrzeugsteuerung übermittelt werden, oder das FTF anhand der vorgegebenen Bahn des Verkehrsleitsystems die Bewegungsabfolge selbst erstellt. Um eine spezifische Transportaufgabe selbständig ausführen zu können, reduziert sich die Funktion der FTF darauf, ihren Ort und ihre Lage im Arbeitsraum er-

fassen zu können. Für diese Kernaufgabe haben sich in der Entwicklungsgeschichte der FTF verschiedene Technologien zur Spurführung und Standortbestimmung herausgebildet.

3.3.1 Spurführung, Standortbestimmung und Lagerfassung Fahrerloser Transportsysteme

Bei FTS kommen zur Standortermittlung unterschiedlichste Verfahren zum Einsatz, die sowohl auf physischen Leitlinien oder künstlichen Bodenmarkern als auch passiven bodenfreien Markern sowie aktiven bodenfreien Markern aufbauen.

Physische Leitlinien

Die kostengünstigsten Alternativen der Spurführung basieren auf physischen Leitlinien und finden ihr Hauptanwendungsfeld in der getakteten Serienmontage. Je nach Ausführungsform kommen mechanische, optische oder induktive Verfahren zum Einsatz, deren Leitlinien auf den Boden aufgebracht oder in den Boden eingelassen werden. Bei einer aktiv-induktiven Spurführung werden am Fahrzeug zwei Spulen im rechten Winkel zu einem im Boden eingelassenen stromdurchflossenen Leiter angebracht. Die Differenz der in den Spulen induzierten Ströme ist ein Maß für die Abweichung und regelt den Lenkmotor des Fahrzeugs. Bei der passiv-induktiven Ausführung wird ein Metallstreifen oder ein Magnetband auf dem Boden verklebt. Unterhalb des FTF werden zwei bis drei Magnetfeldsensoren angebracht, die anhand der Feldänderung die Kanten des Metallbands detektieren. Die dritte Art der physischen Leitlinien basiert auf einer optischen Leitspur, bei der ein Farbstrich, in Sonderfällen auch ein Textilband, mit deutlichem Kontrast zum Untergrund auf den Boden aufgebracht wird. Ein optischer Sensor am Unterboden des Fahrzeugs erkennt die Abweichung zur Fahrspur. Bei sehr einfachen Layouts mit einer geringen Anzahl an Verzweigungen entsprechen vor allem die passiv-induktiven und die optischen Spurführungsverfahren dem Stand der Technik, da die eingesetzten FTF eine preisgünstige Alternative zu allen anderen Ausführungsformen darstellen. Alle Systeme, die auf physischen Leitlinien aufbauen, haben jedoch gemein, dass sie höchst unflexibel gegenüber Änderungen des Layouts sind. Zur Anpassung, Neugestaltung oder Erweiterung des Arbeitsraums müssen in jedem Fall die Leitlinien neu verlegt werden. [9]

Künstliche Bodenmarker

Ein deutlich höheres Maß an Flexibilität bieten Systeme, die virtuelle Leitlinien benutzen. Da die Fahrzeuge nicht mehr an eine physische Leitlinie

gebunden sind, wird diese Verfahren als freie Navigation bezeichnet. Die Fahrzeuge berechnen zum einen ihre aktuelle Pose anhand von integrierten Sensoren wie Inkrementalencodern an den Antriebsmotoren sowie Beschleunigungs- und Drehratensensoren. Bei allen fahrzeuginternen Sensoren summiert sich jedoch aufgrund der relativen Lagebestimmung mit zunehmender Wegstrecke der Fehler zwischen der realen Position im Raum und der berechneten Pose auf. Somit sind Verfahren der Koppelnavigation notwendig, bei der die relative Fahrzeugpose mit absoluten ortsfesten Peilmarkern fusioniert wird. Dies sind z. B. passive Dauermagneten oder quasi-aktive Transponder, die in regelmäßigen Abständen in den Boden eingelassen werden. Die Verlegung erfolgt dabei entweder als Punktefolge oder als Raster. Ein flächendeckendes Raster über den gesamten Arbeitsraum erlaubt dabei die höchste Flexibilität zur Erstellung der virtuellen Leitlinien, erfordert aber gegenüber der Punktefolge einen deutlich höheren Installationsaufwand. In Außenbereichen werden zur Lagepeilung anstatt der Dauermagneten oftmals quasi-aktive Transponder in den Boden eingelassen. Diese werden durch Leseinheiten unter dem Fahrzeug kurzzeitig mit Energie versorgt und senden eine eindeutige Kennung an das Fahrzeug zurück. Allerdings sind die Systemkosten gegenüber der Varianten mit Dauermagneten deutlich kostenintensiver. Alle Peilungssysteme, die mit künstlichen Bodenmarkern arbeiten, erweisen sich im Hinblick auf die Erstellung von neuen Fahrtrouten gegenüber den physischen Leitlinien flexibler. Jedoch zeigen sie Schwächen, wenn der Arbeitsraum des Systems erweitert werden soll, da dies mit aufwändigen Bodenarbeiten verbunden ist. [9]

Passive bodenfreie Marker

Der prominenteste und weitverbreitetste Vertreter der freien Navigation ist die Lasernavigation. In geeigneter Höhe über den Mitarbeitern und Anlagen werden reflektierende Marker an den Wänden und Säulen angebracht, die mit einem rotierenden Laserscanner auf dem FTF vermessen werden. Dieses System funktioniert über große Entfernungen, erfordert aber freie Sicht auf mindestens zwei bzw. drei Marker zur Positionsbestimmung, je nachdem ob mit oder ohne ein direktes Entfernungsmesssystem gearbeitet wird. Vorteilhaft ist, dass aufgrund der kontinuierlichen Vermessung der Peilmarken die Pose des Fahrzeugs durch ein Absolutmesssystem fortlaufend bestimmt und korrigiert wird. Jedoch eignet sich das Verfahren nicht für kleine FTF, da die freie Sicht über die Anlagen nicht umsetzbar ist. Hier bietet es sich an, den Laserscanner zu schwenken und die Positionsbestimmung anhand künstlicher oder natürlicher Marker an der Decke durchzuführen. Die Sicht an die Decke ist in der Regel frei von

Hindernissen und zudem relativ invariant. Eine weitere Möglichkeit ist die Orientierung anhand natürlicher Landmarken wie Pfeilern oder Wänden. Bei dieser Variante ist es wichtig, dass die Marken deutlich zu erkennen sind und sich weder ihre Position, noch ihre Kontur verändert. [9]

Aktive bodenfreie Marker

Eine weitere Klasse von Systemen zur Standortbestimmung stellen aktive bodenfreie Marker dar. Im Außenbereich ist die Selbstortung durch Funkverfahren, bei der die Laufzeitmessung zu stationären codierten Radarreflektoren oder Satelliten durchgeführt wird, etabliert. Die Ortung über Signale des *Global Positioning System* (GPS) ist in Werkshallen jedoch nicht einsetzbar. In bebauter Umgebung und in Gebäuden wird mit stationären Funkbaken ein *Local Positioning Radar* (LPR) aufgebaut. Auch hier ermitteln die Fahrzeuge über die Laufzeit der ausgesendeten Signale ihre Position relativ zu den Funkbaken. Jedoch erweisen sich diese Indoor-GPS-Systeme als kostenintensiv und kommen daher eher selten zum Einsatz. [9]

3.3.2 Weltmodellerstellung intralogistischer mobiler Transportroboter

In der Forschung finden zunehmend FTS Einzug, die durch intelligente Sensorsysteme einen autonomen Betrieb der Fahrzeuge ermöglichen [56–58]. Diese Tendenzen werden nach [9] als die vierte Epoche der FTS bezeichnet, da die Fahrzeuge durch die Sensorik befähigt werden, direkt auf Hindernisse und Veränderungen ihrer Arbeitsumgebung reagieren zu können. Das zentrale Wegstreckennetz ist daher nicht notwendig, da die Verbindung zwischen den auftragsspezifischen Materialquellen und -senken in Abhängigkeit der aktuellen Bedingungen im Arbeitsraum erstellt werden kann. Diese Systeme entsprechen daher nicht den aktuellen Ausführungsformen mit FTF als Automaten, die nur einen vordefinierten Arbeitsvorgang abarbeiten. Es handelt sich bei den Fahrzeugen vielmehr um

„...universell einsetzbare Bewegungsautomaten mit mehreren Achsen, deren Bewegungen hinsichtlich Bewegungsfolge und Wegen bzw. Winkeln frei (d.h. ohne mechanischen bzw. menschlichen Eingriff) programmierbar und gegebenenfalls sensorgeführt sind.“ (VDI-Richtlinie 2860),

und stellen damit Transportroboter dar. Bei dieser neuen Art der FTF ist es neben der reinen Erfassung des Standorts und der Lage im Raum notwendig, auf Basis eines digitalen Abbilds des Arbeitsraums eine Pfadplanung durchführen zu können. Bei der Erstellung des Weltmodells sind zwei grundsätzliche Ansätze zu unterscheiden, einerseits die Digitalisierung der

Umgebung durch Fahrzeug- und andererseits durch Infrastruktursensoren. [59]

Fahrzeugsensoren

Die Methoden zur Weltmodellerstellung auf Basis von Sensoren, die direkt auf dem Fahrzeug integriert sind, entstammen der mobilen Robotik und beruhen darauf, dass ein Roboter eine vorher unbekannte Umgebung kartieren und darin navigieren kann. Es steht das einzelne Fahrzeug und weniger eine Flotte im Fokus der Applikation. Zum Einsatz kommen vorwiegend Laserscanner, die mehrere Aufgaben erfüllen. Zunächst werden ihre Daten verwendet, um eine statische Karte der Umgebung zu erstellen. Auf Grundlage dieser Karte erfolgt die Lokalisierung mit Hilfe eines Kalman- oder Partikelfilters und die aktuellen Daten werden bei der Wegplanung eingesetzt, um Hindernisse zu erkennen und Kollisionen zu vermeiden. Die statische Abbildung des Arbeitsraums wird einmalig mit Hilfe eines Algorithmus zur Lösung des SLAM-Problems (*Simultaneous Localization and Mapping*) erstellt. Der Rao-Blackwellized Partikelfilter [60] verbindet hierzu beispielweise die Daten des Laserscanners mit Messungen der Odometrie zu einer zweidimensionalen Belegungsrasterkarte. Diese Karte kann analog zur Bild 8 als ein Bild beschrieben werden, das weiße Pixel als Freiraum und schwarze Pixel als Hindernisse oder unbekannte Bereiche enthält. [61]



Bild 8: Beispiel eines durch Fahrzeugsensoren und SLAM erstellten Weltmodells

In Abhängigkeit von der Größe des Arbeitsraums und der Auflösung der Karte können sehr große Datenmengen entstehen. Die Daten des Laserscanners werden anschließend in Verbindung mit einer dynamischen Implementierung eines Partikelfilters zur Lokalisierung des Roboters auf der erstellten Karte verwendet. Die *Adaptive Monte-Carlo-Localisation*

(AMCL) [62] führt hierzu durch eine fortlaufende Anpassung der Partikelzahl die Lokalisierung des Roboters durch. Zur kollisionsfreien Pfadplanung werden zwei verschiedene Arten von Belegungsrasterkarten verwendet. Die oben beschriebene statische globale Karte und eine dynamische lokale Karte [63]. Beide Kartentypen enthalten Hindernisse und dienen dem globalen bzw. lokalen Pfadplaner (vgl. Kapitel 3.3.3) als sogenannte *Costmaps*. Die dynamischen lokalen *Costmaps* haben eine feste Größe, sind immer in der aktuellen Position des Roboters zentriert und berücksichtigen nur die aktuellen Daten der Fahrzeugsensoren. Ein Nachteil dieses Verfahrens ergibt sich aus der Verwendung der globalen statischen Karte für die globale Pfadplanung. Verändert sich der Arbeitsbereich des mobilen Roboters, muss diese Karte neu erzeugt werden. Da die Genauigkeit der Karte von der Genauigkeit der Odometriedaten abhängig ist, werden zur Kartenerstellung in einigen Fällen spezielle Kartierungsroboter verwendet. Weiterhin gibt es Methoden, bei der die Roboter neue Hindernisse in der globalen Karte eintragen und somit aktualisieren. Bei diesen Verfahren ist es wichtig, dass temporär vorhandene Hindernisse nach einer definierten Zeit in der globalen Pfadplanung nicht mehr berücksichtigt werden. Dem Nachteil, dass die globale Pfadplanung dennoch auf historischen Umgebungsdaten basiert und daher Schwächen in einem hochvolatilen Umfeld zeigt, wird damit allerdings nicht begegnet. Aktuelle Forschungsprojekte adressieren dieses Problem. Die Fahrzeuge tauschen ihre aktuellen Sensorinformationen zu Hindernissen gegenseitig aus und aktualisieren darauf aufbauend das Weltmodell. Dieses Verfahren funktioniert allerdings erst ab einer großen Anzahl an Fahrzeugen. Zudem ist die Aktualität der Kartensegmente davon abhängig, wie häufig der jeweilige Bereich befahren wird.

Infrastruktursensoren

Eine weitere Methode zur Digitalisierung der Arbeitsumgebung ist der Einsatz von Infrastruktursensoren wie z. B. Deckenkameras. In der Forschung existieren zwei relevante Ansätze für die Umsetzung, das Transportsystem MONAMOVE (*Monitoring and Navigation System for Mobile Vehicles*) [64] der TU Braunschweig und das Vielseitige Transportsystem (VTS) [12] des Lehrstuhls FAPS. MONAMOVE verwendet ein globales Überwachungssystem, um zu jedem Zeitpunkt ein globales und aktuelles Weltmodell zur Pfadplanung verfügbar zu haben. Im Gegensatz zur Weltmodellerstellung mit On-board-Sensoren ist der Planungshorizont der Fahrzeuge nicht auf die lokale Sensorreichweite beschränkt. Durch das Überwachungssystem werden alle zur Bahnplanung relevanten Daten in das Weltmodell eingetragen. Ein zentraler Navigator plant schließlich auf Basis dieser Daten die

Bahn für einen spezifischen Auftrag unter Berücksichtigung der aktuellen Umfeldbedingungen. Im Anschluss überführt der Pilot die errechnete Bahn in ausführbare Fahrbefehle und modifiziert bei Bedarf die resultierende Bahn. Die Fahrzeuge selbst fungieren in diesem Systemansatz, analog zur klassischen Auslegung von FTS als Automaten, welche die übertragenen Fahrbefehle ausführen (siehe Bild 9).

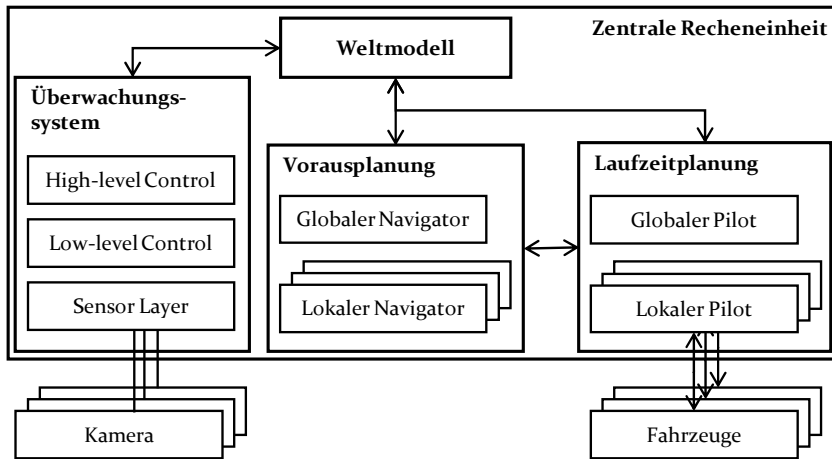


Bild 9: Konzept MONAMOVE zur Weltmodellerstellung mit Infrastruktursensoren (in Anlehnung an [66])

Auch beim VTS erfolgt die globale Lageerfassung von Hindernissen und Zielen durch stationäre 2D-Kameras. Das Ziel ist, mit einer moderaten Investition in Infrastruktursensoren weitgehend sensorlose Vielseitige Transportfahrzeuge (VTF) einzusetzen. Aufgrund ihrer geringen Eigenmasse ist diese Fahrzeugklasse gemäß [65] als inhärent sicher einzustufen, wodurch auf teure Sicherheitssensorik verzichtet werden kann. Die Berechnung der aktuellen Posen erfolgt, im Gegensatz zum Ansatz bei MONAMOVE, direkt auf den Fahrzeugen durch eine Sensordatenfusion aus den Inkrementalgebern der Fahrmotoren und der absoluten Positionsbestimmung anhand der Kameradaten. Die Bilddaten der Deckenkameras werden in die drei Kategorien Zielposition, Hindernisse und Fahrzeuge unterteilt und den Fahrzeugen bereitgestellt. Diese planen auf Basis dieser Informationen autonom ihren Pfad von der aktuellen Position zum fahrauftragsbezogenen Ziel. Sowohl die Daten der Kameras als auch die übertragenen Positionsschätzungen der Fahrzeuge, werden in einem zentralen System zusammengeführt und daraus ein Modell des Arbeitsraums erzeugt (Bild 10). Da die Fahrzeuge selbstständig ihre Pfadplanung vornehmen, ist das Gesamtsystem hinsichtlich der Flottengröße skalierbar, jedoch ist die Größe des

Arbeitsraums aufgrund der zentralen Verarbeitung und Aufbereitung der Bilddaten beschränkt. [66, 14]

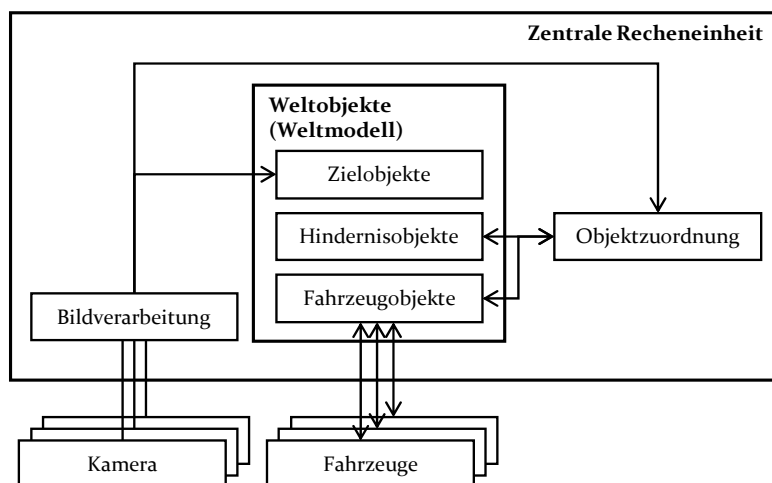


Bild 10: Konzept VTS zur Weltmodellerstellung mit Infrastruktursensoren (in Anlehnung an [14])

3.3.3 Bahnplanung intralogistischer mobiler Transportroboter

Neben der Weltmodellerstellung ist die Bahnplanung von der aktuellen Pose in eine spezifische Zielpose ein Hauptbestandteil mobiler Roboteranwendungen und damit auch von autonomen Transportfahrzeugen. [67]

Arbeitsraum

Der Arbeitsraum W ist der euklidische Umgebungsraum des Roboters und entspricht somit dem in Abschnitt 3.3.2 beschriebenen Weltmodell. Der durch Hindernisse blockierte Bereich wird mit W_{obs} bezeichnet, wobei $W_{obs,i}$ das jeweilige Hindernisobjekt definiert. W_{free} entspricht dem Bereich des Arbeitsraums, der durch den Roboter befahrbar ist.

Konfigurationsraum

Im Konfigurationsraum C wird der Roboter nicht anhand seiner tatsächlichen Kontur, sondern durch einen Punkt repräsentiert. Die Lage des Roboters wird durch die Konfiguration q vollständig beschrieben. Der Konfigurationsraum beschreibt die Gesamtheit aller Konfigurationen, die vom Roboter eingenommen werden können. Die Anzahl der Dimensionen entspricht dabei der Anzahl der DOF und ist somit bei einem mobilen Transportroboter dreidimensional. Der Raum wird zudem in die zwei Bereiche C_{free} und C_{obs} unterteilt. Die Überführung des Arbeitsraums in den

Konfigurationsraum ist ein elementarer Bestandteil aller Bahnplanungsalgorithmen. Hierzu werden die Konturen der Arbeitsraumhindernisse W_{obs} um die Größe des Roboters vergrößert. Das entstehende Polygon C_{obs} beschreibt demnach alle Konfigurationen, die zu einer Kollision des punktförmigen Roboters mit einem Hindernis führen.

Aktion

Der Pfad eines Roboters im Konfigurationsraum C setzt sich aus einer Vielzahl von Übergängen einer Startkonfiguration q_i in eine Zielkonfiguration q_{i+1} zusammen und wird als Aktion bezeichnet.

Vollständigkeit

Ein Bahnplanungsalgorithmus gilt als vollständig, wenn er in einer endlichen Zeit entweder eine Lösung zum Ziel bestimmen kann oder zurückmeldet, dass kein gültiger Pfad existiert. Als auflösungsvollständig werden Verfahren bezeichnet, die eine existierende Lösung in endlicher Zeit finden, jedoch kein Abbruchkriterium enthalten. Die dritte Gruppe basiert auf Zufallszahlen, wobei die Wahrscheinlichkeit, eine Lösung zu finden, mit steigender Anzahl der verwendeten Konfigurationen gegen 100 % strebt. Diese Algorithmen werden als probabilistisch vollständig bezeichnet.

Die Methoden zur Pfadplanung werden in die kombinatorische Bewegungsplanung und die stichprobenbasierte Bewegungsplanung unterteilt [68].

Kombinatorische Algorithmen

Die kombinatorischen Algorithmen finden im Konfigurationsraum den Pfad ohne Approximationen und werden daher auch als exakte Algorithmen bezeichnet. Zudem sind alle kombinatorischen Verfahren vollständig und verwenden zur Pfadplanung überwiegend einen generierten Graphen, auf dem anschließend der kürzeste Weg zum Ziel gesucht wird. Zwei bekannte Vertreter sind der Sichtbarkeitsgraph und die Zellzerlegungsmethode. Der Sichtbarkeitsgraph verbindet die Ecken von Hindernissen sowie die Start- und Zielkonfiguration mit gewichteten Kanten, wenn die Verbindungsstrecke vollständig in C_{free} liegt. Anschließend wird der erzeugte Graph auf unterstützende und trennende Linien reduziert (vgl. Bild 11). Charakteristisch für die Triangulationsmethode der Zellzerlegung ist, dass der Konfigurationsraum in Dreiecke zerlegt wird. Diese bilden sich durch die Verbindungslinien der Eckpunkte der Hindernisse oder dem Rand des Konfigurationsraums. Anschließend werden sowohl die Mittelpunkte jeder

Zelle als auch die Mittelpunkte der Verbindungslinien als Knoten des Graphen betrachtet (vgl. Bild 11). [68, 69]

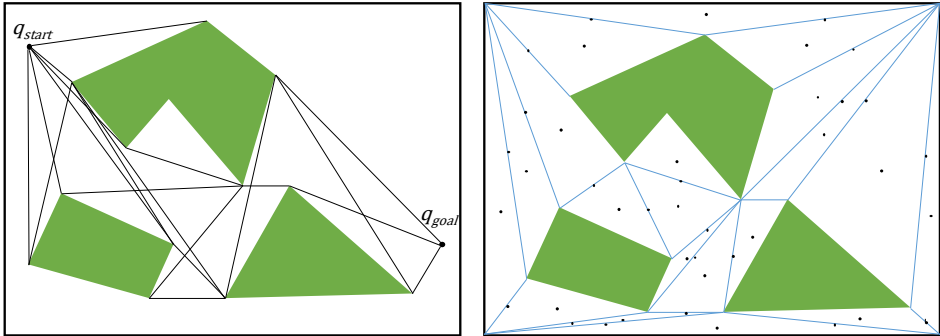


Bild 11: links reduzierter Sichtbarkeitsgraph (in Anlehnung an [71]); rechts Triangulationsmethode der Zellzerlegung (blaue Linien) mit Knotenpunkten des Graphen (in Anlehnung an [70])

Potentialfeldmethode

Bei der Potentialfeldmethode, einem Vorläufer der stichprobenbasierten Varianten, werden den Hindernissen und dem Roboter ein abstoßendes Potential zugewiesen. Zudem wird ein globales Potentialgefälle zwischen der Start- und Zielkonfiguration superpositioniert. Durch dieses erzeugte Potentialfeld ist der Roboter an jeder Stelle des Konfigurationsraums \mathcal{C} in der Lage, die Richtung des maximalen Potentialgefälles zu bestimmen und diesem zu folgen. Dadurch ergibt sich ein Pfad, der zur Zielkonfiguration führt und dabei Hindernissen ausweicht. Bei diesem Verfahren besteht allerdings das Problem von lokalen Minima, bei der der Roboter eine Endpose einnimmt, die nicht der Zielpose entspricht [70]. Weiterhin zeigt die Methode Schwächen beim Durchfahren von Engstellen und Instabilität in der Gegenwart von bewegenden Hindernissen, die den Einsatz in mobilen Robotern schwierig gestalten.

Probabilistische Algorithmen

Die im Hinblick auf die Rechenzeit effizientesten Methoden der Pfadplanung stellen vor allem bei komplexen und höherdimensionalen Problemen die probabilistischen Verfahren dar. Diese sind jedoch nicht vollständig und finden damit nicht zwingend den optimalen Pfad, mit steigender Anzahl der Zufallskonfigurationen strebt der gefundene Pfad jedoch dem optimalen Pfad entgegen. Alle Verfahren haben gemein, dass zufallsbasiert einzelne Konfigurationen q in \mathcal{C} gewählt werden und anschließend auf ihre Kollisionsfreiheit mit den Hindernissen geprüft werden. Ein Hauptunter-

scheidungsmerkmal der verschiedenen Ansätze ist, ob ein problemindividueller Suchbaum (*single-query*) verwendet wird oder der Algorithmus eine Landkarte des Konfigurationsraums (*multi-query*) erzeugt. Ein typischer Vertreter des *single-query* Verfahrens ist der *Rapidly-exploring Random Tree* (RRT) Algorithmus. Beginnend von der Startkonfiguration q_{start} bzw. des bereits bestehenden Baums werden neue Zufallskonfigurationen q_{rand} anhand einer Gleichverteilungsfunktion erzeugt. Diese Funktion kann zudem derart optimiert werden, dass die neuen Konfigurationen mit einer höheren Wahrscheinlichkeit in Richtung der Zielkonfiguration liegen. Anschließend wird im bestehenden Baum die Konfiguration q_{near} ermittelt, die den kürzesten Abstand zu q_{rand} aufweist. In einem definierten Abstand zu q_{near} wird schließlich auf der Verbindungslinie zu q_{rand} die Konfiguration q_{new} auf Kollisionsfreiheit geprüft und gegebenenfalls mit der entsprechenden Kante dem bestehenden Baum hinzugefügt. Dieser Vorgang wird mit einer definierten Anzahl an zu erstellenden Zufallskonfigurationen wiederholt und anschließend geprüft, ob die Zielkonfiguration q_{goal} kollisionsfrei mit dem Suchbaum verbunden werden kann. Ist dies nicht möglich, wird der Suchbaum erneut um die vordefinierte Anzahl an Zufallskonfigurationen erweitert (vgl. Bild 12). Der erzeugte Suchbaum stellt somit eine spezifische Lösung für die Verbindung der aktuellen Startkonfiguration q_{start} mit der auftragsbezogenen Zielkonfiguration q_{goal} dar.

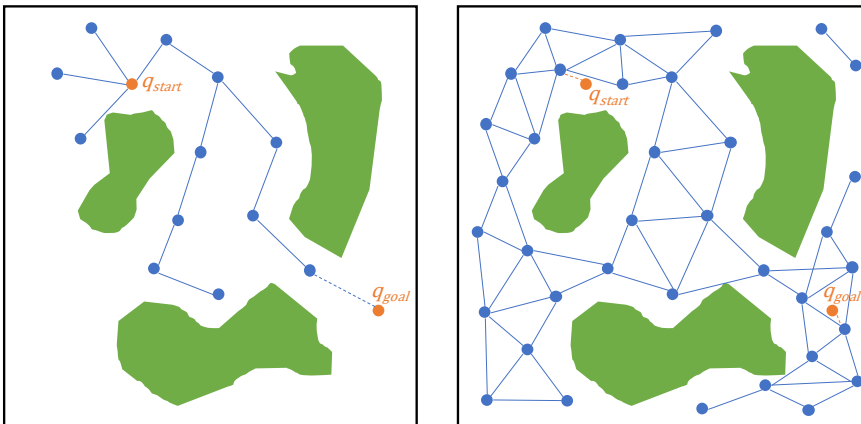


Bild 12: Gegenüberstellung probabilistischer Lösungsverfahren, links: RRT, rechts PRM

Multi-query Algorithmen wie die *Probabilistic Roadmap* (PRM) Methode hingegen unterscheiden eine Lern- und eine Lösungsphase. In der Lernphase wird eine vordefinierte Anzahl an zufälligen und kollisionsfreien Konfigurationen erzeugt. Im Anschluss werden diese Konfigurationen mit

allen, innerhalb eines maximalen Abstands befindlichen Konfigurationen verbunden, wodurch ein Graph im gesamten Arbeitsraum entsteht. Zur spezifischen Problemlösung wird die Startkonfiguration q_{start} und die Zielkonfiguration q_{goal} mit dem Graphen verbunden und die kürzeste Verbindung innerhalb des Graphen ermittelt (vgl. Bild 12).

3.4 Ausführungsformen aktueller Fahrerloser Transportsysteme und mobiler Transportroboter

Wie in den vorangegangenen Abschnitten beschrieben, ist die Entwicklung autonomer intralogistischer Transportsysteme von zwei grundlegenden Ansätzen getrieben. FTS fokussieren im Einsatz die Realisierung einer umfassenden Materialflussaufgabe zwischen definierten Quellen und Senken. Die Koordination der Materialbewegung unter den Fahrzeugen erfolgt dabei zentral über die Leitsteuerung. Die mobilen Transportroboter hingegen adressieren die Materialbewegung durch einzelne autonome Systeme zwischen einer auftragsspezifischen Quelle und Senke. Die individuelle Pfadplanung ist dabei im Transportroboter implementiert.

3.4.1 Kommerziell verfügbare Systeme

Kommerziell verfügbar sind aktuell sowohl FTS als auch mobile Transportroboter für unterschiedliche spezifische Anwendungsfelder der Intralogistik.

Götting KATE

Unter dem Markennamen KATE (Kleine Automatische Transporteinheit) vertreibt die Götting KG ein FTS mit Differenzialantrieb für kleine Nutzlasten bis 50 kg, im Zugbetrieb auch 100 kg. Die Grundfläche der Lastaufnahme ist mit seinen Abmessungen von 40 x 60 cm² zum Transport von Kleinladungsträgern (KLT) optimiert. Das System basiert auf einem optischen Spurführungssystem. Die Transportauftragsabwicklung von Flotten, bestehend aus bis zu 100 Einzelfahrzeugen, erfolgt über eine klassische Leitsteuerung. Aufgrund der Fahrzeuggestaltung und der Abmaße stellt KATE eine kostengünstige Lösung beim Transport von Gütern geringer Größe dar. Jedoch entstehen aus der Systemkonfiguration aus Leitlinientechnik und Leitsteuerung Flexibilitätseinbußen. [71]

Grenzebach L1200S

Das von der Grenzebach GmbH angebotene Unterfah-FTF L1200S ist für Traglasten bis 1.200 kg konzipiert und verfügt über ein Fahrwerk des Typs

(2,0). Es kann sich somit auf der Stelle drehen, die Orientierung des Transportguts zum Fahrwerk allerdings nicht verändern. Aufgrund der geringen Bauhöhe ist es in der Lage, unterschiedliche Lastenträger wie Trolleys, Gestelle oder Regale zu unterfahren und anzuheben. Die Ortung des Fahrzeugs erfolgt über passive Bodenmarker, die in einer Matrixform in den Boden eingelassen sind. Die Personensicherheit wird zusätzlich über zwei Sicherheitslaserscanner realisiert. Im Gegensatz zu KATE gewährt das Spurführungssystem vom L1200S eine höhere Flexibilität. Allerdings erfolgt die Pfadplanung und die Auftragsvergabe über einen proprietären Flottenmanager, der mehrere artgleiche Fahrzeuge innerhalb eines FTS koordiniert. [72]

Serva EVE

Unter dem Produktnamen EVEoutdoor und EVEindoor vertreibt die Serva Transport Systems GmbH ein flächenbewegliches FTF mit einem Fahrwerk des Typs (1,2) bestehend aus vier drehbaren Einzelrädern. Die maximale Transportlast bei diesem System beträgt 1.000 kg und die Lastaufnahme ist für den Transport von Europaletten und Gitterboxen optimiert. Die Ortung des Systems erfolgt mithilfe der Sicherheitslaserscanner anhand natürlicher Marker. Das System operiert meist als Einzelsystem oder in Flotten mit weniger als zehn Fahrzeugen. Zudem werden die Pfade sowie die Materialquellen und -senken in einem proprietären Leitsystem angelegt, das zudem die Verkehrsleitsteuerung mit einem Blockungssystem beinhaltet. Nachteilig an diesem FTF ist außerdem die Konstruktion des Lastaufnahmesystems, das sowohl an der Quelle als auch an der Senke ein spezielles Übergabegestell benötigt. [73]

Bosch Rexroth ActiveShuttle

Der im Forschungsprojekt E|Flow (Energieeffiziente, vielseitige und autonome Transportfahrzeuge für den innerbetrieblichen Materialfluss) entstandene autoBod (autonomer Bodenroller) wird aktuell vom Kooperationspartner Bosch Rexroth unter dem Namen ActiveShuttle an interne Fertigungswerke vertrieben. Der primäre Einsatzzweck ist die Substitution der etablierten Routenzugsysteme durch autonome Transportroboter, die Verbrauchs- und Verbaumaterial direkt vom Zentrallager an die Fertigungsanlagen transportieren. Der Transport erfolgt somit nicht mehr zyklisch, sondern verbrauchsgesteuert. Zudem entfallen durch die direkte Versorgung die fertigungsnahen Zwischenlagerstufen und Mehrfachhandlings. Das Fahrzeug ist gemäß dem spezifischen Anwendungsfall für Traglasten bis 200 kg ausgelegt und mit einem Fahrwerk des Typs (2,0) versehen. Die

Lastaufnahme erfolgt durch eine Hubeinrichtung, mit welcher der Bodenroller unterfahren und angehoben wird. Die Ortung des Systems erfolgt in der kommerziellen Variante mit einem SLAM-Algorithmus, bei dem die Odometriedaten und die Daten des Sicherheitslaserscanners fusioniert werden. Zudem koordiniert ein Flottenmanagementsystem die Fahrzeuge, das sowohl die Aufträge zuordnet, als auch feste Pfade im Weltmodell definiert. Die industrielle Flottenumsetzung entspricht daher einem klassischen FTS mit der einhergehenden unflexiblen Streckenfestlegung.

Omron Adept Lynx

Der mobile Roboter Adept Lynx der Omron Cooperation. Mit einer Nutzlast von 60 kg und einem Fahrwerk des Typs (2,0) stellt ein flexibles und schnell implementierbares Transportsystem für Logistiklösungen im industriellen und medizinischen Umfeld dar. Die Ortung erfolgt mithilfe der verbauten Sicherheitslaserscanner anhand natürlicher Marker auf Basis eines mit SLAM-Algorithmen erzeugten Weltmodells. Da der Roboter selbst über kein Lastaufnahmemittel verfügt, dient die Montageplattform mit einer Abmessung von 70 x 50 cm² zur Aufnahme verschiedener, anwendungsfallbezogener Aufbauten. Das Anwendungsfeld des Transportroboters ist überwiegend die flexible Verknüpfung verschiedener Quellen und Senken im kartierten Arbeitsbereich durch einzelne Fahrzeuge oder sehr kleine Flotten. Die Pfadplanung jedes Roboters erfolgt autonom anhand des historischen Kartenmaterials unter Berücksichtigung der aktuellen On-board-Sensordaten. Dadurch kann eine effiziente Pfadplanung mit Berücksichtigung der aktuellen Bedingungen nicht gewährleistet werden. [74]

MiR100, MiR200

Als MiR100 und MiR200 vertreibt Mobile Industrial Robots ApR zwei mobile Roboter mit einer Traglast von 100 bzw. 200 kg. Es handelt sich dabei um mobile Plattformen ohne aktive Lastaufnahmemittel, die mit Aufsatzmodulen wie Behälter, Regalen oder Förderbändern einsatzbezogen bestückt werden können. Der Roboter verfügt über ein Differenzialfahrwerk des Typs (2,0) und ist somit nicht flächenbeweglich. Das Aufgabenspektrum ist analog zum Adapt Lynx im Bereich der flexiblen Verbindung verschiedener Quellen und Senken mit einzelnen Robotern oder kleinen Flotten. Über eine Fusion der integrierten Laserscanner, der On-board-Kameras und der Odometriedaten erfolgt in einer Erkundungsfahrt die Kartierung des Arbeitsbereichs. Während der Betriebsphase plant der Roboter selbständig den auftragsbezogenen Pfad auf Basis dieser historischen

Umgebungskarte. Vor allem in einem stark volatilen Umfeld mit vielen bewegten Objekten zeigt dieser Ansatz der mobilen Robotik zur Pfadplanung Schwächen. [75, 76]



Bild 13: Kommerziell verfügbare FTS und mobile Transportroboter, 1: Magazino Toru (Quelle: [79]), 2: Serva EVA (Quelle: [75]), 3: Bosch Rexroth ActiveShuttle, 4: MiR100 (Quelle: [77]), 5: Götting KATE (Quelle: [73]), 6: Omron Adept Lynx (Quelle: [76])

Magazino Toru

Ein weiterer Vertreter der mobilen Robotiklösungen ist das System Toru der Magazino GmbH. Das Anwendungsgebiet dieses mobilen Roboters sind Versandlager, die konventionell nach dem Person-zu-Ware-Prinzip gestaltet sind. Anstelle eines Lagerarbeiters, der das gewünschte Objekt aus dem Regal holt, oder der halbautomatisierten Variante, bei der die Ware vom Lagerplatz mittels Fördermittel zum Lagerarbeiter transportiert wird, handelt es sich beim Pick-by-Robot-Ansatz von Toru um eine vollständig automatisierte Methode. Das Stückgut wird direkt von einem mobilen Regalroboter identifiziert, gegriffen und schließlich an seinem Bestimmungsort abgelegt. Das System fährt an das Regal heran, um den Inhalt mit der im Greifsystem eingebauten Kameratechnik zu scannen, bis es das gesuchte Objekt findet. Hierdurch können ungeordnete Objekte in Kommissionierbereichen erkannt und gegriffen werden. Anschließend wird das

Objekt auf einer Ablagefläche zwischengelagert und zur Werkbank oder zur Versandstation transportiert und abgelegt. Der Systemansatz stellt somit eine flexible Einzelsystemlösung im Bereich der Lagertechnik und damit nur einem Teilbereich der Intralogistik dar. [77]

3.4.2 Systeme in der Forschung und Entwicklung

Zusätzlich zu den kommerziellen Lösungen werden in aktuellen Forschungs- und Entwicklungsprojekten neben MONAMOVE und dem VTS (vgl. Abschnitt 3.3.2) weitere FTS und mobile Transportroboter adressiert, welche die Anwendungsfelder stetig erweitern.

Arculee

Mit dem Entwicklungsprojekt Arculee bewirbt die Arculus GmbH eine mobile Plattform mit einer maximalen Traglast von 1.000 kg. Die Transportplattform hat eine Grundfläche von 75 x 70 cm² und soll beim Transport von Karosserien in der Automobilfertigung zum Einsatz kommen und damit das Fließband ersetzen. Aufgrund des frühen Entwicklungsstands ist die finale Gestaltung des Fahrwerks nicht definiert, wobei eine quasi-omnidirektionale Ausführung priorisiert wird. Die Navigation erfolgt dabei über ein Markersystem, dessen Genauigkeit und Flexibilität über ein Funkortungssystem erhöht werden soll. Aufgrund der Systemgestaltung ist eine Leitsteuerung notwendig, wodurch die Anzahl der Fahrzeuge in der Flotte durch die Leistung des Leitrechners beschränkt ist. [78]

Smart Transport Robot

Ein ähnliches System stellt das Entwicklungsprojekt STR (Smart Transport Roboter) der BMW AG dar. Die Traglast der mobilen Plattform mit einer Fahrwerkskinematik des Typs (2,0) beträgt 550 kg. Das FTF soll zukünftig im internen Werksverbund als Einzelfahrzeug zum Einsatz kommen. Daher verfügt es über eine autonome Navigation, verbaute Sicherheitslaser Scanner sowie eine 3D-Kamera zur Objekterkennung. Ein weiterer Anwendungsfall ist die Realisierung größerer Flotten, die ein Funkortungssystem nutzen und durch eine Leitsteuerung koordiniert werden. Der Piloteinsatz stellt ein Versandlager für CKD-Teile (*Completely Knocked Down*) dar, in dem das System die Regale aufgrund seiner geringen Bauhöhe von 220 mm unterfahren kann und zum Werker transportiert (Ware-zur-Person). Aufgrund der systembezogenen Leitsteuerung ist eine Kombination und Synchronisation dieses FTF mit anderen Automatisierungslösungen nur auf Leitrechnerebene möglich. [79]

MultiShuttle move

Ein Forschungsprojekt zur Steigerung der Flexibilität von Logistiklösungen ist das Multishuttle Move von Fraunhofer. Die autonomen Transportfahrzeuge des Systems kommunizieren untereinander über Softwareagenten, um die eingelasteten Transportaufträge zu verteilen, wodurch eine dezentrale Steuerung der Fahrzeuge implementiert ist. In dem zellularen Transportsystem agieren die Transporteinheiten mit einer Abmessung von $113,5 \times 706 \text{ cm}^2$ und einer Nutzlast von 40 kg sowohl als Regalshuttles als auch als FTF außerhalb des automatischen Kleinteilelagers (AKL). Im Arbeitsraum vor dem AKL ist daher neben den Inertialsensoren, der Odometrie und dem Funkortungssystem zusätzlich ein Sicherheitslaserscanner zur Navigation notwendig. Aufgrund des spezifischen Designs und der Spezialisierung der Fahrzeuge für den Transport von KLT ist der Einsatzbereich des Ansatzes auf AKL und deren Dispositionsbereich beschränkt. [80, 81]

KARIS Pro

Ein weiteres Beispiel für ein System mit einer verteilten Auftragsvergabe adressiert das Forschungsprojekt KARIS Pro (kleinskaliges redundantes Intralogistik-System) der Arbeitsgemeinschaft Technologie Intralogistik Baden-Württemberg. Die Auftragsverwaltung erfolgt hier ebenfalls über ein Agentensystem, bei dem eine Auftragsliste generiert wird, die unter allen Fahrzeugen synchronisiert ist. Bei Auftragsvergabe übernimmt ein Fahrzeug den höchst priorisierten Auftrag aus der Liste und startet eine Auktion, indem es eine Anfrage an alle weiteren Fahrzeuge stellt. Jedes Fahrzeug gibt daraufhin ein Angebot an dieses Fahrzeug ab [82]. Es vergleicht die Angebote mit der eigenen Abschätzung und vergibt den Auftrag an das günstigste Angebot. Das mechanische Fahrzeugkonzept beruht auf einer Plattform mit den Abmessungen $65 \times 45 \text{ cm}^2$ bei einer Höhe von 30 cm. Das Fahrwerk des Typs (1,2) bei KARIS Pro besteht im Gegensatz zu seinem Vorgänger KARIS aus vier lenkbaren Antriebsrädern. Vor der Inbetriebnahme muss gemäß der in Abschnitt 3.3.2 beschriebenen Vorgehensweise eine Kartierung des Arbeitsraums erfolgen. Die Routenplanung der Fahrzeuge erfolgt bei der Auftragsdurchführung auf Basis dieser Karte autonom. Zwar bietet dieser Ansatz Vorteile hinsichtlich der freien Routenführung der einzelnen Roboter, jedoch wird die aktuelle Arbeitsumgebung durch das historische Kartenmaterial nur ansatzweise abgebildet. [83, 84]

Doppelkufensystem

Das Doppelkufensystem der Universität Stuttgart ist ein Forschungsansatz, der den Transport von Paletten durch autonome Transportroboter adressiert. Die Nutzlast beträgt 1.000 kg bei einer Transportgeschwindigkeit von bis zu 1 m/s. Durch die Konstruktion der Antriebseinheiten ist es möglich, ein Fahrwerk des Typs (1,2) mit der notwendigen Energieversorgung und den Steuerungskomponenten im Bauraum eines Hubzinkens für Europoolpaletten unterzubringen. Durch dieses Antriebskonzept kann der Momentanpol der Drehbewegung beliebig verschoben werden, was zu einer höheren DOF im Gegensatz zu etablierten Systemen wie Gabelstaplern führt. Zudem können Paletten durch die Umsetzung des Systems mit zwei voneinander getrennten Fahrzeugen komplett durchfahren werden, was einen Transport von Paletten im Verbund ermöglicht. Jedoch wird der aktuelle Prototyp des Systems mit dem Namen LogiMover von Eisemann SE aufgrund der fehlenden Sicherheitssensorik lediglich mannggeführt und ohne eine Auftragsverwaltung betrieben. [85, 86]



Bild 14: FTS und mobile Transportroboter aus der Forschung und Entwicklung, 1: Smart Transport Robot (Quelle: [81]), 2: MultiShuttle move (Quelle: [83]), 3: Doppelkufensystem (Quelle: [87]), 4: KARIS Pro (Quelle: [86]), 5: Arculee (Quelle: [80])

3.5 Zusammenfassung und Handlungsbedarfe

Die aktuellen Entwicklungstendenzen innerbetrieblicher Wertstromstrukturen zeigen einen deutlichen Trend auf, dass sich eine Vereinigung der spezifischen Vorteile der verrichtungs- und prozessfolgeorientierten Fertigungsstrukturen etablieren wird. Alle wissenschaftlichen Ansätze weisen dabei die Gemeinsamkeit auf, dass flexible intralogistische Fertigungsnetzwerke autonome und intelligente Transportmittel benötigen, die nicht für eine spezifische Aufgabe ausgelegt sind, sondern durch ihre Vielseitigkeit unterschiedliche Transportaufgaben übernehmen können. Eine umfangreiche Analyse der aktuellen Ausführungsformen und Forschungsvorhaben im Bereich der FTS und intralogistischen Transportroboter zeigt jedoch auf, dass weder die notwendige Interoperabilität unterschiedlicher Systeme gegeben ist, noch eine Systemarchitektur definiert ist, die einen flexiblen Materialfluss unabhängig der physischen Ausführungsform des Transportmittels ermöglicht.

FTS verfügen über eine zentrale Leitsteuerung, welche die Generierung der Transportaufträge und deren Disposition auf die Fahrzeuge zentral verwaltet. Zudem werden den FTF die Wegstrecken zentral vorgegeben, selbst wenn die Fahrzeuge über eine freie Navigation verfügen. Die einzelne Transporteinheit stellt somit lediglich einen ausführenden Automaten dar, der die zugewiesenen Transportaufträge abarbeitet, ohne auf die Auftragsbelastung oder die Routenplanung Einfluss zu nehmen. Diese Ansätze der FTS verfügen daher nicht über die notwendige Flexibilität und Autonomie, die für ein intralogistisches Fertigungsnetzwerk notwendig sind.

Aktuelle Forschungsvorhaben und kommerzielle Ausführungsformen adressieren diesen Handlungsbedarf durch den Einsatz hochautonomer Transportroboter, die sich in unbekannten Arbeitsumgebungen orientieren und auf Änderungen selbstständig reagieren können. Die industriellen Anwendungsfelder beschränken sich jedoch auf Einzelfahrzeuge und kleine Fahrzeugflotten mit einfachen Transportvorgängen zwischen definierten Materialquellen und -senken, die vom Betreiber eingeplant werden. Bei komplexeren Anwendungsszenarien greifen auch diese Systeme auf eine zentrale Instanz zur Auftragskoordination und Routenführung zurück. Alle aktuellen Ausführungsformen haben zudem gemein, dass sowohl die Hardware als auch die proprietäre Software für spezifische Lösungen appliziert werden und damit eine Interoperabilität zwischen verschiedenen Transportsystemen lediglich durch die Kommunikation übergeordneter, zentraler Instanzen realisiert werden kann. Zielführend ist daher, ein gene-

risches Softwaremodell zu definieren, das durch eine allgemeingültige Beschreibung der Fähigkeiten von Transportrobotern eine hardwareunabhängige Software zur Durchführung und Allokation von Aufträgen erlaubt.

Diesen Handlungsbedarf greift Kapitel 4 auf und beschreibt eine Methode zur Implementierung eines Softwareschichtenmodells auf der Fahrzeugebene, dass eine uniforme Auftragsallokation zwischen heterogenen Systemen erlaubt und damit die Auftragsdisposition herstellerunabhängig macht. Erfolgversprechend ist hierbei die Ausbildung eines MAS, in das die strategische Softwareschicht der Fahrzeuge eingebunden ist. Der Fahrzeugagent agiert somit unabhängig und eigenständig innerhalb des Systems, wodurch eine dezentrale Auftragsvergabe realisiert wird. Für die Ablaufsteuerung in einer flexiblen automatisierten Fertigung beschreiben [87, 88] die Verhandlungsmodelle zwischen einzelnen Agenten von Fertigungsanlagen, zudem zeigen [89, 90] den Transfer dieser Ansätze in die Intralogistik auf. Jedoch ist bislang die Ontologie der jeweiligen Agenten auf die spezifische Zielplattform abgestimmt. Der Handlungsbedarf besteht daher in einer generischen Definition der Ontologie von intralogistischen Transportfahrzeugagenten anhand von logistischen Kennzahlen, nach denen der Agent in den Auftragsverhandlungen agiert. Um die Unabhängigkeit der Zielplattform zu gewährleisten, ist es außerdem notwendig, die individuellen Fähigkeiten der jeweiligen Transportfahrzeuge aufgrund ihrer spezifischen Sensorik und Aktorik zu übergeordneten Hauptfähigkeiten zu aggregieren und dem Agenten zur Verfügung zu stellen. Hierzu erweist sich der Einsatz von Zustandsautomaten, die über das hybride Steuerungsparadigma die Verhandlungsbasis der Agenten bereitstellen, als zielführend [91].

Ein weiterer Handlungsbedarf, der sich aus der Interoperabilität logistischer Systeme und der Abkehr von proprietären Insellösungen ergibt, ist die Beschreibung einer intralogistischen Serviceschicht, die über die Auftragsallokation hinausgehende elementare Funktionen für alle Akteure bereitstellt. Diesen Forschungsinhalt adressiert Kapitel 5, in dem Methoden zur Digitalisierung des Arbeitsraums, der Routenplanung und der Pfadausführung anhand einer SOA beschrieben werden. Die Erfassung des Arbeitsraums mit Hilfe von Infrastruktursensoren bietet im Hinblick auf die Aktualität des Kartenmaterials Vorteile gegenüber der fahrzeuginternen Weltmodellerstellung [14, 92]. Der Handlungsbedarf besteht darin, Methoden zur Erfassung von statischen und dynamischen Hindernissen zu erforschen, die eine Skalierbarkeit des Gesamtsystems erlauben und zugleich in der Lage sind, ein vorverarbeitetes Kartenmaterial für unterschiedliche

Transportfahrzeuge zur Verfügung zu stellen. Im Hinblick auf die Skalierbarkeit der Flottengröße ist es zudem notwendig, eine Methode zu beschreiben, die eine dezentrale Pfadplanung durch die Fahrzeuge selbst ermöglicht und lokale Umgebungsinformationen berücksichtigt. Dabei muss einerseits auf eine zentrale Leitsteuerung verzichtet werden, andererseits muss die Pfadplanung auf unterschiedliche Zielplattformen ohne Implementierungsaufwand anwendbar sein.

Weiterhin ist es notwendig, die Ontologie der Auftragsallokation über das MAS in eine DES zu integrieren, um eine Transparenz der Lösungsfindung zu schaffen. Aufgrund der individuellen Entscheidungen der einzelnen Agenten ist die Systemlösung eines vielschichtigen und großen MAS nicht vorhersagbar [93]. Dieser Umstand wird in Kapitel 6 erforscht und eine Methode beschrieben, wie die Ontologie des MAS bereits während der Planungsphase in eine DES eingebunden werden kann. Zudem ist es in der anschließenden Betriebsphase notwendig, das Simulationsmodell parallel zum realen Betrieb als sogenannten digitalen Zwilling zu betreiben, um die Vorhersagbarkeit der Systemlösung bei diskreten Ereignissen zu gewährleisten und gegebenenfalls Adaptionen vornehmen zu können. Dieser Teilaspekt wird ebenfalls in Kapitel 6 adressiert.

Zusammenfassend ergeben sich sechs Kernanforderungen an ein flexibles intralogistisches Transportsystem mit autonomen intelligenten Entitäten:

1. Uniforme, plattformunabhängige Ontologie für Fahrzeugagenten mit einer kennzahlenbasierten Entscheidungslogik
2. Allgemeingültige Definition von Hauptfähigkeiten der Transportfahrzeuge auf Basis spezifischer Sensorik und Aktorik
3. Weltmodellerstellung und Separation von statischen und dynamischen Hindernissen durch ein skalierbares System aus Infrastruktursensoren
4. Bereitstellung eines aufbereitenden fahrzeugspezifischen Weltmodells über einheitliche Logistiks-services innerhalb einer SOA
5. Zielplattformübergreifend applizierbare Pfadplanung auf den Fahrzeugen ohne zentrale Instanz
6. Sicherstellung der Planbarkeit und Vorhersagbarkeit der Entscheidungen des MAS

Diesen Anforderungen entsprechende Systeme und Dienste werden im Folgenden vorgestellt. Während in den Kapiteln 4, 5 und 6 eine Beschreibung der Methoden im Vordergrund steht, folgt in Kapitel 7 eine Diskussion der Implementierung und eine quantitative, experimentelle Beurteilung.

4 Methode eines Schichtenmodells zur intralogistischen Auftragsallokation und -bearbeitung

Die Umsetzung einer dezentralen und selbstregelnden Auftragsallokation innerhalb eines FTS mit autonomen Fahrzeugentitäten ist seit einigen Jahren der Kern unterschiedlicher wissenschaftlicher Ansätze. Die Restriktionen, die von einer Auftragsverwaltung und -disposition durch eine zentrale Leitsteuerung erfolgen, werden dadurch umgangen. Die Verteilung der Transportaufträge erfolgt innerhalb eines MAS, in dem die Fahrzeugagenten eigenständig eine optimale Zuteilung auf die verfügbaren Transportentitäten vornehmen. Somit wird einerseits die Rechenlast der Disposition bei komplexen und volatilen intralogistischen Szenarien verteilt, andererseits beinhalten diese Systemansätze keine zentrale Schwachstelle (*single point of failure* - SPOF). Die aktuellen wissenschaftlichen Ausführungsformen von Systemen mit einer selbstregelnden Auftragsallokation haben jedoch gemein, dass die Fahrzeugentitäten für einen spezifischen intralogistischen Anwendungsfall konzipiert werden und das FTS daher lediglich im Vorfeld definierte Transportaufgaben löst. Zudem ist die Verhandlungsstrategie der Fahrzeugagenten für das Zielsystem konzipiert. Im Vordergrund steht eine Aufteilung der Transportaufträge auf artgleiche Fahrzeuge nach einer definierten Strategie, um eine optimale Auslastung des spezifischen Transportsystems zu gewährleisten. Die Verwendung unterschiedlicher Fahrzeugkonfigurationen innerhalb eines intralogistischen Schwarms ist daher nicht gewährleistet. Diese Aufgabenstellung, anhand einer uniformen und von der Zielformatplattform unabhängigen Ontologie der Fahrzeugagenten eine plattformübergreifende Systembeschreibung zu realisieren und allgemeingültige Fähigkeiten von Transportrobotern unabhängig der spezifischen Hardware zu beschreiben, wird in den nachfolgend erläuterten Forschungsarbeiten aufgenommen und deren Lösungsmöglichkeiten erörtert. [94–96]

Die Architektur des entwickelten zielformatplattformunabhängigen Schichtenmodells für die Auftragsallokation und -durchführung wird in Abschnitt 4.1 vorgestellt und dessen zentrale Komponenten beschrieben. Die Methode zur generischen Beschreibung der Fähigkeiten von Transportentitäten auf Basis von (quasi-)omnidirektionalen Fahrwerkstypen ist Inhalt von Abschnitt 4.2. Der Abschnitt 4.3 beschreibt die Einbindung der Fahrzeugagen-

ten in ein MAS zur Auftragsallokation innerhalb eines heterogenen Fahrzeugverbunds. Abschließend zusammengefasst werden die gewonnenen Erkenntnisse in Abschnitt 4.4 sowie die mögliche Integration des iLES in die Softwarelandschaft eines Unternehmens aufgezeigt.

Einzelne Aspekte der Ausführungen der Folgeabschnitte 4.1 bis 4.3 sowie des Abschnitts 7.2, der die Umsetzung und Evaluierung der beschriebenen Methode umfasst, sind an den am Lehrstuhl FAPS entstandenen und vom Autor formulierten studentischen Arbeiten [S16–S25] angelehnt. Bestandteile und Vorarbeiten der dargestellten Methoden und Erkenntnisse wurden der internationalen Fachwelt auch in den vom Autor erstellten Veröffentlichungen [P3–P9] zugänglich gemacht.

4.1 Architektur des Schichtenmodells und der Auftragsallokationsebene

Die zentrale Anforderung an das Schichtenmodell und die Auftragsallokationsebene ist deren Einsetzbarkeit in unterschiedlichen intralogistischen Szenarien und die Übertragbarkeit auf jegliche flurgebundene Transportentität mit (quasi-) omnidirektionalem Fahrwerkstyp. Weiterhin ist es notwendig, dass die Architektur eine Ontologie der Fahrzeugagenten bereitstellt, die eine auf intralogistischen Kennzahlen basierende Entscheidungslogik bei der Disposition der Aufträge umsetzt. Die zweite entscheidende Anforderung an die Softwarearchitektur ist eine allgemeingültige Definition von Hauptfähigkeiten der Transportfahrzeuge, die entsprechend der plattformindividuellen Sensorik und Aktorik umgesetzt werden können. Um diese Anforderungen zu erfüllen, ist es notwendig die strategischen Entscheidungen der Transportroboter von der Beschreibung der Fähigkeiten und der Auftragsdurchführung zu entkoppeln und in einem dreischichtigen Softwaremodell aufzuteilen.

4.1.1 Aufbau und Funktionsweise

In der untersuchten Architektur wird ein fahrzeugseitiges Schichtenmodell umgesetzt, dessen Agent in eine Auftragsallokationsebene eingebunden ist. Während die Abschätzung des benötigten Aufwands zur Durchführung eines spezifischen Transportauftrags und dessen anschließende Abwicklung auf dem Fahrzeug selbst durchgeführt wird, erfolgt die Disposition der Aufträge innerhalb eines übergreifenden intralogistischen Systems. Auf Basis dieser Spezifikation beruht die in Bild 15 visualisierte Architektur des fahrzeugseitigen Schichtenmodells und des iLES.

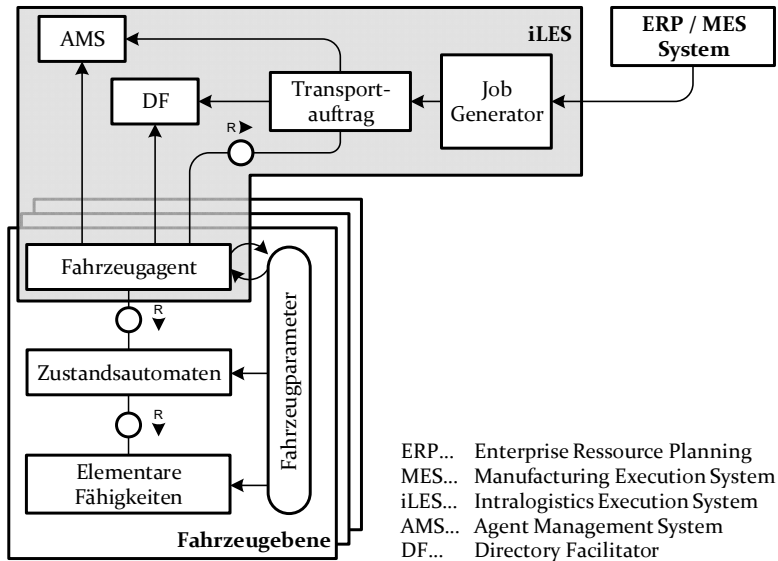


Bild 15: Fundamental Modeling Concepts (FMC)-Block-Diagramm des iLES und des Schichtenmodells der Fahrzeuge

Das fahrzeugseitige Softwareschichtenmodell verfügt über drei Schichten. Die oberste Schicht bildet der Agent, der die Durchführung der dem Fahrzeug zugewiesenen Transportaufträge überwacht und innerhalb des iLES die Verhandlung übernimmt. Darunter befindet sich der Zustandsautomat. Dieser aggregiert einerseits die elementaren Fähigkeiten des Fahrzeugs zu übergeordneten und generischen Hauptfähigkeiten, führt andererseits spezifische Aufträge des Agenten aus und meldet deren Status an diesen zurück. Die dritte Schicht stellt die spezifischen, von der Sensorik und der Aktorik abgeleiteten elementaren Fähigkeiten des Fahrzeugs dar. Diese werden vom Zustandsautomaten zur Durchführung einer komplexeren Handlung kombiniert und entsprechend ausgeführt. Die eingesetzten Aktoren und Sensoren melden folglich ihre Informationen an den Zustandsautomaten. Erfolgt auf Basis eines Verhandlungsergebnisses innerhalb des iLES die Disposition eines Transportauftrags auf das Fahrzeug, weist der Fahrzeugagent den Zustandsautomaten an, diesen Auftrag durchzuführen. Innerhalb der Zustandsautomaten sind die hierzu notwendigen elementaren Fähigkeiten des Fahrzeugs in einem Ablaufplan zusammengefasst. Diese Teilfähigkeiten wiederum sind ebenfalls soweit untergliedert, bis schließlich die elementaren Fahrzeugfähigkeiten erreicht sind. Innerhalb des Zustandsautomaten werden diese untergliederten Funktionen sequenziell abgearbeitet und anhand der Rückmeldung der Sensoren und Aktoren die korrekte Durchführung überwacht. Bei erledigter Abarbeitung oder einem Abbruch

des Auftrags erfolgt eine Rückmeldung des Zustandsautomaten an den Fahrzeugagenten, der in Abhängigkeit des Ergebnisses das weitere Vorgehen plant.

Um die Funktionsweise des Fahrzeugschichtenmodells auf unterschiedlichen Zielplattformen zu ermöglichen, ist in der Architektur ein Fahrzeugparameterserver integriert. Dieser beinhaltet sämtliche Informationen und Eigenschaften, die die jeweilige Zielplattform definieren. Der generische Ansatz des implementierten Parameterservers erlaubt es, sämtliche (quasi)-omnidirektionalen Transportfahrzeuge mit passiven Lastaufnahmemitteln und Greifern zu beschreiben und in das System einzubinden. Anhand der hier implementierten Informationen werden die elementaren Fähigkeiten des Fahrzeugs beschrieben und im Zustandsautomaten zu Hauptfähigkeiten zusammengefasst. Zudem dienen die Parameter dem Fahrzeugagenten als Basis der Auftragsverhandlung innerhalb des iLES. Neben den bei der Applikation eingetragenen Parametern enthält der Server zudem fahrzeugspezifische Kennzahlen, die vom Agenten verwaltet werden und bei der Inbetriebnahme initial festgelegt werden. Hierzu zählt z. B. die Abschätzung der Handlingszeit, die angibt, inwieweit die während der Verhandlungsphase abgeschätzte Zeit zur Lastaufnahme und -abgabe mit der anschließenden tatsächlichen Durchführungszeit korreliert.

In die Auftragsallokationsebene selbst ist fahrzeugseitig lediglich der Fahrzeugagent eingebunden. Die Struktur des iLES ist gemäß dem Management-Referenzmodell des Standards nach FIPA (*Foundation for Intelligent Physical Agents*) konzipiert [97]. Die Plattform verfügt somit über ein obligatorisches *Agent Management System* (AMS) und einen *Directory Facilitator* (DF). Bei der Initialisierung des Fahrzeugagenten und dessen Anmeldung im iLES meldet sich dieser beim AMS an, welches sämtliche aktive Agenten verwaltet und diesen eine eindeutige Adresse, den *Agent-Identifier*, zuordnet [98]. Der DF stellt einen Informationsdienst zur Verfügung, bei dem die jeweiligen Agenten der Plattform nicht nur ihre eigenen Dienste veröffentlichen können, sondern auch gezielt Informationen zu angebotenen Diensten anderer Agenten abrufen [99]. Der Fahrzeugagent trägt folglich bei seiner Initialisierung die relevanten Fahrzeugeigenschaften wie z. B. das zulässige Transportgewicht in den DF ein. Ein weiteres obligatorisches Element des FIPA-Standards ist das *Message Transport System* (MTS), das den Nachrichtentransport zwischen dem Sender und dem Empfänger definiert und sicherstellt. Bei der Einlastung eines Transportauftrags in das iLES ordnet ein zentraler Auftragsagent diesen Auftrag einer spezifischen Auftragsklasse, wie z.B. Eilauftrag, zu. Diese Klasse defi-

niert das anschließende Verhalten und die Ziele des spezifischen Transportauftragsagenten während der Verhandlung mit den Fahrzeugagenten. Zunächst erfolgt die Registrierung des Transportauftragsagenten beim AMS, anschließend ruft dieser die Informationen zu möglichen Transportfahrzeugen beim DF ab, die aufgrund ihrer Fahrzeugeigenschaften prinzipiell in der Lage sind, den Transportauftrag durchzuführen. Somit erfolgt die Anfrage einer Transportabschätzung lediglich an die Fahrzeuge, die aufgrund ihrer baulichen Voraussetzungen den Auftrag generell durchführen können. Die Disposition des Auftrags erfolgt nach dem *FIPA Contract Net Interaction Protocol* [100]. Dabei fragt der Transportauftragsagent die in Frage kommenden Fahrzeugagenten nach einer Abschätzung der Transportkennzahlen an. Diese melden die abgeschätzten Kennzahlen, analog eines Kostenvoranschlags, zurück. Der Auftrag selbst entscheidet schließlich gemäß seinem spezifischen Verhalten und seiner individuellen Ziele, welches Fahrzeug am besten für die Durchführung geeignet ist und bestätigt das Angebot. Alle anderen Fahrzeuge erhalten eine Absage. Somit ist eine direkte, vom Auftrag ausgehende Disposition des Transports gegeben.

Die Beschreibung der Auftragsallokation und deren sequenzielle Abarbeitung durch den Fahrzeugagenten sowie die Methode zur Abschätzung der auftragspezifischen Kennzahlen und die Auftragsdurchführung durch die Hauptfähigkeiten des Zustandsautomaten werden in den folgenden beiden Abschnitten im Detail erörtert.

4.1.2 Beschreibung der Auftragsallokation

Eine der Grundanforderungen an ein flexibles intralogistisches System ist die dezentrale und selbstregelnde Allokation der Transportaufträge auf die jeweiligen Entitäten. Zudem muss das System eine uniforme und somit transportplattformunabhängige Ontologie verwenden, um die Interoperabilität unterschiedlicher Materialflusslösungen zu gewährleisten.

Die erforschte Methode beinhaltet daher ein MAS zur Disposition von Transportaufträgen auf unterschiedliche Transportentitäten. Ausschlaggebend ist dabei, dass die Zuteilung des Auftrags auf ein spezifisches Fahrzeug weder über einen zentralen Broker, noch durch eine Verhandlung der Fahrzeuge untereinander erfolgt, sondern jeder Auftrag in Abhängigkeit seiner individuellen Zielgrößen ein Transportsystem auswählt. Die Disposition des Auftrags erfolgt auf Basis einer Abschätzung der möglichen Transportgrößen jeder einzelnen Transportentität innerhalb eines standardisierten intralogistischen Kennzahlensystems. Durch diese einheitliche

Ontologie entsteht ein Auftragsallokationsalgorithmus, der eine transportplattformunabhängige Disposition ermöglicht. Das in Bild 16 dargestellte iLES beschreibt die Kernkomponenten und den Informationsaustausch innerhalb dieser generischen Auftragsallokationsebene.

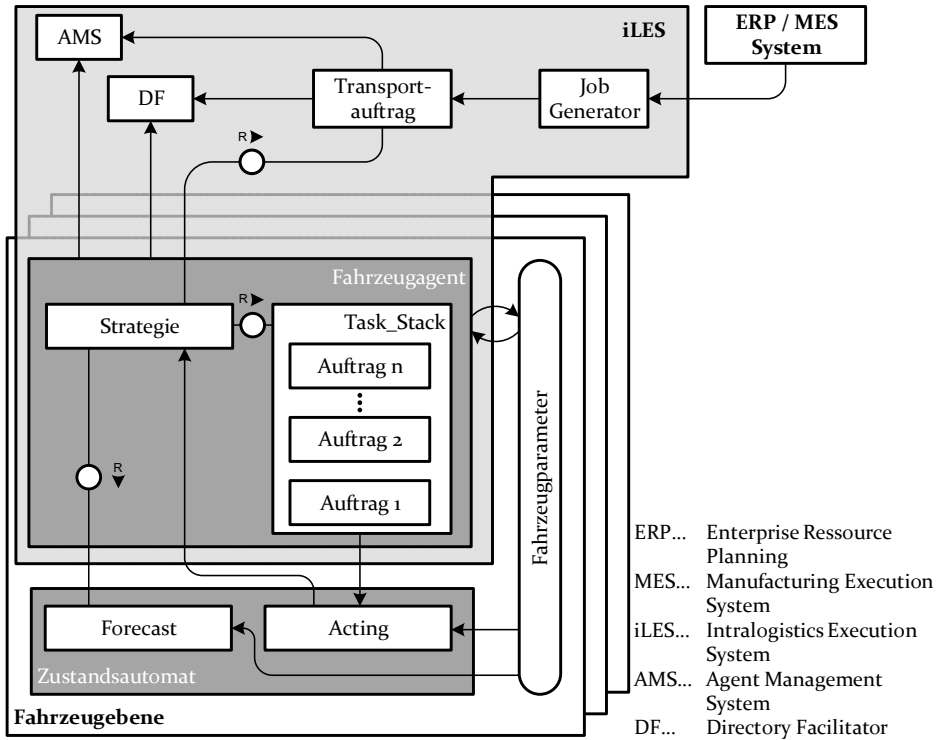


Bild 16: FMC-Block-Diagramm der Auftragsallokation und -bearbeitung

Die Informationen zu einem spezifischen Transportauftrag werden über eine standardisierte Schnittstelle in das iLES eingebracht und vom Transportauftragsgenerator übernommen und interpretiert. Dieser stellt ebenfalls einen Agenten nach dem FIPA-Standard [101] dar. Zur Beschreibung der eingehenden Transportaufträge hält der Transportauftragsgenerator sieben Auftragsklassen vor, in denen das Verhandlungsverhalten in Abhängigkeit der Zielgrößen des spezifischen Auftrags definiert sind. Diese Auftragsklassen sind wie folgt klassifiziert:

- Kostenauftrag: möglichst geringe Auftragskosten; Durchlaufzeit irrelevant; Auftragsbeginn sofort; kein vorgegebener Endzeitpunkt
- Kostenauftrag mit Zeitbedingung: möglichst geringe Auftragskosten; Durchlaufzeit stellt Nebenbedingung dar; Auftrag muss bis zu einem vorgegebenen Zeitpunkt abgeschlossen sein

- Eilauftrag: möglichst geringe Durchlaufzeit; Kosten irrelevant
- Eilauftrag mit Kostenbedingung: möglichst geringe Durchlaufzeit; Kosten stellen Nebenbedingung dar; Maximalkosten dürfen nicht überschritten werden
- Fertigstellungsauftrag: Auftrag muss bis zu einem vorgegebenen Zeitpunkt abgeschlossen sein; möglichst geringe Auftragskosten

Durch Instanziierung der auftragsbeschreibenden Klasse wird ein Transportauftragsagent instanziiert und mit den auftragsartspezifischen Verhandlungsverhalten definiert. Anhand der ihn beschreibenden Eigenschaften überprüft der Transportauftragsagent den DF nach geeigneten Transportentitäten. Ausschlaggebend sind dabei allgemeine Kenngrößen der Transportmittel wie die maximale Zuladung oder das vorhandene Transportaufnahmemittel. Diese Informationen tragen die Fahrzeugagenten selbstständig bei ihrer Initialisierung in den DF ein. Als Basis dienen die Informationen des fahrzeugspezifischen Parameterservers, der bei der Applikation des Schichtenmodells auf die Transportentität erstellt wird und die Eigenschaften der Transporteinheit beschreibt. Nachdem der Transportauftragsagent eine Auswahl geeigneter Transporteinheiten aus dem DF getroffen hat, erfolgt gemäß dem *FIPA Contract Net Interaction Protocol* [100] eine Anfrage an die jeweiligen Fahrzeugagenten, bei der die Informationen zur Materialquelle und -senke übermittelt werden. Die angefragten Fahrzeugagenten erstellen daraufhin ein Angebot anhand der spezifischen Fahrzeugeigenschaften und -strategie. Hierfür stellt der Zustandsautomat die Hauptfähigkeit *Forecast* bereit. Der Fahrzeugagent übergibt dem Zustandsautomaten die Position der auftragsspezifischen Materialquelle und -senke. Dieser plant mit einem Pfadplaner die Fahrstrecke und ermittelt die Transportzeit. Die Rückgabewerte des Zustandsautomaten an den Transportagenten sind folglich die notwendige Transportzeit und die voraussichtlich zurückgelegte Strecke. Die Anfahrt zur Materialquelle wird als gesonderter Fahrauftrag vom Agenten geplant, indem er den Pfad von der aktuellen Position bzw. der Endposition des vorherigen Auftrags zur Materialquelle ebenfalls vom Zustandsautomaten abschätzen lässt. Die Leerfahrt der Transportentität zwischen zwei Aufträgen wird folglich immer dem nachfolgenden Auftrag zugeschrieben, analog zur Rüstkostenzuordnung bei Werkzeugmaschinen. Aus der Aggregation beider Fahraufträge ergibt sich eine auftragsspezifische Gesamtstrecke und -zeit. In Kombination mit den im Parameterserver hinterlegten Eigenschaften ist der Fahrzeugagent in der Lage, die intralogistischen Kennzahlen für diesen Auftrag zu berechnen und an den anfragenden Transportauftragsagenten als *Angebot* zu übermitteln.

Der die Verhandlung anstoßende Transportauftragsagent sammelt die unterschiedlichen Angebote der angefragten Fahrzeugagenten ein und bewertet diese gemäß der von der Auftragsklasse vererbten Strategie. Der Fahrzeugagent, der die gewünschten Anforderungen am besten erfüllt, erhält schließlich den Zuschlag zur Auftragsdurchführung, alle anderen Angebote werden zurückgewiesen. Der Fahrzeugagent bestätigt den Erhalt des Zuschlags und lastet den Transportauftrag sowie die notwendige Anfahrt im fahrzeugeignen *Task_Stack* ein. Im *Task_Stack* werden sämtliche bestätigte Fahraufträge der Transportentität gemäß ihrer Reihenfolge gespeichert und anschließend sequenziell abgearbeitet. Die gespeicherten Fahraufträge enthalten dabei die Zielposition und einen klassierenden Parameter, der die Art des Auftrags beschreibt. Bei der Auftragsart wird zwischen drei Typen unterschieden, deren Eigenschaften im Abschnitt 4.1.3 detailliert erläutert sind. Die Speicherung der Startposition ist hingegen nicht notwendig, da für die Anfahrt ein zusätzlicher Fahrauftrag generiert wird und somit die Zielposition des vorherigen Auftrags stets der Startposition des Folgeauftrags entspricht.

Zur Abarbeitung der Fahraufträge verfügt das Softwareschichtenmodells über die Hauptfähigkeit *Acting*. Der Fahrzeugagent übergibt zur Durchführung eines spezifischen Fahrauftrags dessen Zielposition und den klassierenden Auftragsparameter an diesen Zustandsautomaten. Dieser fügt die notwendigen elementaren Fähigkeiten des Fahrzeugs aneinander und arbeitet diese sequenziell ab, bis der Fahrauftrag abgeschlossen ist. Anhand des klassierenden Auftragsparameters ermittelt der Zustandsautomat zudem, welche Aktion bei Erreichen der Zielposition durchgeführt werden muss. Nach erfolgter Auftragsdurchführung übermittelt der Zustandsautomat die zurückgelegte Wegstrecke und die benötigte Durchlaufzeit an den Fahrzeugagenten zurück. Der Agent lastet anschließend den nächsten Fahrauftrag aus dem *Task_Stack* ein, ermittelt die realen Auftragsparameter und korrigiert die im Parameterserver gespeicherten Schätzparameter. Durch diesen Abgleich ist es möglich, die Güte zukünftiger Angebote kontinuierlich zu optimieren. Abschließend informiert der Fahrzeugagent den Transportauftragsagenten über die erfolgreiche Durchführung des Transportauftrags, der sich folglich aus dem AMS austrägt, selbst terminiert und somit das iLES verlässt.

Aufgrund des Aufbaus und des verwendeten Algorithmus zur Allokation der Aufträge innerhalb des iLES ist eine dezentrale, vom Transportauftrag gesteuerte Disposition der Aufträge gewährleistet. Zudem erfolgt die Einlastung der Transportentitäten in Abhängigkeit ihrer Verfügbarkeit und ih-

rer Eigenschaften selbstregelnd anhand einheitlicher transportplattform-übergreifender Zielgrößen, was die Interoperabilität unterschiedlicher Transporthardware innerhalb eines Systems erlaubt. Zudem wird über den Transportauftragsgenerator eine standardisierte Schnittstelle zu anderen IT-Systemen geschaffen, über die Transportaufträge in das iLES eingelastet werden können.

4.1.3 Beschreibung der Auftragsschätzung und -durchführung

Die Abschätzung sowie Durchführung der Transportaufträge erfolgt direkt auf der jeweiligen Transportentität. Die Ebene der Zustandsautomaten des fahrzeugseitigen Softwareschichtenmodells stellt hierfür zwei Hauptfähigkeiten zur Verfügung, den *Forecast* und das *Acting*. Aufgrund der Anforderungen an ein transportplattformunabhängiges Gesamtsystem zur Auftragsallokation ist für beide Hauptfähigkeiten jeweils ein generischer Zustandsautomat definiert. Dieser kombiniert die Grundfähigkeiten der Transportentität anhand dessen Eigenschaften aus dem Parameterserver.

Der Zustandsautomat für die Hauptfähigkeit *Forecast* enthält den Aufruf eines globalen Pfadplaners, um die notwendigen auftragsspezifischen Kenngrößen, die voraussichtliche Durchlaufzeit und die abgeschätzte Wegstrecke zu berechnen, die zugleich den Rückgabewert an den aufrufenden Fahrzeugagenten darstellen. Zur Abschätzung des globalen Pfads wird dem Zustandsautomaten eine Start- und Zielposition übermittelt. Analog den Ausführungen in Kapitel 5 berechnet der globale Pfadplaner des Fahrzeugs den kürzesten Pfad zwischen beiden Positionen anhand eines Wegenetzes des Arbeitsraums. Auf Basis der berechneten Wegstrecke und der im Parameterserver hinterlegten Geschwindigkeitsparameter kann daraus die voraussichtliche Bearbeitungszeit des Fahrauftrags abgeschätzt werden. Für eine exakte Berechnung der Bearbeitungszeit müssten neben den globalen Informationen über den Arbeitsraum noch lokale Informationen über die Wegstrecke in die Berechnung mit einbezogen werden. Vor allem das Verhalten dynamischer Hindernisse ist bei einer solchen Bewertung jedoch schwer extrapolierbar. Um diese potenziellen, schwer vorher-sagbaren Störungen bei der Auftragsdurchführung bereits bei der Auftragsabschätzung pauschal zu berücksichtigen, wird daher die mittlere Geschwindigkeit aus historischen Auftragsdurchführungen bei der Berechnung der Bearbeitungszeit verwendet. Dieses Vorgehen reduziert die Ausführungszeit des *Forecast*-Zustandsautomaten, wodurch einerseits

eine schnelle Rückmeldung der angeforderten Logistik-KPIs an den Transportauftragsagenten erfolgt und andererseits eine höhere Anzahl an Anfragen bearbeitet werden kann. Zudem berechnet der Algorithmus der Hauptfähigkeit *Forecast* lediglich die reine Fahrzeit zwischen der übermittelten Start- und Zielposition. Eine potenzielle Aktion an der Zielposition bzw. ein Handling zur Lastaufnahme wird vom Fahrzeugagenten in Abhängigkeit der Auftragsart pauschal ergänzt.

Der zweite Zustandsautomat innerhalb des fahrzeugseitigen Software-schichtenmodells beschreibt die Hauptfähigkeit *Acting*, der die Abarbeitung eines spezifischen Fahrauftrags steuert. Der Fahrzeugagent übergibt den anstehenden Auftrag aus dem *Task_Stack* an den Zustandsautomaten. Der Auftrag beinhaltet neben der Zielposition die Beschreibung des Auftrags, bei der folgende drei Arten unterschieden werden:

- *grab_trip*: Anfahrt an die Zielposition mit anschließender Lastaufnahme
- *bring_trip*: Anfahrt an die Zielposition mit anschließender Lastabgabe
- *empty_trip*: Anfahrt an die Zielposition ohne anschließende Tätigkeit

Eine Übergabe der Startposition ist nicht notwendig, da der Auftragsagent jede Anfrage eines Transportauftragsagenten in einzelne Transportaufträge unterteilt und dabei die Anfahrt an die Materialquelle als *grab_trip* und die Abgabe an der Materialsenke als *bring_trip* definiert. Somit entspricht die Zielposition des vorherigen Auftrags der Startposition des aktuellen und muss daher, im Gegensatz zum *Forecast*, nicht explizit übermittelt werden.

Nach der Identifizierung der Auftragsart validiert der *Acting*-Zustandsautomat die Zulässigkeit der Auftragseinlastung durch die Überprüfung des Handlingsystems bzw. des Lastaufnahmemittels. Anschließend wird anhand des globalen Pfadplaners der optimale Pfad zwischen der aktuellen Position und der vom Fahrzeugagenten übergebenen Zielposition ermittelt. Gemäß der verwendeten Servicearchitektur, die in Kapitel 5 umfänglich beschrieben ist, wird der globale Pfad und die Informationen der aktuellen Umgebung des Transportroboters an den lokalen Pfadplaner übergeben. Nach Erreichung der Zielposition wird die definierte Tätigkeit durchgeführt und die Abarbeitung des Auftrags an den Fahrzeugagenten übermittelt.

Aufgrund der Aggregation der elementaren Fahrzeugfähigkeiten und der Etablierung von zwei generischen Hauptfähigkeiten (*Forecast* und *Acting*)

innerhalb getrennter Zustandsautomaten, ist die Kommunikation zwischen Fahrzeugagent und der Ebene der Zustandsautomaten standardisiert und transportplattformunabhängig. Weiterhin ist eine parallele Abschätzung und Durchführung eines Auftrags möglich. Die Durchführung eines spezifischen Auftrags wird komplett innerhalb der Ebene der Zustandsautomaten überwacht und abgearbeitet, wodurch die Rechenkapazitäten des Fahrzeugagenten auf die strategischen Entscheidungen ausgerichtet werden können. Zudem ist die Anforderung an ein transportplattformübergreifendes Schichtenmodell dadurch umgesetzt, dass die Zustandsautomaten, die für die Auftragsabarbeitung relevanten Informationen aus dem Fahrzeugparameterserver beziehen, somit bei der Integration auf einen spezifischen Transportroboter nicht appliziert werden müssen.

4.2 Methode zur Beschreibung allgemeingültiger Hauptfähigkeiten von Transportfahrzeugen

Die Beschreibung des fahrzeugseitigen Schichtenmodells in Abschnitt 4.1 verdeutlicht die Aufgabenteilung der unterschiedlichen Ebenen des generischen Softwaremodells der Fahrzeugentitäten zur Auftragsallokation, -schätzung und -durchführung. Die langfristigen strategischen Entscheidungen der einzelnen Transportentitäten erfolgen in der Agentenschicht. Hierfür stellt die Schicht der Zustandsautomaten die beiden Hauptfähigkeiten *Forecast* und *Acting* zur Verfügung, die eine Aneinanderreihung der fahrzeugspezifischen Basisfähigkeiten darstellen, um intralogistische KPIs von Transportaufträgen abschätzen und Fahraufträge durchführen zu können. Dadurch entsteht eine einheitliche Schnittstelle zwischen dem strategischen Fahrzeugagenten und der Ebenen der Zustandsautomaten.

4.2.1 Umsetzung des hybriden Steuerungsparadigmas

Ein essentieller Bestandteil jedes autonomen Roboters ist ein Algorithmus, der das System in Abhängigkeit von übergebenen Befehlen, der aktuellen Situation und den jeweiligen Anforderungen steuert. Grundsätzlich lassen sich diese Kontrollstrukturen in folgenden drei Paradigmen [102] unterteilen:

- deliberativ [103]
- reaktiv [104]
- hybrid

Verwendetes Schichtenmodell

Das in der Methode des Schichtenmodells umgesetzte, hybride Kontrollparadigma verfolgt den Ansatz der Three-Layer Architecture [105], bei dem die Programme eines autonomen Roboters nach dem benötigten Wissen in die Schichten *Deliberator*, *Sequencer* und *Controller* aufgeteilt sind [106]. Die reaktive Schicht des *Controllers*, die in der beschriebenen Methode die elementaren Fähigkeiten des Fahrzeugs darstellt, beinhaltet Algorithmen, die kein permanentes Wissen benötigen und Sensordaten direkt verarbeiten. Die Abläufe werden beim Auftreten von Fehlern nicht geändert, sondern lediglich entsprechende Informationen an den *Sequencer* weitergeleitet. Die Schicht stellt die elementaren Module wie z.B. einen globalen Pfadplaner zur Verfügung. Der *Sequencer*, die Schicht der Zustandsautomaten, besitzt historische Informationen, die zur Auswahl bestimmter elementarer Fähigkeiten notwendig sind. Der Fahrzeugagent entspricht dem *Deliberator*, der für die Planung und Erreichung übergeordneter Ziele verantwortlich ist und daher dem deliberativen Paradigma folgt. Die Hauptfähigkeiten *Forecast* und *Action* werden in einem hierarchischen Zustandsautomaten (Harel-Automaten) abgebildet, bei dem folgenden Zustände unterschieden werden:

- Basiszustand: Zustand ohne weitere Unterzustände
- Sequentieller Zustand: Anordnung aus mehreren Zuständen, die durch eine Exklusiv-Oder-Verknüpfung verbunden sind
- Orthogonale Region: Anordnung bei der eine Und-Verknüpfung der Unterzustände besteht [107].

4.2.2 Zustandsautomat der Hauptfähigkeit *Forecast*

Der Abschnitt 4.1 erläutert bereits, dass in der Ebene der Zustandsautomaten dem Fahrzeugagenten zwei generische Hauptfähigkeiten zur Verfügung gestellt werden. Der *Forecast* dient dabei zur schnellen Abschätzung der logistischen KPIs Auftragsdurchlaufzeit und Wegstrecke eines potenziellen Transportauftrags, wobei der Zustandsautomat lediglich die reine Fahrstrecke zur Schätzung in Betracht zieht. Die Zeiten für mögliche Andock- oder Handlingsvorgänge am Ziel werden bei der Angebotsabgabe vom Agenten pauschal zur Fahrzeit addiert. Bild 17 zeigt den Aufbau des in der Methode migrierten Zustandsautomaten *Forecast* als Zustandsdiagramm.

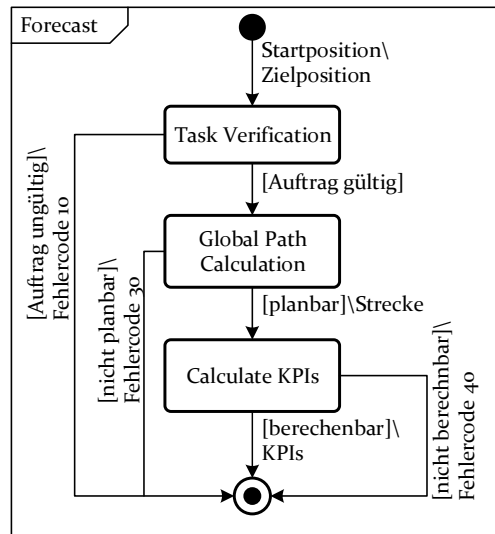


Bild 17: Zustandsdiagramm des Zustandsautomaten Forecast

Der Zustandsautomat erhält vom übergeordneten Fahrzeugagenten den Auftrag zur Abschätzung einer Wegstrecke mit der Übermittlung der auftragsspezifischen Start- und Zielpunkte. Im ersten Schritt validiert der Basiszustand *Task_Verification* die übertragenen Daten und stellt somit sicher, dass lediglich die Koordinaten zweier Punkte vom Agenten übertragen wurden. Die Konzeption des Zustandsautomaten *Forecast* sieht vor, dass die möglichen Fehlerfälle der Basiszustände über einen eindeutigen Fehlercode an die anstoßende Instanz übermittelt werden. Die Interpretation des Fehlers und die Reaktion erfolgen somit ausschließlich auf der Ebene des Fahrzeugagenten. Im Zustandsdiagramm wird dies durch die mit ungültig gekennzeichneten Transitionen und deren jeweiligen Fehlercode als übermittelte Daten dargestellt, die grundsätzlich den Endzustand des Zustandsautomaten auslösen. Der Basiszustand *Global_Path_Calculation*, der anhand eines Weltmodells der Arbeitsumgebung einen Pfad zwischen der übertragenen Start- und Zielposition ermittelt, stellt eine elementare Fähigkeit der Transportentität dar. Die in dieser Forschungsarbeit realisierte Gesamtarchitektur stellt hierfür einen fahrzeugplattformunabhängigen und somit generischen Dienst bereit, dessen Funktionsweise im Kapitel 5 detailliert beschrieben ist. Die Ausgabedaten des globalen Pfadplaners sind im Erfolgsfall die abgeschätzte Strecke und stellen die Transition zum dritten Basiszustand dar. Der Zustand *Calculate_KPIs* berechnet aus der übermittelten Fahrstrecke die voraussichtlich benötigte Transportzeit des Auftrags. Wie bereits beschrieben, wird diese aus Gründen der Performanz anhand der im Fahrzeugparameterservice hinterlegten

mittleren Geschwindigkeit der Transportentität berechnet. Anschließend erreicht der Zustandsautomat seinen Endzustand und übermittelt im Erfolgsfall die ermittelte Strecke und die abgeschätzte Transportzeit an die anstoßende Instanz.

4.2.3 Zustandsautomat der Hauptfähigkeit Acting

Die zweite generische Hauptfähigkeit ist die Durchführung spezifischer Fahraufträge über das *Acting*. Hierbei übermittelt der Agent, wie in Abschnitt 4.1 erläutert, die Zielposition des Auftrags an den Zustandsautomaten. Zudem wird eine Klassierung des Auftrags als *empty*, *grab* oder *bring* durchgeführt. Bild 18 zeigt den Aufbau des umgesetzten Zustandsautomaten *Acting* als Zustandsdiagramm.

Analog zum Zustandsautomaten *Forecast* erfolgt zuerst eine Validierung der vom Fahrzeugagenten übermittelten Daten. Diese bestehen aus den Koordinaten des Zielpunkts und einer Zeichenkette zur Klassifizierung des Fahrauftrags. Wie der *Forecast* verfügt auch das *Acting* über eindeutige Fehlercodes der jeweiligen Basiszustände, die im Fehlerfall dem Agenten eine Interpretation und entsprechende Reaktion erlauben. Im Anschluss an die Validierung der übertragenen Auftragsdaten erfolgt die Identifikation der Auftragsart durch den Basiszustand *Task_Identification* anhand der klassierenden Zeichenkette. Dieser Zustand stellt eine Basisfähigkeit dar, die direkt in der Softwareebene des Zustandsautomaten implementiert ist, da sie sich nicht aus den elementaren Fähigkeiten ableitet. Anhand der Identifizierung der jeweiligen Auftragsart erfolgt die weitere Bearbeitung anhand eines Superzustands, der die orthogonalen Regionen *sm_bring*, *sm_grab* und *sm_empty* enthält, die jeweils selbst einen Zustandsautomaten darstellen. Bild 18 zeigt aus Gründen der Übersichtlichkeit lediglich den Zustandsautomaten *sm_empty*, die beiden anderen sind der Bild 19 zu entnehmen. Da zur Erfüllung des Superzustands die Zustände aller drei orthogonalen Regionen erfüllt sein müssen, werden beim Betreten des Superzustands anhand der Auftragsklassifizierung die beiden nicht durchzuführenden Regionen auf ihren Endzustand gesetzt und lediglich der betreffende Zustandsautomat zur Ausführung gebracht. Alle drei Zustandsautomaten der orthogonalen Regionen haben gemein, dass ihre Abarbeitung anhand von Checkpoints überprüft wird. Somit lässt sich im Fehlerfall einerseits feststellen, an welcher Stelle der Fehler aufgetreten ist und andererseits ermöglicht diese Herangehensweise, dass die Ausführung der Aktion an einer definierten Stelle aufgenommen bzw. wiederholt werden kann.

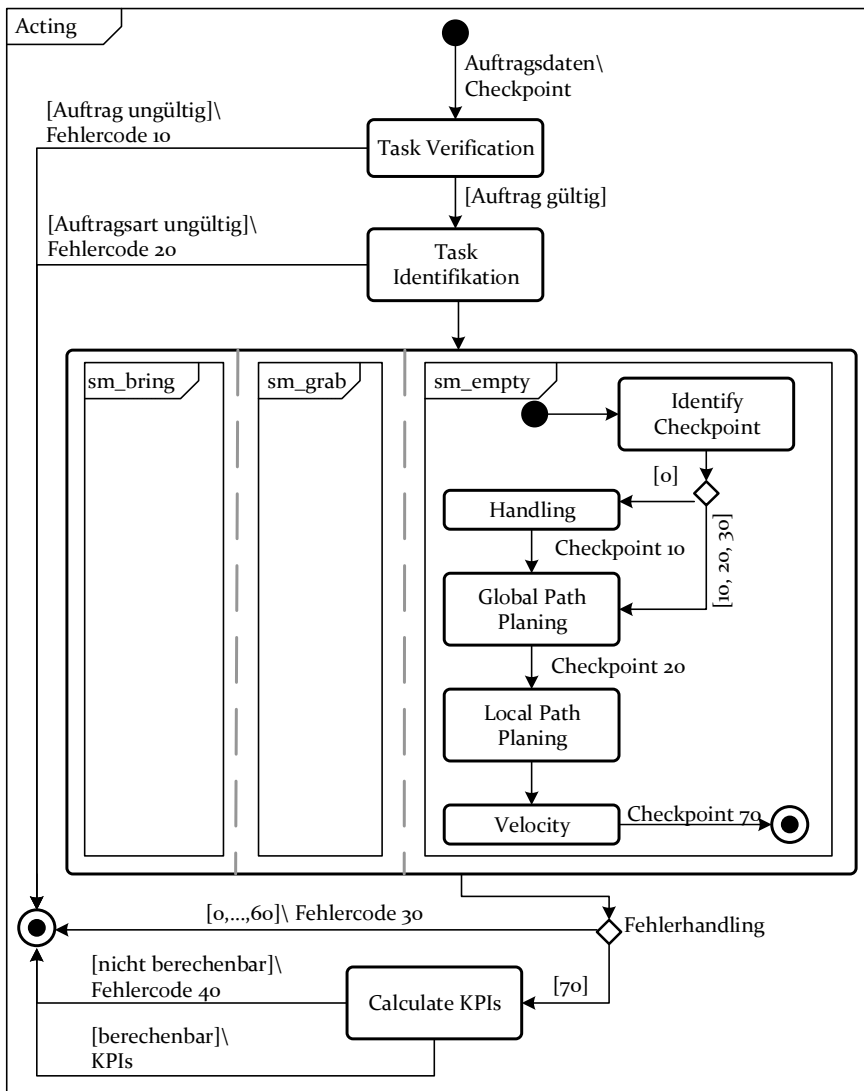


Bild 18: Zustandsdiagramm des Zustandsautomaten Acting

Die Entscheidung, ob im Fehlerfall eine erneute Ausführung des Auftrags auf Basis eines Checkpoints durchgeführt wird, trifft in allen Fällen der Fahrzeugagent. Jedoch verfügt der Zustandsautomat aufgrund des vom Agenten übermittelten Checkpoints über historische Zustandsdaten und reagiert in Kombination mit den aktuellen Sensordaten auf diese, wodurch das Paradigma einer hybriden Struktur erfüllt ist. Der Zustandsautomat *sm_empty* ist zur Bearbeitung von Leerfahrten konzipiert. Diese stellen

eine strategische Auftragsart des Agenten dar, die nicht von einem Transportauftragsagenten initiiert wird. Folglich wird zu Beginn durch den Basiszustand *Handling* sichergestellt, dass der Transportroboter aktuell kein Material geladen hat. Anschließend erfolgt die Transition zum globalen Pfadplaner der identisch zum Pfadplaner des Zustandsautomaten *Forecast* ist. In diesem Fall wird allerdings nicht die Startposition über die Transition übergeben, sondern die aktuelle Position ermittelt. Der Transportroboter berechnet somit stets den globalen Pfad von der aktuellen Position zur Zielposition des Fahrauftrags. Der erzeugte globale Pfad stellt die Transitionsinformation zum lokalen Pfadplaner dar. Die erforschte Gesamtarchitektur stellt auch für den lokalen Pfadplaner einen generischen Dienst bereit, dessen Funktionsweise im Kapitel 5 näher beschrieben ist. Die Ausgabedaten des lokalen Pfadplaners sind im Erfolgsfall die tatsächlich zurückgelegte Strecke und die Transition zum letzten Basiszustand. Der Basiszustand *Velocity* berechnet anhand der zurückgelegten Wegstrecke und der hierfür benötigten Zeit die mittlere Geschwindigkeit des Transportfahrzeugs. Anschließend wird der Zustandsautomat *sm_empty* in den Endzustand gesetzt, wodurch der übergeordnete Superzustand erfüllt ist. Die Rückgabewerte an den Agenten sind die zurückgelegte Wegstrecke, die hierfür benötigte Fahrzeit sowie die mittlere Geschwindigkeit des Transportfahrzeugs.

Bild 19 zeigt die beiden orthogonalen Regionen mit den Zustandsautomaten *sm_bring* und *sm_grab*. Anhand des übertragenen Checkpoints entscheidet der Zustandsautomat, ob es sich bei dem jeweiligen Transportauftrag um einen neuen Auftrag oder einen Wiederholauftrag handelt. Letzteres wird vom Fahrzeugagenten angestoßen, wenn aufgrund einer Fehlermeldung der Agent die Entscheidung trifft, den Auftrag erneut durchzuführen, jedoch die sequentielle Abarbeitung der Basiszustände durch den Zustandsautomaten an einer bestimmten Stelle einsetzen soll. Eine entsprechende Fallunterscheidung wird im Abschnitt 4.3.3 durchgeführt.

Bei einem Neuauftrag wird in beiden Fällen vorerst der Status der Lastaufnahme überprüft. Anschließend erfolgt die Berechnung und Durchführung des globalen und lokalen Pfads zur übermittelten Zielposition. In der erforschten Gesamtarchitektur entspricht die Zielposition von Maschinen zur Lastaufnahme bzw. -abgabe einem definierten Punkt vor der Anlage. Dadurch wird der Andockvorgang von der Pfadplanung entkoppelt, was zu einer höheren Modularisierung des Systemkonzepts führt. Die elementare Fähigkeit des Andockens ist im Basiszustand *Docking* integriert, die Tran-

sition sind die Maschineninformationen zum Andocken. Nach dem erfolgreichen Andocken erfolgt die Lastaufnahme respektive -übergabe und abschließend die Rückfahrt auf die definierte Zielposition vor der Anlage. Abschließend werden die Zustandsautomaten in den Endzustand gesetzt, wodurch der übergeordnete Superzustand erfüllt ist. Die Rückgabewerte an den Agenten sind die zurückgelegte Wegstrecke sowie die hierfür benötigte Fahrzeit. Zudem wird die benötigte Handlingszeit übergeben, die sich aus dem Andockvorgang, der Lastaufnahme bzw. -abgabe und der Freifahrt zusammensetzt.

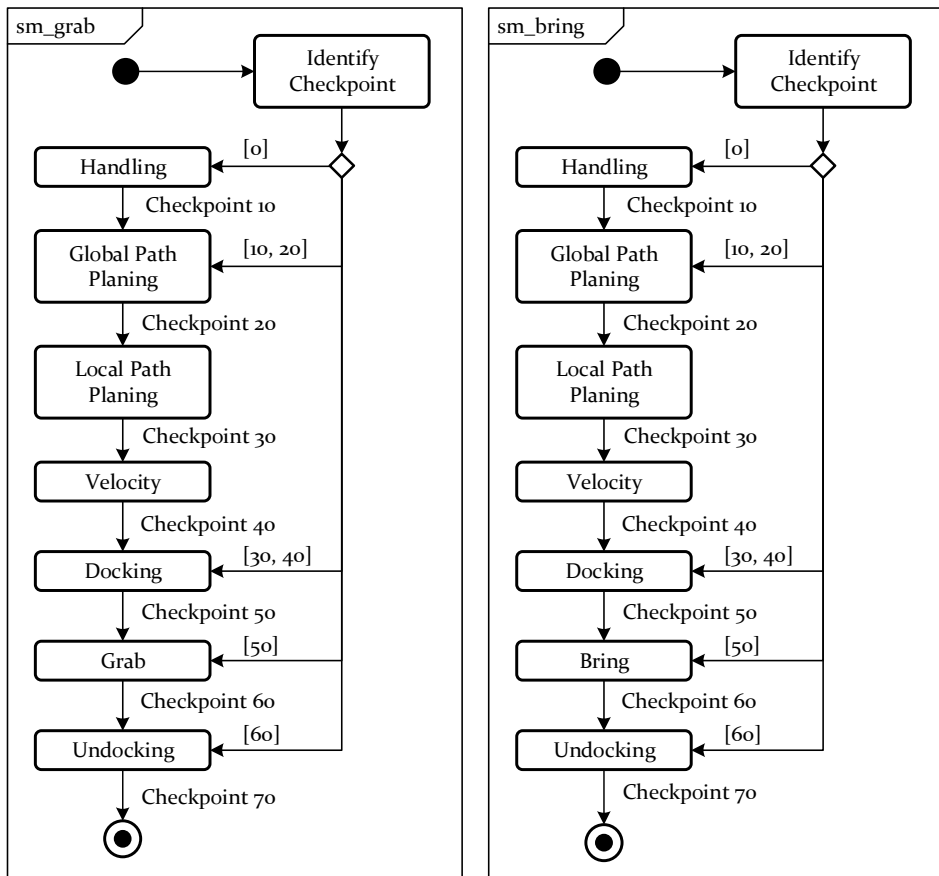


Bild 19: Zustandsdiagramm der orthogonalen Regionen *bring* und *grab*

Durch die beschriebene Umsetzung der beiden generischen Hauptfähigkeiten von Transportfahrzeugen in den hybriden Zustandsautomaten *Forecast* und *Acting* werden die elementaren Fähigkeiten einer Transporteinheit mit entsprechenden Basiszuständen ergänzt, wodurch eine Auftragsabschätzung und -durchführung von einer übergeordneten Instanz

angestoßen werden kann. Die Rückgabewerte bieten zudem die Möglichkeit, eine langfristige strategische Planung in der anstoßenden Instanz durchzuführen.

4.3 Einbindung der Fahrzeugagenten in ein Multi-Agenten-System zur Auftragsallokation

Dem fahrzeugseitigen Schichtenmodell ist zu entnehmen, dass die strategischen Entscheidungen des Fahrzeugs vom Agenten getroffen werden und dieser auf Basis der Rückgabewerte der Hauptfähigkeiten *Forecast* und *Acting* mit seiner Umwelt interagiert. Die Verteilung der Transportaufträge auf die vorhandenen Transportentitäten erfolgt wiederum im iLES, einem MAS, in das die Fahrzeugagenten integriert sind. Die Transportaufträge werden im iLES durch Transportauftragsagenten repräsentiert, die auf Basis der Eigenschaften des Transportauftrags ein spezifisches Verhalten entwickeln und das geeignete Transportmittel auswählen.

4.3.1 Allgemeine Beschreibung von Softwareagenten

Es existiert keine einheitliche Definition eines Softwareagenten, jedoch besteht darin Konsens, dass ein Agent autonom agieren und Entscheidungen treffen können muss [108, 109]. Die VDI-Richtlinie VDI/VDE 2653 [109] gibt folgende Richtlinie an:

"Ein technischer Agent ist eine abgrenzbare (Hardware- oder/und Software-) Einheit mit definierten Zielen. Ein technischer Agent ist bestrebt, diese Ziele durch selbstständiges Verhalten zu erreichen und interagiert dabei mit seiner Umgebung und anderen Agenten."

Der Grad der Autonomie des Programms eines Agenten wird zudem anhand folgender Eigenschaften beschrieben [110]

- autonom: arbeitet unabhängig von Benutzereingriffen
- kognitiv: ist lernfähig und lernt aufgrund zuvor getätigter Entscheidungen bzw. Beobachtungen
- kommunikativ: teilt seine Zustände als Wirkung auf seine Umgebung dieser mit
- modal adaptiv: ändert aufgrund der eigenen Zustände und der Zustände der Umgebung seine eigenen Einstellungen (Parameter und/oder Struktur)
- proaktiv: führt Aktionen aufgrund eigener Initiative aus
- reaktiv: reagiert auf Änderungen der Umgebung

- robust: kompensiert äußere und innere Störungen
- sozial: kommuniziert mit anderen Agenten

Auch in der Beschreibung, Definition und Abgrenzung der Agententypen herrscht eine große nicht standardisierte Begriffsvielfalt. Zusammenfassend lässt sich nach [108] eine Einteilung in drei weitgehend anerkannte Bereiche gemäß Tabelle 3 vornehmen.

Tabelle 3: Einteilung der anerkannten Agententypen (in Anlehnung an [109])

Hauptgruppe	Untergruppe	Kurzbeschreibung
Reaktiver Agent	Einfach	Aktionen werden auf Basis von Sensordaten nach festen Regeln ausgewählt (Bedingungs-Aktions-Regeln)
	Beobachtender Agent	Besitzt Gedächtnis und sammelt Informationen über die Umwelt; Bedingungs-Aktions-Regeln basieren auf Sensordaten und dem Weltbild
	Adaptiver Agent	Verwaltet ein anpassbares Modell der Prozess- und Parameterstruktur; adaptive Regelung möglich
Kognitiver Agent	Zielbasierter Agent	Verwaltet Modell seiner Umwelt; besitzt eine Zielvorgabe; entscheidet auf Basis der Sensordaten, und seines Wissens über Aktionsfolgen; Planung von Teilschritten zur Zielerreichung möglich
	Nutzenbasierter Agent	Besitzt Zielvorgabe; Nutzen aller möglichen und unmöglichen Zustände wird abgebildet; kann bei multipler Zielerreichbarkeit größten Nutzen abschätzen; kann Risikoabschätzungen durchführen

Weiterhin ist die Interaktion der wesentliche Bestandteil eines MAS. Konversationen zwischen Agenten in einem MAS werden nach dem Interaktionsprotokoll strukturiert. Die etablierten Vertreter dieser Protokolle sind [111]:

- Kontraktverfahren: Es werden zwei Agententypen eingesetzt, der Auftraggeber (*Manager*) und der Auftragsnehmer (*Contractor*). Aufgabe des *Managers* ist es, das Problem in Subprobleme zu zerlegen, passende Agenten zur Lösung der Probleme zu finden und auszuschreiben. Jeder an der Ausschreibung interessierte *Contractor* bewirbt sich anschließend auf die Ausschreibung mit einem Angebot. Unter Berücksichtigung des Gesamtzieles bewertet der *Manager* alle bei ihm eingegangenen Bearbeitungsangebote und sendet eine Zuschlagsnachricht an alle *Contractor*, die ihren Beitrag zum Gesamtziel leisten dürfen. Der Vorteil dieses Verfahren besteht darin, dass die Koordination von Aufgaben zwischen Agenten

übernommen wird und eine bestmögliche Zuordnung gewährleistet ist. [112]

- **Auktionsverfahren:** Im Auktionsverfahren wird eine Ware oder Aufgabe in Form einer Versteigerung verteilt, bei der alle interessierten Agenten ein Angebot abgeben dürfen. Die Vergabe erfolgt entweder nach dem Prinzip einer Englischen Auktion, bei der der auktionierende Agent den Preis für die Erledigung einer Aufgabe stetig erhöht, bis ein erster Bieter diesem Preis zustimmt. Die Holländische Auktion wird meist bei der Vergabe von Ressourcen benutzt. Hierbei ruft der auktionierende Agent einen hohen Startpreis für die Ressource aus und verringert diesen stetig, bis der erste Bieter dem Preis zustimmt. Der Vorteil von Auktionen als Koordinationstechnik ist, dass die Menge der ausgetauschten Daten und Informationen zwischen dem auktionierenden und den bietenden Agenten gering ist. Sie eignet sich daher besonders für sehr große Agentenpopulationen.

Tabelle 4: Zusammenfassung der wichtigsten Performativen eines Senders und Empfängers einer Nachricht innerhalb der ACL (in Anlehnung an [114])

Agent	Performativ	Beschreibung
Sender	request	Aufforderung zur Ausführung einer bestimmten Aktion
	inform	Information ohne Absicht einer Aktion
	query_if	Aufforderung zur Durchführung einer Operation bei definierten Startbedingungen
	call for proposal (cfp)	Anfrage eines Vorschlags für eine mögliche Aktion
	propose	Unterbreitung eines Angebots für eine bestimmte Aktion
Empfänger	agree	Übermittlung des Einverständnisses und Rückmeldung des Starts der Aktion
	inform	Information ohne Absicht einer Aktion
	refuse	Ablehnung eines Angebots
	not_understood	Information, dass eine Anfrage nicht verstanden wurde

Bei allen Interaktionsprotokollen wird die Kommunikation der Agenten untereinander durch das Kommunikationsprotokoll standardisiert. Das FIPA-Referenzmodell sieht hierfür die *Agent Communication Language* (ACL) vor, dessen Nachrichten sich aus

- dem Absender der Nachricht,
- der Liste der Empfänger,
- und der Absicht des Senders / Performativen

zusammensetzen. Innerhalb der ACL werden für die Kommunikation zwischen einem sendenden und einem empfangenden Agenten verschiedene

Performative vordefiniert, von denen Tabelle 4 einen Auszug der wichtigsten darstellt. [113]

4.3.2 Ableitung des spezifischen Agentenverhaltens innerhalb des Allokationsalgorithmus

Zur Disposition der Fahraufträge werden neben den obligatorischen Agenten eines MAS, dem AMS und dem DF, im iLES folgende drei Agentenarten verwendet:

- Transportauftragsgenerator (adaptiver Agent): Ist im iLES einmal vorhanden; wird bei der Initialisierung des iLES erzeugt; verwaltet ein anpassbares Modell möglicher Transportauftragsarten; generiert auf Basis eines externen Triggers einen spezifischen Transportauftragsagenten
- Fahrzeugagent (zielbasierter Agent): Anzahl der Fahrzeugagenten entspricht der Anzahl der Transportentitäten; wird bei der Initialisierung einer Transportentität erzeugt; verwaltet ein Modell seiner Umwelt; trifft Entscheidungen auf Basis von Sensordaten und seinem Wissen über die Aktionsfolgen; besitzt Zielvorgaben
- Transportauftragsagent (nutzenbasierter Agent): Anzahl der aktiven Transportaufträge im iLES entspricht Anzahl der aktiven Transportauftragsagenten; wird bei der Einlastung eines Transportauftrags in das iLES vom Transportauftragsgenerator erzeugt; besitzt eine spezifische, der Auftragsart abgeleiteten Zielvorgabe; kann bei multipler Zielerreichung der Transportaufgabe durch mehrere Transportentitäten den größten Nutzen abschätzen

Der Allokationsalgorithmus selbst folgt dem Ansatz des *Contract Net Protocol*, das eine spezifische Form des Kontraktverfahrens darstellt. Die Interaktion der einzelnen Agenten bei der Disposition eines Transportauftrags ist Bild 20 zu entnehmen.

Bei seiner Initialisierung trägt der Fahrzeugagent seine spezifischen Fähigkeiten in den DF ein, indem er eine *inform*-Nachricht sendet. Die spezifischen Fähigkeiten leiten sich dabei aus den Einträgen des fahrzeugspezifischen Parameterservice ab. Übertragen werden

- die maximale Zuladung des Fahrzeugs,
- die maximale Größe des Lastaufnahmemittels,
- die Art des Lastaufnahmemittels inklusive der Spezifikationen,

wodurch die unterschiedlichen, zur Verfügung stehenden Transportentitäten charakterisiert sind.

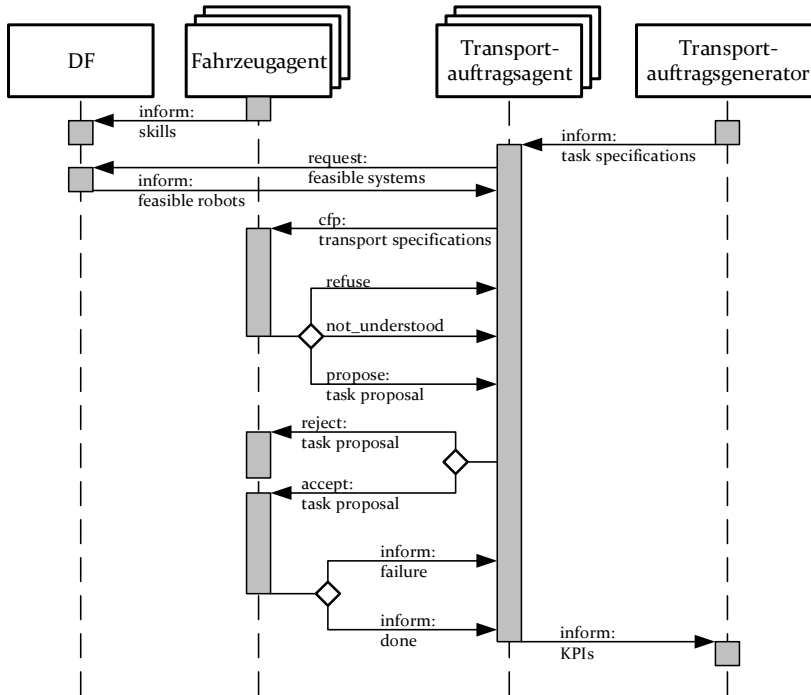


Bild 20: Sequenzdiagramm zur Kommunikation der Agenten innerhalb des iLES zur Disposition von Transportaufträgen auf die Transportentitäten

Transportauftragsgenerator

Durch einen externen Trigger wird der Transportauftragsgenerator dazu veranlasst, auf Basis der Spezifikation eines konkreten Transportauftrags einen Transportauftragsagenten zu erschaffen, dessen Verhalten sich von der vererbten Auftragsklasse (vgl. Abschnitt 4.1.2) ableitet. Jeder in das iLES eingeladete Transportauftrag erhält somit einen Transportauftragsagenten, der ihn bei der anstehenden Disposition repräsentiert und die Vergabe anhand eines adaptierbaren Verhaltens regelt. Die erzeugten Transportauftragsagenten fordern mit Hilfe einer *request*-Nachricht an den DF die Kontaktdaten der Fahrzeugentitäten an, die anhand ihrer Spezifikation für die Durchführung des Auftrags in Frage kommen. Dabei werden die Masse, die Größe und die Anforderungen an das Lastaufnahmemittel mit den Einträgen der Fahrzeuge verglichen. Die Rückgabe des Vergleichs erfolgt über eine *inform*-Nachricht des DF an den anfragenden Transportauftragsagenten.

Transportauftragsagent - Anfrage

Die Disposition erfolgt dann über ein *Contract Net Protocol* welches der Transportauftragsagent anstößt. Dabei wird eine *cfp*-Nachricht an die potenziellen Transportentitäten gerichtet und die Eigenschaften übermittelt, die den Auftrag charakterisieren. Die Materialquelle und -senke sind obligatorisch, die vorgegebene Fertigstellungszeit mit zusätzlichen Informationen, ob eine Wertsteigerung oder Minderung bei Unterschreitung respektive Überschreitung vorliegt, ist optional. Bei Bedarf übermittelt der Agent eine maximale Transportgeschwindigkeit des Auftrags, um somit z. B. normative Vorgaben bei Transporten mit einem erhöhten Gefahrenpotenzial zu erfüllen. Wird keine maximale Transportgeschwindigkeit übermittelt, berechnen die Fahrzeugagenten die Durchführung des Auftrags anhand der im jeweiligen Parameterserver hinterlegten mittleren Geschwindigkeit.

Fahrzeugagent - Anfrage

Die Fahrzeugagenten ergänzen den angefragten Transportauftrag zwischen der Materialquelle und -senke mit einem vorgelagerten Fahrauftrag von der aktuellen Position bzw. der Endposition des potenziell vorausgehenden Transportauftrags, um die Anfahrt zu berücksichtigen. Aus beiden Fahraufträgen ergeben sich über den *Forecast* die zurückgelegten Strecken und die hierfür benötigten Zeiten, die der Agent aggregiert und mit einer pauschalen Handlingszeit für die Lastaufnahme und Abgabe ergänzt. Zur Berechnung der spezifischen Transportkosten werden die Zielgrößen des Auftrags mit den im Parameterservice definierten Koeffizienten des Fahrzeugs verrechnet.

Die einzelnen Koeffizienten können den Wert 0 annehmen, falls diese Kostenart nicht berücksichtigt werden soll. Der Koeffizient k_{meter} berücksichtigt Kosten, die beim Fahrzeug je zurückgelegter Wegstrecke, wie z. B. der Verschleiß von Antriebsteilen, anfallen. Mit dem Koeffizient k_{mass} werden Kosten verrechnet, die aufgrund einer transportierten Masse bezogen auf die Wegstrecke anfallen. Der Koeffizient k_{time} berücksichtigt zeitbezogene Kostenposten, wie z. B. die Abschreibung des Fahrzeugs. Zur Realisierung einer linearen Abschreibung kann dieser Faktor zudem mit einer zeitlichen Degression versehen werden, die vor allem beim Einsatz einer größeren Flotte baugleicher Systeme eine übergeordnete Strategie erlaubt. Alle vom Fahrzeugagenten gesammelten Auftrags-KPIs werden anschließend als *propose*-Nachricht an den anfragenden Transportauftragsagenten versandt. Durch die Belegung der Koeffizienten wird für jeden Fahrzeugtyp ein spezifisches Angebotsverhalten erzeugt. Neben der Abgabe eines Angebots kann der Fahrzeugagent zudem die Anfrage zurückweisen (*refuse*-

Nachricht), falls diese z. B. nicht durchführbar ist, da der globale Pfadplaner keine Lösung findet. Oder mit einer *not_understood* Nachricht anzeigen, dass der empfangene Parametersatz nicht der definierten Ontologie entspricht.

$$c_{task} = (x_{initiate} + x_{operate}) \times k_{meter} + (x_{operate} \times m_{task}) \times k_{mass} + \left(\frac{x_{initiate}}{v_{initiate}} + \frac{x_{operate}}{v_{operate}} + 2 \times t_{handling} \right) \times k_{time} \quad (6)$$

mit

c_{task} : Gesamtkosten Auftrag [€]

$x_{initiate}$: Anfahstrecke zur Materialquelle [m]

$x_{operate}$: Fahrstrecke des Auftrags [m]

m_{task} : Masse des Auftrags [kg]

$v_{initiate}$: mittlere Geschwindigkeit des Fahrzeugs $\left[\frac{m}{s} \right]$

$v_{operate}$: mittlere Transportgeschwindigkeit $\left[\frac{m}{s} \right]$

$t_{handling}$: mittlere Handlingszeit [s]

k_{meter} : Fahrzeugkoeffizient Strecke $\left[\frac{€}{m} \right]$ mit $k_{meter} \in [0 \dots \infty[$

k_{mass} : Fahrzeugkoeffizient Masse $\left[\frac{€}{m \cdot kg} \right]$ mit $k_{mass} \in [0 \dots \infty[$

k_{time} : Fahrzeugkoeffizient Zeit $\left[\frac{€}{s} \right]$ mit $k_{time} \in [0 \dots \infty[$

Transportauftragsagent - Vergabe

Die Vergabe des Auftrags übernimmt der zugewiesene Transportauftragsagent auf Basis der rückgemeldeten Angebote der Fahrzeugagenten und den darin enthaltenen KPIs. Bei den Auftragsarten Kostenauftrag und Eilauftrag geschieht dies trivial anhand der Reihenfolge der Auftragskosten respektive des Endzeitpunkts. Bei einem Kostenauftrag mit Zeitbedingung erfolgt der Zuschlag an die Transportentität mit den geringsten Auftrags-

kosten, die eine Fertigstellung bis zum notwendigen Endzeitpunkt sicherstellt. Bei einem Fertigstellungsauftrag wird hingegen die Einhaltung des vorgegebenen Endzeitpunkts gegenüber den Auftragskosten stärker gewichtet. Ein Eilauftrag mit Kostengewichtung sortiert die übertragenen Angebote nach ihrem Endzeitpunkt und erteilt die Durchführung an die Entität, die einen definierten Kostenrahmen nicht überschreitet. Sowohl der Wertsteigerungs- als auch der Wertminderungsauftrag priorisieren die Einhaltung des definierten Endzeitpunkts. Ersterer gewichtet zudem eine frühere Abarbeitung mit einem zeitlichen Bonus, während letzterer einen Malus vergibt. Dieses auftragsindividuelle Verhalten des Transportauftragsagenten ermittelt somit das für den Auftrag geeignetste Angebot und teilt den Zuschlag an den anbietenden Fahrzeugagenten mit einem *agree-proposal* mit, alle anderen Anbieter erhalten ein *reject-proposal*.

Fahrzeugagent - Auftragsabarbeitung

Bei der Zurückweisung des Angebots oder dem Zuschlag zur Auftragsdurchführung erhält der Fahrzeugagent lediglich die Auftragsnummer vom Transportauftragsagenten, anhand derer die Zu- respektive Absage den bereits vorhandenen Auftragsparametern zugeordnet wird. Der Fahrzeugagent plant sowohl den Fahrauftrag als auch den zugehörigen Anfahrauftrag gemäß dem Angebot im *Task_Stack* ein und bringt diese zum geplanten Zeitpunkt durch Übermittlung an das *Acting* zur Ausführung. Als Rückgabewerte erhält er vom Zustandsautomaten die tatsächlich zurückgelegte Wegstrecke und die Transport- sowie Handlingszeit. Hieraus wird die mittlere Geschwindigkeit ermittelt und die Parameter im Parameterserver aktualisiert. Durch diese Parameteraktualisierung wird die Genauigkeit der Angebotsabgaben des Fahrzeugs permanent gesteigert, ohne dass einzelne Ausreißer ein zu hohes Gewicht erhalten.

Als Mittelungsverfahren kommen drei unterschiedliche Verfahren mit unterschiedlichen Charakteristika hinsichtlich Einschwingverhalten, Robustheit gegen Ausreißer und Überschwinger in Betracht. Bei der ersten Art der Mittelung handelt es sich um einen exponentiell geglätteten Mittelwert [114]. Hierbei wird der neue Mittelwert \bar{x}_k aus dem Messwert x_k , der mit dem Gewichtungsfaktor α bewertet wird, und dem bisherigen Mittelwert \bar{x}_{k-1} , der mit dem Komplement $(1 - \alpha)$ gewichtet wird, berechnet.

$$\bar{x}_k = \alpha \times x_k + (1 - \alpha) \times \bar{x}_{k-1} \text{ mit } \alpha \in [0; 1] \quad (7)$$

Eine Erweiterung des exponentiell geglätteten Mittelwerts stellt die exponentielle Glättung zweiter Ordnung dar [114]. Bei diesem Verfahren werden die Werte zunächst analog einer weiteren Mittelung unterzogen, wobei $\bar{\bar{x}}_k$

dem gemittelten Wert zweiter Ordnung entspricht. Anschließend ergibt sich der Schätzwert \tilde{x}_{k+1} für die nächste Iteration.

$$\bar{\bar{x}}_k = 2 \times \bar{x}_k + (1 - \alpha) \times \bar{\bar{x}}_{k-1} \text{ mit } \alpha \in [0; 1] \quad (8)$$

$$\tilde{x}_{k+1} = 2 \times \bar{x}_k - \bar{\bar{x}}_k \quad (9)$$

Die dritte Art stellt die Berechnung des einfachen gleitenden Mittelwertes dar [114]. Hierbei wird der Mittelwert \bar{x}_k über die letzten n vergangenen Messwerte x_k gebildet.

$$\bar{x}_k = \frac{1}{n} \sum_{i=0}^{n-1} x_{k-i} \quad (10)$$

Eine Empfehlung, welches Mittelungsverfahren für intralogistische Transportentitäten am geeignetsten ist, gibt Abschnitt 7.2.3.

Transportauftragsagent - Abschluss

Nach der Auftragsdurchführung erhält der Transportauftragsagent zudem eine *inform*-Nachricht des Fahrzeugagenten, falls der Auftrag erfolgreich durchgeführt wurde, bzw. eine Fehlermeldung, wenn bei der Ausführung ein Fehler auftritt. Der Transportauftragsagent terminiert sich schließlich selbst.

Der umgesetzte Auftragsallokationsalgorithmus im iLES zeigt, dass die Initiierung einer Disposition stets vom Agenten des Transportauftrags ausgelöst wird und anhand der spezifischen Zielvorgaben erfolgt. Die Fahrzeuge handeln während der Vergabe ebenfalls nach einer eigenen Zielvorgabe und geben ein entsprechendes, für jeden Auftrag individuelles Angebot ab. Die Einlastung ist jedoch ausschließlich dem Transportauftragsagenten vorbehalten.

4.3.3 Ableitung des spezifischen Fahrzeugagentenverhaltens durch Kommunikation mit den Zustandsautomaten

Das Verhalten des Fahrzeugagenten leitet sich neben der Interaktion mit den Agenten des iLES auch aus der Kommunikation mit den Zustandsautomaten *Forecast* und *Acting* und deren Rückgabewerte ab. In Abhängigkeit der Fehlercodes und Checkpoints plant der Agent das weitere Vorgehen bei der Angebotsberechnung, respektive bei der Abarbeitung des Transportauftrags.

Fahrzeugagentenverhalten bei Interaktion mit Forecast

Gemäß des Abschnitts 4.2.2 übergibt der Fahrzeugagent zur Abschätzung eines Fahrauftrags eine Start- und Zielposition an den Zustandsautomaten *Forecast* und erhält als Rückgabewerte entweder eine Wegstrecke und eine Zeit oder einen Fehlercode. Tabelle 5 stellt die potenziellen Ursachen der definierten Fehlercodes und das daraus resultierende Agentenverhalten gegenüber.

Tabelle 5: Ableitung des Verhaltens des Fahrzeugagenten in Abhängigkeit der Rückgabewerte des Zustandsautomaten *Forecast*

Fehler- code	Interpretation und mögliche Ursachen	Reaktion Fahrzeugagent
10	Auftragsparameter ungültig; grundlegender Fehler in der Systemsoftware;	Fahrzeug in Wartungsmodus setzen; Rückgabe einer not_understood- Nachricht an anfragenden Transportauftragsagenten
30	Globaler Pfad nicht berechenbar;	Rückgabe einer refuse-Nachricht an anfragenden Transportauftragsagenten
40	Benötigte Zeit ist aus ermittelter Wegstrecke nicht berechenbar; Fehler bei der Initialisierung der Systemsoftware;	Fahrzeug in Wartungsmodus setzen; Rückgabe einer refuse-Nachricht an anfragenden Transportauftragsagenten

Wie der Tabelle zu entnehmen ist, liegt bei den Fehlercodes 10 und 40 ein Softwareproblem vor, dass durch einen Servicemitarbeiter behoben werden muss. Im ersten Fall besteht ein grundsätzliches Problem bei der Übergabe der Start- und Zielparameter, was einer Überarbeitung des Programmcodes bedarf. Durch die erforschte Methode eines generischen Schichtenmodells auf der Fahrzeugseite ist bei der Applikation der Software jedoch keine Adaption der Kommunikation zwischen dem Fahrzeugagenten und den Zustandsautomaten notwendig, wodurch die Eintrittswahrscheinlichkeit dieses Fehlerfalls sehr gering ist. Im Falle des Fehlercodes 40 wurde bei der Initialisierung des Fahrzeugs und der Befüllung des Fahrzeugparameterservers ein ungültiger Wert für die mittlere Fahrzeuggeschwindigkeit eingetragen, was durch eine Neuparametrisierung behoben werden kann. In beiden Fällen setzt der Fahrzeugagent das Fahrzeug in den Wartungsmodus. Der Fehlercode 30 stellt hingegen einen in der Praxis wahrscheinlichen Rückgabewert dar. In diesem Fall ist der globale Pfadplaner des Fahrzeugs nicht in der Lage, einen gültigen Pfad zwischen der Startposition und dem Ziel zu finden. Der Fahrzeugagent lehnt folglich die Anfrage des Transportauftragsagenten mit einer *refuse*-Nachricht ab.

Fahrzeugagentenverhalten bei Interaktion mit Acting

Analog wird das Fahrzeugagentenverhalten auf Basis der Rückgabewerte des Zustandsautomaten *Acting* beeinflusst. Die Rückmeldung möglicher Fehlerzustände setzt sich gemäß des Abschnitts 4.2.3 aus einem Fehlercode und einem Checkpoint zusammen. In Tabelle 6 werden die potenziellen Ursachen und das daraus resultierende Agentenverhalten gegenübergestellt.

Tabelle 6: Ableitung des Verhaltens des Fahrzeugagenten in Abhängigkeit der Rückgabewerte des Zustandsautomaten Acting

Fehler- code / Check- point	Interpretation und mögliche Ursachen	Reaktion Fahrzeugagent
10	Auftragsparameter ungültig; grundlegender Fehler in der Systemsoftware;	Fahrzeug in Wartungsmodus setzen; Rückgabe einer not_understood-Nachricht an anfragenden Transportauftragsagenten
20	Fahrauftragsart ungültig;	Grundlegender Fehler in der Systemsoftware; Fahrzeug in Wartungsmodus setzen;
30	Lastaufnahmemittel / Handling nicht für Auftragsart vorbereitet	Fahrzeug in Wartungsmodus setzen; Statusüberprüfung auslösen
30 10	Globaler Pfad nicht berechenbar;	Weltmodell aktualisieren; Wiederholung des Auftrags mit Übergabe Checkpoint 10
30 20	Lokaler Pfad nicht durchführbar;	Weltmodell aktualisieren; Wiederholung des Auftrags mit Übergabe Checkpoint 10
30 40	Feinpositionierung an Andockposition nicht erfolgreich	Wiederholung des Auftrags mit Übergabe Checkpoint 10
30 50	Lastaufnahme / -übergabe nicht erfolgreich	Wiederholung des Auftrags mit Übergabe Checkpoint 50
30 60	Freifahrt aus Andockposition nicht erfolgreich	Wiederholung des Auftrags mit Übergabe Checkpoint 60
40	Benötigte KPIs nicht berechenbar;	Fehlende Zugriffsberechtigung; Serviceauftrag auslösen

Die Nomenklatur der Fehlercodes zwischen *Forecast* und *Acting* ist gleich gehalten, damit ähnliche Fehler dieselbe Bezeichnung erhalten und ähnliche Reaktionen des Agenten ausgelöst werden. Eine Sonderform stellt der Fehlercode 30 in Kombination mit den Checkpoints dar. Die Zusatzinformation dient dem Agenten dazu, einen Fehlerfall bei der Auftragsabarbeitung zu identifizieren und seine Reaktion entsprechend zu steuern. Die Checkpoints 10 und 20 zeigen an, dass die Berechnung des globalen oder

lokalen Pfads nicht möglich ist. Im praktischen Umfeld kann dies durch Hindernisse verursacht werden, die bei der Planung nicht kartiert waren. In beiden Fällen erfolgt eine Wiederholung des Auftrags mit der Berechnung eines globalen Pfads auf Basis des aktualisierten Weltmodells. Durch den Checkpoint 40 wird eine fehlerhafte Positionierung des Systems zur Andockstelle gemeldet. Auch in diesem Fall wird der Auftrag erneut mit dem Checkpoint 10 ausgeführt. Dadurch positioniert sich die Transportentität erneut vor der Anlage und wiederholt den Andockvorgang. Dieses Verhalten wird zudem nach einer bestimmten Anzahl an Wiederholungen abgebrochen und das System in einen Störungsmodus versetzt. Checkpoint 50 signalisiert eine fehlerhafte Lastaufnahme bzw. -abgabe und hat die Wiederholung des Vorgangs zur Folge. Der Checkpoint 60 zeigt an, dass die Freifahrt der Transportentität aus der Andockstation nicht erfolgreich war. Dieser Vorgang wird einmalig wiederholt. Im erneuten Fehlerfall beendet der Transportagent den Transportauftrag und versucht den Folgeauftrag auszuführen.

Strategische Fahraufträge des Fahrzeugagenten

Neben der Abschätzung von Fahraufträgen und dessen Durchführungen, ist der Fahrzeugagent zudem befähigt, strategische Fahraufträge durchzuführen. In der erforschten Methode des Schichtenmodells sind folgende strategische Fahrten definiert:

- Ladefahrt: Anhand der kontinuierlichen Überprüfung des Ladezustands wird die Kapazität des Energiespeichers überprüft und bei Unterschreitung eines Mindestladezustands eine Fahrt zur Ladeposition eingeplant. Auf diese folgt im *Task_Stack* ein Auftragsblocker, der der voraussichtlichen Ladezeit entspricht.
- Positionierungsfahrt: Erhält der Auftragsagent bei Auftragsverhandlungen wiederholt Zurückweisungen seiner Angebote und gilt in allen Fällen $x_{initiate} > x_{operate}$ steht die Transportentität mit hoher Wahrscheinlichkeit an einer strategisch ungünstigen Ausgangsposition. In diesem Fall definiert der Fahrzeugagent eigenständig eine Fahrt an eine zuvor definierte Position, um bei den folgenden Verhandlungen eine günstigere Ausgangsposition zu erreichen. Ebenso kann bei einer fehlenden Funkverbindung eine alternative Position im Arbeitsraum angefahren werden.

Beide strategischen Fahraufträge werden durch die Einlastung einer Leerfahrt direkt vom Agenten ohne externen Trigger angelegt.

4.4 Zusammenfassung und Beurteilung

Grundlage des erforschten Schichtenmodells zur dezentralen und selbstregelnden Auftragsallokation bildet das in Abschnitt 4.1 vorgestellte Softwareschichtenmodell der Fahrzeuge und die Interaktion der Agenten im iLES. Die zentrale Eigenschaft der Architektur ist die generische Beschreibung der Eigenschaften von mobilen Transportentitäten mit einem Lastaufnahmemittel oder einem Greifersystem, um den Applikationsaufwand der Fahrzeugsoftware auf ein Minimum zu reduzieren. Durch diese Vorgehensweise wird eine eindeutige Schnittstelle zwischen dem Fahrzeugagenten und der Ebene der Zustandsautomaten geschaffen, wodurch die Entscheidungslogik und Strategie des Agenten von der physischen Plattform entkoppelt ist. Somit entsteht eine uniforme Softwarearchitektur für sämtliche Arten von intralogistischen Transportrobotern. Weiterhin wird die Entscheidung zur Disposition eines Transportauftrags auf eine spezifische Transportentität vom Transportauftragsagenten getroffen, wodurch die Strategie und Zielvorgabe des Auftrags bei der Vergabe im Vordergrund steht. Durch diesen Ansatz entsteht innerhalb des iLES ein Auftragsallokationsalgorithmus, der sich durch seine Universalität und Vergabestrategie von den bekannten Ansätzen abhebt.

Das Kernelement der in Abschnitt 4.2 erläuterten Methode zur Beschreibung und Definition von übergeordneten Hauptfähigkeiten eines Transportroboters führt zur Ausbildung der hybriden Zustandsautomaten *Forecast* und *Acting*. Diese Fähigkeiten sind notwendig, damit einerseits die auftragsspezifischen logistischen KPIs abgeschätzt werden können und der Fahrzeugagent in die Lage versetzt wird, an Auftragsvergaben mitzuwirken und sein Verhalten zu planen. Andererseits ermöglicht das *Acting* die Durchführung von spezifischen Transportaufträgen und eine Rückkopplung zu den KPIs des Angebots, um die Angebotsqualität der jeweiligen Transportentität kontinuierlich zu verbessern. Die Definition der hybriden Zustandsautomaten ist zudem von der physischen Ausführungsform des Transportroboters losgelöst. Die Applikation auf das reale Zielsystem erfolgt anhand eines einheitlichen fahrzeugseitigen Parameterservers, der das Verbindungselement zwischen den generischen Basiszuständen und den Aktoren sowie Sensoren herstellt. Somit bildet die Schicht der Zustandsautomaten eine allgemeingültige Beschreibung der Fähigkeiten eines Transportfahrzeugs und verfolgt damit den Ansatz einer übergreifenden, nicht proprietären Fahrzeugsteuerung.

Konsequent wird dieser Grundsatz auch bei der Definition und Implementierung der an der Disposition beteiligten Softwareagenten umgesetzt. Der

Abschnitt 4.3 beschreibt die Verhandlung zwischen den Agenten und die Disposition der Aufträge anhand logistischer KPIs. Sowohl die Transportauftragsagenten, als auch die Fahrzeugagenten verfolgen dabei eigenständige Ziele, wobei stets die Auftragsziele für die Vergabe entscheidend sind. Der Fahrzeugagent ist zudem in Kombination mit den Zustandsautomaten in der Lage, strategische Entscheidungen zu treffen und eigenständige Fahrten anzuweisen, um mittelfristig die Attraktivität des Transportmittels zu steigern. Durch die Segmentierung der Systemstrategie auf die einzelnen Agenten entsteht eine Architektur, bei der jedes Einzelsystem zu einem individuellen Optimum strebt. Auch wenn die einzelnen Entscheidungen der Entitäten innerhalb des Systems trivial und nachvollziehbar sind, ist ab einer gewissen Anzahl an Transportsystemen und -aufträgen und der damit einhergehenden Fülle an sich gegenseitig beeinflussenden Einzelentscheidungen, das Gesamtsystem nicht mehr deterministisch. Abhilfe für diese Restriktion bietet die Einbindung des Allokationsalgorithmus in eine ereignisdiskrete Simulationsumgebung, wie in Kapitel 6 beschrieben. Weiterhin ist durch den Entfall einer übergeordneten Instanz gewährleistet, dass unterschiedliche Transportsysteme mit unterschiedlichen Strategien und Eigenschaften in einer gemeinsamen Architektur existieren können. Der Transportauftrag kann somit selbst das für ihn geeignetste Transportmittel auswählen, ohne dass hierfür eine explizite Regel besteht. Durch diese standardisierte Kommunikation und Softwarearchitektur ist das iLES sowohl in der Anzahl der Transportentitäten als auch der eingesetzten Fahrzeugarten skalierbar, was sich in dieser Form von den bekannten Ansätzen von FTS grundsätzlich unterscheidet.

5 Verteilte Servicearchitektur zur Digitalisierung des Arbeitsumfelds, der Routen- und Pfadplanung

Die Abkehr von einer klassischen Leitsteuerung bei FTS zur Realisierung eines flexiblen und autonom agierenden Intralogistiksystems steht seit einigen Jahren im Fokus wissenschaftlicher Bemühungen. Neben der Auftragsverteilung adressieren die Ansätze ebenfalls eine verteilte Erfassung des Arbeitsraums als Basis der Routen- und Pfadplanung. Den Restriktionen der zentralen Fahrzeugkoordination wird durch dezentrale Sensorkonzepte begegnet. Dabei erfassen hochautonome Transportroboter ihr Umfeld und gleichen die Daten mit einem historischen Weltmodell ab, wodurch eine Selbstortung im Arbeitsraum realisiert wird. Bei der Routen- und Pfadplanung zeigen die bekannten Systeme jedoch Schwächen, da die Planung in den Bereichen, welche die On-board-Sensoren temporär nicht erfassen, auf historischen Daten beruht. Diesen Nachteil adressieren Systeme mit einem verteilten, in der Infrastruktur integrierten Sensorkonzept, da sie ein echtzeitbasiertes Abbild der Arbeitsumgebung generieren. Die aktuellen Konzepte erweisen sich jedoch als Insellösungen, da die Sensordaten, die von einer zentralen Instanz zu einem Weltmodell zusammengefügt werden, für eine spezifische Fahrzeugapplikation aufbereitet sind und innerhalb eines definierten Transportsystems für die Routen- und Pfadplanung verwendet werden. Eine plattformübergreifende Nutzung des Weltmodells ist nicht gegeben. Die Aufgabenstellung, anhand eines infrastrukturellen Sensorkonzeptes das Arbeitsumfeld zu digitalisieren und mittels einer verteilten Serviceschicht eine plattformübergreifende Routen- und Pfadplanung zu realisieren, wird in den nachfolgend erläuterten Forschungsarbeiten aufgenommen und deren Lösungsmöglichkeiten erörtert. [115]

Die verteilte Architektur der entwickelten intralogistischen Serviceschicht zur Routen- und Pfadplanung wird in Abschnitt 5.1 vorgestellt und die zentralen Komponenten und Dienste beschrieben. Die Methode zum Generieren eines Weltmodells mit Hilfe der dezentralen Infrastruktursensoren ist Inhalt von Abschnitt 5.2. Die Dienste, die eine typspezifische Routenplanung anhand der Parameter des Transportroboters erzeugen und diesem zur Verfügung stellen, werden in Abschnitt 5.3 erläutert. Der Abschnitt 5.4

beschreibt die generische, fahrzeugtypspezifische Pfadplanung. Abschließend zusammengefasst werden die gewonnenen Erkenntnisse in Abschnitt 5.5.

Einzelne Aspekte der Ausführungen der Folgeabschnitte 5.1 bis 5.4 sowie des Abschnitts 7.3, der die Umsetzung und Evaluierung der beschriebenen Methode umfasst, sind an den am Lehrstuhl FAPS entstandenen und vom Autor initiierten studentischen Arbeiten [S26–S49] angelehnt. Bestandteile und Vorarbeiten der dargestellten Methoden und Erkenntnisse wurden der internationalen Fachwelt auch in den vom Autor erstellten Veröffentlichungen [P10–P17] zugänglich gemacht.

5.1 Architektur des verteilten Systems

Die zentrale Anforderung an die Systemarchitektur ist deren Unabhängigkeit von proprietären Lösungen einzelner Hersteller und die Interoperabilität zwischen unterschiedlich konzipierten Transportentitäten. Zudem ist es notwendig, dass die Architektur sowohl im Bereich der dezentralen Erfassung des Arbeitsraums skalierbar ist, als auch die generischen Dienste so verteilt sind, dass die Größe der Transportflotten und die Anzahl unterschiedlicher Typen von Transportentitäten nicht beschränkt ist. Um diese Anforderungen zu erfüllen, müssen die Hauptfunktionen Arbeitsraumerfassung, Routen- und Pfadplanung in Teilfunktionen getrennt und diese als übergreifende Dienste im System orchestriert werden.

5.1.1 Aufbau und Funktionsweise der Dienste

In der untersuchten verteilten Servicearchitektur wird das Arbeitsumfeld der mobilen Transportroboter mit Infrastruktursensoren erfasst und optional durch On-board-Sensoren erweitert. Wie bereits in Abschnitt 3.3.2 erläutert, bietet der Einsatz von Infrastruktursensoren den Vorteil, dass ein umfassendes Abbild des Arbeitsraums erstellt werden kann, das nicht auf historischen Sensordaten beruht, sondern die aktuelle Szenerie in Echtzeit wiedergibt. Dadurch ist es möglich, die Routen der Fahrzeuge situationsbedingt zu gestalten. Zudem können die Fahrzeuge auf Objekte reagieren, die ihre Bahn kreuzen aber mit On-board-Sensoren nicht erfasst werden können. Ein typisches Beispiel sind Kreuzungsbereiche, die von den Fahrzeugen durchfahren werden müssen, allerdings von den Fahrzeugsensoren aufgrund von Hindernissen nicht eingesehen werden können. Das Fahrzeug muss somit mit verringerter Geschwindigkeit in den Kreuzungsbe-

reich einfahren bis die Sensoren den Kreuzungsbereich vollständig erfassen. Durch diesen Verzögerungs- und Beschleunigungsvorgang wird einerseits der relative Energieverbrauch des Fahrzeugs erhöht und andererseits die Durchlaufzeit des Auftrags verlängert. Die Erfassung des Arbeitsraums durch Infrastruktursensoren behebt diesen Umstand, da dem Fahrzeug die notwendigen Informationen über Objekte im Kreuzungsbereich zur Verfügung gestellt werden und es damit frühzeitig auf mögliche Kollisionssituationen reagieren kann. In Anlehnung an die wissenschaftlichen Vorarbeiten am Lehrstuhl FAPS [14] basiert der in Bild 21 dargestellte Ansatz der Servicearchitektur auf einer Erfassung des Arbeitsraums mit Deckenkameras.

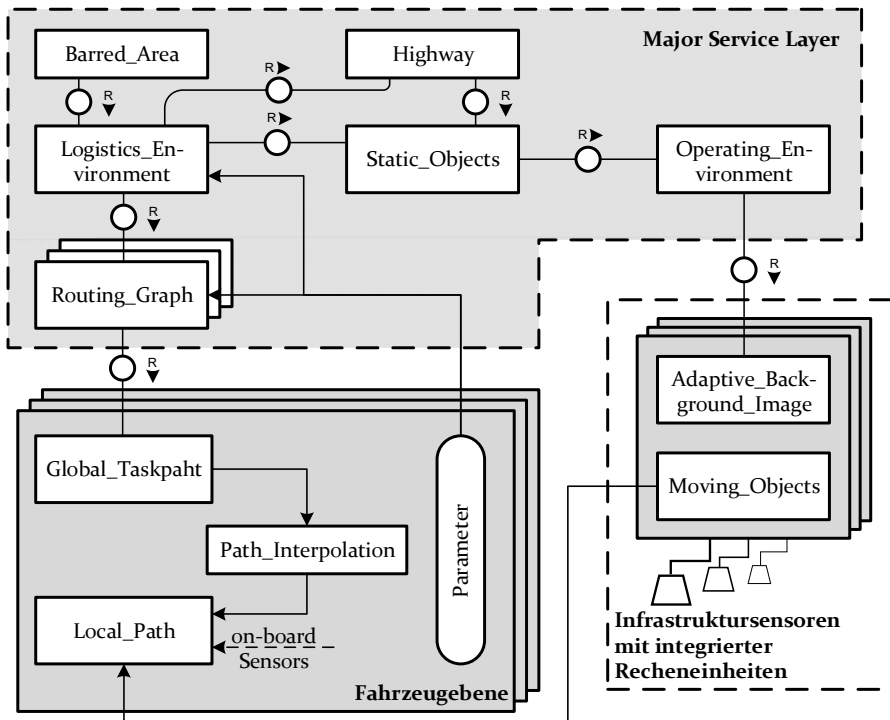


Bild 21: Systemarchitektur zur Erfassung des Arbeitsraums, der Routen- und Pfadplanung

Abhängig von der Fläche, der vorhandenen Deckenhöhe sowie der notwendigen Auflösung, wird der Arbeitsraum durch eine definierte Anzahl von Deckenkameras erfasst. Eine sensornahe Auswerteeinheit separiert im spezifischen Bereich jeder einzelnen Kamera die dynamischen Objekte vom Hintergrund. Diese werden mit einer umschließenden geometrischen Fläche beschrieben, über mehrere Bilder getrackt und die Trajektorie ermittelt. Das Ergebnis dieser Berechnung stellt der Dienst *Moving_Objects* zur

Verfügung. In diesem Verfahren wird zudem ein adaptives Hintergrundbild generiert, welches der Service *Adaptive_Background_Image* bereithält. Da jede Deckenkamera mit einer eigenen Auswerteeinheit bestückt ist, ist der Systemansatz hinsichtlich der Fläche des Arbeitsraums skalierbar.

Zentraler Bestandteil der Servicearchitektur ist die *Major_Service_Layer* (MSL), die die fahrzeugübergreifenden Dienste beinhaltet. Der Service *Operating_Environment* fügt die adaptiven Hintergrundbilder zu einem Gesamtbild des Arbeitsraums zusammen, welches lediglich die statischen Objekte enthält. Der nachfolgende Algorithmus zur Objekterkennung separiert die Objekte im Arbeitsraum vom Hallenboden und überführt das Gesamtbild in ein Binärbild, in welchem unbefahrbare und befahrbare Bereiche definiert werden. Das Ergebnis stellt der Dienst *Static_Objects* bereit. Zur Umsetzung eines performanten Intralogistiksystems ist es notwendig, das Binärbild des Arbeitsraums mit weiteren Informationen anzureichern. Hierzu werden in der MSL die Dienste *Barred_Area* und *Highway* integriert. Ersterer dient dazu, manuell Sperrflächen beschreiben zu können, die von Transportrobotern nicht befahren werden sollen. Der zweite Dienst dient zur Definition von Hauptverkehrswegen für die intralogistischen Transportentitäten. Hintergrund ist, Bereiche im Arbeitsraum beschreiben zu können, in denen die Routenführung einem standardisierten Verteilungsmuster folgt, um Kreuzungen und Gegenverkehr zu vermeiden. Ziel ist es, in diesen Bereichen eine höhere mittlere Geschwindigkeit der Systeme zu gewährleisten. Prädestiniert sind hierfür vorhandene Hauptgänge im Werkslayout und Verbindungskorridore zwischen einzelnen Fertigungsbereichen. Die Informationen aus den Diensten *Static_Objects*, *Barred_Areas* und *Highways* werden schließlich zu einem digitalen Gesamtbild des Arbeitsraums, dem *Logistics_Environment*, aggregiert. Die Aufteilung der Dienste der MSL auf unterschiedliche Recheneinheiten und ihre redundante Auslegung ist bezogen auf die notwendige Rechenleistung und die gewünschte Sicherheit im spezifischen Applikationsumfeld zu definieren. Aufgrund des standardisierten Datenaustauschs zwischen den Diensten der MSL ist die physische Auslegung der Plattform für die Funktion der einzelnen Dienste unerheblich.

Wie bereits im Abschnitt 4.1 erläutert, verfügt die Software der Transportentitäten über einen Parameterserver, in dem die applikationsspezifischen Eigenschaften und Fähigkeiten des Fahrzeugs beschrieben sind. Bei der Anmeldung des Fahrzeugs beim *Routing_Graph*-Dienst übermittelt es die notwendigen Informationen zur Generierung eines typspezifischen Routenplans. Ab diesem Zeitpunkt erstellt der *Routing_Graph* aus den Informationen des *Logistics_Environment* einen Routenplan in Form einer

Adjazenzmatrix für diese Fahrzeugklasse und stellt diesen zur Verfügung. Er beinhaltet folglich einen fahrzeugtypspezifischen kantengewichteten Graphen, der sämtliche Informationen über statische Hindernisse, Sperrflächen, Hauptverkehrswege und infrastrukturelle Zusatzinformationen berücksichtigt. Dieser Graph wird auf der Transportentität im *Global_Taskpath* mit dem auftragsbezogenen Ziel aus dem iLES (vgl. Abschnitt 4.3) ergänzt, der kürzeste globale Pfad berechnet und durch *Path_Interpolation* lokale Stützpunkte zwischen den Knoten eingefügt. Der lokale Pfadplaner vervollständigt das Weltmodell mit den Informationen des *Moving_Objects*-Dienst der Kameras und der optionalen On-board-Sensordaten, errechnet den auszuführenden lokalen Pfad des Fahrzeugs und stellt diesen im Dienst *Local_Path* bereit. Anhand der Daten des lokalen Pfadplaners erfolgt schließlich die Ansteuerung der Antriebseinheiten des Fahrzeugs. Da der beschriebene Ansatz der Systemarchitektur vorsieht, dass die Dienste *Global_Taskpath*, *Task_Interpolation* und *Local_Path* auf den Transportentitäten selbst implementiert sind, sind die Forderungen nach einer Skalierbarkeit des Systems im Hinblick auf die Größe der Fahrzeugflotten, der Vermeidung einer zentralen Routeninstanz und die Autonomie der einzelnen Fahrzeuge realisiert.

Die Vorteile des zentralen Systemansatzes, statische und dynamische Objekte in einer Auswerteeinheit direkt am Infrastruktursensor zu separieren und unterschiedlich in die systemübergreifende Routenplanung und die auftragsbezogene Pfadplanung zu integrieren, werden in den folgenden Abschnitten im Detail erörtert.

5.1.2 Beschreibung der systemübergreifenden Routenplanung

Ausschlaggebend für die sensornahe Verarbeitung der Kameradaten und der direkten Separation in dynamische und statische Objekte ist die notwendige Echtzeitfähigkeit des Gesamtsystems bei der Reaktion auf Hindernisse. Die Echtzeitfähigkeit definiert, dass die Verarbeitungsergebnisse der Dienste innerhalb einer vorgegebenen Zeitspanne verfügbar sein müssen, um auf Änderungen des Umfelds reagieren zu können [116]. Es ist offensichtlich, dass die Informationen über ein dynamisches Hindernis im Gegensatz zu statischen Objekten kurzzyklischer aktualisiert und mit einer geringeren Latenzzeit bereitgestellt werden müssen.

Vor allem im industriellen Umfeld sind statische Objekte wie Maschinen, Anlagen und Einrichtungen über längere Zeitspannen ortsfest. Eine kurzzyklische Aktualisierung und Verarbeitung der Objektdaten würde daher

keinen Mehrwert liefern und unnötig Rechenressourcen binden. Aus diesem Grund arbeitet die beschriebene verteilte Servicearchitektur nach dem in Bild 22 dargestellten Funktionsprinzip zur Erstellung des fahrzeugtypspezifischen Routenplans.

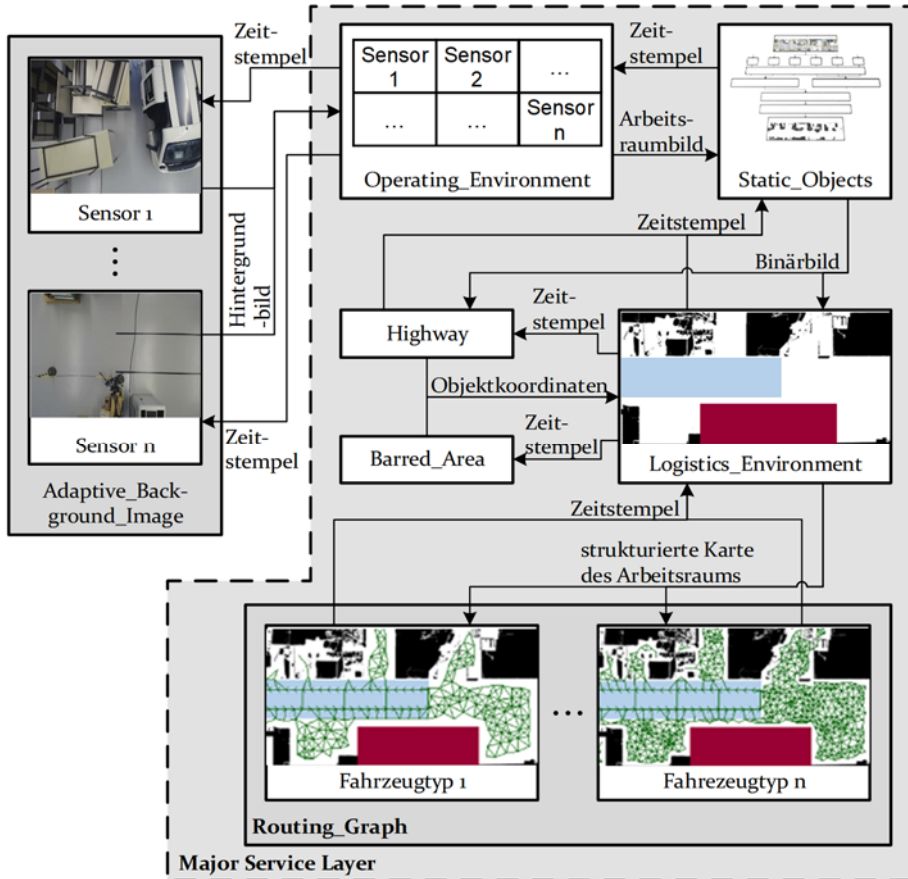


Bild 22: Funktionsprinzip der fahrzeugtypübergreifenden Routenplanung unter Berücksichtigung der statischen Hindernisse

Der Dienst *Operating_Environment* bezieht die adaptiven Hintergrundbilder der Infrastruktursensoren, fügt diese zu einem Gesamtbild des Arbeitsraums zusammen und stellt dieses mit einem eindeutigen Zeitstempel innerhalb der MSL zur Verfügung. Der Algorithmus des *Static_Objects*-Dienstes fordert zunächst diesen Zeitstempel an und vergleicht die übermittelten Daten mit dem Zeitstempel des zuletzt bearbeiteten Gesamtbildes. Sind die Zeitstempel identisch, terminiert der Algorithmus und beginnt nach Ablauf einer definierten Wartezeit erneut. Unterscheiden sich die Zeitstempel, erfolgt die Anforderung des Gesamtbilds, die Berechnung

des daraus resultierenden Binärbilds und dessen Bereitstellung über den Dienst *Static_Objects*. Das Vorgehen des Algorithmus des *Logistics_Environment* ist analog, jedoch aufgrund der beteiligten Dienste komplexer. Hier werden bei der ersten Anfrage die Aktualität des Binärbildes, der definierten Hauptverkehrsstrecken und Sperrflächen mit den vorhandenen Daten anhand von Zeitstempeln verglichen und bei Bedarf angefordert. Wiederrum erfolgt bei unveränderter Datenlage eine Beendigung des Algorithmus respektive bei Abweichungen die Berechnung des digitalen Abbilds des intralogistischen Gesamtarbeitsraums. Final erzeugt *Routing_Graph* auf Basis dieses digitalen Abbilds und den im Einsatz befindlichen Transportentitäten fahrzeugtypspezifische Routenpläne. Auch dieser Algorithmus arbeitet nach dem beschriebenen Zeitstempelprinzip.

Es zeigt sich, dass die Gesamtarchitektur der MSL den Grundsatz einer bedarfsgeregelten Verarbeitung der Infrastruktursensordaten in Bezug auf statische Objekte verfolgt. Dadurch wird die Rechenlast des Gesamtsystems auf ein möglichst niedriges Maß reduziert und zugleich die Echtzeitanforderung gewährleistet.

5.1.3 Ableitung eines auftragsbezogenen globalen Pfads aus der fahrzeugtypspezifischen Adjazenzmatrix

Ausgangspunkt der auftragsbezogenen Pfadplanung ist der fahrzeugtypspezifische Routenplan des *Routing_Graph*-Dienstes, der eine Adjazenzmatrix darstellt. Das Transportfahrzeug erhält aus dem iLES einen konkreten Transportauftrag, indem eine neue Zielpose und eine Aktion am Zielpunkt übergeben wird. Die Methode und Struktur dieser Auftragseinlastung vom Fahrzeugagenten in den Zustandsautomaten zur Ausführung der Transportaufgabe ist ausführlich im Abschnitt 4.3 erläutert. Vor der Berechnung des Pfads prüft der *Global_Taskpath* die Aktualität der lokal gespeicherten Adjazenzmatrix mit der bereitgestellten fahrzeugtypspezifischen Adjazenzmatrix des *Routing_Graph*-Dienstes anhand des eindeutigen Zeitstempels und ersetzt diese gegebenenfalls (vgl. Bild 23). Zusätzlich werden der Startpunkt, der der momentanen Ist-Position entspricht, und der Zielpunkt des Auftrags in der Matrix ergänzt.

Wie bereits erläutert, beschreibt die Adjazenzmatrix welche Knoten des Graphen durch eine Kante verbunden sind. Der verwendete Graph ist dabei aufgrund der Integration der Hauptverkehrswege gerichtet und gewichtet. Zur Berechnung einer optimalen Lösung zwischen Start- und Zielpunkt im gegebenen diskreten Zustandsraum werden in der Literatur unter anderem

der A*- und Dijkstra-Algorithmus, sowie deren Spezialisierungen und Erweiterungen diskutiert [117]. Übergreifend werden bei allen Verfahren die Knoten der Matrix in folgende drei Kategorien [68] unterschieden:

- unbesucht: Knoten die noch nicht besucht wurden
- lebend: Besuchte Knoten, bei denen noch nicht alle Nachfolger besucht wurden
- tot: Besuchte Knoten, von denen alle Nachfolger besucht wurden

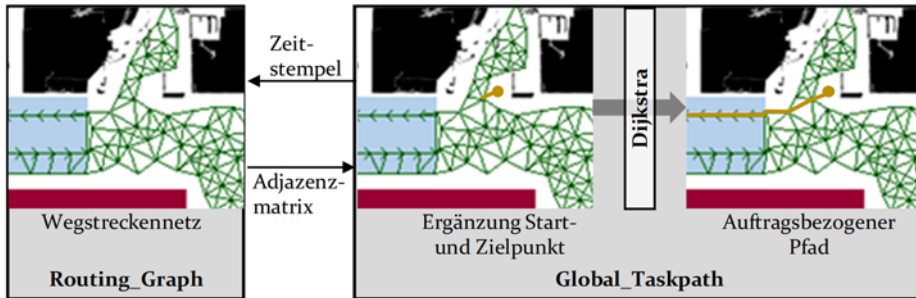


Bild 23: Funktionsprinzip der Erstellung eines auftragsbezogenen globalen Pfads

Der im *Global_Taskpath* implementierte Dijkstra-Algorithmus speichert alle vorhandenen lebenden Knoten in der Prioritätsliste Q mit den zugehörigen Kosten $C_e(q)$, die zum Erreichen dieses Knotens veranschlagt werden. Anhand dieser Kostenwerte wird die Prioritätsliste aufsteigend sortiert, wobei anfänglich lediglich der Startknoten q_S mit $C_e(q_S) = 0$ enthalten ist. Die relativen Kantengewichte der Adjazenzmatrix beschreiben dabei die Kosten $l(e)$ zwischen den Knoten. Ausgehend vom ersten Knoten q in der Prioritätsliste Q werden alle angrenzenden Knoten q' nach folgender Fallunterscheidung untersucht:

- Ist der betrachtete Knoten q' noch unbesucht, erfolgt die Aufnahme in Q . Der Knoten wird somit als lebend markiert und mit $C_e(q') = C_e(q) + l(e_{q'-q})$ bewertet
- Ist der betrachtete Knoten q' bereits in Q enthalten, prüft der Algorithmus, ob $C_e(q')$ kleiner als der aktuell veranschlagte Kostenwert ist. Ist dies der Fall wird der aktuelle Wert ersetzt.

Dieser Vorgang wiederholt sich, bis alle Kanten zum Zielpunkt q_G untersucht wurden, womit der Algorithmus immer den kostengünstigsten Weg durch den Graphen findet.

In Bild 24 wird dieses Vorgehen anhand eines vereinfachten Beispielgraphen verdeutlicht. Das Beispiel zeigt ausgehend vom Startknoten q_S die

Kosten, um die Knoten q_1 bis q_4 sowie den Zielknoten q_G zu erreichen. Zudem werden die Kanten, die höhere Kosten zum Erreichen des Knoten verursachen, gestrichen und die günstigsten Verbindungen im Lösungsbaum visualisiert. Ausgehend vom Zielknoten q_G ergibt sich durch dieses Vorgehen ein eindeutiger Pfad zum Startknoten. Aus diesem Grund wird bei der Implementierung die Zuordnung der realen Start- und Zielpunkte mit denen im Algorithmus vertauscht, damit die Ausführung des Pfades auf einem eindeutigen Ergebnis aufbaut.

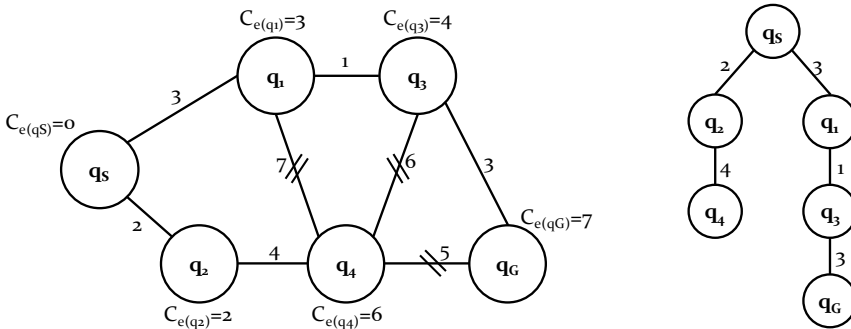


Bild 24: Vereinfachter Lösungsgraph des Dijkstra-Algorithmus (links) mit entsprechendem Lösungsbaum (rechts)

Der A*-Algorithmus stellt eine Erweiterung des Vorgehens nach Dijkstra dar. Ziel der Optimierung ist es, die Gesamtzahl an untersuchten Knoten zu reduzieren, indem eine heuristische Schätzfunktion $G(q)$ die Kosten vom zurzeit betrachteten Knoten bis zum Zielknoten mit einbezieht. Diese Kosten entsprechen der euklidischen Distanz bis zum Zielknoten. Die Sortierung der Knoten in der Prioritätsliste findet nun nach der Summe der optimalen Kosten vom Startpunkt bis zum gegenwärtigen Knoten $C_e(q')$ und $G(q')$ statt. Der A*-Algorithmus ist vollständig und findet, wie auch der Dijkstra-Algorithmus, immer die optimale Lösung im Konfigurationsraums, falls eine solche existiert. Das größte Potenzial entfaltet der Algorithmus jedoch bei Graphen, deren Kanten eine ähnliche Gewichtung aufzeigen. Durch die Implementierung der Hauptverkehrsstrecken mit ihren signifikant niedrigeren Kantengewichten sollen die Transportentitäten dazu bewegt werden, diese den übrigen Wegen vorzuziehen, auch wenn dadurch längere Strecken in Kauf genommen werden müssen. Durch die Schätzfunktion $G(q')$ des A*-Algorithmus wird diese Vorgabe jedoch nicht zwingend erfüllt. Auf Basis der Validierungsergebnisse in Abschnitt 7.3.4, die dieses Verhalten zeigen, ist daher die Verwendung des Dijkstra-Algorithmus vorzuziehen. [118]

5.1.4 Beschreibung der auftragsbezogenen Pfadplanung

Wie in Abschnitt 5.1.1 erwähnt, ist der vom Dienst *Routing_Graph* erzeugte fahrzeugtypspezifische Graph die Grundlage der globalen Pfadplanung. Aufgrund der beschriebenen Verarbeitung der Infrastruktursensordaten erfolgt sowohl die Grapherstellung als auch die globale Pfadplanung von der aktuellen Pose zur Zielpose ohne Berücksichtigung von dynamischen Hindernissen. Der fahrzeugtypspezifische Graph stellt in der Analogie zur klassischen Softwarearchitektur von FTS mit einer Leitsteuerung somit einen Fahrwegeplan dar. Trotz der Anforderung, dass sich der Graph an sich verändernde Rahmenbedingungen selbstständig anpasst, ist eine Verarbeitung der dynamischen Hindernisse in der Phase der Routenplanung nicht sinnvoll, da diese temporäre Hindernisse auf der Fahrstrecke der Transportentitäten darstellen. Die Routenplanung beinhaltet daher lediglich Objekte, die ein dauerhaftes Umfahren der Fahrzeuge bedingen. Auch bei der Planung des globalen Pfads erweist sich eine Berücksichtigung von dynamischen Hindernissen als nicht zielführend, da die Güte der Bewegungsvorhersage mit zunehmendem Abstand zwischen der aktuellen Position des Fahrzeugs und dem Hindernis abnimmt. Zudem können für die Anpassung der Pfadplanung auf statische und dynamische Hindernisse im Arbeitsraum unterschiedliche Strategien angewendet werden. Während statische Objekte stets umfahren werden müssen, genügt es bei einem Kollisionskurs mit einem dynamischen Hindernis die Geschwindigkeit des Transportroboters zu verändern, um den Kreuzungspunkt der Pfade zeitlich versetzt zu befahren. Diese Überlegungen aufgreifend arbeiten die auf der Transportentität integrierten Dienste der SOA nach dem in Bild 25 dargestellten Funktionsprinzip, um eine auftragsbezogene Pfadplanung durchzuführen.

Durch diese direkte Integration der Infrastruktursensordaten im lokalen Pfadplaner ist eine kurzzyklische Abtastrate in Bezug auf dynamische Hindernisse des roboternahen Umfelds gewährleistet. Weiterhin bezieht die Transportentität lediglich die Daten von den Auswerteeinheiten, die zur Erstellung der lokalen Umgebungskarte notwendig sind. Dadurch wird die Datentransferrate des Gesamtsystems auf das geringstmögliche Maß beschränkt.

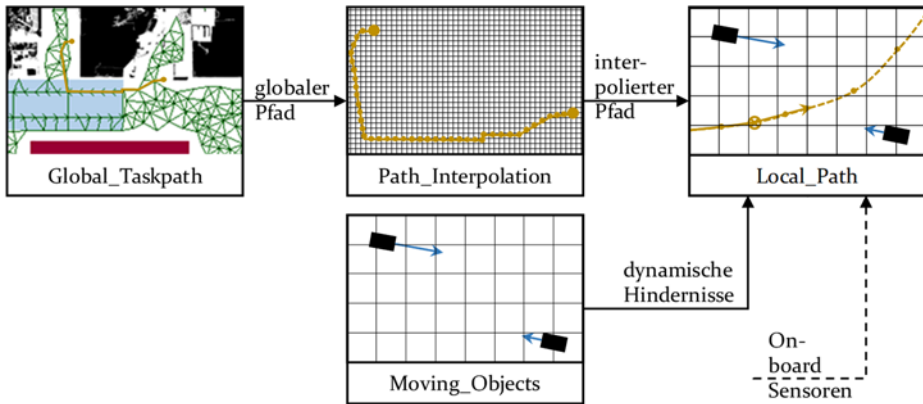


Bild 25: Funktionsprinzip der lokalen Pfadplanung unter Berücksichtigung dynamischer lokaler Hindernisse

5.2 Erfassung des Arbeitsraums durch dezentrale Infrastruktursensoren

Wie in der Beschreibung der verteilten Systemarchitektur erörtert, stellt die dezentrale Erfassung des Arbeitsraums durch die Deckenkameras und die Detektion der dynamischen und statischen Objekte einen zentralen Bestandteil des Gesamtsystems dar. Lütke beschreibt, dass vor der Inbetriebnahme der statischen Hintergrundsubtraktion eines VTS sämtliche Hindernisse aus der erfassten Szenerie entfernt werden müssen, um ein Referenzbild des befahrbaren Hallenbodens zu erstellen [14]. Dieses Vorgehen ist im industriellen Umfeld nicht praktikabel. Eine optimierte Herangehensweise ist die erforschte sensor-nahe Separation der dynamischen Hindernisse anhand eines adaptiven Hintergrundbilds. [119]

5.2.1 Methode zur Separation von dynamischen und statischen Hindernissen

Ziel der ersten Operation ist, aus dem Livebild der Deckenkameras ein adaptives Hintergrundbild und ein binäres Bild der dynamischen Objekte zu selektieren. Hierbei wird ausgenutzt, dass sich die Position der Sensoren und somit die Pixelwerte von statischen Objekten (*Background* - BG), im Gegensatz zu dynamischen Objekten (*Foreground* - FG), über die Einzelbilder hinweg nicht verändern. Nach [58] beschreibt $p(\vec{x}^{(t)}|BG)$ ein Modell des Hintergrunds bei dem jeder Wert eines Pixel $\vec{x}^{(t)}$ zum Zeitpunkt t definiert wird. Dieses Hintergrundmodell wird anhand eines Trainingssets χ

geschätzt und durch $\hat{p}(\vec{x}^{(t)}|\chi, BG)$ repräsentiert. Im konkreten Anwendungsfall kann sich der Hintergrund z. B. durch veränderte Lichtverhältnisse über die Zeit verändern, wodurch eine zeitliche Adaption der Trainingsdaten notwendig ist. Hierzu wird über eine repräsentative Zeitperiode T der Trainingssatz zum Zeitpunkt t wie folgt definiert:

$$\chi_T = \{\vec{x}^{(t)}, \dots, \vec{x}^{(t-T)}\} \quad (11)$$

In der erforschten Architektur wird für jedes Einzelbild der Trainingssatz χ_T aktualisiert und auf dessen Basis das Hintergrundmodell neu geschätzt. Jedoch können einige Pixel des Models Informationen zu Vordergrundobjekten enthalten, was nach [57] mit einem *Gaussian Mixture Model* (GMM) mit M Komponenten berücksichtigt wird:

$$\hat{p}(\vec{x}^{(t)}|\chi_T, BG + FG) = \sum_{m=1}^M \hat{\pi}_m N(\vec{x}; \hat{\mu}_m, \hat{\sigma}_m^2 I) \quad (12)$$

mit

$$\sum_{m=1}^M \hat{\pi}_m = 1$$

$$0 \leq \hat{\pi}_m \leq 1$$

Die Komponenten $\hat{\mu}_1, \dots, \hat{\mu}_M$ beschreiben dabei die Abschätzung der arithmetischen Mittelwerte, $\hat{\sigma}_1, \dots, \hat{\sigma}_M$ die Abschätzung der Varianzen des GMM. Zur detaillierten mathematischen Herleitung sei an dieser Stelle auf die Literatur verwiesen.

Mit Hilfe des Schwellwerts c_f kann anhand der abnehmenden Zugehörigkeitswahrscheinlichkeiten $\hat{\pi}_m$ der Einzelpixel zum Hintergrund über

$$B = \arg \min_b \left(\sum_{m=1}^b \hat{\pi}_m > (1 - c_f) \right) \quad (13)$$

die maximale Anzahl der Vordergrundinformationen bestimmt werden, ohne dass eine Änderung des Hintergrundmodells erfolgt. Das bedeutet, dass sich die Zugehörigkeitswahrscheinlichkeit der Pixel eines Objekts, welches sich für längere Zeit an einem Ort befindet, kontinuierlich erhöht. Bei Erreichen des Schwellwerts c_f werden die Pixel des Objekts schließlich

als Hintergrund interpretiert. Der Schwellwert legt somit die Sensitivität des Algorithmus fest, wobei ein Optimum zwischen langzyklischen Veränderungen der Hintergrundszenerie und sich langsam bewegenden Vordergrundobjekten gefunden werden muss (vgl. Bild 26). Durch dieses Vorgehen wird der von Lütteke beschriebene Umstand umgangen, dass bei einer Änderung der Hintergrundszenerie, z. B. durch ändernde Lichtverhältnisse, zwingend ein neues Hintergrundbild erzeugt werden muss.

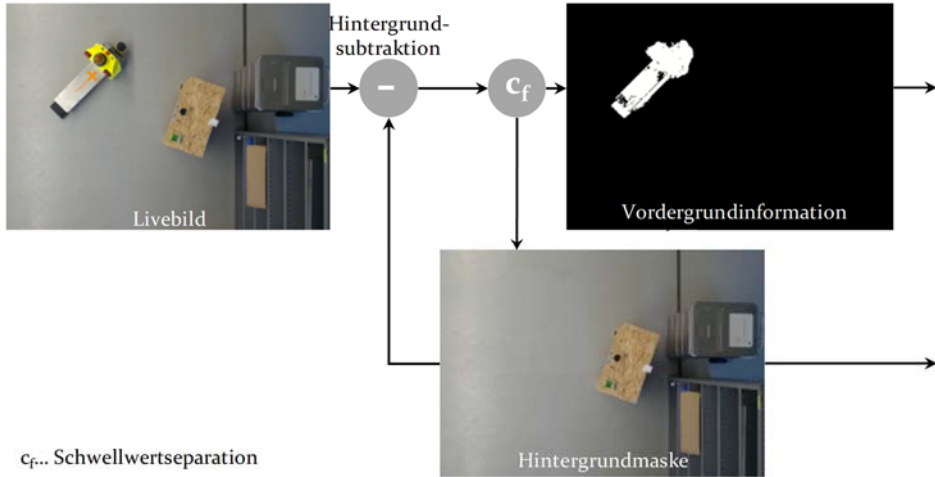


Bild 26: Separation des Deckenkamerabilds in Vorder- und Hintergrundinformationen

5.2.2 Methode zur Verfolgung dynamischer Hindernisse

Die im binären Bild detektierten Vordergrundobjekte stellen sämtliche dynamische Objekte innerhalb der digitalisierten Szenerie dar und können durch den Zustandsvektor

$$\vec{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (14)$$

mit

x, y : Positionskoordinaten

\dot{x}, \dot{y} : Geschwindigkeitskomponenten

repräsentiert werden. Wie im Abschnitt 5.1.4 beschrieben, stellen diese Informationen und die Abmessung der dynamischen Hindernisobjekte die

Basis für das Verhalten des lokalen Pfadplaners dar. Während anhand eines einzelnen Bildes die Position und Abmessung eines spezifischen Objekts ermittelt werden kann, ermöglicht das *Multi-Object-Tracking*-Verfahren durch die Sequenz der aufeinanderfolgenden binären Vordergrundbilder die Bestimmung der Geschwindigkeitskomponenten. Hierbei muss neben der Detektion der Objekte eine korrekte Zuweisung ihrer detektierten Position zu einer Trajektorie erfolgen. Im vorliegenden Fall bietet sich die Verwendung einer Markov-Kette erster Ordnung zum Lösen des Tracking-Problems an. Bei diesem rekursiven Ansatz erfolgt die Schätzung des folgenden Punkts der Trajektorie ausschließlich auf dem aktuellen Schätzwert des Objektzustands, z. B. Pose und Geschwindigkeit. Weiter in der Vergangenheit liegende Trajektorienpunkte werden zur Errechnung des folgenden Trajektorienpunkts nicht einbezogen. Die Zuordnung der Messungen zu den prädizierten Trajektorienpunkten erfolgt im Sinne kleinster Distanzen zwischen Messung und Prädiktion. Die a priori Zustandsschätzung der dynamischen Hindernisobjekte erfolgt dabei mithilfe eines Kalmanfilters. Zur Zuweisung der Positionsmessungen zu den Objektpositionsprädiktionen wird die Ungarische Methode verwendet. [120, 121]

Kalmanfilter

Zur mathematischen Herleitung des Kalmanfilters sei an dieser Stelle auf die Literatur [122] verwiesen. Im konkreten Anwendungsfall wird die zur a priori Zustandsschätzung verwendete Dynamikmatrix A der Vordergrundobjekte durch

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

beschrieben, wobei dt der Zykluszeit zur Erzeugung eines binären Vordergrundbilds entspricht. Die Rauschsignale des Systems werden als weiß, mittelwertfrei und unkorreliert vorausgesetzt, wodurch die Prozessrauschkovarianzmatrix Q zeitlich unveränderlich ist. Die Eingangsmatrix B und das Eingangssignal u für Hindernisobjekte betragen 0. Da von den Deckenkameras lediglich die Position der Objekte erfasst wird, lautet die Messmatrix

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (16)$$

und die Messrauschkovarianzmatrix R reduziert sich zudem zu einer 2x2-Matrix.

Betrifft ein Hindernisobjekt erstmalig die von der Deckenkamera erfasste Szenerie, wird dessen Zustandsvektor mit der Messung der Position und einer Geschwindigkeit initialisiert und ihm eine eindeutige Identifizierung zugewiesen. Zudem wird die Kovarianzmatrix P entsprechend der Messunsicherheit im Kamerarandbereich gesetzt. Für die folgende a priori Schätzung des Objekts gilt für die Objektposition

$$x_{k+1} = Ax_k \quad (17)$$

und für die Kovarianz

$$P_{k+1} = AP_kA^T + Q \quad (18)$$

Anhand der Messwerte z_k des Folgebildes wird das Kalman-Gain K_k bestimmt,

$$K_k = P_kH^T(HP_kH^T + R)^{-1} \quad (19)$$

der Systemzustand x_k korrigiert

$$x_k = x_k + K_k(z_k - Hx_k) \quad (20)$$

und mithilfe der Einheitsmatrix I die Kovarianzmatrix P_k für die anschließende a priori Schätzung angepasst

$$P_k = (I - K_kH)P_k \quad (21)$$

Durch den Algorithmus des Kalman-Filters wird somit eine stetige Verbesserung der Positions- und Geschwindigkeitsschätzung der Hindernisobjekte im Bereich eines Deckensensors gewährleistet.

Ungarische Methode

Da innerhalb eines Sensorbereichs mehrere dynamische Hindernisobjekte erfasst werden, müssen die Zustandsprädiktionen den -messungen zugeordnet werden. Ziel ist es, dass jedem unabhängigen Element des Messvorgangs m_j ein Element der Prädiktion e_i zugeordnet wird. Für jedes Paar x_{ij} folgen zudem die Zuordnungskosten c_{ij} , die den euklidischen Abstand beider Positionen darstellen. Die Ungarische Methode liefert für die Zuordnungsmatrix x eine optimale Lösung mit exakt einer 1 in jeder Zeile und Spalte, indem der Ausdruck

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij} \quad (22)$$

minimiert wird, für den gilt:

$$x_{ij} = \begin{cases} 1 & \text{wenn } e_i \text{ } m_j \text{ zugewiesen wird} \\ 0 & \text{sonst} \end{cases}$$

Für eine ausführliche mathematische Herleitung der Ungarischen Methode sei auf die Literatur verwiesen [123].

Das Ergebnis aus der Fusion von Kalman-Filter und Ungarischer Methode erlaubt die detektierten Vordergrundobjekte zu tracken und die Information zur Trajektorie jedes bewegten Hindernisobjekts innerhalb des Sichtbereichs des Deckensensors zu erlangen.

5.2.3 Übermittlung der dynamischen Objektdaten

In der Gesamtarchitektur werden die Informationen der dynamischen Hindernisse von den lokalen Pfadplanern der Fahrzeuge verarbeitet, um bei einer drohenden Kollision in der Lage zu sein, entsprechende Gegenmaßnahmen einzuleiten. Daher ist eine performante Verarbeitung des Livebilds sowie eine möglichst verzögerungsfreie Übermittlung der dynamischen Objektdaten an die Fahrzeuge entscheidend. Die Verarbeitung des Kamerabilds sowie das Tracken der dynamischen Hindernisse und der Abschätzung ihrer Trajektorie durch die sensornahe Auswerteeinheit ist in Bild 27 dargestellt.

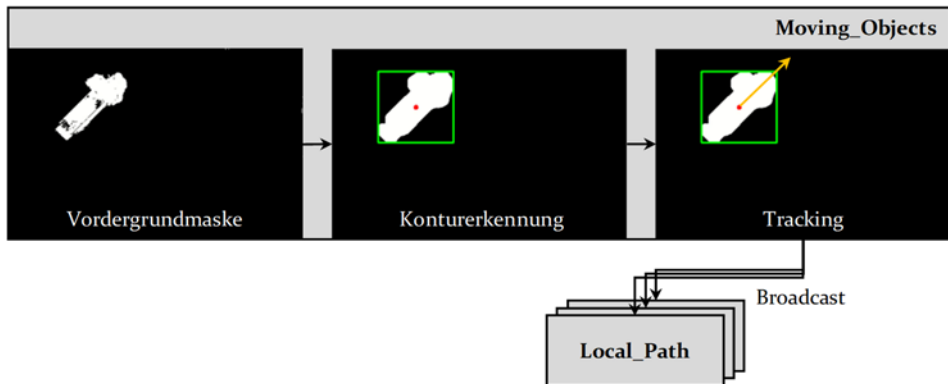


Bild 27: Funktionsprinzip der Übermittlung der dynamischen Objektdaten vom Infrastruktursensor an die Transportentitäten per Broadcast

Es wird ersichtlich, dass die Informationen der dynamischen Hindernisobjekte über einen Broadcast publiziert werden und somit nicht der üblichen Umsetzung des Informationsaustauschs der MSL entsprechen. Hintergrund ist, dass diese Information ohne eine zusätzliche Latenzzeit an die

Fahrzeuge übermittelt werden sollen. Die Sensoren übermitteln daher proaktiv die Position, die Ausmaße anhand der umschließenden Kontur sowie die geschätzte Trajektorie jedes bewegten Hindernisses innerhalb ihres Bereiches. Die Fahrzeuge, die diese Informationen zur aktuellen lokalen Pfadplanung benötigen, können diese verwerten, ohne zuvor eine Anfrage stellen zu müssen.

Der Grundsatz der Gesamtarchitektur, eine geringstmögliche Datenmenge zu versenden, wird durch *Moving_Objects* erfüllt. Die Informationsbereitstellung durch Dienste wird an dieser Stelle jedoch zu Gunsten der Reaktionsfähigkeit der Fahrzeuge durch eine kontinuierliche Datenübermittlung ersetzt.

5.2.4 Fusionierung der adaptiven Hintergrundbilder zu einem statischen Arbeitsraum

Das adaptive Hintergrundbild, welches bei der Verarbeitung der Szene entsteht, wird über den Dienst *Adaptive_Background_Image* bereitgestellt und ist die Grundlage für die Generierung des statischen Weltmodells. Die Fusion der Einzelbilder der Infrastruktursensoren erfolgt durch *Operating_Environment* (vgl. Bild 28). Hierfür haben sich Stitching-Algorithmen etabliert, die Einzelbilder anhand von Merkmalen zu einem Gesamtbild zusammenfügen. Dabei ist es notwendig, in beiden Bildern eine *Feature*-Suche durchzuführen, um robuste Übereinstimmungen zu detektieren und die notwendige Transformation der beiden Bildmatrizen zueinander abzuschätzen. Als etablierte Algorithmen gelten die skaleninvariante Merkmalstransformation (*Scale-Invariant Feature Transform* - SIFT) [124], und SURF (*Speeded Up Robust Features*) [125]. Bei beiden Verfahren der Bildfusion ist die Detektion der Merkmale rechenzeitbestimmend, womit die Erstellung des Gesamtbildes mit der Anzahl der Einzelbilder stark zunimmt. Durch die feste Installation der Infrastruktursensoren im Produktionsumfeld ist jedoch eine Varianz der Transformationsfunktion über die Zeit nicht zu erwarten. Der Algorithmus des *Operating_Environment* verfolgt daher den Ansatz, die Transformation der Bilder zueinander lediglich bei der Installation und Inbetriebnahme, bzw. bei einer manuell angestoßenen Kalibrierung des Systems durchzuführen und die relative Position im Raum jeder einzelnen Kamera zu speichern. Dadurch reduziert sich im laufenden Betrieb die Erstellung des Gesamtbildes auf eine Verknüpfung der Einzelbildmatrizen zur Gesamtmatrix.

5 Verteilte Servicearchitektur zur Digitalisierung des Arbeitsumfelds, der Routen- und Pfadplanung

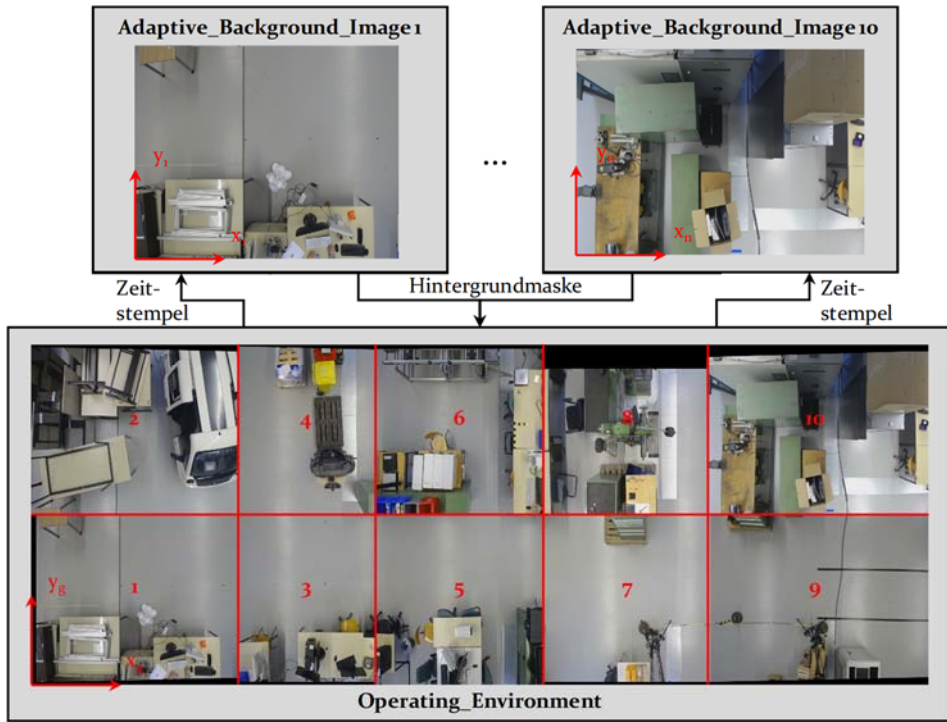


Bild 28: Fusion der adaptiven Hintergrundbilder zu einem Abbild des gesamten Arbeitsraums

Mathematisch werden zuerst die einzelnen Pixel der Einzelbilder zeilenweise verknüpft und die Zeilen anschließend zum Gesamtbild fusioniert. Der Algorithmus des *Operating_Environment* fusioniert die Einzelbilder im RGB-Farbraum, wodurch mathematisch eine dreidimensionale Matrix entsteht, die die Bildinformationen des gesamten Arbeitsraums enthält.

5.3 Bereitstellung einer fahrzeugtypspezifischen Routenplanung

Wie in Abschnitt 5.2 beschrieben, setzt sich das vom *Operating_Environment* bereitgestellte Gesamtbild des Arbeitsraums lediglich aus den einzelnen adaptiven Hintergrundbildern der Infrastruktursensoren zusammen und enthält keine Informationen über Objekte. Zur Generierung von Verkehrsrouten ist es jedoch unumgänglich, den für die Fahrzeuge befahrbaren Bereich vom nicht befahrbaren Bereich unterscheiden zu können.

5.3.1 Komplettierung des Weltmodells

Die Umwandlung des fusionierten adaptiven Hintergrundbildes in ein Binärbild, das die befahrbaren Flächen in Weiß und nicht befahrbare Flächen in Schwarz kennzeichnet, erfolgt in einem mehrstufigen Verfahren (vgl. Bild 29). Dazu wird das Farbbild in verschiedene Farb Räume überführt und in spezifischen Farbkanälen eine Kantendetektion vorgenommen. Abschließend werden die Kanten, mit Hilfe von morphologischen Operatoren zu in sich geschlossenen Körperkanten überführt.

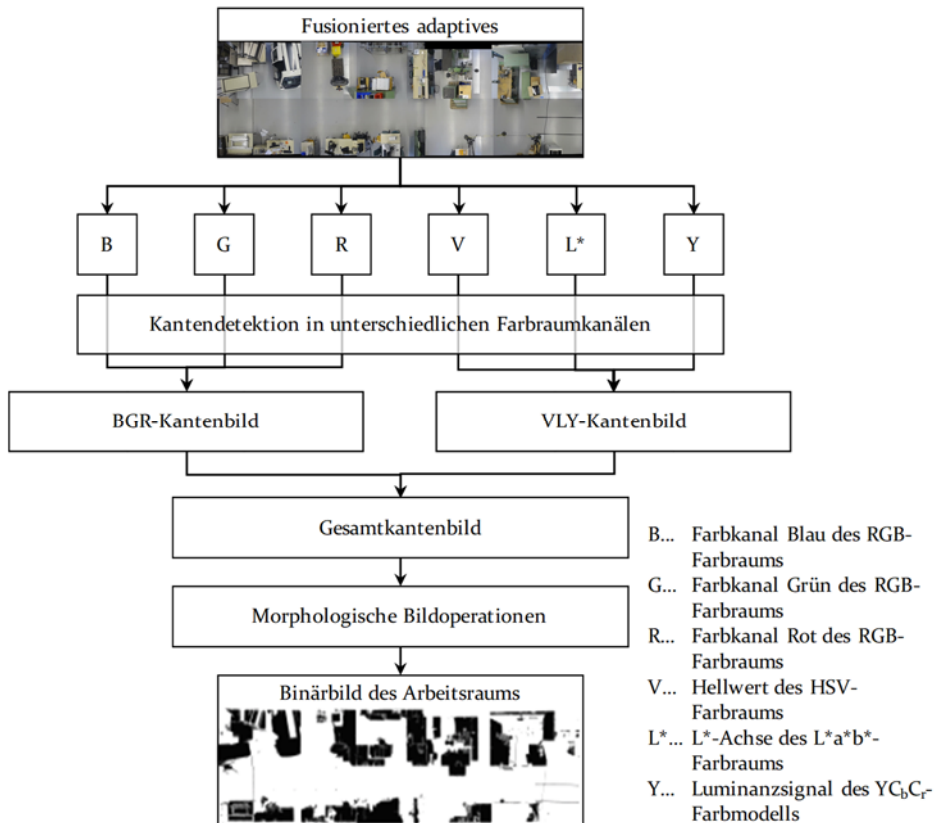


Bild 29: Methode zur Erzeugung eines Binärbilds des Arbeitsraums

Vorverarbeitung des Bildes

Im HSV-Farbraum wird ein Bild mit dem Farbwert (*Hue*), der Farbsättigung (*Saturation*) und dem Hellwert (*Value*) beschrieben. Der Wert des für die Kantendetektion relevanten Hellwertkanals ergibt sich aus dem Maximum der einzelnen Farbkanäle des RGB-Farbraums. [126]

$$V = \max\{R, G, B\} \quad (23)$$

Der $L^*a^*b^*$ Farbraum ist Teil einer Reihe von verschiedenen Farbräumen der *Commision Internationale de l'Éclairage* (CIE). Um den für die Kantendetektion notwendigen Helligkeitswert der L^* -Achse zu ermitteln, muss zunächst eine Konvertierung in den CIE XYZ-Farbraum erfolgen. Hierfür gibt es keine allgemeingültige Umrechnungsformel, da ein Weißpunkt als Bezugspunkt benötigt wird und dieser je nach Ausprägung des RGB Farbsystems unterschiedlich definiert ist. In der erforschten Architektur wird hierzu der RGB-Farbraum Adobe RGB 1998 mit dem Weißpunkt D65 zugrunde gelegt. Y_N entspricht bei der Berechnung dem Y-Wert des Referenzweißpunkts [127].

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0,5767309 & 0,1855540 & 0,1881852 \\ 0,2973769 & 0,6273491 & 0,0752471 \\ 0,0270343 & 0,0706872 & 0,9911085 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (24)$$

$$L^* = 116 \sqrt[3]{\frac{Y}{Y_N}} - 16 \quad (25)$$

Im YC_bC_r -Farbmodell beschreibt Y das Luminanzsignal und die beiden Chrominanzsignale C_b und C_r repräsentieren die Farbkomponenten. Die Berechnung des YC_bC_r Wertetripels erfolgt in Abhängigkeit des anzuwendenden Standards und kann somit nicht verallgemeinert werden. Nachfolgende Gleichung bezieht sich auf die im Algorithmus implementierten Werte zur Konvertierung der Pixelwerte des RGB-Farbraums in das YC_bC_r -Farbmodell zur Berechnung des Y Wertes. [127]

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65,481 & 128,553 & 24,966 \\ -37,797 & -74,203 & 112,0 \\ 112,0 & -93,786 & -18,214 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (26)$$

Kantendetektion

Die Kantendetektion erfolgt auf den sechs ausgewählten Bildkanälen (R, G, B, V, L^*, Y) analog. In diesen Bildern sind Kanten durch große Unterschiede benachbarter Pixelwerte charakterisiert, die jedoch auch durch Bildrauschen verursacht werden können. Daher wird das Ausgangsbild $g(x, y)$ mit einem Gaußfilter geglättet, bevor es mit Hilfe eines Sobeloperators in X- und Y-Richtung gefaltet wird um vertikale bzw. horizontale Kanten zu betonen, wodurch sich die partiellen Ableitungen g_x und g_y ergeben. Diese Bildoperatoren stellen Masken dar, bei denen sich der neue

Wert aus den gewichteten Werten der umgebenden Pixel ergibt und im Algorithmus wie folgt hinterlegt sind: [128, 129]

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \text{ Gaußfilter} \quad (27)$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ Sobeloperator} \\ X \text{ Richtung} \quad (28)$$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \text{ Sobeloperator} \\ Y \text{ Richtung} \quad (29)$$

Die Kantenrichtung θ lässt sich anhand der ermittelten partiellen Ableitungen mittels

$$\theta = \arctan2(g_y, g_x) \quad (30)$$

berechnen, wobei die Richtungen auf 0° , 45° , 90° und 135° gerundet werden. Die absolute Kantenstärke ergibt sich aus dem euklidischen Betrag der beiden partiellen Ableitungen, der hinsichtlich der Effizienzsteigerung des Algorithmus folgendermaßen approximiert wird: [128, 129]

$$G(x, y) = |g_x(x, y)| + |g_y(x, y)| \quad (31)$$

Anschließend wird mit der *Non-Maximum Suppression* (NMS) Methode sichergestellt, dass lediglich die Maxima entlang einer Kante erhalten bleiben. Dazu wird jeder Wert von $G(x, y)$ mit den beiden Pixeln rechts und links von sich verglichen und auf 0 gesetzt, falls einer der Werte größer ist. Abschließend wird bei der Kantendetektion mit dem Hysterese-Verfahren mit zwei Schwellwerten $T_1 < T_2$ festgestellt, ab welcher Kantenstärke ein Pixel zu einer Kante zählt. Im Bild wird dabei ein Pixel gesucht, dessen Wert über T_2 liegt, und folgt anschließend der Kante beidseitig. Alle Pixel entlang der Kante, deren Werte über T_1 liegen, werden dabei als Kantenelement markiert. [128, 129]

Morphologische Operatoren

Durch die Kantendetektion in sechs verschiedenen Farbkanälen und deren Kombination in einem Gesamtbild entsteht ein für intralogistische Szenarien robustes Kantenbild der Arbeitsumgebung. Durch eine Invertierung und Normierung wird das Kantenbild in ein Binärbild aus weißen Flächen und schwarzen Kanten überführt. Der erste morphologische Bildoperator,

das *Closing*, ist notwendig, um offene Kanten zu schließen. Dabei wird das Binärbild mit einem rechteckigen Strukturelement der Kantenlänge 5 Pixel zuerst einer Dilatation und anschließend einer Erosion unterzogen, die wie folgt definiert ist:

$$A \cdot B = (A \oplus B) \ominus B \quad (32)$$

mit

A: Menge der selektierten Pixel des Eingangsbildes $G(x, y)$

B: Menge der selektierten Pixel des Strukturelements

In einer abschließenden Operation erfolgt das Schließen von zusammenhängenden Konturen und das Filtern nach ihrer beschreibenden Fläche anhand eines oberen Schwellwertes. Der Schwellwert ist notwendig, damit der befahrbare Boden nicht als Hindernisfläche interpretiert wird, da dieser mit den umgebenden Hindernissen eine geschlossene, jedoch verhältnismäßig große Kontur darstellt. Das Resultat dieser Operationen ist ein Binärbild, welches den Arbeitsraum in frei befahrbare Bereiche und Bereiche mit Hindernissen unterteilt und über den Dienst *Static_Objects* in der MSL publiziert wird.

5.3.2 Anreicherung des Weltmodells mit Zusatzinformationen

Das Binärbild des *Static_Objects*-Dienstes würde bereits als Datenbasis ausreichen, um einen intralogistischen Routenplan zu erzeugen. Die erforschte MSL bietet jedoch mit *Logistics_Environment* einen Dienst an, der das Weltmodell mit intralogistischen Zusatzinformationen anreichert, um die Performanz der Routen zu erhöhen. Hierzu werden Hauptverkehrswege, Sperrflächen sowie Materialquellen und -senken integriert.

Hauptverkehrswege

Wie im Abschnitt 5.1 erläutert, haben die Hauptverkehrswege das Ziel, die mittlere Geschwindigkeit der Fahrzeuge zu erhöhen, in dem eine Bewegungsvorzugsrichtung vorgeschrieben wird. Ein Hauptverkehrsbereich wird dabei als Viereck beschrieben, dessen umschließende Fläche frei von statischen Hindernisobjekten ist. In einem intralogistischen Szenario zeichnet sich ein Hauptverkehrsweg als gestrecktes Viereck aus, und wird vom Algorithmus des *Highway*-Dienstes als Parallelogramm *ABCD* angenähert. Anhand des größten Sicherheitsabstands der aktuell registrierten

Fahrzeugtypen werden den Außenkanten um $\frac{a_{s,max}}{2}$ verschoben, womit das Parallelogramm $A'B'C'D'$ entsteht. Die Eckpunkte dieses Parallelogramms stellen die Knotenpunkte der Hauptverkehrsstrecke, die Kanten die Wegstrecken dar, wobei die Vorzugsfahrtrichtung gegen den Uhrzeigersinn definiert ist. Um neben den Eckpunkten weitere Auf- und Abfahrtpunkte des Hauptverkehrswegs zu definieren, werden im Folgenden die Kanten im Abstand f mit Zwischenknotenpunkten versehen. Zur Reduzierung der Wege innerhalb der Hauptverkehrsstrecke wird zudem jeder zweite Zwischenknotenpunkt mit dem gegenüberliegenden Knotenpunkt verbunden. Diese Verbindungskanten erhalten jedoch keine Vorzugsrichtung, wodurch sie in beide Richtungen befahrbar sind (vgl. Bild 30). Jeder definierte Bereich einer Hauptverkehrsstrecke erhält zudem eine eindeutige Identifikationsnummer (ID), anhand derer der Hauptverkehrsweg vom Bediener wieder gelöscht werden kann.

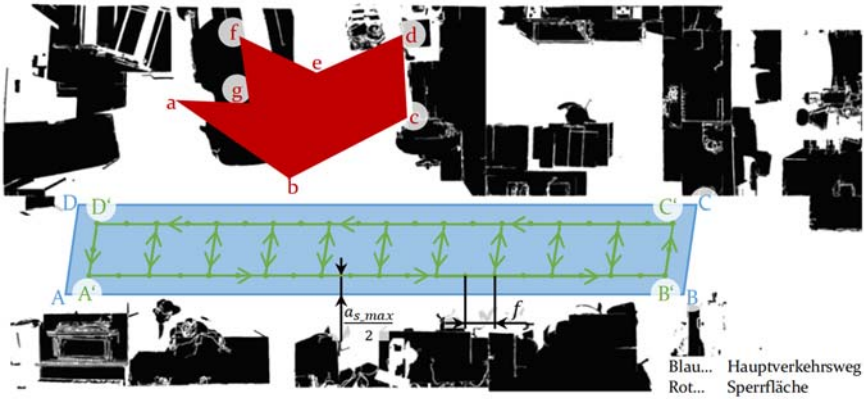


Bild 30: Prinzipskizze eines Hauptverkehrsbereichs (ABCD) und einer Sperrfläche (abcdefg)

Sperrflächen

Analog zu den Hauptverkehrswegen werden die Sperrflächen durch den Anwender über den Dienst *Barred_Areas* definiert, ebenfalls mit einer ID versehen und haben das Ziel, Bereiche im Weltmodell zu beschreiben, in die keine Transportroboter einfahren dürfen. Hierdurch können beispielsweise sicherheitskritische Werksbereiche dauerhaft gesperrt werden. Im Gegensatz zum Hauptverkehrsweg ist gemäß Bild 30 eine Sperrfläche als Polygon definiert und kann zudem statische Hindernisobjekte ganz oder teilweise beinhalten (roter Bereich).

Das durch die Sperrflächen ergänzte Binärbild des Arbeitsraums und die Fläche der Hauptverkehrsstrecken mit den Knotenpunkten und Kanten beschreiben ein umfassendes Weltmodell der intralogistischen Arbeitsumgebung und werden vom Dienst *Logistics_Environment* angeboten.

5.3.3 Durchführung einer fahrzeughtypspezifischen Routenplanung

Zur konsequenten und ganzheitlichen Umsetzung der Vorgabe an die Gesamtarchitektur, den Informationsfluss zwischen den Diensten auf ein Minimum zu reduzieren, erfolgt die Erstellung der Routenpläne für jede Fahrzeugklasse in einem zentralen Dienst. Bei der Initialisierung eines Transportroboters stellt dieser eine Anfrage an den Dienst *Routing_Graph* zur Übermittlung eines passenden Routenplans anhand des fahrzeughtypspezifischen Sicherheitsabstands a_s . Falls für diesen Fahrzeugtyp kein Routenplan zu Verfügung steht, wird dieser vom erforschten *Multi-Query-Roadmap* (MQR)-Algorithmus, einer Sonderform des PRM (vgl. 3.3.3), erzeugt.

Im ersten Schritt generiert der Algorithmus einen Zufallspunkt im übermittelten Arbeitsraums des *Logistics_Environment*-Dienstes. Es folgt eine Prüfung, ob dieser Punkt innerhalb des befahrbaren Bereichs liegt und folglich außerhalb eines Hindernisses, einer Sperrfläche oder einer Hauptverkehrsstrecke. Die zweite Prüfbedingung ist, dass der minimale Abstand zur nächsten Sperrfläche, zum nächsten Hindernis oder vorhandenen Knotenpunkt größer als a_s ist. Dadurch wird sichergestellt, dass das Fahrzeug in diesem Punkt jede Orientierung einnehmen kann, ohne mit einem Hindernis oder einer Sperrfläche zu kollidieren. Dies ist notwendig, da aufgrund der Systematik der MSL die Hindernisse lediglich im Arbeitsraum W und nicht im Konfigurationsraum C beschrieben werden. Sind beide Bedingungen erfüllt, wird der Knotenpunkt eingetragen, andernfalls verworfen. Im Gegensatz zum etablierten *Multi-Query-Tree* ist das Abbruchkriterium für die Wiederholung der Zufallspunktgenerierung nicht durch die Anzahl der erzeugten Zufallspunkte, sondern durch die Anzahl der Fehlversuche zwischen zwei Eintragungen definiert. Dies ist notwendig, da die Anzahl der möglichen Zufallspunkte im gegebenen Arbeitsraum vom Sicherheitsabstand a_s abhängig ist. Im Anschluss erfolgt die Verbindung der zufallsgenerierten Knotenpunkte über Kanten, die einerseits auf Kollisionsfreiheit mit Sperrflächen und Hindernissen geprüft werden und deren Länge andererseits $2a_s$ nicht überschreiten darf. Abschließend werden die vom *Logistics Environment* übermittelten Knoten- und Zwischenknotenpunkte der

Hauptverkehrswege ins Streckennetz integriert. Der Algorithmus verbindet dabei alle Knoten und Zwischenknoten der Hauptverkehrswege mit allen Knotenpunkten des Streckennetzes, wenn der Abstand innerhalb $2,5a_s$ und die Verbindung kollisionsfrei ist. Ein graphisch aufbereitetes Beispiel dieses Algorithmus zeigt Bild 31.

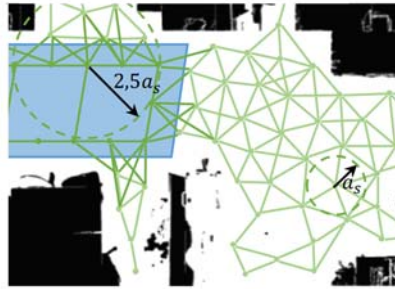


Bild 31: Erstellung des Routenplans mithilfe des MQR-Algorithmus

Der Dienst *Routing_Graph* hält somit für jeden Fahrzeugtyp eine Streckenkarte mit spezifischen Zufallspunkten im Arbeitsraum in Abhängigkeit des Sicherheitsabstands a_s bereit. Die Information über das befahrbare Streckennetz ist komplett über die Knotenpunkte und Kanten definiert, die einen gerichteten und gewichteten Graphen ergeben, der durch die globale Adjazenzmatrix abgebildet ist. Um bei der Generierung eines globalen Pfads, die Verwendung der Hauptverkehrswege zu forcieren, werden die Kantengewichte zwischen den Knoten und Zwischenknoten um den Faktor 1,5 reduziert. Der Datentransfer zu den anfragenden Transportrobotern reduziert sich durch dieses Vorgehen auf das Übermitteln der spezifischen Adjazenzmatrix, dem Streckennetz. Das datenintensive intralogistische Weltmodell wird dagegen nicht übertragen. Zudem erfolgt die rechenaufwendige Generierung der fahrzeugtypspezifischen Adjazenzmatrix für jeden Fahrzeugtyp einmalig und damit unabhängig von der Anzahl der eingesetzten Entitäten. Dadurch wird der Grundsatz der Architektur, möglichst geringe Datenmengen zwischen den Diensten zu transportieren und die Gesamtrechenlast zu reduzieren, maßgeblich erfüllt.

5.4 Fahrzeuginterne Planung des Pfads

Die Beschreibung der verteilten Systemarchitektur in Abschnitt 5.1 erörtert die Vorteile einer differenzierten Bearbeitung der Daten der Infrastruktursensoren in der MSL und auf der Fahrzeugseite. Die Verarbeitung des adaptiven Hintergrundbilds erfolgt gemäß Abschnitt 5.3 innerhalb der

MSL, die den Transportentitäten eine fahrzeugtypspezifische Adjazenzmatrix des Arbeitsraums zur Verfügung stellt. Diese Informationen aufgreifend, erzeugt das Transportfahrzeug anhand des auftragsindividuellen Zielpunktes autonom einen globalen Pfad, ausgehend von der aktuellen Position. Während der Durchführung des Pfads werden zusätzliche lokale Informationen der aktuellen Umgebung eingebunden, um auf dynamische Hindernisse reagieren zu können.

5.4.1 Glättung des globalen Pfads durch Interpolation

Durch die erforschte Systematik der MSL, den globalen Fahrzeugpfad anhand der MQR zu erstellen, entsteht an den Knotenpunkten eine Unstetigkeit der Fahrzeugtrajektorie. Dies bedeutet, dass im Stillstand eine Umorientierung des Fahrzeugs erfolgen muss und somit die mittlere Geschwindigkeit sinkt. Es ist daher notwendig, den auftragsspezifischen globalen Pfad vor der Weitergabe an den lokalen Pfadplaner zu glätten. In der Literatur werden hierzu im Bereich der autonomen Robotik verschiedene Ansätze diskutiert, deren bekannteste Vertreter auf

- dem *Elastic Band Planner* [130]
- und der *Dynamic Window Approach* [131]

aufbauen, die eine Bahnglättung im lokalen Pfadplaner integrieren. Die Stärken beider Ansätze sind, dass neben der reinen Pfadglättung zudem eine Pfadoptimierung stattfindet, falls Hindernisse, die vom globalen Pfadplaner nicht berücksichtigt wurden, eine Umplanung erfordern. Durch die Methode der erforschten Architektur sind jedoch bei der globalen Pfadplanung sämtliche statischen Hindernisse berücksichtigt, da nicht auf historischen Karten des Arbeitsraums geplant wird. Zudem wird der Ansatz verfolgt, dass auf eine drohende Kollision mit dynamischen Hindernissen durch Anpassung der Geschwindigkeit (vgl. Abschnitt 5.4.2) reagiert wird und zunächst keine Umplanung des Pfads erfolgt. Die Funktion des *Path_Interpolation*-Dienstes reduziert sich daher auf die Glättung des globalen Pfads, und lehnt sich an den Regelalgorithmus von Buschhaus [132] an. Bild 32 illustriert die Herangehensweise, die Trajektorie des globalen Pfads zu interpolieren und von der aktuellen Position einen vorlaufenden Interpolationspunkt anzupeilen.

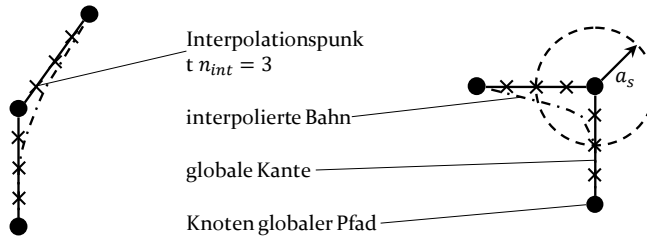


Bild 32: Interpolation der globalen Pfadkanten in Anlehnung an [133]

Der Abstand zwischen zwei Knotenpunkten des globalen Pfads wird durch $n_{int} \geq 1$ Interpolationspunkte unterteilt, deren Anzahl im Parameterserver des Fahrzeugs festgelegt ist. Sämtliche Knoten und Interpolationspunkte der Trajektorie, deren euklidische Distanz a_{int} zur aktuellen Fahrzeugposition den Sicherheitsabstand a_s unterschreiten, gelten als besucht. Das Fahrzeug peilt folglich stets den Punkt auf der Soll-Trajektorie an, der gerade außerhalb des fahrzeugtypspezifischen Sicherheitsabstands liegt. Aufgrund der Ausprägung der erforschten Gesamtarchitektur beträgt im globalen Pfad der Winkel zwischen zwei Kanten maximal 90° . Für die maximale Umorientierung ergeben sich somit 45° , wenn für n_{int} der Minimalwert initialisiert ist. Bei der Pfadausführung des Fahrzeugs kann dieser Extremfall durch eine Kreisbogenfahrt angenähert werden, ohne den Sicherheitsbereich a_s zu verlassen. Dies ist notwendig, da das Fahrzeug lediglich über die Informationen der Adjazenzmatrix verfügt und somit keine Informationen zu Hinderniskanten erhält.

Durch das Vorgehen des Algorithmus und die Parametrisierung der Interpolation werden einerseits Unstetigkeiten des globalen Pfads geglättet und andererseits statische Umorientierungen während der Fahrt vermieden, was zu einer höheren mittleren Geschwindigkeit bei der Pfadausführung führt. Erst bei Erreichen der Zielposition ist eine Ausrichtung des Fahrzeugs durch eine reine Drehbewegung zur Zielpose notwendig.

5.4.2 Einbindung lokaler Informationen bei der Pfadausführung

Die anschließende Durchführung des lokalen Pfads setzt sich aus zwei elementaren Bausteinen zusammen. Der

- Ortung und
- Reaktion auf dynamische Hindernisse,

die beide die Einbindung von lokalen Informationen der Infrastruktursensoren und optional der On-board-Sensoren voraussetzen. Die Lösungsansätze beider Aufgaben setzen auf der Übermittlung der dynamischen Objektdaten des *Moving_Objects*-Broadcasts der sensornahen Auswerteeinheiten auf. Für die Problemstellung der Ortung von VAV beschreibt Lütteke [14] ein Verfahren, bei dem die Fahrzeuge eigenständig auf Basis der vom Infrastruktursensor übermittelten Hindernisstrajektorien im Vergleich zur eigenen Trajektorie eine Selbsttortung durchführen. Dieses Verfahren ist in der erforschten Systemarchitektur unverändert integriert, bezüglich der exakten Beschreibung des Algorithmus wird daher an die entsprechende Literatur verwiesen.

Der geglättete Pfad des *Path_Interpolation*-Dienstes berücksichtigt aufgrund des Datenflusses der Architektur lediglich die statischen Hindernisobjekte. Erst bei der Generierung der expliziten Fahrbefehle an die Antriebssteuerung durch den fahrzeugeigenen *Local_Path*-Dienst werden die Daten von *Moving_Objects* integriert und damit die dynamischen Hindernisse in die Pfadausführung einbezogen. Optional können weitere On-board-Sensoren wie Ultraschallsensoren oder LiDAR-Systeme das Abbild der lokalen Fahrzeugumgebung komplettieren. Die implementierte Strategie beruht auf dem Grundsatz, dass die Fahrzeuge die geplante Strecke nicht verlassen sollen, sondern mögliche Kollisionen mit dynamischen Hindernissen durch Anpassung der Geschwindigkeit vermeiden. Einerseits wird dadurch der optimal geplante Pfad auch bei kurzzeitigen Störungen beibehalten, andererseits wird die Reaktion des Transportfahrzeugs für Personen im Arbeitsraum vorhersehbar. Der implementierte Algorithmus zur Berechnung der Fahrzeuggeschwindigkeit beruht auf der Auswertung der Hindernisstrajektorien im Fahrbereich mit einer überlagerten Lagepeilung. Hierzu werden gemäß Bild 33 die Konturen und Trajektorien-schätzungen der dynamischen Hindernisse über den *Moving_Objects* Broadcast der entsprechenden Deckenkameras in die lokale Karte fusioniert.

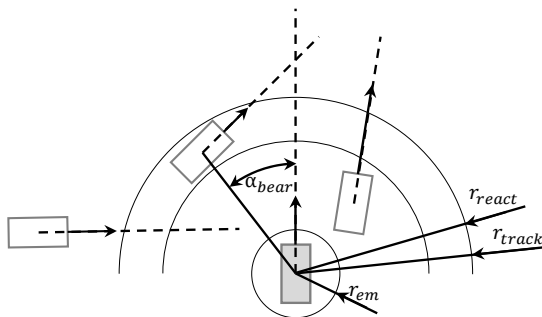


Bild 33: Geschwindigkeitsanpassung aufgrund lokaler Hindernisinformationen

Im Parameterserver des Transportroboters sind die Parameter r_{track} , r_{react} und r_{em} definiert, die das Verhalten des Roboters regeln. Der Halbkreis mit dem Radius r_{track} beschreibt eine Fläche, in der die Trajektorien der vorhandenen dynamischen Hindernisse ausgewertet und mit der eigenen verglichen werden. Findet sich dabei ein Schnittpunkt, ermittelt der Algorithmus den Peilungswinkel α_{bear} zum Objekt. Falls sich dieser über die Zeit nicht verändert, besteht potenziell eine Kollision mit dem dynamischen Hindernis. Unterschreitet eines dieser Objekte den Radius r_{react} , verringert das Fahrzeug schrittweise die aktuelle Fahrgeschwindigkeit, um die Kollision zu vermeiden. Der Sicherheitsradius r_{em} dient lediglich als Notstoppanweisung, falls ein Objekt den definierten Sicherheitsabstand zum Fahrzeug unterschreitet. Handelt es sich bei dem dynamischen Hindernis ebenfalls um eine autonome Transportentität die an die MSL angebunden ist, gelten zusätzlich die folgenden Regeln um ein gegenseitiges Blockieren zu verhindern:

- Vorfahrende Fahrzeuge haben Vorrang.
- Sich entgegenkommende Fahrzeuge mit $-45^\circ \leq \alpha_{bear} \leq 45^\circ$ weichen stets nach rechts aus.
- Bei sich entgegenkommenden Fahrzeugen mit $|\alpha_{bear}| > 45^\circ$ gilt die Rechts-vor-Links-Regel.

Durch das Einbinden der vorverarbeiteten Informationen und der Beschränkung auf die nähere Umgebung des spezifischen Fahrzeugs wird der Grundsatz der Gesamtarchitektur, möglichst geringe Datenmengen auszutauschen, erfüllt. Zudem zeigt sich das Potenzial der Gesamtlösung in Kreuzungsbereichen, die durch reine On-board-Sensorik nicht einsehbar sind. Über die Infrastruktursensoren sind die Fahrzeuge befähigt, den Kreuzungsbereich ohne Geschwindigkeitsreduzierung zu durchfahren oder gegebenenfalls frühzeitig auf Kreuzungsverkehr zu reagieren, da die notwendigen Informationen dem Algorithmus frühzeitig zur Verfügung stehen.

5.5 Zusammenfassung und Beurteilung

Grundlage des erforschten Servicemodells zur Digitalisierung des Arbeitsraums, sowie der Routenplanung und Pfadausführung bildet die in Abschnitt 5.1 vorgestellte verteilte Systemarchitektur der MSL. Die zentrale Eigenschaft der MSL ist, dass die Algorithmen zur Erfüllung der notwendigen Einzelaufgaben in unabhängige Dienste ausgelagert werden. Diese lassen sich in drei Hauptgruppen unterscheiden: die Dienste der sensornahen Auswerteeinheit, die Fahrzeugdienste und die zentralen Dienste. Zudem

wird durch die Systemgestaltung der Grundsatz verfolgt, den Datenaustausch zwischen den einzelnen Instanzen auf ein Minimum zu reduzieren und stets bedarfsorientiert auszulösen. Das System lässt zudem zu, den Ort und die Anzahl der physischen Einheiten, auf denen die Algorithmen implementiert sind, frei zu wählen. Vor allem die zentralen Dienste können somit bei einer Erweiterung des Systems auf unterschiedliche Rechner aufgeteilt werden, damit die Gesamtperformanz erhalten bleibt. Weiterhin ist das System sowohl in der Fläche als auch in der Anzahl der verwendeten Transportroboter unbegrenzt skalierbar, da die Gesamtarchitektur vorsieht, dass die zur Skalierung notwendigen Dienste sowohl bei den Infrastruktursensoren als auch auf den Fahrzeugen integriert sind. Lediglich die einheitlichen fahrzeugübergreifenden Dienste werden durch eine zentrale Dienstebene abgebildet. Aufgrund der definierten Kommunikationsstruktur zwischen den Diensten ist es zudem möglich, dass bereits verwendete FTS Zwischeninformationen nutzen, um ihre Wirtschaftlichkeit bei der Pfadplanung zu erhöhen. Weiterführend kann dies genutzt werden, um logistikferne Aufgaben, wie z. B. die Konformität des aktuellen Layouts, mit den vorhandenen Hallenplänen zu prüfen. Die Architektur bietet daher neben der praktischen Notwendigkeit einen übergreifenden Mehrwert, der in vergleichbaren Systemen nicht gegeben ist.

In Abschnitt 5.2 wird eine Methode zur dezentralen Erfassung des Arbeitsraums durch Infrastruktursensoren erörtert. Hierbei wird die erfasste Szenerie direkt von einer sensornahen Auswerteeinheit in ein adaptives Hintergrundbild und einen Broadcast, der die Trajektorie und Position von dynamischen Hindernissen sendet, überführt. Die adaptiven Hintergrundbilder der einzelnen Sensoren beinhalten sämtliche statische Hindernisse und werden anschließend zu einem Gesamtbild des Arbeitsraums fusioniert. Zudem wird bei der Umsetzung die Restriktion bekannter Ansätze behoben, damit eine robuste Separation der Vordergrund- und Hintergrundinformationen gegenüber wechselnden Lichtverhältnissen möglich ist. Um die Latenzzeit zur Übermittlung von dynamischen Hindernissen an die Fahrzeuge gering zu halten, wird an dieser Stelle der Grundsatz der Architektur auf Dienste zu setzen verlassen und ein Broadcast eingesetzt.

Das innovative Kernelement stellt die Bereitstellung eines fahrzeugtypspezifischen Streckennetzes innerhalb des gesamten Arbeitsraums der Transportentitäten dar. Abschnitt 5.3 erläutert die Methode, wie anhand einer Kantendetektion in spezifischen Farbkanälen des Bildes eine Unterscheidung zwischen dem befahrbaren Bereich und den statischen Hindernissen getroffen werden kann. Dieses Weltmodell wird zudem mit Informationen

angereichert, wodurch eine effektivere Nutzung des verfügbaren Fahrbereichs erlangt wird. In diesem Bereich werden abschließend zentral die Fahrstrecken für alle logistischen Fahrzeugtypen erstellt. Das Gesamtsystem ist somit nicht wie etablierte Ausführungsformen auf eine spezifische Anwendung beschränkt, sondern erlaubt eine parallele Verwendung unterschiedlichster intralogistischer Lösungen in einem gemeinsamen Arbeitsraum.

Konsequent wird der Grundsatz der Interoperabilität und generischen Definition der Dienste auch bei der fahrzeuginternen Pfadplanung und -durchführung verfolgt, was Abschnitt 5.4 aufzeigt. Die globale auftragsspezifische Pfadplanung erfolgt auf Basis der zur Verfügung gestellten Streckenkarte und wird vor der Ausführung geglättet, um eine höhere mittlere Geschwindigkeit der Transportroboter zu erzielen. Zudem werden bei der Ausführung die Informationen der Deckensensoren zu dynamischen Hindernissen im Nahbereich des jeweiligen Fahrzeugs berücksichtigt.

6 Transfer der intralogistischen Auftragsallokation in eine ereignisdiskrete Simulation

Ein zentraler Kritikpunkt an der dezentralen Disposition von Fertigungsaufträgen durch ein MAS ist, dass die Vorhersagbarkeit des Verhaltens und der Wechselwirkungen mit der Anzahl der beteiligten Agenten abnehmen, obwohl die Einzelentscheidungen häufig trivial und nachvollziehbar sind [93]. Dies führt dazu, dass große Materialflusssysteme, die durch eine Vielzahl an Agenten repräsentiert werden, keinem Determinismus unterliegen und die Reaktion des Systems auf diskrete Störeinflüsse im Vorfeld nicht abgeschätzt werden kann [133]. Somit ist eine Simulation des Zielsystems und die iterative Optimierung der Entscheidungsalgorithmen der an der Disposition beteiligten Agenten während der Planungsphase unumgänglich. Dadurch kann einerseits das Systemverhalten generell und die Reaktion auf diskrete Störungen im Speziellen getestet und hinsichtlich übergeordneter Ziele optimiert werden. Die Simulation der Entscheidungsmechanismen eines intralogistischen MAS sind Gegenstand zahlreicher MAS-Simulationstools, jedoch ist die Komplexität einer industriellen Fertigung mit ihren ereignisdiskreten Störungen hier nicht abbildbar [134]. In Softwareprogrammen, die auf die Simulation von industriellen Wertströmen spezialisiert sind, können die Entscheidungsmechanismen eines intralogistischen MAS auf der anderen Seite nur sehr bedingt implementiert werden [135]. Zudem ist eine Umsetzung der in der Simulation optimierten Strategie von der Planungsphase in den realen Betrieb nicht gewährleistet. Die Aufgabenstellung, einerseits die Ontologie eines intralogistischen MAS in der Planungsphase einer flexiblen Fertigungsstruktur in eine ereignisdiskrete Simulationsumgebung zu integrieren und dadurch in die Lage versetzt zu werden, den Allokationsalgorithmus zu optimieren, sowie einen verlustfreien Transfer in den operativen Betrieb zu ermöglichen, wird in den nachfolgend erläuterten Forschungsarbeiten aufgenommen und deren Lösungsmöglichkeiten erörtert. [136, 137]

Etablierte Einsatzfelder ereignisdiskreter Simulationssoftware und der Lösungsansatz, die Entscheidungsmechanismen eines MAS bereits in der Planungsphase mit der Simulation zu koppeln, wird in Abschnitt 6.1 vorgestellt sowie die Überführung von der Planungsphase in den operativen Betrieb skizziert. Die Methode zur Optimierung des Allokationsalgorithmus innerhalb einer DES als *Software-in-the-Loop* (SiL) adressiert der Abschnitt

6.2. Abschnitt 6.3 beschreibt die Methode zum Transfer der Simulationstologie in die realen Entitäten und die kontinuierliche Verbesserung des Modells durch den Einsatz von *Hardware-in-the-Loop* (HiL) und der Abschnitt 6.4 validiert das Modell. Abschließend werden die gewonnenen Erkenntnisse in Abschnitt 6.5 zusammengefasst.

Einzelne Aspekte der Ausführungen der Folgeabschnitte 6.1 bis 6.3 sowie des Abschnitts 7.4, der die Umsetzung und Evaluierung der beschriebenen Methode umfasst, sind an den am Lehrstuhl FAPS entstandenen und vom Autor angeleiteten studentischen Arbeiten [S50–S60] angelehnt. Bestandteile und Vorarbeiten der dargestellten Methoden und Erkenntnisse wurden der internationalen Fachwelt auch in den vom Autor erstellten Veröffentlichungen [P5, P18, P19, P8] zugänglich gemacht.

6.1 Architektur zur Kopplung eines MAS mit einer DES

Die elementare Anforderung zur Integration des Auftragsallokationsmechanismus des iLES in eine ereignisdiskrete Materialflusssimulation besteht darin, die verwendete Ontologie des MAS in der Simulationsumgebung abzubilden. Weiterhin muss das Verhalten der einzelnen Agenten auf diskrete Ereignisse abgestimmt werden können, um eine Optimierung des Gesamtsystems zu ermöglichen. Es entsteht folglich ein digitales Abbild der Transportentitäten innerhalb der Simulationsumgebung, welches ohne Implementierungsaufwand in den realen Betrieb überführbar ist. Um diese Anforderungen vollumfänglich zu erfüllen, muss der Algorithmus zur Auftragsdisposition als SiL in die Simulation eingebunden werden.

6.1.1 Einsatzfelder ereignisdiskreter Materialflusssimulationen

Mit Hilfe einer Materialflusssimulation sollen Produktionsprozesse nachgebildet werden, um die dynamischen Prozesse in einem Modell zu testen, zu optimieren und auf die reale Produktion zu übertragen [138]. Dafür muss ein Modell gewählt werden, dessen Detaillierungsgrad einerseits aussagekräftige Ergebnisse im Hinblick auf das Simulationsziel liefert, andererseits der notwendige Aufwand zum Erstellen des Modells im wirtschaftlich sinnvollen Rahmen bleibt [139]. Im industriellen Umfeld werden für die verschiedenen Anwendungsdomänen spezifische Simulationsmodelle verwendet, wobei sich für die Modellierung des Materialflusses ereignisdiskrete Simulationsumgebungen als de-facto Standard etabliert haben [140]. Gemäß Bild 34 werden die Aktivitäten von konkreten Ereignissen ausgelöst

und folglich die Zustandsveränderung des Modells berechnet. Dabei sind vor allem anlagenbezogene Aspekte wie Maschinenauslastungen sowie Lagerkapazitäten von Interesse. In aktuellen Forschungsvorhaben rücken zudem energetische Anlagenparameter wie zustandsabhängige Lastprofile einer Produktionsanlage in den Fokus [141]. [142]

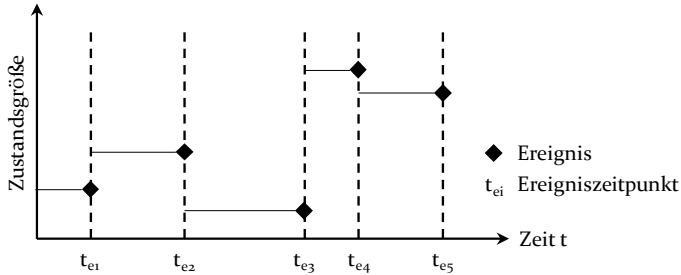


Bild 34: Schaubild zu Zuständen, Ereignissen und Zustandswechseln einer ereignisdiskreten Simulation (in Anlehnung an [144])

6.1.2 Lösungsansatz zur Integration des iLES in ereignisdiskrete Materialflusssimulationen

Bei der Abbildung von intralogistischen Materialflusssystemen sind ereignisdiskrete Simulationsumgebungen auf die etablierten Lösungsansätze zur Auftragsdisposition beschränkt. Dies bedeutet, dass die Auftragsaufteilung unter den Transportfahrzeugen nach vordefinierten Kriterien erfolgt und somit dem Allokationsprinzip klassischer FTS nach Abschnitt 3.2.2 folgt. Zwar bieten Simulationsumgebungen die Möglichkeit das Verhandlungsverhalten eines MAS durch frei programmierbare Methoden abzubilden, jedoch erweist sich einerseits die Abbildung der Funktionalität und andererseits der Transfer in das reale System als schwierig. [143]

Der erforschte Lösungsansatz basiert daher darauf, nicht die Funktionalität des Allokationsalgorithmus in der Simulationsumgebung abzubilden, sondern das iLES direkt in die Simulation als SiL einzubinden. In einem ersten Schritt entsteht anhand der Planungsdaten ein Simulationsmodell der Fertigungsumgebung, in das die Transportroboter mit dem initialen Parameterserver eingebunden werden. Die Fahrzeugagenten werden in einem Prototyp des iLES repräsentiert. Die angestoßenen Transportaufträge stellen ein diskretes Ereignis der Simulation dar, welches die Berechnung der Zustandsänderungen bedingen. Hierzu wird ein Agent des Transportauftrags innerhalb des iLES-Prototyps erstellt, die darauffolgende Verhandlung der Agenten folgt dem Allokationsverhalten nach Abschnitt 4.1.2. Das Simulationsmodell erhält lediglich die Rückmeldung der Fahrzeugagenten und

bringt diese zur Ausführung. Der Vorteil dieses Ansatzes ist, dass das Verhalten der Agenten während der Planungsphase hinsichtlich übergeordneter Ziele optimiert werden kann. Ein Transfer der simulierten Fahrzeugagenten ist zudem ohne Applikationsverluste gegeben, da lediglich der Ort, nicht jedoch die Syntax oder die Ontologie verändert wird. Die zu übertragenden Daten zwischen der Simulationsumgebung und dem iLES sowie die erforderliche Architektur sowohl während der Planungsphase, als auch im operativen Betrieb, sind Inhalt der Abschnitte 6.2 und 6.3.

6.2 Methode zur Einbindung des Allokationsalgorithmus in der Planungsphase

Eine Integration des Auftragsallokationsalgorithmus in die DES durch die Realisierung einer SiL ist umsetzbar, da die Generierung von Aufträgen ein diskretes Ereignis in einer Materialflusssimulation darstellt. Die folgende Berechnung der Systemänderung erfolgt nicht innerhalb der Simulationsumgebung, sondern über den ausgelagerten Algorithmus des iLES. Die Stärken der Materialflusssimulationen, wie die Berechnung von Maschinenauslastungen und energetischen Anlagenparametern, bleiben davon unberührt. Im Hinblick auf die Verwendung des Lösungsansatzes während der Planungsphase ergeben sich zwei Forschungsfragen: die Integration der iLES-Ontologie mit der Kommunikation zwischen den Systemen, sowie die Interpretation der Ergebnisse und Optimierung des Systems.

6.2.1 Einbindung der intralogistischen Ontologie in eine Simulationsumgebung

Während der Planungsphase stellen das iLES und die DES zwei getrennte Systeme dar, die entweder auf einer Recheneinheit oder über ein Netzwerk miteinander kommunizieren. Die Simulationsvariante des iLES verfügt über dieselben Strukturen wie das operative Pendant, mit der Ausnahme, dass auf der Ebene der Zustandsautomaten nur der *Forecast* implementiert ist (vgl. Bild 35). In der DES werden die Produktionsanlagen gemäß dem geplanten Layout positioniert und die geschätzten Wegstrecken zwischen den potentiellen Auftragsquellen und -senken für jede Fahrzeugklasse in der entsprechenden Wegstreckentabelle eingetragen. Anhand dieser Tabelle wird für jede Fahrzeugklasse innerhalb des iLES eine Adjazenzmatrix erstellt und über einen simulierten *Routing_Graph*-Dienst angeboten. Zudem sind graphische Modelle der Transportfahrzeuge in die Simulation eingebunden, deren Charakterisierung klassenweise anhand einer Parame-

tertabelle erfolgt. Bei Simulationsstart wird im iLES eine Instanz des Fahrzeugagenten für jedes simulierte Transportfahrzeug erzeugt, mit einer eindeutigen Fahrzeug-ID versehen und der Parameterserver anhand der Werte in der Parametertabelle der entsprechenden Fahrzeugklasse initialisiert.

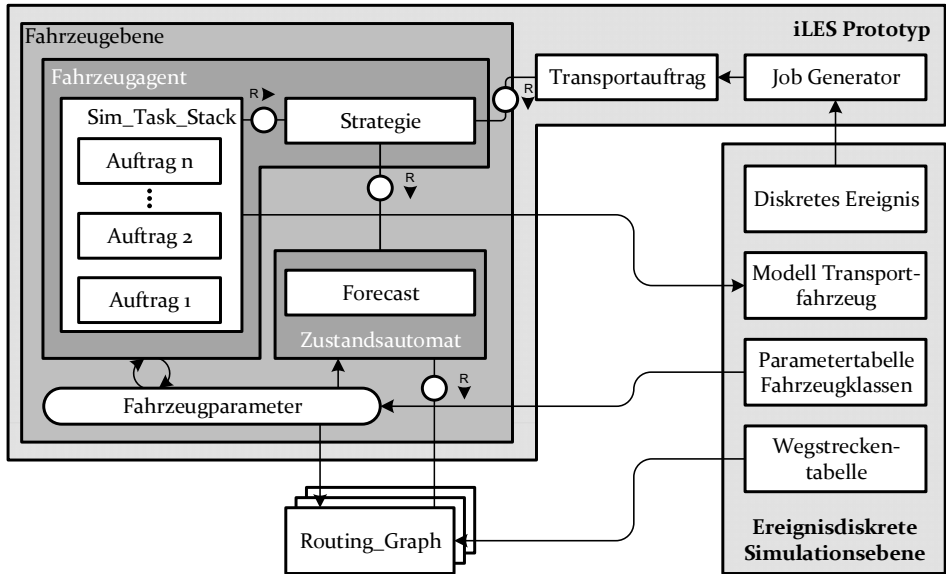


Bild 35: FMC-Block-Diagramm zur Kommunikation zwischen dem iLES und der DES in der Planungsphase

Bei Eintritt eines diskreten Ereignisses in Form eines Transportauftrags übermittelt die Simulationssoftware die Auftragsdaten an das iLES, auf deren Basis ein Transportauftragsagent instanziiert wird. Die Auftragsverhandlung des Transportauftragsagenten mit den Fahrzeugagenten erfolgt gemäß den Ausführungen in Abschnitt 4.1. Das Abschätzen des globalen Pfads beruht dabei allerdings auf den Werten der Wegstreckentabellen und der Transportauftragsagent disponiert den Auftrag. Die Information zur Auftragsdurchführung hingegen übermittelt der Fahrzeugagent aus dem *Sim_Task_Stack* durch Paarung der Fahrzeug-ID und der geschätzten Auftragsparameter mit den spezifischen Auftragsdaten an die Simulation. Hier wird der Auftrag vom simulierten Transportfahrzeug unter Berücksichtigung der abgeschätzten Auftragsdaten durchgeführt.

Durch dieses Vorgehen kann einerseits der Auftragsallokationsalgorithmus in die DES integriert werden. Andererseits liefert die Simulation dem iLES die notwendigen Eingangsgrößen, um einen realen Betrieb zu simulieren.

Die Verhandlungs- und Dispositionsmechanismen unterscheiden sich folglich nicht zu einem operativen Betrieb.

6.2.2 Optimierung des Layouts und der Auftragsdefinition anhand von KPIs während der Planungsphase

Die beschriebene SiL-Lösung ermöglicht es einerseits das Fertigungslayout und die benötigten Ressourcen innerhalb der DES zu validieren und andererseits die Strategie der einzelnen Auftragsagentenklassen des iLES für den spezifischen Anwendungsfall zu optimieren. Kenngrößen zum Vergleich und der Bewertung unterschiedlicher Gesamtkonstellationen sind übergeordnete Zielgrößen wie die mittlere Auftragsdurchlaufzeit, Liefererfüllungsgrad und die benötigten Maschinen- und Anlagenressourcen. Die Variation der Eingangsparameter erfolgt dabei ausschließlich über die DES, in der iLES wird lediglich die Zuordnung der im Produktionsplan hinterlegten Aufträge zu den sieben Auftragsarten nach Abschnitt 4.1.2 verändert, um unterschiedliche Strategien abzubilden. Eine Variation des Fahrzeugagentenverhaltens ist nicht vorgesehen, da hierdurch eine Optimierung des einzelnen Transportfahrzeugs hinsichtlich der Amortisation oder Transportleistung möglich ist, was jedoch nicht zu einer Optimierung der Gesamtsystemkenngrößen führt.

Durch die beschriebenen Parameter des Modells ist es möglich, neben den unterschiedlichen Produktionslayouts und den eingesetzten Ressourcen zudem das Verhalten des Produktionssystems auf die Zuteilungs- und Vergabestrategie der einzelnen Transportaufträge des Produktionsplans zu simulieren. Zudem kann aus der Reaktion des Systems auf diskrete Störeinflüsse wie Maschinenstillstände die Robustheit des Produktionsnetzwerks abgeschätzt und das nicht deterministische Verhalten der Auftragsdisposition abgebildet werden.

6.3 Überführung des Allokationsalgorithmus aus der Planungsphase in die Betriebsphase

Neben der Integration des Auftragsallokationsalgorithmus in die DES stellt der Transfer der MAS-Ontologie in den produktiven Betrieb im Allgemeinen und die Migration der Fahrzeugagenten auf die Transportentitäten im Speziellen den zweiten Kern der Forschungsarbeiten dar. Zudem soll das Simulationsmodell der Planungsphase während des operativen Betriebs kontinuierlich anhand der Rückgabewerte der Transportentitäten aktualisiert werden. Somit erstreckt sich die Verwendung des Modells von der Planungsphase über den Produktionsstart bis zum Produktionsende.

6.3.1 Transfer der Ontologie auf die physischen Transporteinheiten

Die Architektur des iLES sowie die Interaktion zwischen den Agenten und die Auftragsdisposition im produktiven Einsatz erfolgt analog zu den Ausführungen in Kapitel 4. Zur Initialisierung der Fahrzeugparameterserver dienen die Simulationsdaten aus der Fahrzeugklassentabelle. Die Zuweisung der realen Transportaufträge zur jeweiligen Transportauftragsklasse erfolgt anhand der simulierten Zuordnungsstrategie. Der Aufbau des produktiven iLES ist identisch zu dem in der Simulation verwendeten Prototyp, jedoch erfolgt die Einlastung von spezifischen Transportaufträgen nicht über die Simulationsschnittstelle, sondern über ein übergeordnetes Managementsystem unter Beibehaltung der Ontologie. Der Fahrzeugagent wird aus der iLES herausgelöst und im Softwarepaket mit den Zustandsautomaten auf die entsprechende Transportentität migriert. Dabei wird weder der Quellcode des Fahrzeugagenten verändert, noch die Kommunikationsstruktur zu Transportauftragsagenten verändert. Lediglich die Rückmeldung an die DES nach einer erfolgreichen Verhandlung entfällt und wird durch die Ablage im *Task_Stack* ersetzt. Gemäß Bild 36 verändern sich beim Transfer der iLES-Ontologie und des Auftragsallokationsalgorithmus vom Prototyp in den operativen Betrieb nur die Absender und Adressaten von eingehenden und ausgehenden Informationen. Die Strategie der Auftragsdisposition bleibt jedoch von Änderungen oder Anpassungen unberührt. Somit ist im realen Betrieb ein zur Simulation identisches Verhalten der Agenten während der Verhandlung gewährleistet.

Die Systemarchitektur zeigt, dass ein zur realen Produktion paralleler Weiterbetrieb der Simulation möglich ist. Dadurch können die Reaktion des Produktionsnetzwerks auf zukünftige Produktionspläne getestet und im Vorfeld Anpassungen an der Zuordnung der Auftragsarten oder der Parameter vorgenommen werden. Der simulierte Fahrzeugagent greift dazu auf den *Forecast* und den Parameterserver seines physischen Pendants zu, weist dasselbe Verhalten wie der entsprechende Fahrzeugagent auf, legt die disponierten Fahraufträge jedoch weiterhin im separaten *Sim_Task_Stack* ab. Die Übermittlung der Transportaufträge aus der Simulation erfolgt, wie die des übergeordneten Fertigungsmanagementsystems, über die iLES mit dem Zusatz, dass es sich um einen simulierten Transportauftrag handelt. Die Dispositionsverhandlung erfolgt dann ausschließlich mit den simulierten Fahrzeugagenten. Hierdurch entsteht ein digitaler Zwilling jeder Entität des flexiblen Materialflusssystems, der nach dem Grundsatz des HiL die Algorithmen der Zielplattform nutzt. Es ist dabei unabhängig, auf welcher

physikalischen Recheneinheit der Algorithmus der simulierten Fahrzeuga-
genten implementiert ist, dem Simulationsrechner, dem Produktionsrech-
ner oder der Recheneinheiten der Transportentitäten. Hinsichtlich der ent-
stehenden Datenraten zwischen Agent und Zustandsautomat ist letzteres
jedoch zu bevorzugen.

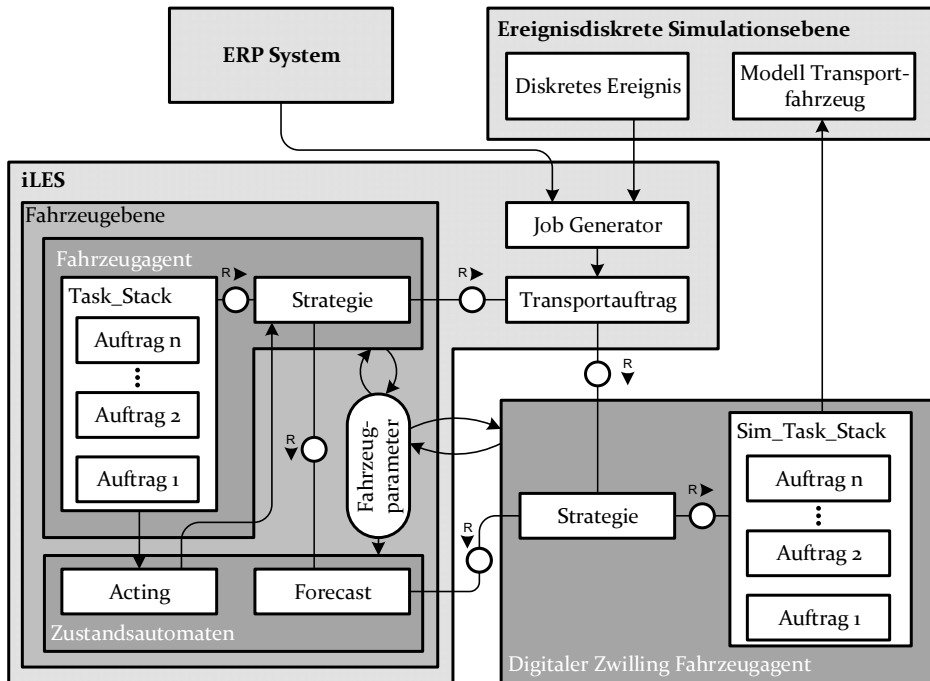


Bild 36: FMC-Block-Diagramm zur Kommunikation zwischen dem iLES, dem ERP und der DES in der Planungsphase

6.3.2 Kontinuierliche Verbesserung des Simulationsmodells durch die Rückkopplung der operativen Entitäten

Die Visualisierung der Architektur in Bild 36 verdeutlicht, dass sowohl der Fahrzeugagent als auch sein für die Simulation verantwortlicher digitaler Zwilling denselben Fahrzeugparameterserver nutzen. Die Optimierung bzw. Korrektur der Fahrzeugparameter erfolgt jedoch ausschließlich durch den realen Fahrzeugagenten. Es findet somit eine Rückkopplung der realen Performanz der Transportentitäten in die Simulationsumgebung statt. Bei der Abarbeitung realer Transportaufträge erzeugen die Entitäten mit den erfassten Daten und Informationen demnach einen digitalen Schatten, anhand dessen das Simulationsmodell und dessen Güte in Bezug auf die Vorhersage des Systemverhaltens permanent verbessert wird.

Der zweite essentielle Vorteil der Architektur ergibt sich aus der Interaktion des realen und simulierten Fahrzeugagenten mit demselben Zustandsautomaten zur Auftragsabschätzung. Gemäß Abschnitt 4.2.2 erfolgt die Abschätzung durch den *Forecast* mit Hilfe eines globalen Pfads, der auf der Adjazenzmatrix des *Routing_Graph*-Dienstes basiert. Dieser bildet nach Abschnitt 5.3 die reale statische Arbeitsumgebung ab und generiert für die jeweiligen Fahrzeugklassen individuelle Fahrwege. Die Abschätzung der Durchführzeiten für Transportaufträge ergibt somit in der Simulationsumgebung die gleichen Ergebnisse wie in der realen Fertigungsumgebung. Zudem werden Änderungen am Layout, Sperrflächen und Hauptverkehrswegen, die die Fahrwege der Transportentitäten beeinflussen, in der Abschätzung und Durchführung von simulierten Transportaufträgen berücksichtigt, ohne dass das Modell manuell angepasst werden muss. Das für die Simulation verwendete Wegstreckennetz entspricht somit stets den aktuellen statischen Gegebenheiten des Arbeitsraums und wird dadurch kontinuierlich aktualisiert.

6.4 Validierung des Simulationsmodells

Ein wichtiger Bestandteil bei der Erstellung von Simulationsmodellen ist deren Validierung. Aufgrund der Gesamtkomplexität der Simulationen von Fertigungsstrukturen und -methoden erfolgt die Validierung meist innerhalb abgesteckter Systemgrenzen in denen die Ergebnisse der Simulation im Vorfeld abgeschätzt werden können und somit ohne einen direkten Vergleich zu einem realen System bewertbar sind. [144]

Auf dieser Basis erfolgt die Validierung des Simulationsmodells mit integriertem Allokationsalgorithmus der iLES anhand eines idealisierten Beispielszenarios. Bewertet wird anschließend, ob sich die Ergebnisse mit den Erwartungen an das Modell decken.

6.4.1 Beschreibung des Simulationsszenarios

Das idealisierte Simulationsszenario ist an das Beispiel der RoboCup-Regularien des Jahres 2018 angelehnt. Das Szenario beschreibt die Montage von Produkten die aus einem Unterteil, bis zu 3 Zwischenringen und einem Deckel bestehen. Die Komplexität der Produkte definiert sich über die Anzahl der benötigten Zwischenringe, die Variantenkomplexität entsteht durch die Kombination der jeweiligen Bauteilvarianten, die sich durch Farben unterscheiden (vgl. Bild 37). Dadurch sind über 240 unterschiedliche Produktvarianten abbildbar. [145]

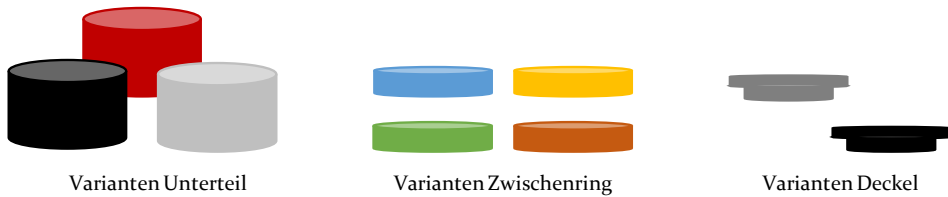


Bild 37: Varianten der einzelnen Bauteile im Beispielszenario (in Anlehnung an [147])

Der Bezug der Unterteile erfolgt an einer zentralen Station und stellt von Fertigungsseite einen entkoppelten Supermarkt zum Vorprozess dar. Die Unterteile sind folglich in jeglicher Variante jederzeit ohne Zeitverzug verfügbar. Die Montage der Zwischenringe benötigt 50 ± 10 sec, wobei eine Anlage zwei unterschiedliche Varianten montieren kann. Zur Abbildung des Produktionsszenarios sind folglich mindestens zwei Montageanlagen für Zwischenringe notwendig. Die Montage der Deckel benötigt 20 ± 5 sec. Im Produktionsszenario sind zwei Deckelmontageanlagen vorgesehen, die jeweils beide Varianten montieren können. Die Materialsenke wird durch eine Ablieferstation repräsentiert, die als Entkopplungspunkt in Form eines Supermarkts ausgeführt ist. [145]

6.4.2 Produktionsszenarien und Bewertung der Simulationsergebnisse

Zur Validierung des Simulationsmodells mit der integrierten iLES ist es zielführend, das beschriebene Produktionsszenario in folgenden Versuchsreihen zu simulieren und gegenüber den zu erwartenden Ergebnissen zu bewerten:

1. Vergleich der mittleren Durchlaufzeit der Produkte im Fertigungsnetzwerk zu einer konventionellen Fertigungsline bei zufälliger Variantenreihenfolge
2. Simulation der über der Simulationszeit maximalen Anzahl von Produkten vor den Anlagen bei unterschiedlichen Auslastungsgraden des Fertigungsnetzwerks

Die zu erwartenden Simulationsergebnisse der einzelnen Versuchsreihen ist wie folgt:

1. Durch die fehlende Austaktung und die nicht eindeutige Prozessfolge wird erwartet, dass das flexible Fertigungsnetzwerk gegenüber einer konventionellen Linienstruktur eine kürzere mittlere Durchlaufzeit der Produkte aufweist.

2. Aufgrund der Warteschlangentheorie wird erwartet, dass mit zunehmendem Auslastungsgrad des Fertigungsnetzwerks die maximale Anzahl der Produkte vor den Anlagen exponentiell ansteigt.

Simulationsergebnisse Versuchsreihe 1

In jeweils zehn Simulationsläufen werden das Fertigungsnetzwerk und die Fertigungslinie mit einem Produktionsplan bestehend aus 50 Produkten beaufschlagt. Um die Produktvarianz hinsichtlich der Zwischenringe in der Fertigungslinie abbilden zu können, ist es notwendig einen Rücklauf der Produkte bei der Montage der Zwischenringe einzufügen. Die mittlere Transportzeit zwischen zwei Anlagen wird im Fertigungsnetzwerk mit 20 sec, bei der Fertigungslinie mit 5 sec angesetzt. Die Maschinenverfügbarkeit jeder Anlage wird mit 100 % simuliert und alle 30 sec ein zufälliger Fertigungsauftrag generiert.

Die Simulation ergibt eine mittlere Durchlaufzeit der Produkte in der konventionellen Linienstruktur von 253 sec, im flexiblen Fertigungsnetzwerk von 182 sec. Damit ist die mittlere Durchlaufzeit im Fertigungsnetzwerk um ca. 28 % und damit signifikant geringer. Dies lässt sich auf die nicht eindeutige Prozessreihenfolge durch die hohe Produktvarianz zurückführen, die eine optimale Positionierung der Montageanlagen der Zwischenringe im Wertstrom verhindert und somit die Vorteile einer prozessfolgeorientierten Organisationsstruktur nicht zum Tragen kommen. Die Simulationsergebnisse decken sich daher vollumfänglich mit der Erwartung.

Simulationsergebnisse Versuchsreihe 2

Das Fertigungsnetzwerk wird in aufeinanderfolgenden Simulationsläufen mit einem variablen Produktionsplan beaufschlagt bis das System eingeschwungen ist. Hierbei wird die mittlere Zeit zwischen zwei Fertigungsaufträgen in den Simulationsläufen variiert um einen unterschiedlichen Auslastungsgrad der Anlagen zu generieren.

Aus den Auswertung der mittleren Anzahl der Produkte im Puffer der Zwischenringmontageanlagen ist die typische Charakteristik der Warteschlangentheorie aus Kapitel 2.2.4 erkennbar (vgl. Bild 38). Unterschreitet der mittlere Abstand von zwei aufeinanderfolgenden Fertigungsaufträgen im Mittel 30 sec, steigt der Pufferbestand vor den Anlagen exponentiell an. Der Auslastungsgrad der Zwischenringmontageanlagen beträgt hierbei ca. 0,8. Da die Zwischenringmontage den Engpass im Produktionsnetzwerk darstellt, ist die Charakteristik der Warteschlangentheorie speziell im Pufferbestand vor diesen Anlagen zu erwarten. Die Ergebnisse der Versuchsreihe 2 decken sich somit ebenfalls mit der Erwartung.

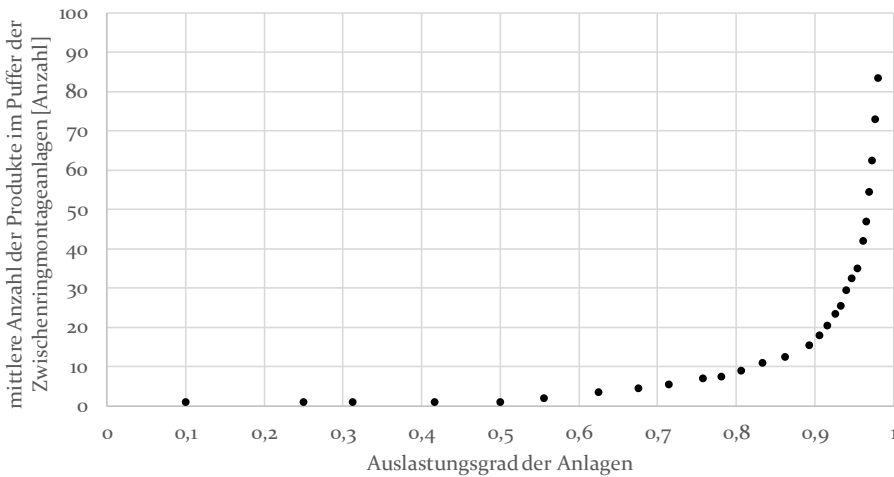


Bild 38: Anzahl der Produkte im Anlagenpuffer über dem Anlagenauslastungsgrad im Simulationsmodell

6.5 Zusammenfassung und Beurteilung

Grundlage der erforschten Methoden zur Verwendung einer DES, um die Vorhersage des Auftragsallokationsverhaltens eines intralogistischen MAS während der Planungs- und Betriebsphase einer flexiblen Fertigungsstruktur zu ermöglichen, stellt der in Abschnitt 6.1 vorgestellte Lösungsansatz dar. Dabei wird der Grundsatz verfolgt, nicht die Systematik der Auftragsdisposition in der Simulation abzubilden, sondern einen Prototyp des iLES als SiL direkt in die Simulation einzubinden. Gegenüber den etablierten Ansätzen zeichnet sich dieser Lösungsansatz vor allem dadurch aus, dass das iLES als zentrale und elementare Komponente zur Disposition der Transportaufträge bereits in der Planungsphase der flexiblen Fertigungsstruktur berücksichtigt wird und hinsichtlich der übergeordneten Zielstellungen optimiert werden kann. Zudem wird eine parallele Weiterverwendung des Simulationsmodells während der operativen Phase ermöglicht. Weiterhin entfällt beim Transfer des Quellcodes der simulierten Fahrzeugagenten auf die im operativen Betrieb verwendeten Transportentitäten der Migrationsaufwand, da die Softwareumgebung der Agenten unverändert bleibt.

Diesen Lösungsansatz aufgreifend wird in Abschnitt 6.2 ein Vorgehensmodell zur Einbindung des Allokationsalgorithmus in die DES und zur Optimierung des Systemverhaltens in der Planungsphase erörtert. Dabei wird

ein Prototyp des iLES mit einem Simulationsmodell der Fertigung gekoppelt. Die in der Simulation entstehenden diskreten Fertigungsaufträge erzeugen analog zum Fertigungsmanagementsystem im realen Betrieb auf Seiten des iLES-Prototyps Transportauftragsagenten. Dadurch wird der Algorithmus zur Verhandlung innerhalb des iLES angestoßen und lediglich das Ergebnis der Disposition an die Simulation zurückgemeldet. Da sich die Eingangsinformationen des iLES während der Simulationsphase zum operativen Betrieb nicht unterscheiden, ist auch ein identisches Verhalten des Systems auf die diskrete Generierung von Transportaufträgen gewährleistet. Die Reaktion des Gesamtsystems auf spezifische Konstellationen wird dadurch abschätzbar.

Den innovativen Kern des Lösungsansatzes stellt die in Abschnitt 6.3 beschriebene Methode zur Überführung der in der Simulation optimierten Agenten in den operativen Betrieb und die parallele Weiterverwendung des Simulationsmodells dar. Hierzu wird auf der Fahrzeugseite neben dem Fahrzeugagenten ein digitaler Zwilling für Simulationszwecke integriert. Dieser greift auf die gleichen Daten und Algorithmen zur Auftragsabschätzung zu wie sein für die physische Entität verantwortliches Pendant. Die von der Simulationsumgebung stammenden Transportaufträge werden lediglich mit einem Parameter versehen, dass der Auftrag nach der Disposition nicht durchgeführt, sondern die abgeschätzten Auftragsdaten an die Simulation zurückgegeben werden. Durch kontinuierliche Aktualisierung der Fahrzeugparameter durch den Fahrzeugagenten sowie der Verwendung des realen *Routing_Graph*-Dienstes ist sichergestellt, dass die Rückgabewerte an die Simulation dem realen Systemverhalten entsprechen. Zu den etablierten Ansätzen ergibt sich daher ein signifikant geringerer Aufwand, das Simulationsmodell mit dem realen Fertigungslayout fortlaufend zu synchronisieren. Der Hauptnutzen des parallelen Betriebs der Simulation zum realen Fertigungsbetrieb stellt die Vorhersagbarkeit des Systemverhaltens auf Basis von Simulationsergebnissen dar. Dadurch wird die größte Restriktion des Einsatzes von MAS in Fertigungsbetrieben, der fehlende Determinismus des Gesamtsystems, umgangen.

Zusammenfassend stützt die Validierung des Simulationsmodells in Abschnitt 6.4 die Funktion und Wirksamkeit der erforschten Methoden und die Einbindung des Allokationsalgorithmus der iLES in die Simulationsumgebung.

7 Beispielhafte Umsetzung und Validierung des Systemansatzes

Die vorangegangenen Kapitel beschreiben eine Systemarchitektur für autonome intralogistische Transportentitäten, die Methoden zur Auftragsallokation (Kapitel 4), zur servicebasierten Routenplanung (Kapitel 5) und zur Kopplung an eine DES (Kapitel 6) adressiert.

Tabelle 7: Anforderungen und Bewertungskriterien zur Evaluation der implementierten Methoden

Anforderung an Methode	Bewertungskriterium
Universalität der Software-schichten (Fahrzeugagent, Zustandsautomaten, Parameterserver)	Nachweis der Funktionsfähigkeit auf unterschiedlichen Transport-entitäten: <ul style="list-style-type: none"> • Fahrantrieb mit Mecanumräder • (Quasi-)omnidirektionaler Fahrantrieb • Differenzialantrieb • Greifersystem • Aktives Lastmittel • Hubplattform • Passives Lastmittel
Funktionalität der Auftrags-allokation im MAS	<ol style="list-style-type: none"> 1. Umsetzbarkeit des MAS in einem Softwareframework und Einbindung der Fahrzeugfähigkeiten in zentrale Agenten 2. Erfolgreiche Auftragsverhandlung und -vergabe anhand der Auftragsziele und Fahrzeugfähigkeiten
Skalierbarkeit der verteilten Servicearchitektur	Einbindung in ein Systemkonzept mit verteilten dezentralen Infrastruktursensoren (Deckenkameras)
Nachweis der Echtzeitfähigkeit bei der Detektion von dynamischen Objekten	Laufzeitmessung der Algorithmen an den dezentralen Infrastruktur-sensoren
Nachweis einer validen Arbeitsraumdigitalisierung	Binärbilderzeugung eines industrienahen Szenarios und Bewertung der Detailerfassung
Nachweis der fahrzeugtypspezifischen Wegstreckennetzbildung	Berechnung fahrzeugtypspezifischer Netze im industrienahen Szenario und Bewertung der Anzahl der Netzkpunkte unterschiedlicher Fahrzeugklassen
Integrierbarkeit des iLES in eine DES	<ol style="list-style-type: none"> 1. Analyse und Bewertung der Kommunikation zwischen dem iLES und der Simulation auf Basis von industriellen Kommunikationsstandards 2. Bewertung der Simulationsergebnisse bei Variation der <ul style="list-style-type: none"> • Auftragsarten • Fahrzeugparameter

Um die Funktionsfähigkeit der Einzelkomponenten und die Leistungsfähigkeit des Gesamtsystems quantitativ zu evaluieren, wird im Folgenden die Implementierung der Methoden in ausgewählten Softwaretools und

mithilfe geeigneter Frameworks vorgestellt (Abschnitt 7.1) und anhand dieser die Forschungsergebnisse validiert. Abschnitt 7.2 fokussiert die Applikation des generischen Schichtenmodells auf unterschiedliche Transportplattformen und die Auftragsdisposition innerhalb des MAS. Die Digitalisierung einer industrienahen Arbeitsumgebung zur Generierung eines plattformübergreifenden fahrzeugtypspezifischen Routenplans und der Bereitstellung von lokalen Umgebungsinformationen zur Pfadausführung wird in Abschnitt 7.3 diskutiert. Die Umsetzung des Informationsaustauschs zwischen den realen Entitäten und der Simulation beschreibt Abschnitt 7.4. Die Anforderungen an die implementierten Methoden werden anhand spezifischer Kriterien gemäß Tabelle 7 evaluiert.

Einzelne Aspekte der Ausführungen der Folgeabschnitte 7.1 bis 7.4 sind an die am Lehrstuhl FAPS entstandenen und vom Autor angewiesenen studentischen Arbeiten [S61–S82] angelehnt.

7.1 Beschreibung der prototypischen Gesamtarchitektur

Die prototypische Gesamtarchitektur gliedert sich in sechs funktionale Teilbereiche, in denen vier Softwareframeworks für die spezifischen Teilfunktionen eingesetzt werden. Bild 39 zeigt den aggregierten Aufbau, der zur Validierung und Evaluierung der Methoden dient.

MSL

Die Basis stellt die intralogistische Serviceebene dar, die die elementaren Dienste zur Digitalisierung der Arbeitsumgebung mit Infrastruktursensoren beinhaltet. Die Implementierung aller Funktionen der Serviceebene erfolgen im Softwareframework ROS (*Robot Operating System*) in der Version Kinetic Kame. Die Vorteile des ROS vor allem bei robotischen Anwendungen beruhen darauf, dass eine Vielzahl an Programmbibliotheken und Treibern zur Anbindung von Sensoren und Aktoren zur Verfügung gestellt werden. Zudem ist der Nachrichtenaustausch zwischen Programmen bzw. Programmteilen standardisiert. Das Framework ist weiterhin nach BSD (*Berkeley Software Distribution*) lizenziert und zählt somit zur Gruppe der Open-Source-Lizenzen [146]. Jedoch ist im Gegensatz zur GPL (*GNU General Public License*) kein Copyleft enthalten, wodurch die Herstellung kommerzieller teilproprietärer Produkte ermöglicht ist [147]. Die Dienste der intralogistischen Serviceebene sind als eigenständige ROS-Packages ausgeführt und stellen somit eigenständige Programme innerhalb der Serviceebene dar.

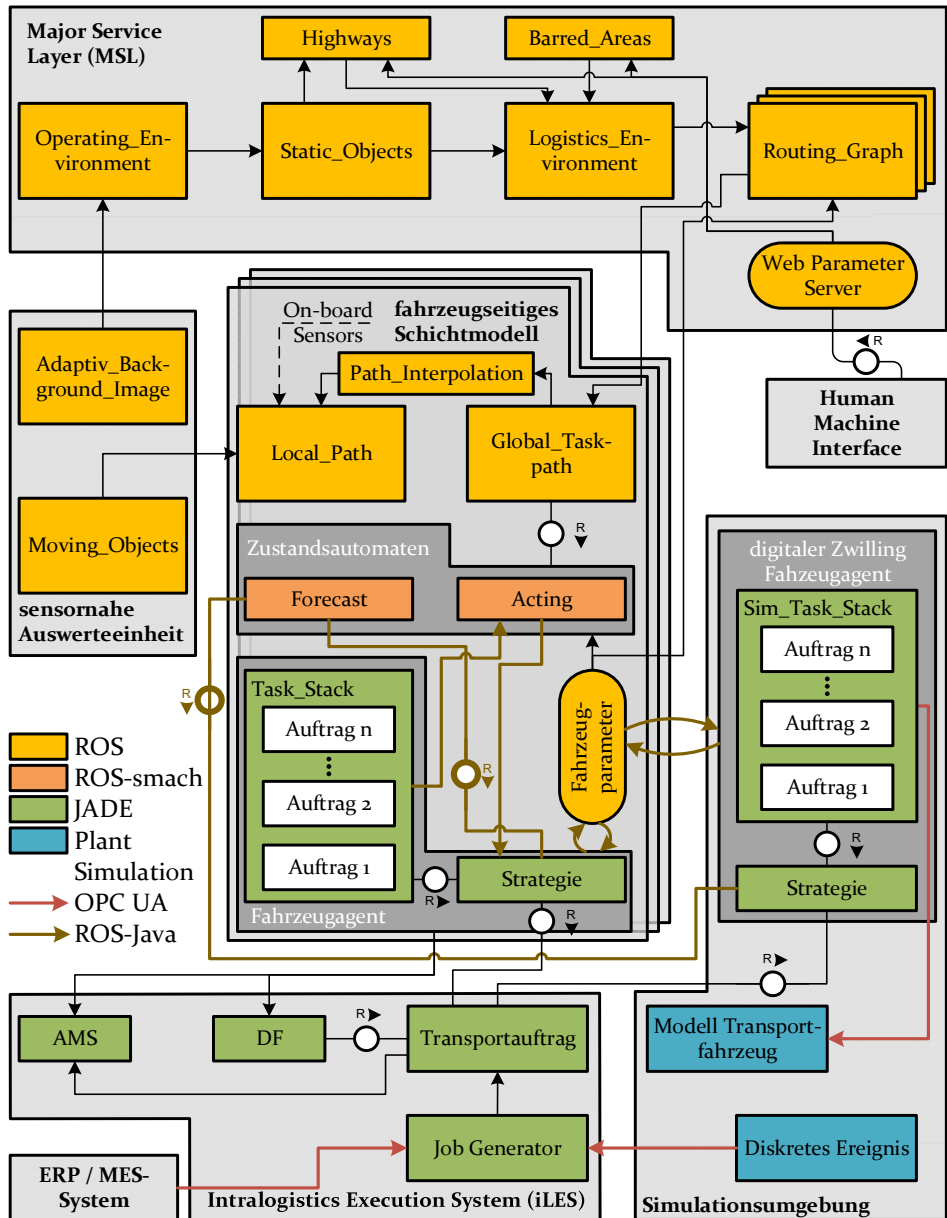


Bild 39: FMC-Block-Diagramm der Gesamtarchitektur zur prototypischen Realisierung einer autonomen und servicebasierten Auftragsallokation und durchführung flexibler intralogistischer Transportsysteme

Dies ermöglicht einzelne Funktionalitäten unabhängig voneinander zu testen, zu optimieren oder zu substituieren, ohne dass sich die Gesamtfunktionalität ändert. Der bedarfsorientierte Informationsaustausch zwischen

den einzelnen Programmteilen der Serviceebene erfolgt über *ROS_Services*, bei denen die Interprozesskommunikation nach dem RPC-Modell (*Remote Procedure Call*) realisiert ist.

Sensornahe Auswerteeinheit

Die Digitalisierung der Arbeitsumgebung erfolgt durch die sensornahen Auswerteeinheiten ebenfalls unter Verwendung des ROS-Frameworks. Der Informationsaustausch zwischen der Serviceebene und den einzelnen Sensoren als verteilte Anwendung ist durch die Funktionalität des Frameworks gewährleistet. Der *ROS_Master* der Serviceebene stellt dabei den Dienst zur Benennungskonvention und Registrierung der einzelnen Programmteile dar. Die Vergabe eines eindeutigen Namensraums für die sensornahen Auswerteeinheiten ermöglicht es, dass die Quellcodes der Programme auf den Auswerteeinheiten identisch sind. Bei der Implementierung ist somit lediglich der Namensraum, der sich idealerweise an der fortlaufenden ID der Kameras orientiert, anzupassen. Zur Bildverarbeitung wird die ebenfalls BSD-lizenzierte Programmbibliothek OpenCV in das ROS-Framework eingebunden [148].

iLES

Die übergeordnete Funktion der Auftragsgenerierung und -allokation, das iLES, ist in der quelloffenen Middleware JADE (*Java Agent Development Framework*) implementiert. JADE ist speziell für den agentenbasierten Betrieb von *Peer-to-Peer* Anwendungen konzipiert und folgt dem Standard der FIPA [149]. Die Middleware ist nach LGPL (*Lesser General Public License*) lizenziert und ermöglicht somit ebenfalls die Entwicklung von proprietärer Software [150]. Jedoch fordert die Lizenz, dass die verwendeten Teile des Quellcodes, die der Lizenz unterliegen, veröffentlicht werden. Daher ist eine Trennung zwischen dem proprietären und dem quelloffenen Teil der entwickelten Software z. B. durch dynamische Programmbibliotheken wie DLL (*Dynamic Link Library*) bei einer kommerziellen Umsetzung notwendig [151]. Ein JADE-Agent entspricht einer Java-Klasse und erhält durch Vererbung die elementaren Funktionen wie die Registrierung und Methoden zum Senden und Empfangen von Nachrichten. Die Implementierung der einzelnen Agenten ist dabei nicht an physische Grenzen gebunden und kann zur Laufzeit auf andere Recheneinheiten verschoben werden, was die Modularität und Skalierbarkeit des Systems erhöht. Hinsichtlich der Optimierung der im Netzwerk ausgetauschten Datenmenge sind die Fahrzeugagenten daher auf den jeweiligen Transportentitäten implementiert. [152]

Uniforme transportplattformunabhängige Softwareschicht

Die zentrale Komponente der Gesamtarchitektur stellen die Transportentitäten selbst und das implementierte Softwareschichtenmodell dar. Die elementaren Fähigkeiten der Fahrzeuge sind als ROS-Pakete implementiert. Die einzelnen Pakete selbst sind auf jeder Fahrzeugentität identisch und werden in der Gesamtarchitektur lediglich durch die Verwendung eines Namensraums, der an die Fahrzeug-ID gekoppelt ist, unterschieden. Im Gegensatz zu den sensornahen Auswerteeinheiten verfügen die Fahrzeuge über einen eigenen *ROS_Master*. Dadurch wird sichergestellt, dass jede Transportentität selbst eine abgeschlossene Programmeinheit darstellt und ohne die Anbindung an die Serviceebene funktionsfähig ist. Einerseits kann dadurch eine alternative Gesamtarchitektur auf Basis der Forschungsarbeiten umgesetzt werden, andererseits ist die Funktionsfähigkeit der Transportroboter auch ohne eine Verbindung zu den Basisdiensten der MSL gewährleistet. Der aktuelle Arbeitsauftrag kann somit auch bei einem Abriss des Informationskanals, z. B. aufgrund schlechter Funkverbindung, abgeschlossen werden, respektive die Positionierungsfahrt nach Abschluss eines Auftrags autonom initialisiert werden. Um den Informationsaustausch innerhalb einer Architektur zu gewährleisten, hält das Framework das Metapaket *Multimaster* bereit. Der Fahrzeugparameterserver ist als *ROS_Parameterserver* implementiert. Damit können alle Programme innerhalb des Namensraums, und damit lediglich die Programme der jeweiligen Entität, zur Laufzeit auf die gespeicherten Informationen zugreifen und diese ändern. Wie bereits im Abschnitt 5.2.3 erläutert, ist die Übermittlung der Objektdaten der bewegten Hindernisse zwischen den Infrastruktursensoren und den Fahrzeugen als Broadcast auszuführen, um eine angemessene Reaktionsfähigkeit zu gewährleisten. Die Implementierung innerhalb des Frameworks erfolgt über *ROS_Topics*, die nach dem *Publish-Subscribe* Modell arbeiten. Die sensornahen Auswerteeinheiten veröffentlichen folglich permanent die Objektinformationen der bewegten Hindernisse über ein eindeutiges Thema, das die Fahrzeuge nach Bedarf abonnieren. Die eindeutige Kennung des Themas erfolgt über die ID des zugehörigen Infrastruktursensors. [153, 154]

Das Bindeglied zwischen den ROS-Paketen und dem Fahrzeugagenten stellen die Zustandsautomaten dar, die in *ROS-smach* realisiert sind. *ROS-smach* bietet eine Reihe von Klassen und Containern zum Beschreiben von Zustandsautomaten und die Umsetzung eines Fehlercodehandlings. Die Kommunikation zwischen den Zustandsautomaten *Forecast* und *Acting* und den Programmen der Basisfähigkeiten erfolgt nach dem RPC-Modell und damit reaktiv. Der Bruch zwischen den Frameworks ROS und JADE

verläuft zwischen dem Fahrzeugagenten und den Zustandsautomaten und damit innerhalb der physischen Transportentität. Den Informationsaustausch beider Systeme ermöglicht das ROS-seitig implementierte Plug-In *ROS-Java*, das die Java-Nachrichten aus JADE in ein ROS-konformes Datenformat übersetzt.

Nach außen bietet die uniforme Softwareschicht folglich zwei eindeutige Schnittstellen, bei der die eine die Grundfähigkeit eines mobilen Transportroboters, im Arbeitsraum navigieren zu können, adressiert. Die JADE-Schnittstelle ermöglicht dem Fahrzeug hingegen an Transportauftragsverhandlungen teilzunehmen. Da JADE zudem FIPA-konform ist, ist dabei irrelevant, ob die übrigen Agenten des iLES ebenfalls in JADE oder einer alternativen MAS-Middleware implementiert sind, da die Nachrichtenstruktur zwischen den Agenten genormt ist. Ein operativer Einsatz der Transportroboter ist somit auch in einer Umgebung gewährleistet, die von der erforschten Gesamtarchitektur abweicht.

Simulationsmodell der Arbeitsumgebung

Parallel zum übergeordneten Fertigungsmanagementsystem können durch das digitale Abbild der Fertigung simulierte Transportaufträge im iLES erzeugt werden. Als Simulationsumgebung dient die objektorientierte Materialflusssimulationssoftware Plant Simulation. Der Datenaustausch zwischen Plant Simulation und den JADE-Agenten erfolgt über OPC UA. [155]

Digitaler Zwilling des Fahrzeugagenten

Um das identische Verhalten jeder Transportentität innerhalb der Simulation und der realen Fertigung zu realisieren, wird innerhalb der Gesamtarchitektur ein digitaler Zwilling jedes Fahrzeugagenten integriert. Dieser greift, wie das reale Pendant, auf den selben *ROS-Parameterserver* und den *Forecast* zu. Um den Informationstransport innerhalb des Netzwerks zu minimieren, ist der digitale Zwilling ebenfalls auf der Fahrzeugebene implementiert.

HMI

Abschließend bietet ein integriertes HMI die Möglichkeit, die Informationen innerhalb der Serviceebene für den Bediener sichtbar zu machen. Zudem erfolgt hierüber die Manipulation der Sperrflächen und der Hauptverkehrswege. Auf Seite der Serviceebene wird hierzu ein Parameterserver im *Rosbridge Protocol* eingerichtet, der die Informationen der jeweiligen Dienste verwaltet. Im Frontend dient ein *WebSocket Server* als Interaktionsmöglichkeit für unterschiedliche Endgeräte. Somit ist die Konfiguration

und die Visualisierung der Arbeitsumgebung nicht auf einen spezifischen Arbeitsplatz beschränkt, sondern im gesamten Netzwerk nutzbar.

7.2 Implementierung des generischen Fahrzeugschichtenmodells auf Zielplattformen

Die Leistungsfähigkeit der uniformen transportplattformunabhängigen Softwareschicht lässt sich durch die Integration auf unterschiedliche intra-logistische Transportentitäten unter Beweis stellen. Die durchgeführten Forschungsarbeiten zeigen, dass Transportplattformen für differenzierte Anwendungsbereiche mit dem entwickelten Softwareschichtenmodell betrieben werden können (vgl. Bild 40).



Bild 40: Transportroboter mit der implementierten uniformen Softwareschicht zum prototypischen Funktionstest; 1: autoBod, 2: FAPS-VAV, 3: Robotino (Applikation Elektronikproduktion), 4: Robotino (Applikation RoboCup)

autoBod (ActiveShuttle)

Der autoBod stellt einen, in Forschungskooperation mit der Robert Bosch GmbH konzipierten Prototyp eines Transportroboters für KLT dar. Ziel ist es, standardisierte Rollwagen, auf denen sich gestapelte KLT befinden, im Taxibetrieb zwischen einer Quelle und einer Materialsenke zu transportieren. Die Ladefläche beträgt $60 \times 22 \text{ cm}^2$ und hat eine maximale Traglast von 200 kg. Das Lastaufnahmemittel verfügt über einen Hubmechanismus zur

Aufnahme der Bodenroller. Sowohl der Hubmechanismus als auch der Differentialantrieb wird durch Maxon EC 60 Flat Motoren realisiert. Zur Detektion des roboternahen Umfelds und der Einhaltung bestehender industrieller Sicherheitsbestimmungen verfügt der autoBod über den Sicherheitslaserscanner OS32C von Omron. Die zentrale Recheneinheit des Systems ist ein Pokini i2 von Exone mit einem Intel Core i5-4300U Dual Core Prozessor, einer Taktrate von 2,9 GHz, 16 GB Arbeitsspeicher und einer Intel HD 4400 Graphics Grafikkarte. Die Ansteuerung der Antriebsmotoren und des Hubmotors erfolgt durch die Recheneinheit über einen CAN-Bus (*Controller Area Network*).

FAPS-VAV

Das FAPS-VAV stellt einen, im Rahmen der Forschungsarbeiten entwickelten, Funktionsdemonstrator eines Kleinstransportroboters dar, der aufgrund seiner geringen Eigenmasse und Zuladung von 10 kg inhärent sicher ist. Das passive Lastaufnahmemittel ist für Werkstückträger mit einer maximalen Kantenlänge von 30 cm ausgelegt. Der Differentialantrieb des quasiomnidirektionalen Fahrwerks ist mit den Gleichstrommotoren DSMP360-24-14 von Drive-System Europe angetrieben. Zur Digitalisierung des Nahbereichs verfügt der Roboter über den Laserscanner RPLIDAR A2 von Slamtec. Zusätzlich überwachen Ultraschallsensoren den unmittelbaren Frontbereich des Fahrzeugs. Als zentrale Recheneinheit fungiert ein Odroid-XU4 von Hardkernel. Die Umwandlung der Fahrbefehle erfolgt über den Mikrocontroller Arduino Mega 2560, die Ansteuerung der Motoren über ein Dual VNH5019 Motor Driver Shield von Pololu.

Robotino (Applikation Elektronikproduktion)

Zur flexiblen Vernetzung einer Elektronikproduktion wurde während der Forschungsarbeiten ein Transportroboter konzipiert. Die Basis des Systems ist ein Robotino von Festo Didactic mit einem omnidirektionalen Fahrwerk. Die Ansteuerung der Motorregelung übernimmt ein integrierter 32-Bit Mikrocontroller. Die zentrale Recheneinheit ist ein Intel Core i5 mit 2,4 GHz Taktung und 8 GB RAM. Zur Digitalisierung des Nahfelds ist die Plattform mit dem Laserscanner RPLIDAR A2 von Slamtec ausgestattet. Das applizierte Lastaufnahmemittel ist speziell für den Einsatz in einer flexiblen Elektronikproduktion zum Transport von einzelnen Leiterkarten konstruiert. Es ermöglicht einen Höhenausgleich des Lastmittels zur Anlage von 30 mm und eine Breitenanpassung des Förderbands auf Leiterkarten von 29 bis 418 mm. Die Lastaufnahme und -übergabe erfolgt aktiv. Die Ansteuerung des Lastaufnahmемoduls erfolgt über einen Mikrocontroller Arduino Mega 2560 mit Schrittmotormodul.

Robotino (Applikation RoboCup)

Die vierte Transportplattform zur Validierung des uniformen Software-schichtenmodells basiert ebenfalls auf dem Fahrmodul Robotino von Festo Didactic in der gleichen Konfiguration wie der vorherige Versuchsträger. Der Unterschied liegt im Lastaufnahmemittel, das für den Einsatz der Plattform in der Logistics League des RoboCup ausgelegt ist. In diesem industrienahen Szenario werden zylindrische Bauteile mit einem Durchmesser von 40 mm von Transportrobotern anhand eines individuellen Arbeitsplans zwischen unterschiedlichen Montagestationen transportiert. Das Lastaufnahmemodul verfügt über einen anwendungsspezifischen Greifer zur Aufnahme der Bauteile. Die Ansteuerung des Lastaufnahmemoduls erfolgt über ein eigenentwickeltes Board mit Mikrocontrollersteuerung.

7.2.1 Umsetzung des typspezifischen Parameterservers in Kombination mit einer allgemeingültigen Fähigkeitsbeschreibung

Bei der Auswahl der erforschten und konzipierten Plattformen ist eine Varianz (quasi-)omnidirektionaler Fährantriebe mit passiven und aktiven Lastaufnahmemitteln abgebildet. Hierdurch ist gewährleistet, dass die Aggregation der Basisfähigkeiten innerhalb der Zustandsautomaten *Acting* und *Forecast* unter Zuhilfenahme der Fahrzeugkonfiguration im Parameterserver bei unterschiedlichen Zielplattformen durchführbar ist.

Parameterserver

Der implementierte Parameterserver gliedert sich in die drei Bereiche mit unterschiedlichen Zugriffsrechten für den Fahrzeugagenten und den Zustandsautomaten:

- Systemdefinition (lesend): Es erfolgt die Auswahl der Fahrwerksart sowie die Dimensionierung des Gesamtsystems. Weiterhin werden die Sicherheits- und Bahnplanungsparameter festgelegt und die Kommunikation zwischen dem Fährantrieb und des zentralen Fahrzeugrechners ausgewählt.
- Auftragsberechnung (lesend/schreibend): Die Berechnungen und Abschätzung von Transportaufträgen erfolgt auf Basis der definierten Parameter und fahrzeugspezifischen Kostenfaktoren.
- Lastaufnahme (lesend): Die Definition des Lastaufnahmemittels erfolgt anhand der eingesetzten Bauart, sowie der Positionierung und Ansteuerung des Systems.

Die Zugriffsrechte der Zustandsautomaten und des Fahrzeugagenten sind überwiegend lesend, lediglich die Parameter zur Auftragsberechnung und die Schätzfaktoren zum Handling werden kontinuierlich optimiert.

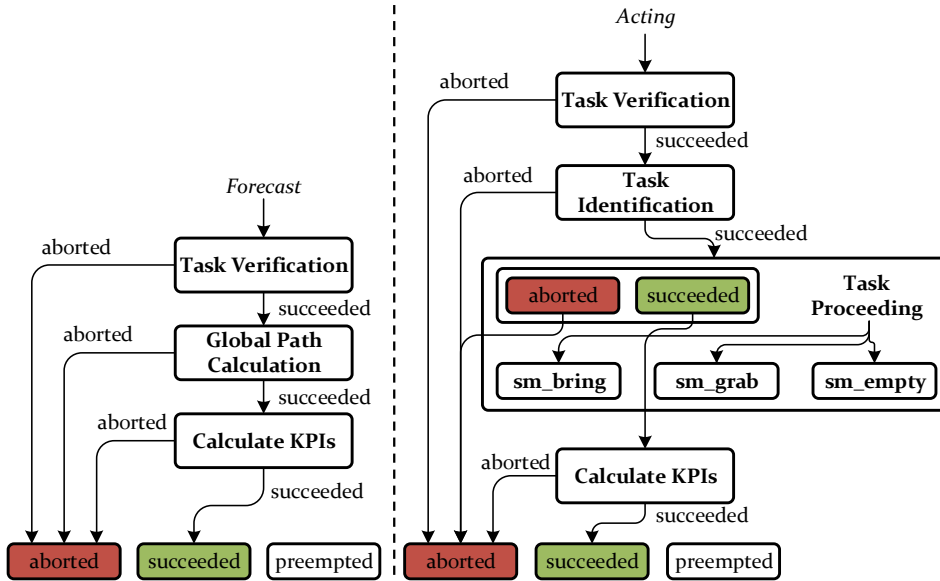


Bild 41: Umsetzung der generischen Zustandsautomaten Forecast und Acting in ROS-smach

Zustandsautomaten

In Bild 41 sind die in *ROS-smach* implementierten Zustandsautomaten *Forecast* und *Acting* visualisiert, die für sämtliche Versuchsplattformen identisch sind. Die einheitliche Gestaltung ist möglich, da die plattformspezifischen Parameter zur Berechnung der Route und der Durchführung des Pfads von den entsprechenden Basisfähigkeiten aus dem Parameterserver gezogen werden. Die Lastaufnahme und -abgabe hingegen sind in spezifischen Routinen implementiert, die nach der Feinpositionierung aufgerufen werden. Diese Routinen sind auf die jeweilige Konfiguration des Lastaufnahmемittels abgestimmt und nicht Teil des allgemeingültigen Zustandsautomaten. Ebenso erfolgt die Umwandlung der Fahrbefehle des lokalen Pfadplaners in die Drehraten der Antriebsmotoren im Motorsteuermodul. Der Pfadplaner gibt hierzu lediglich die Trajektorie des Fahrzeugdrehpunktes vor. Die hinterlegten Daten im Parameterserver definieren den Kommunikationskanal zu den Steuermodulen, die anhand der Fahrzeugdimensionen die Drehrate der einzelnen Motoren berechnen und diese ansteuern. Im erforschten Fahrzeugschichtenmodell sind hierzu eine USB-

Schnittstelle und die Kommunikation innerhalb eines CAN-Bus umgesetzt, mit der sämtliche Versuchsträger angesteuert werden können.

7.2.2 Analyse der Kommunikation zwischen Zustandsautomaten und Fahrzeugagent

Die in *ROS-smach* implementierten allgemeingültigen Zustandsautomaten *Forecast* und *Acting* stellen die Trennung zwischen dem Planungshorizont eines Fahrauftrags und der strategisch langfristigen Planung der Agenten dar. Dadurch entsteht bei der Realisierung zwischen beiden Schichten eine Schnittstelle der verwendeten Frameworks JADE und ROS und damit der Programmiersprachen. Diese Schnittstelle wird auf Seiten der Zustandsautomaten durch das Plug-In *ROS-Java* geschlossen, das die Zustandsautomaten dazu befähigt, Java-Nachrichten zu interpretieren und auszuführen.

Die Antwort der Zustandsautomaten enthält zudem die Übermittlung der Fehlercodes und Checkpoints bei der Ausführung der Hauptfähigkeiten, anhand derer der Agent die in Abschnitt 4.3.3 aufgeführten Verhaltensweisen ausführt. Die Realisierung dieser Strategie zeigt der in Bild 42 dargestellte Auszug des Zustandsautomaten der orthogonalen Region *sm_empty*.

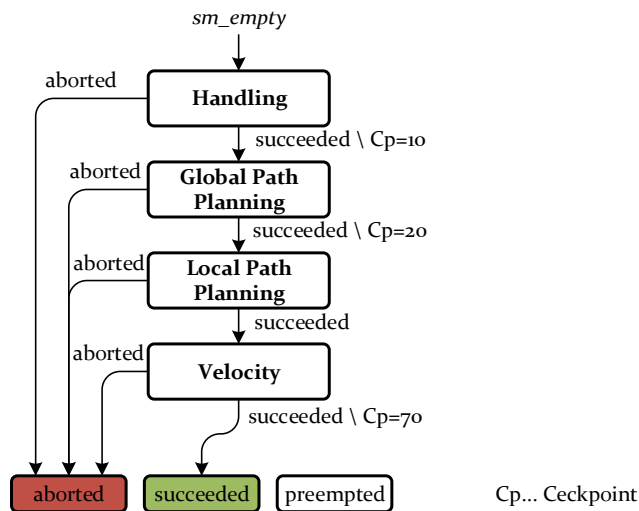


Bild 42: Umsetzung der Checkpoints in ROS-smach am Beispiel der Leerfahrt

Es ist zu entnehmen, dass bei der erfolgreichen Abarbeitung eines Unterzustands neben dem *succeeded* zudem der Checkpoint übergeben wird. Beim Eintritt in den Unterzustand überprüft dieser den übergebenen Checkpoint und entscheidet anhand dessen, ob eine Ausführung notwendig ist, oder der Zustand aufgrund des übergebenen Checkpoints bereits

erfüllt ist. Dadurch kann, gemäß der erläuterten Strategie in Abschnitt 4.3.3, bei einer wiederholten Ausführung des Fahrauftrags vom Fahrzeugagenten ein definierter Startpunkt im Programmablauf vorgegeben werden.

7.2.3 Analyse der Auftragsdisposition im Multi-Agenten-System

Die übergeordnete Funktion der Agentenkommunikation innerhalb des iLES ist die Disposition von Fahraufträgen. Der gesamte Informationsaustausch zwischen den Agenten sowie die Durchführung der Auftragsverhandlung nach dem *Contract Net Protocol* ist im Rahmen der prototypischen Umsetzung im Framework JADE realisiert. Das iLES stellt hierbei eine verteilte Anwendung dar, da die Erzeugung der Transportaufträge, der AMS und der DF auf einem zentralen Rechner umgesetzt ist, die Fahrzeugagenten jedoch auf den Transportentitäten implementiert sind.

Eintragung der Fahrzeugagenten in DF

Nach dem Start der zentralen Recheneinheit trägt der Fahrzeugagent die Fähigkeiten des Fahrzeugs gemäß den Angaben im Parameterserver in den DF auf dem zentralen Rechner ein.

1	Robot: rob1@172.26.113.11:1099/JADE	1	Robot: rob2@172.26.113.21:1099/JADE
2	Reading parameters from server ...	2	Reading parameters from server ...
3	MaxWeigt: 10.0kg	3	MaxWeigt: 200.0kg
4	AvgTransferSpeed: 1.2m/sec	4	AvgTransferSpeed: 1.6m/sec
5	AvgTransportSpeed: 1.0m/sec	5	AvgTransportSpeed: 1.1m/sec
6	AvgTimeHandleSource: 7.0sec	6	AvgTimeHandleSource: 10.0sec
7	AvgTimeHandleTarget: 5.0sec	7	AvgTimeHandleTarget: 7.0sec
8	CostsPerDist: 0.01Euro/m	8	CostsPerDist: 0.1Euro/m
9	CostsPerMass: 0.001Euro/(kg*m)	9	CostsPerMass: 0.05Euro/(kg*m)
10	CostsPerTime: 0.0Euro/sec	10	CostsPerTime: 0.00005Euro/sec
11	... done	11	... done
12	Publication of following information:	12	Publication of following information:
13	robot_type = transporter	13	robot_type = transporter
14	max_weight_greater_1000kg = no	14	max_weight_greater_1000kg = no
15	max_weight_smaller_1000kg = no	15	max_weight_smaller_1000kg = no
16	max_weight_smaller_500kg = no	16	max_weight_smaller_500kg = yes
17	max_weight_smaller_100kg = no	17	max_weight_smaller_100kg = yes
18	max_weight_smaller_50kg = no	18	max_weight_smaller_50kg = yes
19	max_weight_smaller_10kg = yes	19	max_weight_smaller_10kg = yes
20	max_weight_smaller_5kg = yes	20	max_weight_smaller_5kg = yes
21	max_weight_smaller_1kg = yes	21	max_weight_smaller_1kg = yes
22	Registering the services in DF ...	22	Registering the services in DF ...
23	... finished	23	... finished

Bild 43: Ausführungsprotokoll der Registrierung von zwei Fahrzeugagenten am DF; links: FAPS-VAV, rechts: autoBod

Im prototypischen Szenario wird der Übersichtlichkeit halber lediglich eine Fähigkeitsunterscheidung in Bezug auf die maximale Transportmasse umgesetzt. Eine Eigenheit des verwendeten Frameworks ist, dass die Abfrage der Fahrzeugeigenschaften im DF durch die Transportauftragsagenten nur über Gleichheitsbeziehungen erfolgen kann. Daher ist die Definition von Gewichtsklassen notwendig, in die sich die Transportentitäten bei der Anmeldung und Übermittlung der Fahrzeugparameter eintragen.

Angebotsanfrage und -vergabe

Vor der Angebotsanfrage bezieht der Transportauftragsagent vom DF die IDs der Transportentitäten, die für die Durchführung des Auftrags generell in Frage kommen. Hierzu wird anhand des Auftragsgewichts die notwendige Gewichtsklasse festgelegt und die Fahrzeuge mit der entsprechenden Zuladung ermittelt.

TransportAgent Auftragsl@localhost:1099/JADE EvaluationType: Cheapest Limit Costs: 500,00€ Limit Duration: 60s Limit EndTimestamp: 1539613262059 MaterialWeight: 0.5kg	Definition der Auftragsparameter
Required: max_weight_smaller_1kg == yes Found 2 robots: - VAV2 - VAV1 Searching done Calling for proposals ... Done. Called 2 robots	Suche der geeigneten Transportroboter im DF
Received valid proposal from robot: VAV1 Received valid proposal from robot: VAV2 Stopped listening for proposals. Received 2 valid proposals Selecting best proposal for evaluation type 'Cheapest': - VAV1 Duration (propose): 25sec End Timestamp (propose): 1539613193307 Costs (propose): 70,61€ - VAV2 Duration (propose): 29sec End Timestamp (propose): 1539613196751 Costs (propose): 74,05€	Rückmeldung der Transportentitäten mit geschätzter Transportdauer, -beginn und -kosten
Best proposal: VAV1 Duration (propose): 25sec End Timestamp (propose): 1539613193307 Costs (propose): 70,61€	Auswahl des besten Angebots
Replying proposals - Accept Proposal of VAV1 - Reject Proposal of VAV2 Waiting for VAV1 to confirm proposal ... Received proposal confirmation	Vergabe des Auftrags und Absage an übrige Anbieter
Waiting for VAV1 to confirm execution ... Received transport execution confirmation Robot finished transport successfully Proposal vs. Execution Duration (real / propose): 20sec / 25sec (-5sec) End Timestamp (real / propose): 1539613185845 / 1539613193307 (-7sec) Costs (real / propose): 69,37€ / 70,61€ (-1,23€)	Rückmeldung nach Durchführung und Bewertung gegenüber Schätzung
Conversation ID: Auftragsl finished	Beendigung des Auftrags

Bild 44: Angebotsvergabe und -durchführung aus Sicht des Transportauftragsagenten in JADE

Bei der anschließenden Angebotsanfrage an die Entitäten ist die Angabe der Start- und Zielposition sowie das Auftragsgewicht verpflichtend, eine maximale Transportgeschwindigkeit sowie der gewünschte Start- bzw. Endzeitpunkt optional. Die Antwort der angefragten Fahrzeugagenten strukturiert sich in drei Parameter: der geschätzten Auftragsdurchlaufzeit, dem voraussichtlichen Endzeitpunkt und den veranschlagten Transportkosten. Die Auftragsvergabe durch den Transportauftragsagenten erfolgt gemäß den Zielvorgaben der Auftragsart (vgl. Abschnitt 4.1.2). Bild 44 zeigt die exemplarische Auftragsvergabe an eine spezifische Entität auf Basis der Auftragsart Kostenauftrag mit Zeitbindung. Aufgrund der Auftragsart erhält das Angebot mit den geringsten Kosten und eines zulässigen Endzeitpunkts den Zuschlag.

Der Großteil der Auftragsarten bezieht die entstehenden Transportkosten in die Vergabe mit ein. Die Kosten werden gemäß Formel (6) ermittelt. Zielführend ist dabei, den Zeitkostenfaktor k_{time} mit einer linearen Degression zu versehen. Dies hat zur Folge, dass älterer Fahrzeuge im System gegenüber ihren baugleichen Mitbewerbern eine erhöhte Chance zum Erhalt des Zuschlags bekommen. Das führt zu einer höheren Betriebsleistung älterer Transportroboter und somit zu verschobenen Wartungszeitpunkten. Dieses einfache Beispiel zeigt, dass die Abbildung unterschiedlicher Gesamtsystemstrategien durch die Veränderung einzelner Parameter möglich ist.

Korrektur der Fahrzeugparameter

Ein wesentlicher Bestandteil für die Performanz der Auftragsdisposition innerhalb des iLES ist die Güte der Auftragsschätzung der einzelnen Entitäten. Hierzu vergleicht der Fahrzeugagent nach der Durchführung eines spezifischen Auftrags die im Parameterserver hinterlegten Einflussfaktoren zur Schätzung der einzelnen Tätigkeiten Anfahrt, Feinpositionierung und Teilehandling mit den tatsächlichen Werten. Gemäß der Korrekturformeln (vgl. 4.3.2) werden diese schließlich angepasst und die Güte der Auftragschätzung kontinuierlich optimiert.

Der Parameterwert nähert sich je nach Korrekturverfahren in Abhängigkeit des Initialwertversatzes und der Schwankungsbreite der Messwerte unterschiedlich schnell dem Realwert an (vgl. Bild 45). Die Auswertungen zeigen die auf 1 normierte Geschwindigkeit einer Transportentität, da die tatsächliche Fahrgeschwindigkeit typspezifisch ist. Statistische Versuchsfahrten mit den eingesetzten Demonstratoren zeigen zudem, dass die Streuung der mittleren Geschwindigkeiten bei gleichen Fahraufträgen zwischen 5 und

20 % liegt. Die Diagramme zeigen somit das Verhalten der unterschiedlichen Korrekturverhalten hinsichtlich der zu erwartenden Streuungen und eines initial falsch festgelegten Geschwindigkeitsparameters. Im Hinblick auf die Robustheit des Korrekturverfahrens in unterschiedlichen Szenarien wird die Verwendung einer gleitenden Mittelwertbildung empfohlen und im Versuchssystem implementiert.

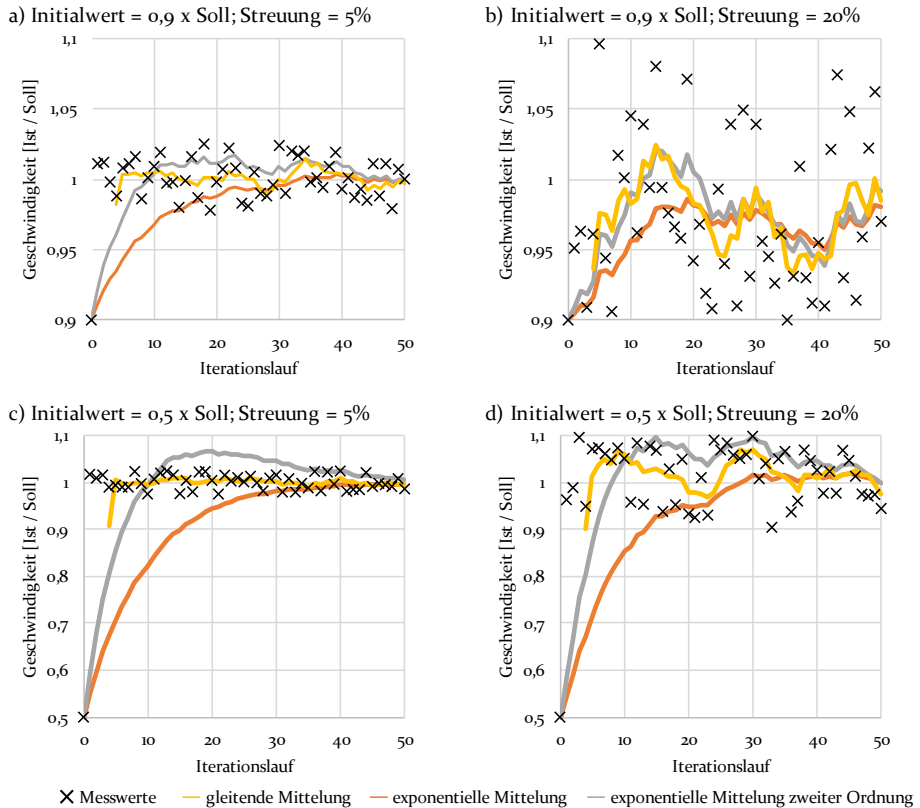


Bild 45: Gegenüberstellung der Verfahren zur Korrektur des initialen Fahrzeuggeschwindigkeitsparameters anhand simulierter Transportfahrten

7.3 Umsetzung der verteilten Architektur und der intralogistischen Serviceebene

Die Modularität der intralogistischen Serviceebene zur Digitalisierung des Arbeitsraums lässt sich durch die Implementierung in einem fabrikähnlichen Testfeld validieren. Das Testszenario befindet sich in der Laborhalle des Lehrstuhl FAPS und umfasst eine Fläche von 220 m², das durch Anla-

gen, Mischverkehr und Personen geprägt ist. Die Erfassung des Arbeitsraums erfolgt durch zehn in 4,5 m Höhe installierte Logitech C930e Webcams, die jeweils einen Bereich von $7,4 \times 4,15 \text{ m}^2$ digitalisieren. Als sensornahe Auswerteeinheit dient ein Raspberry Pi 2 Model B. Die Netzwerkinfrastruktur sieht vor, dass die Informationen der Raspberry Pis an einem Switch gebündelt und über einen WLAN-Router im Netzwerk verteilt werden. Die übergeordneten Dienste sind auf einem zentralen Server implementiert, der über Link-Aggregation am Router angebunden ist, um die notwendige Datenrate zu gewährleisten.

7.3.1 Analyse der sensornahen Informationsseparation

Die Algorithmen zur Interpretation der erfassten Szene jeder Kamera sind auf den sensornahen SBC implementiert. Die vom Sensor übermittelten Daten werden gemäß den Ausführungen in Abschnitt 5.2 in ein adaptives Hintergrundbild sowie die Position und Trajektorien der dynamischen Hindernisse separiert. Bild 46 zeigt das Ergebnis dieses Algorithmus anhand einer industrienahen Beispielszenerie. Die berechneten Positionsdaten und Trajektorien der dynamischen Objekte zeigt die Abbildung zum besseren Verständnis als Bild. Im realen Betrieb hingegen erfolgt die Übermittlung als Array.

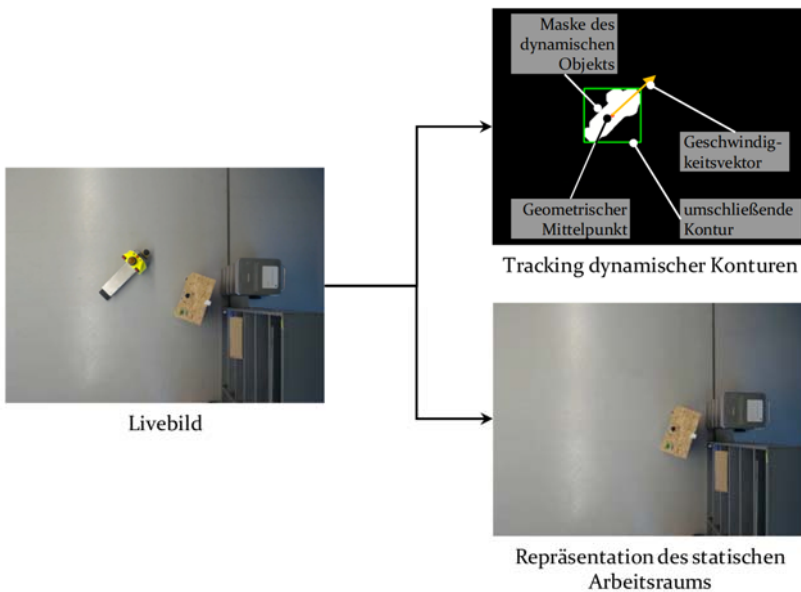


Bild 46: Ergebnis der sensornahen Informationstrennung mit Logitech C930e Webcams und Raspberry Pi 2 Model B

Die Ergebnisse zeigen die Funktionsfähigkeit des erforschten Algorithmus in einem industriellen Szenario. Die dynamischen Objekte werden eindeutig erfasst und verfolgt sowie durch das adaptive Hintergrundbild die statische Szenerie des Arbeitsraums repräsentiert. Die ermittelten Positionen und Trajektorien der dynamischen Hindernisse werden in der weiteren Verarbeitung zur Kollisionsvermeidung der Transportroboter verwendet. Daraus ergibt sich die Forderung, die Laufzeit des erforschten Algorithmus hinsichtlich der benötigten Echtzeitfähigkeit zu validieren.

Implementierung des Quellcodes im SBC

Da die Verarbeitung des Livebilds maßgeblich von Matrizenberechnungen dominiert ist, ist eine direkte Proportionalität zwischen der gewählten Auflösung der Kamera und der Laufzeit zu erwarten. Im Testszenario werden drei unterschiedliche Auflösungen verglichen und die Laufzeit des Algorithmus anhand 50 ausgewählter Einzelframes ermittelt, die zwischen 0 bis 5 Einzelobjekte enthalten. Das Versuchsergebnis ist Tabelle 8 zu entnehmen.

Tabelle 8: Versuchsergebnis der Informationsseparation mit SBC an 50 Einzelbildern

Auflösung [Pixel]	Repräsentierte Kantenlänge [cm]	Mittlere Laufzeit [s]	Streuung [s]	Latenz bei 2m/s [m]	Latenz bei 5m/s [m]
640 x 360	1,15	0,29	0,02	0,6	1,45
960 x 540	0,77	0,67	0,03	1,34	3,35
1920 x 1080	0,38	2,61	0,02	5,22	13,05

Die Auswertung zeigt, dass die Laufzeit des Algorithmus von der Matrizenkalkulation bestimmt wird und somit direkt proportional zur Auflösung ist. Im Hinblick auf die Echtzeitfähigkeit ist anzustreben, eine möglichst geringe Auflösung des Sensors zu wählen. Dem gegenüber steht das abnehmende Abbildungsverhältnis mit einer reduzierten Detailtreue des adaptiven Hintergrundbildes. Aufgrund des gegebenen Öffnungswinkels entspricht jeder Pixel durch die Montage in 4,5 m Höhe bei einer Auflösung von 640 x 360 Pixel einem Quadrat mit einer Kantenlänge von 1,15 cm am Boden. Als Grenzwert für qualitativ sinnvolle Abbildungen sollte nach [14] ein Pixel maximal einer Fläche von 1 cm² entsprechen.

Die Versuchsergebnisse zeigen, dass bei der hierfür notwendigen Auflösung von 940 x 540 Pixel bereits eine mittlere Laufzeit des Algorithmus von 0,67 s erreicht wird. In dieser Zeit legt ein System mit einer Bewegungsgeschwindigkeit von 2 m/s, z. B. Mitarbeiter oder Transportroboter, eine Strecke von 1,34 m zurück. Bei Staplern ist nach der DGUV [156] sogar mit

Geschwindigkeiten bis 5 m/s zu rechnen, was einer Strecke von 3,35 m entspricht. Diese Verzögerung bei der Positionsbestimmung und Übermittlung kann bei der Kollisionsberechnung des Fahrzeugs kompensiert werden. Hierzu wird die Geschwindigkeit des Objekts entlang der ermittelten Trajektorie als konstant angesetzt und die reale Position anhand der übermittelten Positionsdaten geschätzt [14]. Bei Objekten mit variierender Geschwindigkeit und nicht abschätzbaren Trajektorien, wie z. B. Menschen, stößt diese Verfahren jedoch an Grenzen.

Die Echtzeitfähigkeit des erforschten Algorithmus ist hinsichtlich einer Kollisionsvermeidung bei der Sensorapplikation in 4,5 m Höhe unter Verwendung eines Raspberry Pi 2 als kritisch zu beurteilen.

Implementierung des Quellcodes im FPGA

Dieses Laufzeitproblem lässt sich potenziell durch die Verwendung eines FPGA (*Field Programmable Gate Array*) beheben. In dem integrierten Schaltkreis können logische Schaltungen geladen werden und durch die Definition der Schaltungsstruktur der Algorithmus zur Auswertung der Sensordaten konfiguriert werden. Durch die vorgegebene Verschaltung ist vor allem im Bereich der Bildverarbeitung mit einer signifikanten Reduzierung der Laufzeiten zu rechnen. Aufgrund der komplexen Integration des FPGA in das ROS-Framework wird im Rahmen einer Versuchsreihe der erforschte Algorithmus im FPGA konfiguriert, jedoch nicht in die Servicearchitektur integriert. Zum Einsatz kommt das Zedboard Zynq-7000 Development Board von Xilinx mit einem DDR-SDRAM zur Speicherung des adaptiven Hintergrundbilds. Die Ergebnisse der zum Vorversuch analogen Versuchsreihe sind der Tabelle 9 zu entnehmen.

Tabelle 9: Versuchsergebnis der Informationsseparation mit FPGA an 50 Einzelbildern

Auflösung [Pixel]	Repräsentierte Kantenlänge [cm]	Mittlere Laufzeit [ms]	Streuung [ms]	Latenz bei 2m/s [mm]	Latenz bei 5m/s [mm]
640 x 360	1,15	2,36	0,03	4,7	11,8
960 x 540	0,77	5,28	0,04	10,6	26,4
1920 x 1080	0,38	21,24	0,04	42,5	106,2

Auch in dieser Versuchsreihe sind die Matrizenberechnungen die laufzeitbestimmenden Operationen des Algorithmus. Jedoch zeigt sich eine deutliche Reduzierung der mittleren Laufzeiten um den Faktor 100. Die potenziell zurückgelegte Wegstrecke eines Menschen bei einer Auflösung von 960 x 540 Pixel beträgt somit 10,6 mm. Auch die Objekte mit einer Geschwindigkeit von 5 m/s legen in dieser Zeit lediglich 26,4 mm zurück. In der beschriebenen Konfiguration entspricht dies 4 Pixel.

Die Echtzeitfähigkeit der optimierten Hardwarekonfiguration mit einem Xilinx Zedboard Zyng-7000 hinsichtlich der Kollisionsvermeidung erweist sich daher als positiv. Sichergestellt werden muss hingegen die Integration der Hardware in das Softwareframework ROS.

7.3.2 Analyse der binären Darstellung des Gesamtarbeitsraums

Die Binarisierung des Arbeitsraums erfolgt auf dem zentralen Server. Die erforschte Methodik sieht vor, zuerst die einzelnen adaptiven Hintergrundbilder zu einem Gesamtbild des Arbeitsraums zu aggregieren, bevor die segmentierte Maske für das Gesamtabbild erzeugt wird.

Interpretation anhand von Einzelbildern

Um spezifische Effekte des eingesetzten Sensorkonzepts beurteilen zu können, diskutiert der folgende Abschnitt die Erzeugung der Binärmaske anhand des Szenenausschnitts in Bild 47. Das Referenzbild zeigt eine industrietypische Szenerie mit unterschiedlich gearteten statischen Objekten.

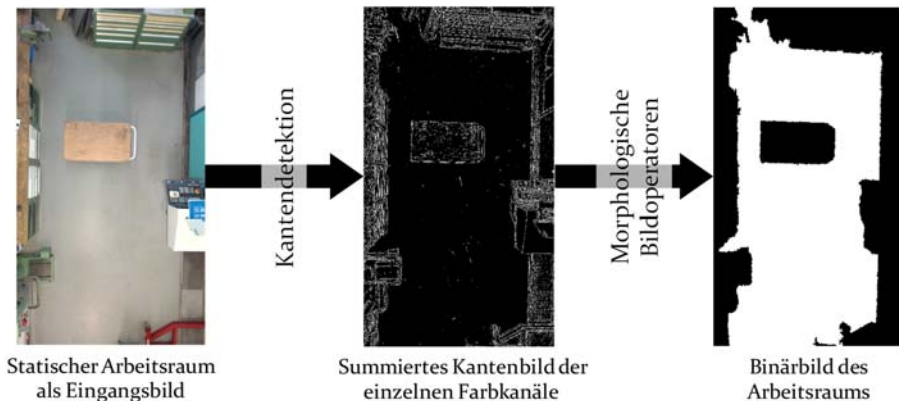


Bild 47: Erzeugung eines binären Abbilds des Arbeitsraums mittels Kantendetektion und morphologischen Operatoren für ein Einzelbild

Wie in Abschnitt 5.3 umfassend erläutert, erfolgt im ersten Schritt eine Kantendetektion in unterschiedlichen Farbkanälen des adaptiven Hintergrundbilds. Das Kantenbild zeigt durch die Kantendetektionsrate von 98,6 % die hohe Zuverlässigkeit des erforschten Ansatzes, unterschiedlichste, auch reflektierende Objektkanten zu detektieren, wohingegen Farbverläufe und Schattenbildung am Boden nicht als Kanten detektiert werden. Die implementierten morphologischen Operatoren werden auf das Kantenbild angewendet und erzeugen die segmentierte binäre Maske,

in der die befahrbaren Bereiche weiß dargestellt sind. Durch die eingesetzten Filtermasken und Grenzwerte werden kleinste Fehldetektionen im Bodenbereich des Kantenbilds nicht als Hindernisobjekte in die Binärmaske integriert. Zudem werden Objektkanten durch die Fülloperation geschlossener Kantenzüge in die Gesamtobjektkontur integriert.

Jedoch zeigt das Referenzbild auch die Schwächen der zweidimensionalen Szenenerfassung durch Deckenkameras auf. Der Arbeitsraum unter ausragenden Anlagenteilen wie Bedienpaneelen wird als belegt interpretiert, da aufgrund der fehlenden Tiefeninformation nicht ermittelt werden kann, in welcher Höhe sich das Hindernis befindet. Weiterhin werden Objektkanten auf den Boden projiziert. Dadurch wird die Fläche vor allem von hohen Objekten außerhalb der Bildmitte mit zunehmenden Bildwinkel größer interpretiert. Diese prinzipiell befahrbaren Bereiche werden folglich als nicht befahrbar eingestuft, wodurch sich der Operationsbereich der Fahrzeuge einschränkt. Da das Gesamtsystem andererseits auch für Fahrzeugtypen verwendet werden kann, die über keine On-board-Sensorik verfügen, ist eine pauschale Freigabe dieser Bereiche nicht zulässig, da durch die Abschattung keine Informationen vorliegen. Auch das Unterfahren von ausragenden Maschinenteilen ist im realen Einsatz nicht zielführend, da eine Einzelfallunterscheidung zwischen der maximalen Durchfahrtshöhe und der aktuellen Fahrzeughöhe erfolgen müsste. Die Steigerung der Systemkomplexität würde somit nicht im Verhältnis zum Zugewinn der Abbildungsgenauigkeit und des Zugewinns an befahrbarer Fläche stehen.

Interpretation anhand der Gesamtabbildung

Im prototypischen Systembetrieb erfolgt die Generierung der binären Maske des Arbeitsraums auf Basis der zusammengesetzten adaptiven Hintergrundbilder der Gesamtszenerie (siehe Bild 48). Hierbei werden weitere systembedingte Effekte der Arbeitsraumdigitalisierung ersichtlich. Durch die Aufnahme des Arbeitsraums durch zehn Einzelkameras entsteht aufgrund der Projektionsfehler eine verzerrte Darstellung der erfassten Objekte. Vor allem im Übergangsbereich zwischen zwei Einzelbildern erzeugt der beschriebene Einfluss bei hohen Objekten unterschiedliche projizierte Objektabmaße. Weiterhin zeigt die Berechnung der Binärmaske in Bereichen mit einer hohen Dichte und Anzahl an statischen Objekten eine Fehlinterpretation der Hindernisflächen.

Neben den realen Hindernissen werden auch Bodenmarkierungen von der Software erfasst und als nicht befahrbare Bereiche interpretiert. Dieser Umstand befähigt das Gesamtsystem jedoch, optische Markierungen wie eingezeichnete Sperrflächen zu detektieren und diese in der Pfadplanung

zu berücksichtigen. Weiterhin besteht die Möglichkeit, die binäre Maske des intralogistischen Arbeitsraums manuell nachzubearbeiten.

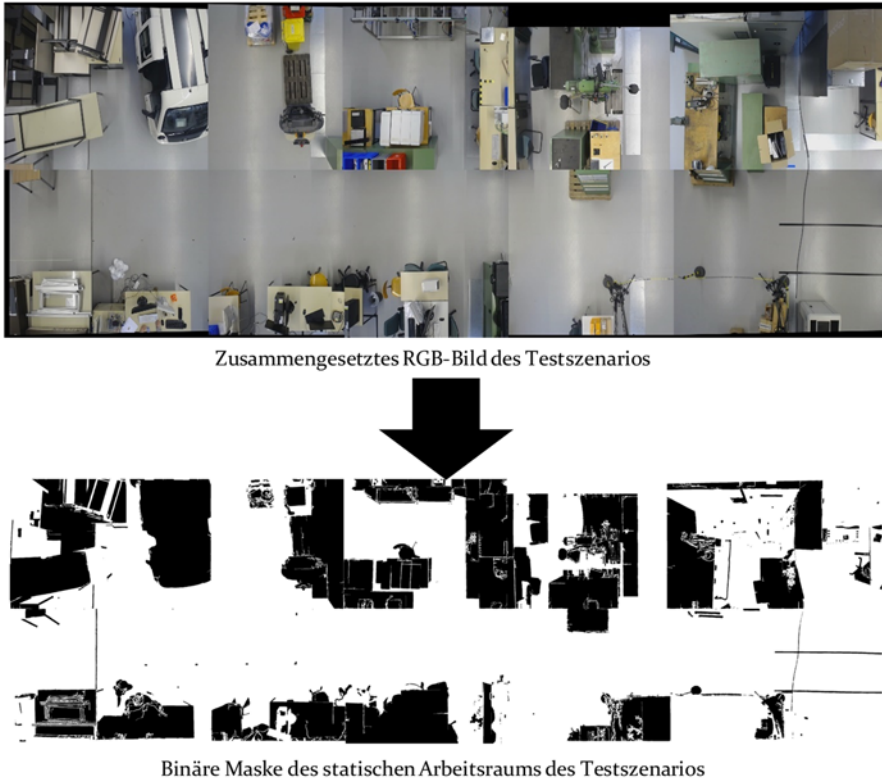


Bild 48: Binäre Maske des Arbeitsraums für die Gesamtszenerie der Testumgebung

7.3.3 Analyse der globalen Routengenerierung

Aufbauend auf der binären Maske des Arbeitsraums werden die Routenetze erzeugt, die auf die jeweiligen Fahrzeugspezifika der unterschiedlichen Transportentitäten angepasst sind und globale Zusatzinformationen berücksichtigen. An dieser Stelle erfolgt somit der Bruch von einer systemweiten Interpretation des Arbeitsraums hin zu einer fahrzeugetypspezifischen Aufbereitung der Informationen. Folglich sind, vor allem bei sehr differenzierten Transportroboterkonzepten, die Laufzeiten der Algorithmen systemrelevant.

Einfluss der Variation der Zufallspunkte

Hauptinflussfaktor der Laufzeit ist neben der gegebenen Größe des Arbeitsraums die Erzeugung der Zufallspunkte innerhalb der befahrbaren Bereiche. Der Abschnitt 5.3 beschreibt die erforschte Methodik, welche die

Fehlversuche zwischen der Erzeugung von zwei zulässigen Zufallspunkten ausgewertet und bei Erreichen des Grenzwertes abbricht. Dadurch lässt sich unabhängig von der Arbeitsraumgröße ein einheitliches und fahrzeugtyp-übergreifendes Abbruchkriterium definieren.

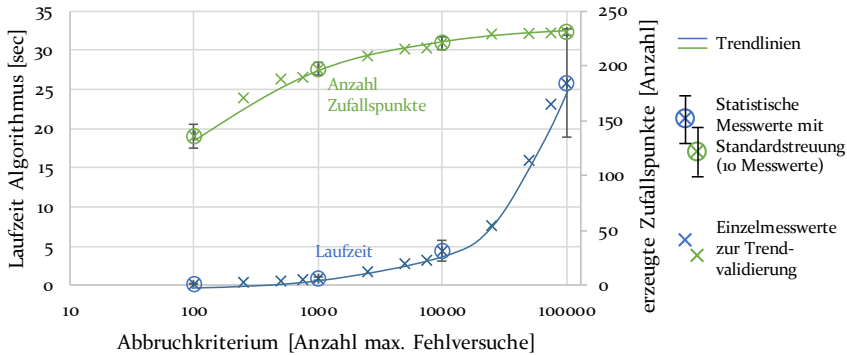


Bild 49: Variation des Abbruchkriteriums zur Erzeugung von Zufallspunkten innerhalb der Routengenerierung

In der prototypischen Anwendung kann durch Variation des Abbruchkriteriums ein zielführender Wertebereich für industrielle Arbeitsräume definiert werden. Die Auswertung erfolgt ohne Zusatzinformationen wie Sperrflächen und Hauptverkehrswegen auf Basis der in Bild 48 dargestellten Binärmaske. Der Sicherheitsradius des verwendeten Fahrzeugtyps beträgt 40 cm. Wie in Bild 49 erkennbar, steigt die Laufzeit des Algorithmus erwartungsgemäß bei Erhöhung des Abbruchkriteriums exponentiell an. Die Anzahl der erzeugten Zufallspunkte, die ein Indikator der Feinmaschigkeit und damit Güte des erzeugten Routenplans darstellen, nähern sich hingegen logarithmisch einem Grenzwert von 240 Punkten an. Ein Optimum aus Performanz und Qualität ist daher in einem Wertebereich zwischen 10.000 und 25.000 Fehlversuchen in Folge erreicht.

Einfluss von Sperrgebieten und Hauptverkehrswegen

Ein wichtiger Bestandteil der erforschten Systemarchitektur ist die Komplettierung des Weltmodells durch Hauptverkehrswege und Sperrgebiete, die über die HMI zur Laufzeit des Systems definiert werden. Die Vorteile des Einsatzes zeigen sich anhand des idealisierten Beispielszenarios in Bild 50, bei dem zur besseren Übersicht die unterschiedlichen Bereiche eingefärbt wurden. Das Beispiel zeigt die Versuchsumgebung mit unterschiedlichen Hindernisobjekten und einem ausgeprägten Mittelgang. Bei einer rein zufallspunktbasierten Wegstreckenplanung entstehen vor allem im Mittelgang eine Vielzahl an Kreuzungspunkten, die potenziell die mittlere

Geschwindigkeit der Transportfahrzeuge reduzieren. Abhilfe schafft der definierte Hauptverkehrsbereich. Die in diesem Bereich erzeugten Kanten sind gerichtet und daher nur in eine Richtung befahrbar. Dadurch entsteht eine gerichtete Verkehrsführung mit einer reduzierten Anzahl an Kreuzungspunkten, wodurch eine höhere mittlere Fahrgeschwindigkeit erreicht wird. Weiterhin werden die gerichteten Kanten im Außenbereich der definierten Fläche erzeugt. Dadurch reduziert sich die potenzielle Flächennutzung der autonomen Transportentitäten im Hauptverkehrsbereich, wodurch der Betrieb im Mischverkehr erleichtert wird.

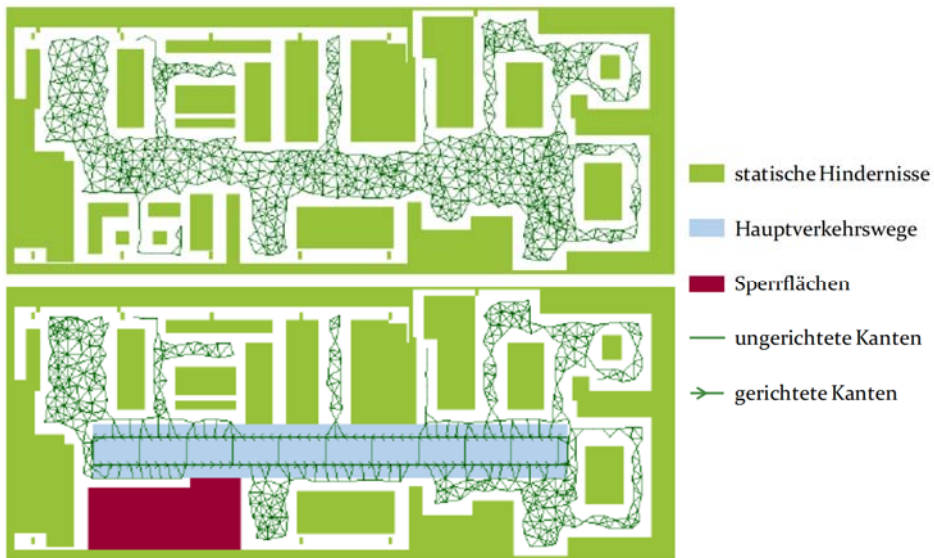


Bild 50: Einfluss von Hauptverkehrswegen und Sperrflächen auf die Ausprägung eines Wegstreckennetzes im idealisierten Layout

Zudem ist im Beispiel der Einfluss von Sperrflächen auf die Generierung des Streckennetzes erkennbar. Durch Sperrflächen kann die Erzeugung von Zufallspunkten innerhalb von potenziellen Freiflächen verhindert werden. Die Sperrfläche selbst wird wie ein statisches Hindernis behandelt, wodurch auch die individuellen Sicherheitsabstände der Fahrzeugtypen eingehalten werden. Sowohl die Anzahl als auch die Dimension der eingesetzten Hauptverkehrsbereiche und Sperrflächen bewirkt erwartungsgemäß keine signifikante Änderung der Laufzeit des Algorithmus gegenüber der zufallsbedingten Streuung.

Einfluss unterschiedlicher Fahrzeugtypen

Der entscheidende Einflussfaktor für die Feinmaschigkeit des Wegstreckennetzes und die Anzahl der erzeugten Zufallspunkte innerhalb eines gegebenen Szenarios ist die Spezifikation des Fahrzeugtyps und der daraus abgeleitete Sicherheitsabstand. Bild 51 zeigt die Streckennetze zweier unterschiedlicher Fahrzeugtypen in der Versuchsumgebung. Die Sicherheitsabstände der beiden Systeme betragen 15 und 30 cm was bei einem gleichen Abbruchkriterium von 10.000 verworfenen Zufallspunkten zu 943 respektive 295 erzeugten Knotenpunkten führt. Dies resultiert in einem signifikant feineren Streckennetz für das kleinere Fahrzeug. Zudem können enge Bereiche befahren werden und die Kanten verlaufen näher zu den statischen Objekten. Bei den Hauptverkehrsbereichen zeigt sich ein weiterer Vorteil der Bereitstellung des Streckennetzes durch einen übergeordneten Dienst. Die Definition der Hauptverkehrswege orientiert sich stets an den Abmessungen des größten Fahrzeugtyps innerhalb des Schwarms. Dadurch bilden sich diese unabhängig von den individuellen Fahrzeugtypparametern aus und sind von allen eingesetzten Fahrzeugen befahrbar und systemübergreifend strukturiert.

Da der Einfluss der erzeugten Zufallspunkte im Verhältnis zum Abbruchkriterium für die Laufzeit des Algorithmus vernachlässigbar ist, ist der eingesetzte Fahrzeugtyp nicht rechenzeitbestimmend. Aufgrund der seriellen Generierung der einzelnen Streckenkarten steigt hingegen die Gesamtlaufzeit des Algorithmus direkt proportional mit der Zahl der eingesetzten Fahrzeugtypen an.

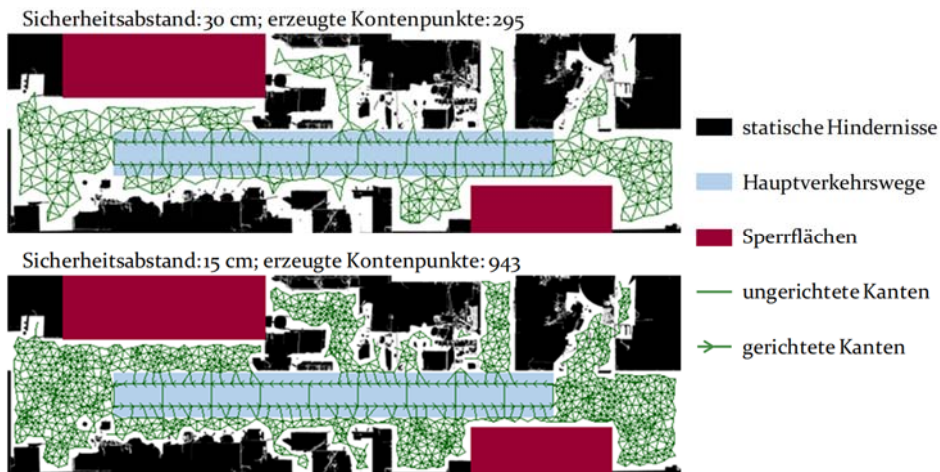


Bild 51: Automatische Generierung von fahrzeugtypspezifischen Streckennetzen innerhalb eines gemeinsamen Arbeitsraums

7.3.4 Analyse der fahrzeugindividuellen Pfadausführung

Im Bereich der elementaren Dienste erfolgt die globale und lokale Pfadplanung auf den Transportentitäten. Die Validierung der Algorithmen erfolgt auf Basis der prototypischen Arbeitsumgebung.

Berechnung des globalen Pfads

Vom übergeordneten Dienst erhält das Transportfahrzeug die Adjazenzmatrix des typspezifischen Streckennetzes. Informationen zu statischen Hindernissen oder weiteren Objekten werden nicht übermittelt. Die globale Pfadplanung erfolgt gemäß Abschnitt 5.4 durch Verwendung des Dijkstra-Algorithmus.

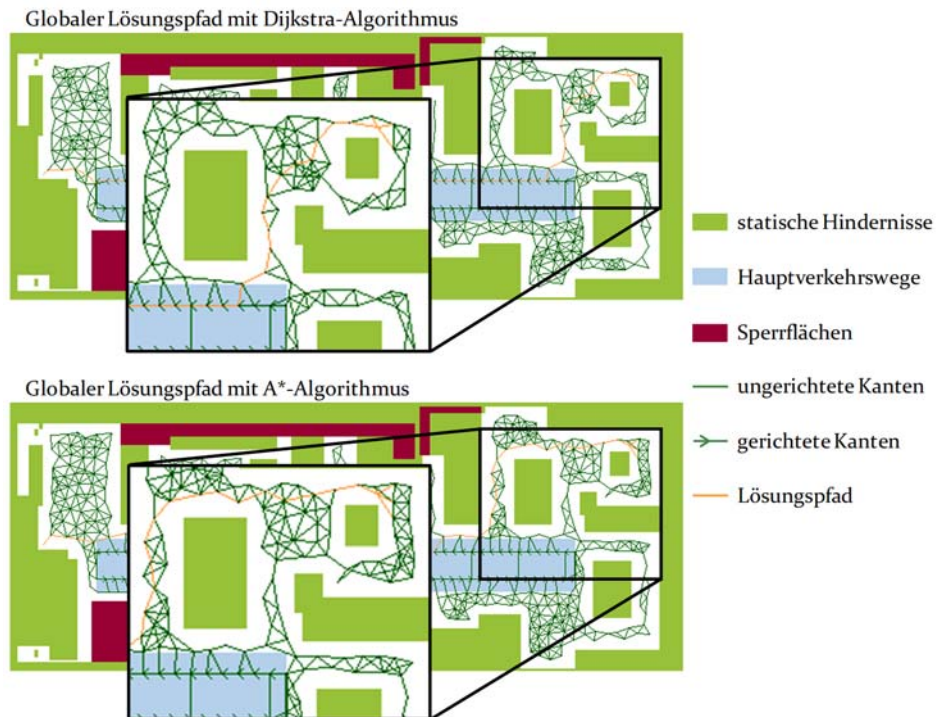


Bild 52: Visualisierung der globalen Pfadplanung mit A*- und Dijkstra-Algorithmus

In der Bild 52 zeigt, dass der Lösungsgraph des Dijkstra-Algorithmus im Gegensatz zur A*-Lösung überwiegend den Hauptverkehrswege nutzt. Die Intention, anhand der Hauptverkehrswege einen möglichst vorhersehbaren und strukturierten Verkehrsfluss zu schaffen, wird daher durch den Einsatz des lauffeiztintensiveren Dijkstra-Algorithmus besser erfüllt. Zur Berechnung des dargestellten Lösungspfads benötigt die rechenleistungsschwächste Versuchsplattform, das FAPS-VAV, im Mittel 2,2 ms. Dies

zeigt, dass die Berechnung von großen Pfaden im Verhältnis zur Durchführungszeit des Gesamtauftrags vernachlässigbar ist. Die Verwendung des Dijkstra-Algorithmus erfüllt somit die Vorgaben einer schnellen Pfadberechnung und primären Verwendung von Hauptverkehrswegen.

Berechnung und Durchführung des lokalen Pfads

Der globale Pfad wird zur Glättung mit Interpolationspunkten versehen und vom Fahrzeug während der Pfadausführung nach dem Prinzip des vorseilenden Stützpunktes nach Buschhaus [157] angepeilt. Zur Validierung des Systems ist die Verwendung einer Plattform mit Differenzialantrieb zielführend, da diese bei zu großer Bahnunstetigkeit funktionsbedingt dazu neigen, eine Drehbewegung auf der Stelle durchzuführen. Die Versuchsergebnisse mit dem autoBod zeigen, dass selbst bei der maximal möglichen Unstetigkeit im globalen Pfad, in jedem Bahnpunkt der rotatorischen eine translatorische Bewegung überlagert ist. Dadurch ist eine möglichst zeiteffiziente Durchführung des globalen Pfads in jedem Fall gewährleistet.

7.4 Einbindung der Fahrzeugagenten in die ereignisdiskrete Simulationsumgebung

Die Funktionsfähigkeit des simulationsbasierten digitalen Zwillings der Produktionsumgebung lässt sich durch die Einbindung der Auftragsallokation in eine industriennahe Materialflusssimulation (vgl. Bild 53) validieren und bewerten. Das Modell beschreibt eine flexible Fertigungsumgebung, in der der Materialtransport der Verbau- und Verbrauchsmittel sowie der Materialtransport zwischen den Anlagen vollständig durch Transportroboter durchgeführt wird. Anhand des Produktionsplans werden die Fertigungsaufträge generiert, deren spezifische Arbeitspläne differenzierte Arbeitsfolgen und Priorisierungen vorsehen. Da das Verhalten der Produktionsanlagen in einem hochflexiblen Fertigungsnetzwerk kein Bestandteil der Forschung ist, werden in den Arbeitsplänen die einzelnen Maschinen und nicht die Maschinengruppen vorgegeben. Für die Durchführung eines Materialtransports werden die auftragspezifischen Parameter in einen OPC UA Server eingetragen. Die Auftragsallokation erfolgt in JADE und die verhandelte Auftragszuweisung wird unter Angabe der Auftragsnummer auf dem Server abgelegt. Plant Simulation führt anschließend gemäß den Transportauftragsdaten den Auftrag aus.

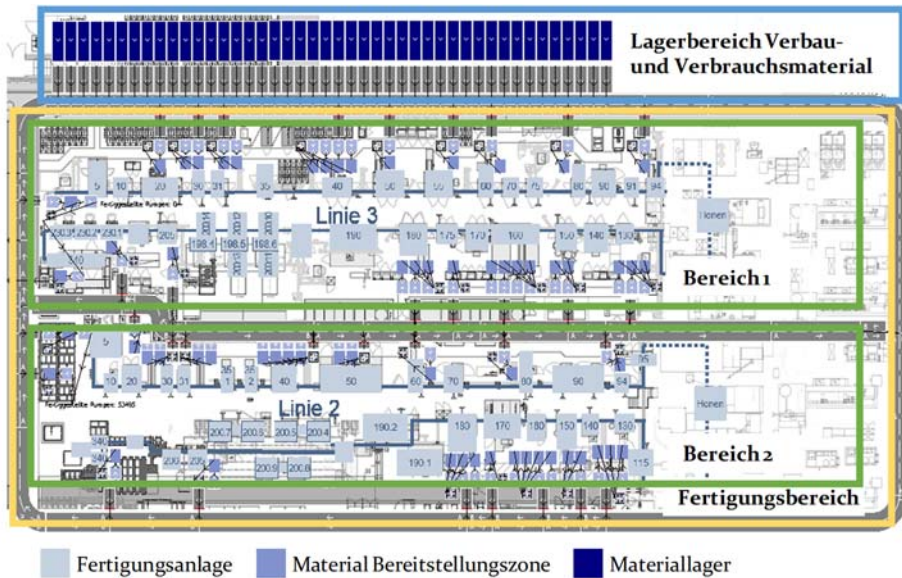


Bild 53: Modell einer Materialflusssimulation zur Validierung der Kopplung von Plant Simulation und JADE

7.4.1 Umsetzung der Schnittstelle zwischen dem Multi-Agenten-System und der ereignisdiskreten Simulation

Zentraler Bestandteil des Informationsaustausches zwischen Plant Simulation und JADE ist die Definition des OPC UA Servers als Schnittstelle. In Bild 54 ist die realisierte Architektur zur Systemvalidierung dargestellt. Die Schnittstelle beider Systeme befindet sich auf dem Simulationsrechner. Dieser enthält das Simulationsmodell, den OPC UA Server und die Agenten zur Generierung von Transportaufträgen sowie die Transportauftragsagenten selbst. Die digitalen Zwillinge der Fahrzeugagenten sind auf den realen Versuchsträgern implementiert, damit ein Zugriff auf die Zustandsautomaten gewährleistet ist. Der Datenaustausch zwischen dem OPC UA Server und JADE erfolgt über einen RESTful-Client (*Representational State Transfer*) und nicht über einen OPC UA Client [158]. Die Größe der Datenmenge erfordert bei der beschriebenen Architektur die zustandslose Verbindung über einen *RESTful Web Service* um den Kommunikationsaufwand zu senken [159]. JADE bietet zur Anbindung des MAS an einen Web Service das Add-on WSIG (*Web Service Integration Gateway*) [149]. Die *Client*-Anfragen werden vom WSIG *Servlet* aufbereitet und an den WSIG Agenten weitergeleitet, der als Kommunikationsschnittstelle zu den Agenten des iLES fungiert. [160]

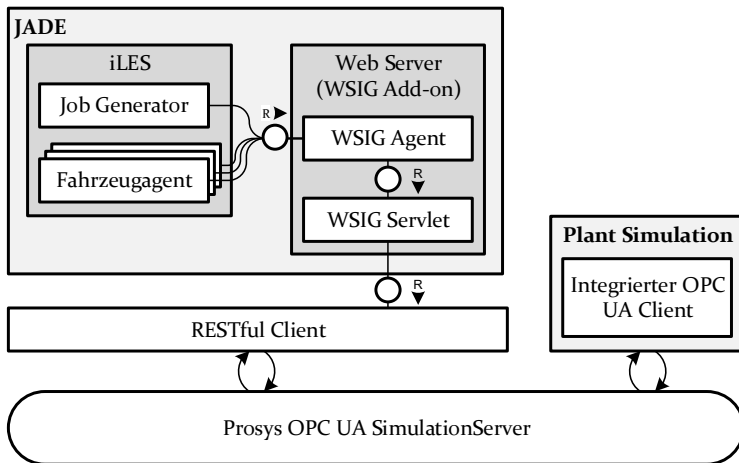


Bild 54: FMC-Block-Diagramm der realisierten Kommunikationsarchitektur zwischen Plant Simulation und JADE unter Verwendung eines OPC UA Servers

Bei der Erzeugung eines Transportauftrags in der Simulation werden über die integrierte OPC UA Schnittstelle von Plant Simulation die spezifischen Auftragsdaten in den OPC UA Server eingetragen. Dabei wird eine eindeutige Auftrags-ID vergeben und die Positionen der Quelle und Senke übertragen. Zudem wird das Transportgewicht und die Auftragsart übermittelt. Die Definition einer maximalen Transportgeschwindigkeit sowie des Startzeitpunkts als eindeutige Unixzeit sind optional und von der Auftragsart abhängig (vgl. Tabelle 10).

Tabelle 10: Auftragsplatzierung von Plant Simulation über den OPC UA Server; Auftragsstruktur und Beispielauftrag; Parameter in runden Klammern sind optional

ID	Quelle [m, m, °] x, y, Winkel	Senke [m, m, °] x, y, Winkel	Gewicht [kg]	(v_{\max}) [m/s]	(Startzeitpunkt) [Unix]	Art	Sim-Flag
4	124,5 147,3 22	78,2 99,6 45	55	2	1534749638	5	1

Der Aufbau des Auftrags ist identisch zu realen Transportaufträgen, mit dem Unterschied, dass der boolesche Wert zur Identifikation von simulierten Aufträgen (*Sim-Flag*) gesetzt ist. Aus den übermittelten Informationen leitet der Auftragsgenerator einen Transportauftrag ab, der die Allokation mit den digitalen Fahrzeugagenten anstößt. Als Basis zur Schätzung dient im beschriebenen Versuch eine manuell generierte Adjazenzmatrix der simulierten Fertigungsumgebung. Bei erfolgter Disposition ergänzt der Transportauftragsagent die Angebotsdaten auf dem OPC UA Server (vgl. Tabelle 11).

Tabelle 11: Ergebnis der Auftragsallokation von JADE an den OPC UA Server; Datenstruktur und Beispielauftrag

ID	Quelle [m, m, °] x, y, Winkel			...	Sim- Flag	Auftragsdurch- laufzeit [sec]	Endzeitpunkt [UNIX]	Kosten [€]	Fahrzeug ID
4	124,5	147,3	22	...	1	189	1534749929	3,24	27

Aus der Auftragsdurchlaufzeit und dem Endzeitpunkt ermittelt Plant Simulation den Start des Auftrags und führt diesen unter Einhaltung der Lösungsinformationen aus JADE mit dem zugehörigen Modell des Transportfahrzeugs durch. Weitere Informationen aus der Allokationsverhandlung stellen die entstehenden Transportkosten dar.

Neben den Transportaufträgen übermittelt Plant Simulation zudem den aktuellen Simulationszeitpunkt an den OPC UA Server. Da die simulierten Transportaufträge im *Sim_Task_Stack* des digitalen Fahrzeugagenten nicht zur Ausführung kommen, werden diese zyklisch auf Basis der aktuellen Simulationszeit gelöscht.

7.4.2 Analyse der Simulationsergebnisse

Zur Analyse der Funktionsfähigkeit der Kommunikationsschnittstelle und des Allokationsalgorithmus werden folgende Versuche mit je fünf Simulationsläufen je Variationsszenario durchgeführt (Bild 55):

- Versuch 1: Variation der Auftragsartverteilung;
- Versuch 2: Variation der Fahrzeugparameter; zwei Fahrzeugklassen mit unterschiedlichen Wegstreckenkosten; nur Kostenaufträge;

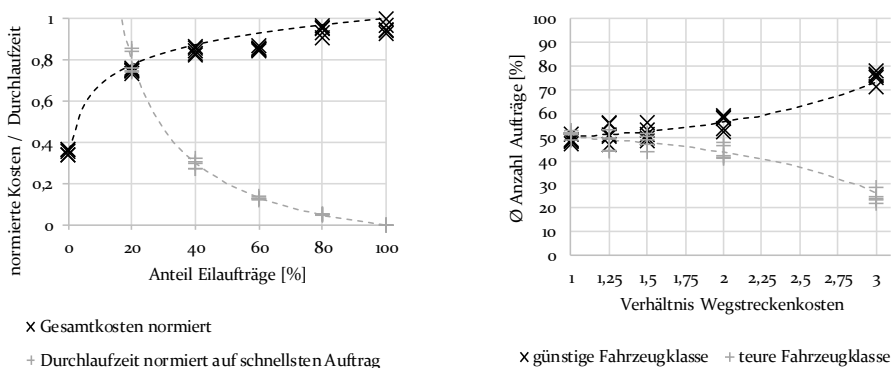


Bild 55: Versuchsergebnisse aus der ereignisdiskreten Simulation; links: Variation der Auftragsarten; rechts: Variation der Fahrzeugparameter

Die Anlagenverfügbarkeit während der Simulationsläufe wird mit 100 % angesetzt, um Einflüsse aus diskreten Maschinenstillständen zu vermeiden.

Durch die Variation der Auftragsarten zeigt sich, dass eine Priorisierung von Eilaufträgen gegenüber den Kostenaufträgen erfolgt. Die mittlere Durchlaufzeit aller Aufträge ist bei einem Verhältnis von 20 % Eilaufträgen ca. um den Faktor 0,8 höher als die der kürzesten Durchlaufzeit. Mit einer zunehmenden Anzahl an Eilaufträgen fällt dieser Effekt jedoch hyperbolisch ab, da der Anteil an Eilaufträgen im System zunimmt und diese vermehrt den Durchschnittswert bilden. Zudem steigt die individuelle mittlere Durchlaufzeit der Eilaufträge mit der Zunahme an Eilaufträgen, da der Allokationsvorteil gegenüber den anderen Aufträgen auf die Ressource Transportfahrzeug geringer wird. Im Gegenzug steigen die Gesamtkosten für die Durchführung aller Aufträge logarithmisch, da Eilaufträge nur die Durchlaufzeit und nicht die Kosten für den Transport in die Entscheidung einbeziehen.

Durch die Variation der Fahrzeugparameter können unterschiedliche Konfigurationen des Fahrzeugschwarms getestet werden. Die Diagramme zeigen die Simulationsergebnisse mit zwei Fahrzeugklassen, eine mit günstigen Wegstreckenkosten und eine mit hohen. Die Versuchsergebnisse verdeutlichen, dass die Disposition von Aufträgen auf die günstigere Fahrzeugklasse mit zunehmendem Unterschied der definierten Wegstreckenkosten exponentiell zunimmt, da die Attraktivität der günstigeren Fahrzeugklasse für Kostenaufträge mit zunehmenden Delta der Wegstreckenkosten zunimmt. Dadurch werden sogar längere Wegstrecken und Durchlaufzeiten egalisiert.

Aus den Ergebnissen der Simulationsläufe lässt sich ableiten, dass die Änderung der Auftragsstruktur und des Fahrzeugverhaltens die Systemantwort signifikant beeinflussen. Somit ist der Informationsfluss zwischen dem Modell in Plant Simulation und dem Vergabeverhalten des iLES gewährleistet, wodurch eine Optimierung der Allokationsstrategie und des Gesamtsystems in der Simulation erfolgen kann. Außerdem wirken sich Änderungen der Transportentitäten und damit der Auftragsallokation in JADE im Modell aus, wodurch die Rückkopplung historischer Daten in die Simulation gewährleistet ist.

7.5 Zusammenfassung und Beurteilung

Nach einer Beschreibung der erforschten Gesamtarchitektur und der zur Evaluierung verwendeten Softwareframeworks in Abschnitt 7.1, wird im

Abschnitt 7.2 die Implementierung des generischen Fahrzeugschichtenmodells adressiert. Die Ergebnisse zeigen, dass die erforschten Zustandsautomaten des Schichtenmodells auf allen Fahrzeugen mit (quasi-)omnidirektionalen Fahrwerk implementiert werden können. Über den integrierten und durch eine standardisierte Schnittstelle konfigurierbaren Fahrzeugparameterserver erfolgt die Applikation der Software auf einer Zielpattform, ohne dass die Algorithmen selbst angepasst werden müssen. Dies ermöglicht einen plattformübergreifenden Einsatz der Fahrzeugsoftware. Weiterhin bietet die erforschte Methodik die Möglichkeit, die Fahrzeugdimensionen während der Laufzeit zu verändern, was die Basis für einen Kooperationsbetrieb baugleicher Transportplattformen darstellt. Die prototypische Implementierung des Schichtenmodells zeigt die Vorteile, die Schnittstellen zwischen den einzelnen Softwareframeworks im Transportfahrzeug zu realisieren. Durch die Kommunikation zwischen dem umgesetzten Fahrzeugagenten und der Zustandsautomaten auf Basis standardisierter Checkpoints und Fehlercodes sind strategische Entscheidungen ableitbar. Weiterhin kann der Zugriff des Agenten auf den Fahrzeugparameterserver gewährleistet werden, um die Auftragsschätzungen anhand der KPIs durchgeführter Transporte stetig zu verbessern.

Die Vorteile der erforschten Methode einer agentenbasierten Auftragsallokation konnte durch die Implementierung innerhalb des MAS-Frameworks JADE aufgezeigt werden. Die Disposition eines spezifischen Transportauftrags richtet sich nicht nach den Zielgrößen eines übergeordneten Leitsystems, sondern nach den individuellen Zielen des jeweiligen Auftrags. Zudem erfolgt die Angebotsanfrage an alle im Materialflusssystem verfügbaren Entitäten. Dadurch wird eine plattformübergreifende Dispositionsstrategie realisiert, die eine Interoperabilität verschiedener Fahrzeugtypen und eine standardisierte Informationsübermittlung ermöglicht.

Ganzheitlich betrachtet ist neben der plattformübergreifenden Auftragsdisposition auch eine standardisierte, dienstbasierte Bereitstellung eines Weltmodells notwendig. Um dies zu ermöglichen, zeigt der Abschnitt 7.3 die Realisierung der hierfür notwendigen essentiellen Dienste in einer verteilten SOA. Die von Deckenkameras erfassten Arbeitsbereiche werden nach dynamischen und statischen Hindernissen untersucht und darauf aufbauend die globale Routenplanung und Pfadausführung transportplattformübergreifend realisiert. Das Gesamtsystem ist darauf ausgerichtet, eine möglichst geringe Datenübermittlung zwischen den Diensten zu realisieren. Die Versuchsergebnisse zeigen, dass die erforschte Methodik für unterschiedliche Fahrzeugkonfigurationen optimierte Routenplanungen

liefert. Zudem werden dynamische Objekte bei der Pfadausführung in einer Weise berücksichtigt, wie es durch eine On-board-Sensorkonfiguration nicht möglich wäre.

Aufgrund des verteilten Systemansatzes und der Verwendung von Deckenkameras ist das erforschte Konzept hinsichtlich des Arbeitsraums beliebig skalierbar. Durch die standardisierte und transportplattformunabhängige Schnittstelle zwischen den Diensten ist zudem gewährleistet, dass das Gesamtsystem in Bezug auf die Anzahl der Transportroboter und Fahrzeugtypen unbeschränkt ist.

Die in Abschnitt 7.4 umgesetzte Kopplung des Systems an eine ereignisdiscrete Simulation zeigt die Möglichkeit auf, eine Vorhersage des Systemverhaltens bei unterschiedlichen Strategien und Auftragsarten zu ermöglichen. Durch den Datenaustausch des Simulationsmodells mit den Fahrzeugagenten über OPC UA ist einerseits die Dispositionsstrategie der Aufträge in der Simulation abbildbar. Andererseits erfolgt eine stetige Aktualisierung des Modellverhaltens durch die Integration der realen Parameterdaten der Transportroboter.

Zusammenfassend stützen die experimentellen Untersuchungen die Funktion und Wirksamkeit der erforschten Methoden und der entwickelten Serviceschicht zur autonomen Auftragsallokation und -durchführung flexibler intralogistischer Transportroboter.

8 Zusammenfassung und Ausblick

FTS sowie intralogistische Transportroboter zeigen hinsichtlich der Realisierung einer flexiblen physischen Vernetzung von Produktionsanlagen das höchste Potenzial. Durch die Entkopplung der Maschinenposition im Raum und der produktspezifischen Reihenfolge des Fertigungsschritts im Arbeitsplan, können die Vorteile der verrichtungs- und prozessfolgeorientierten Struktur von Fertigungsbetrieben vereint werden. Diese Verschmelzung ist aufgrund des Trends zu individualisierten Massenprodukten erforderlich, um weiterhin von den Produktivitätsvorteilen einer Großserienproduktion profitieren zu können, aber zugleich die Flexibilität und Redundanz der Anlagenkapazitäten einer Werkstattfertigung zu gewährleisten, die aufgrund der steigenden Variantenzahl erforderlich sind. Die aktuell verfügbaren und erforschten Transportsysteme zeigen jedoch konzeptionelle Schwächen, die einen Einsatz in einem flexiblen intralogistischen Produktionsnetzwerk nicht ermöglichen. Die Ausführungsformen und Forschungsvorhaben im Bereich der FTS und intralogistischen Transportroboter weisen weder die Interoperabilität der einzelnen Systeme noch die Systemarchitektur auf, die für eine flexible plattformübergreifende Materialflusslösung notwendig sind. FTS verfügen standardmäßig über eine zentrale Leitsteuerung, welche die Transportaufträge auf die FTF disponiert, ein zentrales Wegstreckennetz vorgibt und die Fahrzeuge lediglich als Automaten zur Auftragsdurchführung integriert. Die industriellen Anwendungsfelder von autonomen Transportrobotern beschränken sich wiederum auf Einzelfahrzeuge und kleine Fahrzeugflotten mit einfachen Transportvorgängen zwischen definierten Materialquellen und -senken. Bei komplexeren Szenarien kommt auch hier eine zentrale Leitsteuerung für die Auftragsdisposition zum Einsatz. Sowohl die kommerziellen als auch die wissenschaftlichen Ausführungsformen von autonomen Transportrobotern haben zudem gemein, dass die Hardware sowie die Software spezifische intralogistische Einsatzfelder adressiert und eine Interoperabilität unterschiedlicher Systeme lediglich durch übergeordnete zentrale Instanzen realisiert werden kann. Demzufolge adressiert die vorliegende Dissertationsschrift eine autonome und servicebasierte Auftragsallokation und -durchführung, die nicht von einer Applikation oder spezifischen Systemlösung getrieben ist, sondern die fundamentalen Dienste eines flexiblen intralogistischen Transportsystems erforscht und deren Migration auf unterschiedliche Fahrzeugtypen fokussiert.

Ein wesentlicher Aspekt der Forschungsarbeiten bildet die Bereitstellung einer Auftragsallokationsmethode, bei der der Transportauftrag selbst eine dezentrale und selbstregelnde Disposition auf die Transportentitäten vornimmt. Der elementare Bestandteil ist hierbei eine generische Beschreibung der Eigenschaften von Transportrobotern, um auf einer zweiten Ebene mit Zustandsautomaten übergeordnete und einheitliche Fähigkeiten von Transportsystemen definieren zu können. Diese stellen die Basis für die strategische Ebene der Fahrzeugagenten dar, die in ein MAS eingebunden sind und die Auftragsallokation mit den Transportaufträgen verhandeln. Die erforschte Allokationsstrategie in Zusammenhang mit der universellen Hardwareabstraktion der Transportentitäten hebt sich dabei signifikant von den bekannten Ansätzen von FTS und autonomen Transportrobotern ab, da es die generische Auftragsvergabe und nicht die spezifische Applikation in den Fokus stellt. Zudem ist das System durch die standardisierte Kommunikation und Architektur in der Anzahl der Fahrzeuge und Fahrzeugarten unbegrenzt skalierbar, was sich in dieser Form von den bekannten Ansätzen fundamental unterscheidet. Die Forschungsarbeiten münden somit in einem iLES, das ein logistisches Pendant zu den etablierten MES darstellt.

Neben der Auftragsvergabe stellt zusätzlich die Erforschung eines Servicemodells zur Digitalisierung des Arbeitsraums, der Routenplanung und Pfadausführung einen Eckpfeiler zur Erhöhung der Systemflexibilität und Interoperabilität dar. In der vorgestellten SOA sind die einzelnen Dienste in der Form orchestriert, dass der Datenaustausch zwischen den einzelnen Instanzen auf ein Minimum reduziert wird und stets bedarfsorientiert erfolgt. Durch die Definition von elementaren plattformübergreifenden Funktionen und der gezielten Integration von Diensten in den Infrastruktursensoren und Fahrzeugen ist die erforschte Architektur zudem sowohl in der Größe des Arbeitsraums als auch der Anzahl der eingesetzten Transportroboter unbegrenzt skalierbar. Auf Basis des einheitlichen Weltmodells aller Fahrzeuge und des plattformübergreifenden Informationsflusses ist weiterhin der Einsatz unterschiedlicher mobiler intralogistischer Transportlösungen in einem gemeinsamen Arbeitsraum möglich, was etablierte Systeme in dieser Form nicht gewährleisten. Der Anforderung aus der Verschmelzung der verrichtungs- und prozessfolgeorientierten Struktur nach einer interoperablen Gesamtsystemlösung wird daher in einer bislang nicht verfügbaren Form entsprochen.

Eine wesentliche Herausforderung bei der Einführung einer Gesamtarchitektur bestehend aus iLES und SOA stellt die Vorhersagbarkeit des Systemverhaltens dar. Vor allem mit steigender Anzahl an Transportentitäten und

aktiven Transportaufträgen sind die Einzelentscheidungen und deren Wechselwirkungen nicht deterministisch. Die erforschte Methode zur Integration der Vergabestrategie und Ontologie in eine ereignisdiskrete Simulation liefert in diesem Kontext eine Antwort. Der Grundsatz ist hierbei, nicht die Funktion des iLES abzubilden, sondern den Allokationsalgorithmus als SiL direkt in die Simulation mit einzubinden. Durch die Schaffung eines direkten Transfers des simulierten Systemverhaltens in den realen Betrieb und die Möglichkeit, das Simulationsmodell als HiL auch im produktiven Einsatz fortzuführen, weist die beschriebene Lösung ein Alleinstellungsmerkmal auf. Des Weiteren stellt die realisierte automatische Synchronisation der aktuellen Gegebenheiten des Arbeitsraums und der Eigenschaften der Transportentitäten mit dem Simulationsmodell eine signifikante Steigerung gegenüber den etablierten Ansätzen dar.

Die erforschten Methoden und Algorithmen zur systemübergreifenden und generisch formulierten Auftragsallokation und -durchführung münden in einer prototypischen Realisierung in den Softwareframeworks ROS und JADE sowie der Simulationssoftware Plant Simulation. Der modulare Aufbau der Architektur gewährleistet eine einfache Erweiterbarkeit sowohl im Hinblick auf die Integration weiterer Fahrzeugtypen, als auch zusätzlicher intralogistischer Dienste. Zudem ermöglichen die Lizenzierungsmodelle der verwendeten Tools eine kommerzielle Umsetzung des Gesamtsystems oder einzelner Funktionen und damit das Potenzial einer Verfügbarmachung für den Markt. Die sukzessive Erweiterung des Funktionsumfangs bildet dabei die Grundlage für eine nachhaltige und zukunftsfähige Einsetzbarkeit. Bei einer kommerziellen Version sollte daher schrittweise die Anzahl der möglichen Fahrzeug- und Fahrwerkstypen gesteigert und deren Fähigkeiten, wie z. B. durch die Integration von komplexen Manipulatoren, erweitert werden.

Ausblickend ist es zielführend, die Funktionalität des iLES in der Form zu erweitern, dass aus den Produktionsaufträgen oder Bestellungen direkt eine Ressourceneinlastung der benötigten Fertigungsanlagen und Transportsysteme erfolgen kann. Dazu muss die Auftragsart für Fertigungs- und Transportaufträge aus der Bestellung und der Unternehmensstrategie abgeleitet werden. Weiterhin ist die Weiterentwicklung der integrierten Auftragsverwaltung zu einem Ressourcenkalender notwendig, um ein Optimum aus Fertigungs- und Logistikaufwand zu erreichen. Im Hinblick auf die MSL weist die Detektion von statischen Objekten im Arbeitsraum das größte Forschungsfeld auf. Durch den Einsatz von Bilderkennungsalgorithmen aus dem Bereich *deep learning* besteht das Potenzial, die komplexe

Methode der Binärbilderstellung zu substituieren und zugleich die Definition von Hauptverkehrswegen anhand der vorhandenen Strukturen direkt zu integrieren. Gleiches gilt für die Methoden zur Selbstortung der Fahrzeuge nach Lütke [14]. Auch in diesem Fall besteht durch den Einsatz des maschinellen Lernens ein hohes Potenzial zur Effizienzsteigerung der Algorithmen. Durch die realisierte Modularität der MSL und der definierten Schnittstellen zwischen den Diensten kann die Optimierung und sogar Substitution eines Moduls erfolgen, ohne die generelle Funktionsweise der SOA überarbeiten zu müssen. In diesem Zusammenhang wäre weiterführend die Integration der Maschinenpositionen als zusätzlicher Dienst in das Weltmodell sinnvoll. Dadurch können die fahrzeugtypspezifischen Routenpläne auf die notwendigen und optimalen Quellen- und Senkenverbindungen reduziert und somit die Größe der versendeten Adjazenzmatrizen verringert werden. Ein weiteres Einsatzfeld von Algorithmen der Künstlichen Intelligenz bietet die Optimierung des Agentenverhaltens, der Allokationsstrategie und des Fertigungslayouts auf Basis spezifischer Auftrags- und Systemszenarien. Die Schnittstelle zur DES stellt hierbei die Möglichkeit, unterschiedliche Antworten auf die jeweiligen Herausforderungen im Simulationsmodell empirisch zu testen, gegeneinander zu bewerten und ein Optimum der Parameter zu finden. Durch diesen Ansatz des *data farming* könnte das iLES durch folgende Forschungsvorhaben in die Lage versetzt werden, die Systemstrategie im Hinblick auf übergeordnete Unternehmensziele selbstständig anzupassen und damit eine bislang unerreichbare Flexibilität und Autonomie von intralogistischen Wertströmen zu erreichen.

9 Summary and Outlook

AGVs and intralogistics transport robots show the highest potential regarding the realization of flexible physical networking of production plants. By decoupling the machine position in the space and the product-specific sequence of the production step in the work plan, the advantages of the process-oriented and sequence-oriented structure of production plants can be combined. This fusion is necessary due to the trend to individualized mass products in order to be able to continue to benefit from the productivity advantages of mass production, but at the same time to guarantee the flexibility and redundancy of the plant capacities of workshop production that are required due to the increasing number of variants. The transport systems currently available and being researched, show conceptual weaknesses, which do not allow their use in a flexible intralogistics production network. The design forms and research projects in the field of AGVs and intralogistics transport robots show neither the interoperability of the individual systems nor the system architecture necessary for a flexible cross-platform material flow solution. AGVs are equipped normally with a central control system, which schedules the transport orders to the AGV, defines a central route network and integrates the vehicles as machines for order execution only. The industrial application fields of autonomous transport robots are limited to single vehicles and small vehicle fleets with simple transport tasks between defined material sources. In more complex scenarios, a central control system for order planning is used. Both the commercial and the scientific forms of autonomous transport robots have in common that the hardware and the software address specific intralogistics fields of application and the interoperability of different systems can only be realized by higher-level central instances. Consequently, this dissertation addresses an autonomous and service-based order allocation and execution that is not driven by an application or specific system solution, but investigates the fundamental services of a flexible intralogistics transport system and focuses their migration on different vehicle types.

An essential aspect of the research work is to provide an order allocation method in which the transport order itself executes a decentralized and self-regulating disposition to the transport entities. The elementary component is a generic description of the features of transport robots in order to be able to define higher-level and uniform skills of transport systems on a second level with state machines. These are the basis for the strategic level of the vehicle agents, which are integrated in a MAS and negotiate the order

allocation with the transport orders. The investigated allocation strategy in combination with the universal hardware abstraction of the transport entities is significantly different from the known approaches of AGVs and autonomous transport robots, as it focuses on generic order allocation and not on a specific application. In addition, the system is unlimitedly scalable in the number of vehicles and vehicle types due to the standardized communication and architecture, which fundamentally differs from the known approaches. Thus, the research leads to an iLES, which is a logistic pendant to the established MES.

Additionally, the research of a service model for digitizing the workspace, routing and pathing is a key element to increase the system flexibility and interoperability. In the presented SOA, the individual services are orchestrated in a way that the data exchange between the individual instances is reduced to a minimum and is always demand-oriented. Through the definition of elementary cross-platform functions and the targeted integration of the services in the infrastructure sensors and vehicles, the researched architecture is also infinitely scalable in terms of both the size of the workspace and the number of transport robots used. Based on the uniform world model of all vehicles and the cross-platform information flow, it is possible to use different mobile intralogistics transport solutions in a shared workspace, which is not provided by established systems. The requirement for an interoperable overall system solution resulting from the fusion of the process-oriented and sequence-oriented structure is hence met in a previously unavailable version.

The predictability of the system performance represents a major challenge for the introduction of an overall architecture consisting of iLES and SOA. Especially with an increasing number of transport entities and active transport orders, the individual decisions and their interactions are not deterministic. The researched method for integrating the allocation strategy and ontology into a discrete-event simulation provides an answer in this context. The main aim is not to simulate the function of the iLES, but to integrate the allocation algorithm as SiL directly into the simulation. This solution has a unique selling point due to the creation of a direct transfer of the simulated system behaviour into the real operation and the possibility to continue the simulation model as HiL even in productive use. Furthermore, the realized automated synchronization of the current conditions of the workspace and the features of the transport entities with the simulation model represents a significant advancement compared to the established approaches.

The researched methodology and algorithms for cross-system and generically defined order allocation and execution lead to a prototypical realization in the software frameworks ROS and JADE as well as the simulation software Plant Simulation. The modular structure of the architecture guarantees a simple expandability with regard to the integration of further vehicle types as well as additional intralogistics services. In addition, the licensing models of the used tools enable a commercial implementation of the entire system or individual functions and thus the potential of making it available for the market. The continuous extension of the functionality is the basis for a sustainable and future-proof application. A commercial version should therefore incrementally increase the number of possible vehicle and chassis types and expand its features, e.g. by integrating complex manipulators.

Prospectively, it is expedient to expand the functionality of the iLES in a way that the production orders or purchase orders can be used directly to allocate resources to the required production systems and transport units. For this purpose, the order type for production and transport orders must be derived from the purchase order and the corporate strategy. Furthermore, the further development of the integrated order management into a resource calendar is necessary in order to achieve an optimum of production and logistics costs. With regard to the MSL, the detection of static objects in the workspace has the largest scope of research. The use of image recognition algorithms out of the field of deep learning has the potential to substitute the complex method of binary image creation and at the same time to integrate the definition of highways directly based on the existing structures. Similarly, the methods for the self-location of vehicles according to Lütteke [14] are also applicable. The implementation of machine learning has a high potential to increase the efficiency of the algorithms. Due to the realized modularity of the MSL and the defined interfaces between the services, the optimization and even substitution of a module can be done without revising the general functionality of the SOA. In this context, the integration of the machine positions as an additional service into the world model would make sense. Thus, the vehicle type-specific routes can be reduced to the necessary and optimal source and sink links and thus the size of the sent adjacency matrices can be reduced. Another field of application for artificial intelligence algorithms is the optimization of agent behaviour, allocation strategy and production layout on the basis of specific order and system scenarios. The interface to DES provides the possibility to empirically test different solutions to the respective challenges in the simulation model, to evaluate them against each other and to find an optimum of the

parameters. Through this approach of data farming, the iLES could be enabled by following research projects to autonomously adapt the system strategy with regard to superordinate company targets and thus achieve a so far unattainable flexibility and autonomy of intralogistics value streams.

Literaturverzeichnis

- [1] VDI. 2870 Blatt 1, *Ganzheitliche Produktionssysteme*. Berlin: Beuth Verlag GmbH
- [2] HÄRDLER, J. und T. GONSCHOREK. *Betriebswirtschaftslehre für Ingenieure. Lehr- und Praxisbuch : mit 52 Tabellen und zahlreichen Übungsaufgaben*. 6., neu bearbeitete Auflage. München: Fachbuchverlag Leipzig im Carl Hanser Verlag, 2016. ISBN 9783446443648
- [3] BLOHM, H., T. BEERMANN, U. SEIDENBERG und H. SILBER. *Produktionswirtschaft. Potenziale, Prozesse, Produkte*. 5., vollständig überarbeitete Auflage. Herne: NWB Verlag GmbH & Co.KG, 2016. nwb Studium. ISBN 9783482630255
- [4] GÜNTHER, H.-O. und H. TEMPELMEIER. *Produktion und Logistik*. Berlin/Heidelberg: Springer Berlin Heidelberg, 2012. ISBN 9783642251641
- [5] WANNENWETSCH, H. *Integrierte Materialwirtschaft, Logistik und Beschaffung*. 5., neu bearb. Aufl. Berlin: Springer Vieweg, 2014. Springer-Lehrbuch. ISBN 9783642450228
- [6] SCHÖNSLEBEN, P. *Integrales Logistikmanagement. Operations und Supply Chain Management innerhalb des Unternehmens und unternehmensübergreifend*. 7., bearbeitete und erweiterte Auflage. Berlin: Springer Vieweg, 2016. ISBN 9783642203800
- [7] HANISCH, S. *Das Konzept der Mass Customization. Grundlagen, Praxis und Perspektiven für die Anbieter-Kunden-Beziehung*. Zugl.: Göttingen, Univ., Diplomarb. Saarbrücken: VDM Verl. Dr. Müller, 2006. ISBN 3865502814
- [8] SCHLICK, C., Hg. *Megatrend Digitalisierung - Potenziale der Arbeits- und Betriebsorganisation*. Berlin: GITO, 2016. Schriftenreihe der Wissenschaftlichen Gesellschaft für Arbeits- und Betriebsorganisation (WGAB) e.V. ISBN 9783955451851
- [9] ULLRICH, G., Hg. *Fahrerlose Transportsysteme. Eine Fibel - mit Praxisanwendungen - zur Technik - für die Planung*. 2., erw. und überarb. Aufl. Wiesbaden: Springer Vieweg, 2014. Fortschritte der Robotik. ISBN 978-3-8348-2591-9

- [10] VDI. 2510 Blatt 1, *Infrastruktur und periphere Einrichtungen für Fahrerlose Transportsysteme (FTS)*. Berlin: Beuth Verlag GmbH
- [11] SCHOLZ, M. und J. FRANKE. Bitte ein Taxi! Autonomer Materialtransport in der Intralogistik [online]. *Handling*, 2015, 52-53. Verfügbar unter: <https://www.handling.de/epaper/handling1215/blaetterkatalog/index.html>
- [12] FRANKE, J. und F. LÜTTEKE. Versatile autonomous transportation vehicle for highly flexible use in industrial applications [online]. *CIRP Annals*, 2012, 61(1), 407-410. ISSN 00078506. Verfügbar unter: doi:10.1016/j.cirp.2012.03.048
- [13] SCHULTE, C. *Logistik. Wege zur Optimierung der Supply Chain*. 5., überarb. und erw. Aufl. München: Vahlen, 2009. Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften. ISBN 380063516X
- [14] LÜTTEKE, F. *Vielseitiges autonomes Transportsystem basierend auf Weltmodellerstellung mittels Datenfusion von Deckenkameras und Fahrzeugsensoren*. Bamberg: Meisenbach, 2014. Fertigungstechnik - Erlangen. 259. ISBN 3875253760
- [15] GROSS, D., Hg. *Forbes' greatest business stories of all time*. New York: John Wiley & Sons, 1996. ISBN 0471196533
- [16] WEMHÖNER, N. *Flexibilitätsoptimierung zur Auslastungssteigerung im Automobilrohbau*. Zugl.: Aachen, Techn. Hochsch., Diss., 2005. Aachen: Shaker, 2006. Berichte aus der Produktionstechnik. 2006,12. ISBN 3832251111
- [17] KUHNERT, F., J. MASER und P. FUNDA. How to stay No. 1! Impulse für die zentralen Herausforderungen in der Automobilindustrie, 2013
- [18] PILLER, F.T. *Mass Customization. Ein wettbewerbsstrategisches Konzept im Informationszeitalter*. Zugl.: Würzburg, Univ., Diss., 1999 u.d.T.: Kundenindividuelle Massenproduktion (mass customization) als wettbewerbsstrategisches Modell industrieller Wertschöpfung in der Informationsgesellschaft. 4., überarb. und erw. Aufl., Nachdr. Wiesbaden: Dt. Univ.-Verl., 2008. Gabler Edition Wissenschaft Markt- und Unternehmensentwicklung. ISBN 3835003550

- [19] DIEZ, W. *Automobil-Marketing. Erfolgreiche Strategien, praxisorientierte Konzepte, effektive Instrumente*. 6., vollständig überarbeitete Auflage. München: Verlag Franz Vahlen, 2015. ISBN 9783800646401
- [20] COLETTI, P. und T. AICHNER. *Mass Customization. An Exploration of European Characteristics*. Berlin, Heidelberg: Paolo Colletti, 2011. SpringerBriefs in Business. ISBN 9783642183898
- [21] NEBL, T. *Produktionswirtschaft*. 7., vollst. überarb. und erw. Aufl. München: Oldenbourg, 2011. Lehr- und Handbücher der Betriebswirtschaftslehre. ISBN 3486596691
- [22] KRAMPE, H., H.-J. LUCKE und M. SCHENK, Hg. *Grundlagen der Logistik. Einführung in Theorie und Praxis logistischer Systeme*. 4. völlig neu bearbeitete und erweiterte Auflage. München: Huss-Verlag GmbH, 2012. Logistik Wissen. ISBN 978-3-941418-80-6
- [23] VAHRENKAMP, R. und C. SIEPERMANN. *Produktionsmanagement*. 6., 6., überarbeitete Aufl. München: De Gruyter, 2008. ISBN 3486587846
- [24] ZÄPFEL, G. *Strategisches Produktions-Management*. 2., 2., unwesentlich veränderte Aufl. München: De Gruyter Oldenbourg, 2000. ISBN 3486254502
- [25] JACOBS, H.-J. und H. DÜRR. *Entwurf und Gestaltung von Fertigungsprozessen. Planung und Steuerung der spanenden Teilefertigung*. München: Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2002. ISBN 3446217487
- [26] WILDEMANN, H. *Fertigungsstrategien. Reorganisationskonzepte für eine schlanke Produktion und Zulieferung*. 3., überarb. Aufl. München: Transfer-Centrum-Verl., 1997. TCW. 6. ISBN 3929918897
- [27] SCHUH, G. und V. STICH, Hg. *Produktionsplanung und -steuerung 2. Evolution der PPS*. 4., überarb. Aufl. Berlin: Springer, 2012. VDI-Buch. ISBN 978-3-642-25426-0
- [28] GROCHOWSKI, E., R. NAGELE und P. OHLHAUSEN. *Interdisciplinary, cross-company new product development in the future automotive industry-the case ARENA2036*, 2015

- [29] NYHUIS, P., Hg. *Wandlungsfähige Produktionssysteme. Heute die Industrie von morgen gestalten*. Hannover: Technische Informationsbibliothek u. Universitätsbibliothek; PZH Produktionstechnisches Zentrum, 2008. ISBN 3939026964
- [30] GOLDMAN, S.L., R.N. NAGEL, K. PREISS und H.-J. WARNECKE. *Agil im Wettbewerb. Die Strategie der virtuellen Organisation zum Nutzen des Kunden*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. ISBN 3540606440
- [31] LLAMAS, V., T. COUDERT, L. GENESTE, J.C. ROMERO-BEJARANO und A. de VALROGER. Proposition of an agile knowledge-based process model [online]. *IFAC-PapersOnLine*, 2016, 49(12), 1092-1097. ISSN 24058963. Verfügbar unter: doi:10.1016/j.ifacol.2016.07.588
- [32] WARNECKE, H.-J. und M. HÜSER. *Die Fraktale Fabrik. Revolution der Unternehmenskultur*. Neuaufl. Reinbek bei Hamburg: Rowohlt, 1996. rororo science. 9708. ISBN 3499197081
- [33] LEITÃO, P. und F. RESTIVO. ADACOR: A holonic architecture for agile and adaptive manufacturing control [online]. *Computers in Industry*, 2006, 57(2), 121-130. ISSN 01663615. Verfügbar unter: doi:10.1016/j.compind.2005.05.005
- [34] WARNECKE, H.-J., Hg. *Vom Fraktal zum Produktionsnetzwerk. Unternehmenskooperationen erfolgreich gestalten*. Berlin: Springer, 1999. ISBN 354064525X
- [35] RYU, K. und M. JUNG. Agent-based fractal architecture and modelling for developing distributed manufacturing systems [online]. *International Journal of Production Research*, 2003, 41(17), 4233-4255. ISSN 0020-7543. Verfügbar unter: doi:10.1080/0020754031000149275
- [36] THARUMARAJAH, A., A.J. WELLS und L. NEMES. Comparison of emerging manufacturing concepts. In: *1998 IEEE International Conference on Systems, Man and Cybernetics*. Piscataway: IEEE, July 1998, S. 325-331. ISBN 0-7803-4778-1
- [37] BOTTI, V. und A. GIRET. *ANEMONA. A Multi-agent Methodology for Holonic Manufacturing Systems*. s.l.: Springer Verlag London Limited, 2008. Springer series in advanced manufacturing. ISBN 978-1-84800-309-5

- [38] CAMARINHA-MATOS, L.M., H. AFSARMANESH und V. MARIK, Hg. *Intelligent Systems for Manufacturing. Multi-Agent Systems and Virtual Organizations*. Boston, MA: Springer, 1998. IFIP - The International Federation for Information Processing. 1. ISBN 978-1-4757-5547-3
- [39] THARUMARAJAH, A. Comparison of the bionic, fractal and holo-
nic manufacturing system concepts [online]. *International Journal of Computer Integrated Manufacturing*, 1996, 9(3), 217-226. ISSN 0951-192X. Verfügbar unter:
doi:10.1080/095119296131670
- [40] MÖNCH, L., Hg. *Agentenbasierte Produktionssteuerung komplexer Produktionssysteme*. Wiesbaden: Deutscher Universitäts-Verlag, 2006. Wirtschaftsinformatik. ISBN 9783835002494
- [41] KLAUS, P. und W. KRIEGER. *GABLER LEXIKON LOGISTIK*. Wiesbaden: Gabler, 2009. ISBN 978-3-8349-0149-1
- [42] KIENCKE, U. *Ereignisdiskrete Systeme. Modellierung und Steuerung verteilter Systeme*. 2., überarb. und erw. Aufl. München: Oldenbourg, 2009. ISBN 3-486-58011-6
- [43] MARQUARDT, H.-G. und K. TUREK. Beitrag zur Dimensionierung von automatischen Sortiersystemen (Sorter) [online]. *Logistics Journal nicht-referierte Veröffentlichungen*, 2007. Verfügbar unter: doi:10.2195/LJ_Not_Ref_Turek_072007
- [44] BAUM, D. *Grundlagen der Warteschlangentheorie*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. Springer-Lehrbuch Masterclass. ISBN 978-3642396311
- [45] LIEBEROTH-LEDEN, C., J. FISCHER, J. FOTTNER und B. VOGEL-HEUSER. Control Architecture and Transport Coordination for Autonomous Logistics Modules in Flexible Automated Material Flow Systems. In: *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*: IEEE, 20. August 2018 - 24. August 2018, S. 736-743. ISBN 978-1-5386-3593-3
- [46] PFEIFER, H., G. KABISCH und H. LAUTNER. *Fördertechnik. Konstruktion und Berechnung*. 7., verb. Aufl. Braunschweig u.a.: Vieweg, 1998. Viewegs Fachbücher der Technik. ISBN 3-528-64061-8
- [47] CAMPION, G., G. BASTIN und B. DANDREA-NOVEL. Structural properties and classification of kinematic and dynamic models of

- wheeled mobile robots [online]. *IEEE Transactions on Robotics and Automation*, 1996, 12(1), 47-62. Verfügbar unter: doi:10.1109/70.481750
- [48] JACOBS, T., C.P. CONNETTE, M. HÄGELE und e. AL. Steigerung der Flexibilität mobiler Roboter und Transportsysteme durch den Einsatz kompakter, omnidirektionaler Antriebe. In: *Konferenzbeiträge / Internationales Forum Mechatronik 2010. Winterthur 3. - 4. November 2010*. Winterthur: ZHAW Zürcher Hochschule für Angewandte Wissenschaften School of Engineering IMS Institut für Mechatronische Systeme, 2010, S. 15. ISBN 9783033026551
- [49] ADASCALITEI, F. und I. DOROFTEI. Practical applications for mobile robots based on Mecanum wheels - a systematic survey, 2011, 21-29
- [50] TLALE, N. und M. de VILLIERS. Kinematics and Dynamics Modelling of a Mecanum Wheeled Mobile Platform. In: T. MOIR, Hg. *15th International Conference on Mechatronics and Machine Vision in Practice, 2008. M2VIP 2008 ; 2 - 4 Dec. 2008, Auckland, New Zealand*. Piscataway, NJ: IEEE, 2008, S. 657-662. ISBN 978-1-4244-3779-5
- [51] VDI. 4451 Blatt 7, *Kompatibilität von Fahrerlosen Transportsystemen (FTS)*: Beuth Verlag GmbH
- [52] COLLING, D., S. IBRAHIMPASIC, A. TRENKLE und K. FURMANS. *Dezentrale Auftragserzeugung und -vergabe für FTF*: Wissenschaftliche Gesellschaft für Technische Logistik, 2016
- [53] SCHRECKER, A. Systemübersicht und Funktionen von Fahrerlosen Transportsystemen. In: A. SCHRECKER, Hg. *Planung und Steuerung Fahrerloser Transportsysteme. Ansätze zur Unterstützung der Systemgestaltung*. Gabler Edition Wissenschaft. Wiesbaden: Deutscher Universitätsverlag, 2000, S. 5-13. ISBN 978-3-8244-7086-0
- [54] ANGERBAUER, R. Fahrerlose Transportsysteme. In: R. ANGERBAUER, Hg. *Anwenderorientierte Programmierung fahrerloser Transportsysteme*. Berlin: Springer, 1996, S. 15-22. ISBN 978-3-540-61371-8

- [55] ANGERBAUER, R. Anwenderorientierte Programmierung fahrerloser Transportsysteme. In: R. ANGERBAUER, Hg. *Anwenderorientierte Programmierung fahrerloser Transportsysteme*. Berlin: Springer, 1996, S. 23-41. ISBN 978-3-540-61371-8
- [56] SCHWAB, M. *A Decentralized Control Strategy for High Density Material Flow Systems with Automated Guided Vehicles*: KIT Scientific Publishing
- [57] ZIVKOVIC, Z. Improved adaptive Gaussian mixture model for background subtraction. In: J. KITTLER, Hg. *Proceedings of the 17th International Conference on Pattern Recognition. August 23 - 26, 2004, Cambridge, UK*. Los Alamitos, Calif.: IEEE Computer Soc, 2004, 28-31 Vol.2. ISBN 0-7695-2128-2
- [58] ZIVKOVIC, Z. und F. VAN DER HEIJDEN. Efficient adaptive density estimation per image pixel for the task of background subtraction [online]. *Pattern Recognition Letters*, 2006, 27(7), 773-780. ISSN 01678655. Verfügbar unter: doi:10.1016/j.pat-rec.2005.11.005
- [59] BEN-ARI, M. und F. MONDADA. *Elements of Robotics*. Cham: Springer International Publishing, 2018. ISBN 978-3-319-62532-4
- [60] GRISETTI, G., C. STACHNISS und W. BURGARD. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters [online]. *IEEE Transactions on Robotics*, 2007, 23(1), 34-46. ISSN 1552-3098. Verfügbar unter: doi:10.1109/TRO.2006.889486
- [61] KRUSE, E. *Bewegungsplanung für mobile Roboter in dynamischen Umgebungen auf Basis automatisch erzeugter statistischer Daten*. Als Ms. gedr. Aachen: Shaker, 1998. Fortschritte in der Robotik. Bd. 4. ISBN 9783826537011
- [62] FOX, D. Adapting the Sample Size in Particle Filters Through KLD-Sampling [online]. *The International Journal of Robotics Research*, 2016, 22(12), 985-1003. ISSN 0278-3649. Verfügbar unter: doi:10.1177/0278364903022012001
- [63] LU, D.V., D. HERSHBERGER und W.D. SMART. Layered cost-maps for context-sensitive navigation. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS 2014) ; Chicago, Illinois, USA, 14 - 18 September 2014*. Piscataway, NJ: IEEE, 2014, S. 709-715. ISBN 978-1-4799-6934-0

- [64] KRUSE, E. und F.M. WAHL. Camera-based observation of obstacle motions to derive statistical data for mobile robot motion planning. In: *1998 IEEE International Conference on Robotics and Automation*. Piscataway: IEEE, May 1998, S. 662-667. ISBN 0-7803-4300-X
- [65] DIN EN. 1525, *Sicherheit von Flurförderzeugen - Fahrerlose Flurförderzeuge und ihre Systeme*. Berlin: Beuth Verlag GmbH
- [66] GUTSCHE, R. *Fahrerlose Transportsysteme. Automatische Bahnplanung in dynamischen Umgebungen*. Wiesbaden: Vieweg+Teubner Verlag, 1994. Fortschritte der Robotik. 22. ISBN 352806658X
- [67] SPIEKER, S. *Bahnplanung autonomer Transportfahrzeuge*. München: AVM, 2012. ISBN 3869242817
- [68] LAVALLE, S.M. *Planning algorithms*. Cambridge: Cambridge University Press, 2006. ISBN 9780521862059
- [69] CHOSET, H.M. *Principles of robot motion. Theory, algorithms, and implementation*. Cambridge, Mass: MIT Press, 2005. Intelligent robotics and autonomous agents. ISBN 0262033275
- [70] KOREN, Y. und J. BORENSTEIN. Potential field methods and their inherent limitations for mobile robot navigation. In: *Robotics and Automation, '91 International Conference On*. Los Alamitos: IEEE Computer Society Press, Oct. 1991, S. 1398-1404. ISBN 0-8186-2163-X
- [71] Götting KG | *Komponenten für FTF | Fahrzeugautomatisierung* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <https://www.goetting.de/>
- [72] *Intralogistik - Grenzebach Maschinenbau GmbH* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <https://www.grenzebach.com/de/produkte-maerkte/intralogistik/>
- [73] *EVE – Serva* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://serva-ts.com/de/intralogistik/eve/>
- [74] *LD Series | Omron, United States* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <https://industrial.omron.us/en/products/ld-series#features>
- [75] ROBOTS, M.I. *MiR100™ | Mobile Industrial Robots* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://www.mobile-industrial-robots.com/en/products/mir100/>

- [76] ROBOTS, M.I. *MiR200 more powerful autonomous mobile robot | Mobile Industrial Robots* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://www.mobile-industrial-robots.com/en/products/mir200/>
- [77] *Magazino Produkte - Übersicht über alle Roboter und Lösungen* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <https://www.magazino.eu/toru/>
- [78] KOPYTYNSKI, W. *arculus GmbH – Modular Production* [online]. 5 Juni 2018 [Zugriff am: 7. Juni 2018]. Verfügbar unter: <https://www.arculus.de/>
- [79] »Smart Transport Robot« – *FTS im Kofferformat - Fraunhofer IML* [online]. 7 Juni 2018 [Zugriff am: 7. Juni 2018]. Verfügbar unter: https://www.iml.fraunhofer.de/de/presse_medien/pressemitteilungen/-smart-transport-robot---fts-im-kofferformat-.html
- [80] GÜNTHER, W. und M. ten HOMPEL. *Internet der Dinge in der Intralogistik*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010. VDI-Buch. ISBN 9783642048951
- [81] *MultiShuttle* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: https://www.iml.fraunhofer.de/de/abteilungen/b1/maschinen_und_anlagen/Forschungsprojekte/Multishuttle_Move_Fahrerloses_Transportsystem.html
- [82] JUNG, H. *Konzept einer agentenbasierten Transportsteuerung. Für komplexe, dynamische und multimodale Logistiknetzwerke*: KIT Scientific Publishing, 2016
- [83] FURMANS, K., F. SCHÖNUNG, C. NOBBE und D. BERBIG. *Wandlungsfähiger Materialfluss für die Produktion von Morgen. ATZproduktion*, 2010, 2010(Volume 3), 54-57
- [84] *Karis Pro Abschlussbericht* [online] [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://karispro.de/Abschlussbericht%20KA-RIS%20PRO.pdf>
- [85] COLOMB, A. und K.-H. WEHKING. *Materialflusssystem für die JIS-Teilebereitstellung. Hebezeuge Fördermittel 6/2016, Forschungskatalog Flurförderzeuge 2016*, 2016, 41
- [86] *Gold für das Doppelkufensystem von Eisenmann - MindSphere World auf der Hannover Messe* [online] [Zugriff am: 7. Juni 2018].

- Verfügbar unter: http://www.eisenmann.com/de/m Medien/presse/presseinformationen/2014/2014_02_25_Logi-MAT_BestesProdukt.html
- [87] POSSEL-DÖLKEN, F. *Projektierbares Multiagentensystem für die Ablaufsteuerung in der flexibel automatisierten Fertigung*. Zugl.: Aachen, Techn- Hochsch., Diss., 2006. Aachen: Shaker; Techn. Univ, 2006. Berichte aus der Produktionstechnik. 2006,10. ISBN 3-8322-5077-8
 - [88] WEBER, M. *Unterstützung der Wandlungsfähigkeit von Produktionsanlagen durch innovative Softwaresysteme*. Zugl.: Erlangen-Nürnberg, Univ., Diss., 2007. Bamberg: Meisenbach, 2007. Fertigungstechnik - Erlangen. 188. ISBN 3875252691
 - [89] WOLF, W. *Entwicklung eines agentenbasierten Steuerungssystems zur Materialflussorganisation im wandelbaren Produktionsumfeld*. Zugl.: Erlangen-Nürnberg, Univ., Diss., 2009. Bamberg: Meisenbach, 2009. Fertigungstechnik - Erlangen. 205. ISBN 9783875252934
 - [90] OVERMEYER, L., Hg. *Tagungsband / 6. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik (WGTL). 29. und 30. September 2010*. Garbsen: PZH Produktionstechnisches Zentrum, 2010. ISBN 9783941416710
 - [91] KRESS-GAZIT, H., T. WONGPIROMSARN und U. TOPCU. Correct, Reactive, High-Level Robot Control [online]. *IEEE Robotics & Automation Magazine*, 2011, 18(3), 65-74. ISSN 1070-9932. Verfügbar unter: doi:10.1109/MRA.2011.942116
 - [92] KRUSE, E. und F.M. WAHL. Camera-based monitoring system for mobile robot guidance. In: *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Proceedings : innovations in theory, practice, and applications : October 13-17, 1998, Victoria Conference Centre, Victoria, B.C., Canada*. Piscataway, N.J.: IEEE, 1998, S. 1248-1253. ISBN 0-7803-4465-0
 - [93] HOMPEL, M. ten, T. SCHMIDT und L. NAGEL, Hg. *Materialflusssysteme. Förder- und Lagertechnik*. 3., völlig neu bearbeitete Auflage. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007. Intralogistik. ISBN 978-3-540-73235-8

- [94] ARNOLD, D. *Intralogistik. Potentiale, Perspektiven, Prognosen*. Berlin Heidelberg: Springer-Verlag GmbH, 2006. VDI-Buch. ISBN 978-3540296577
- [95] BALTES, J., M.G. LAGOUDAKIS, T. NARUSE und S.S. GHIDARY, Hg. *RoboCup 2009: Robot Soccer World Cup XIII*. Berlin: Springer, 2010. Lecture notes in computer science Lecture notes in artificial intelligence. 5949. ISBN 978-3-642-11875-3
- [96] AICHER, T., D. DOTSCHE, M. SPINDLER, J. FOTTNER und B. VOGEL-HEUSER. A feasibility study regarding the implementation and application of a two-layer architecture for automated material flow systems. In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*: IEEE, 29. Oktober 2017 - 1. November 2017, S. 2821-2827. ISBN 978-1-5386-1127-2
- [97] DIRK GEHRKE, J. Relevanzbasierte Informationsbeschaffung für die informierte Entscheidungsfindung intelligenter Agenten, 2011
- [98] PAWLASZCZYK, D. *Skalierbare agentenbasierte Simulation. Werkzeuge und Techniken zur verteilten Ausführung agentenbasierter Modelle*. Zugl.: Ilmenau, Techn. Univ., Diss., 2009. Ilmenau: Univ.-Verl., 2009. ISBN 3939473596
- [99] TELGEN, D., E. PUIK, L. VAN MOERGESTEL, T. BAKKER und J.-j. MEYER. Reconfigurable Equiplets Operating System - A Hybrid Architecture to Combine Flexibility and Performance for Manufacturing, 2015, 8, 309-326
- [100] FIPA. SC00029H, *Contract Net Interaction Protocol Specification* [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://www.fipa.org/specs/fipa00029/SC00029H.html>
- [101] FIPA. SC00023K, *Agent Management Specification* [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://www.fipa.org/specs/fipa00023/SC00023K.html>
- [102] NAKHAEINIA, D., S.H. TANG, S. NOOR und O. MOTLAGH. A review of control architectures for autonomous navigation of mobile robots, 2011, 6, 169-174
- [103] KORTENKAMP, D., R.P. BONASSO und R. MURPHY, Hg. *Artificial intelligence and mobile robots. Case studies of successful robot systems*. Menlo Park, Calif.: AAAI Press, 1998. ISBN 0262611376

- [104] MURPHY, R.R. *Introduction to AI Robotics*. Cambridge, Mass.: MIT Press, 2000. Intelligent robotics and autonomous agents. ISBN 0262133830
- [105] ERANN GAT. On Three-Layer Architectures. In: , 1997
- [106] NIEMÜLLER, T., A. FERREIN und G. LAKEMEYER. A Lua-based Behavior Engine for Controlling the Humanoid Robot Nao. In: J. BALTES, M.G. LAGOUDAKIS, T. NARUSE und S.S. GHIDARY, Hg. *RoboCup 2009: Robot Soccer World Cup XIII*. Berlin: Springer, 2010, S. 240-251. ISBN 978-3-642-11875-3
- [107] HAREL, D. Statecharts: a visual formalism for complex systems [online]. *Science of Computer Programming*, 1987, 8(3), 231-274. ISSN 01676423. Verfügbar unter: doi:10.1016/0167-6423(87)90035-9
- [108] WOOLDRIDGE, M. Intelligent Agents: The Key Concepts. In: V. MAŘÍK, O. ŠTĚPÁNKOVÁ, H. KRAUTWURMOVÁ und M. LUCK, Hg. *Multi-Agent Systems and Applications II. 9th ECCAI-ACAI / EASSS 2001, AEMAS 2001, HoloMAS 2001 Selected Revised Papers*. Berlin: Springer, 2002, S. 3-43. ISBN 978-3-540-43377-4
- [109] VDI/VDE. 2653 Blatt 1, *Agentensysteme in der Automatisierungstechnik - Grundlagen*. Berlin: Beuth Verlag GmbH
- [110] MÖNCH, L. *Agentenbasierte Produktionssteuerung komplexer Produktionssysteme*. s.l.: DUV Deutscher Universitäts-Verlag, 2006. ISBN 383500249X
- [111] MURPHY, R.R. *Introduction to AI robotics*. Cambridge, Mass.: MIT Press, 2000. Intelligent robotics and autonomous agents. ISBN 9780262133838
- [112] MARZOUGUI, B. und K. BARKAOUI. Interaction Protocols in Multi-Agent Systems based on Agent Petri Nets Model [online]. *International Journal of Advanced Computer Science and Applications*, 2013, 4(7). ISSN 2158107X. Verfügbar unter: doi:10.14569/I-JACSA.2013.040723
- [113] FIPA. SC00061G, *ACL Message Structure Specification* [Zugriff am: 7. Juni 2018]. Verfügbar unter: <http://www.fipa.org/specs/fipa00061/SC00061G.html>

- [114] BROWN, R.G. *Smoothing, forecasting and prediction of discrete time series*. Mineola, NY: Dover Publ, 2004. ISBN 978-0486495927
- [115] ROMER, M., J. ZEILER, S. MEIßNER und J. FOTTNER. The Internet of Load Carriers - Design of a Cloud-based Service System for Smart and Connected Load Carriers. In: *ICETE 2018. Proceedings of the 15th International Joint Conference on e-Business and Telecommunications : Porto, Portugal, July 26-28, 2018*. Setúbal, Portugal: SCITEPRESS - Science and Technology Publications Lda, 2018, S. 166-173. ISBN 978-989-758-319-3
- [116] ISO/IEC. 2382, *Information technology - Vocabulary*. Berlin: Beuth Verlag GmbH
- [117] HART, P., N. NILSSON und B. RAPHAEL. A Formal Basis for the Heuristic Determination of Minimum Cost Paths [online]. *IEEE Transactions on Systems Science and Cybernetics*, 1968, 4(2), 100-107. ISSN 0536-1567. Verfügbar unter: doi:10.1109/TSSC.1968.300136
- [118] MARDER-EPPSTEIN, E., E. BERGER, T. FOOTE, B. GERKEY und K. KONOLIGE. The Office Marathon: Robust navigation in an indoor office environment. In: *IEEE International Conference on Robotics and Automation (ICRA)*, 2010. 3 - 7 [i.e. 3 - 8] May 2010, Anchorage, Alaska, USA. Piscataway, NJ: IEEE, 2010, S. 300-307. ISBN 978-1-4244-5038-1
- [119] STAUFFER, C. und W.E.L. GRIMSON. Adaptive background mixture models for real-time tracking. In: *1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. June 23-25, 1999 Fort Collins, Colorado*. Los Alamitos, Ca: IEEE Computer Society, 1999, S. 246-252. ISBN 0-7695-0149-4
- [120] MAGEE, D.R. Tracking multiple vehicles using foreground, background and motion models [online]. *Image and Vision Computing*, 2004, 22(2), 143-155. ISSN 02628856. Verfügbar unter: doi:10.1016/S0262-8856(03)00145-8
- [121] PERERA, A.G.A., C. SRINIVAS, A. HOOGS, G. BROOKSBY und W. HU. Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions. In: C. SCHMID, S. SOATTO und C. TOMASI, Hg. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition workshops*.

- CVPR workshops ; New York, New York, USA, 17 - 22 June 2006. Piscataway, NJ: IEEE, 2006, S. 666-673. ISBN 0-7695-2597-0
- [122] ISERMANN, R. *Digitale Regelsysteme*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977. ISBN 978-3-642-96342-1
- [123] DEMPE, S. und H. SCHREIER. *Operations Research. Deterministische Modelle und Methoden*. Wiesbaden: Teubner, 2006. Teubner Studienbücher Wirtschaftsmathematik. ISBN 9783519004486
- [124] BROWN, M. und D.G. LOWE. Automatic Panoramic Image Stitching using Invariant Features [online]. *International Journal of Computer Vision*, 2007, 74(1), 59-73. ISSN 0920-5691. Verfügbar unter: doi:10.1007/s11263-006-0002-3
- [125] BAY, H., A. ESS, T. TUYTELAARS und L. VAN GOOL. Speeded-Up Robust Features (SURF) [online]. *Computer Vision and Image Understanding*, 2008, 110(3), 346-359. ISSN 10773142. Verfügbar unter: doi:10.1016/j.cviu.2007.09.014
- [126] FELLNER, D.W. *Computer-Grafik*. Mannheim: BI-Wissenschaftsverl., 1988. Reihe Informatik. 58. ISBN 978-3411031795
- [127] FAIRCHILD, M.D. *Color appearance models*. Third edition. Chichester, West Sussex: Wiley, 2013. Wiley-IS&T series in imaging science and technology. ISBN 9781118653098
- [128] WANG, B. und S. FAN. An Improved CANNY Edge Detection Algorithm. In: Q. LI, Hg. *Second International Workshop on Computer Science and Engineering*, 2009. WCSE '09 ; 28 - 30 Oct. 2009, Qingdao, China ; proceedings. Piscataway, NJ: IEEE, 2009, S. 497-500. ISBN 978-1-4244-5285-9
- [129] GUPTA, A., R.K. DALAL, R. GUPTA und P. WADHWA. DGW-canny. An improvised version of Canny edge detector. In: *International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, 2011. 7 - 9 Dec. 2011, Chiang Mai, Thailand. Piscataway, NJ: IEEE, 2011, S. 1-6. ISBN 978-1-4577-2166-3
- [130] QUINLAN, S. und O. KHATIB. Elastic bands. Connecting path planning and control. In: *Robotics and Automation, '93 International Conference On*. Los Alamitos: IEEE Computer Society Press, Oct. 1993, S. 802-807. ISBN 0-8186-3450-2

- [131] FOX, D., W. BURGARD und S. THRUN. The dynamic window approach to collision avoidance [online]. *IEEE Robotics & Automation Magazine*, 1997, 4(1), 23-33. ISSN 1070-9932. Verfügbar unter: doi:10.1109/100.580977
- [132] BUSCHHAUS, A. *Hochpräzise adaptive Steuerung und Regelung robotergeführter Prozesse*. Dissertation, 2017. Bericht aus dem Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik. 300. ISBN 9783875254273
- [133] TIMO, D.-I.F., R.-L. MÓNICA, B. CHRISTIAN, M. FRIEDRICH, K. BERND, B. URLICH und S. WALDEMAR. Agent-based communication to map and exchange shop floor data between MES and material flow simulation based on the open standard CMSD [online]. *IFAC-PapersOnLine*, 2016, 49(12), 1526-1531. ISSN 24058963. Verfügbar unter: doi:10.1016/j.ifacol.2016.07.796
- [134] KERWITZ, M. und M. CLASEN. Evaluation von Entwicklungs-umgebungen für Multi-Agenten-Simulationen. In: A. RUCKELSHAUSEN, A. MEYER-AURICH, T. RATH, G. RECKE und B. THEUVSEN, Hg. *Informatik in der Land-, Forst- und Ernährungswirtschaft 2016*. Bonn: Gesellschaft für Informatik e.V, 2016, S. 85-88
- [135] LABITZKE, N. *Wertorientierte Simulation zur taktischen Planung logistischer Prozesse der Stahlherstellung*. Zugl.: Braunschweig, Techn. Univ., Diss., 2010. Wiesbaden: Gabler Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden, 2011. Produktion und Logistik. ISBN 9783834928955
- [136] ARNOLD, D., H. ISERMANN, A. KUHN, H. TEMPELMEIER und K. FURMANS, Hg. *Handbuch Logistik*. 3., neu bearb. Aufl. Berlin: Springer, 2008. VDI-Buch. ISBN 9783540729280
- [137] LIENERT, T. und J. FOTTNER. Routing-based Sequencing Applied to Shuttle Systems. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*: IEEE, 4. November 2018 - 7. November 2018, S. 2949-2954. ISBN 978-1-7281-0321-1
- [138] VDI. 3633, *Simulation von Logistik-, Materialfluss- und Produktionssystemen - Begriffe*. Berlin: Beuth Verlag GmbH
- [139] KUMMER, S., O. GRÜN und W. JAMMERNEGG, Hg. *Grundzüge der Beschaffung, Produktion und Logistik*. 2., aktualisierte Aufl.,

- [Nachdr.]. München: Pearson Studium, 2013. wi - Wirtschaft. ISBN 9783827373519
- [140] KÜHN, W. *Digitale Fabrik. Fabriksimulation für Produktionsplaner*. s.l.: Carl Hanser Fachbuchverlag, 2006. ISBN 3446406190
- [141] KOHL, J. *Automatisierte Datenerfassung für diskret ereignisorientierte Simulationen in der energieflexiblen Fabrik*. Dissertation. Bericht aus dem Lehrstuhl für Fertigungsautomatisierung und Produktionssystematik. 282. ISBN 9783875254068
- [142] BANGSOW, S. *Praxishandbuch Plant Simulation und SimTalk. Anwendung und Programmierung in über 150 Beispiel-Modellen*. [Elektronische Ressource]. München: Hanser Verlag, 2011. ISBN 9783446427822
- [143] HABL, A., O. KIPOURIDIS und J. FOTTNER. Deploying micro-services for a cloud-based design of system-of-systems in intralogistics. In: *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. University of Applied Science Emden/Leer, Emden, Germany, 24-26 July 2017 : proceedings. Piscataway, NJ: IEEE, 2017, S. 861-866. ISBN 978-1-5386-0837-1
- [144] RABE, M., S. SPIEKERMANN und S. WENZEL. *Verifikation und Validierung für die Simulation in Produktion und Logistik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN 978-3-540-35281-5
- [145] *Rules and Regulations - RoboCup Logistics League* [online]. 20 Februar 2019 [Zugriff am: 12. März 2019]. Verfügbar unter: <http://www.robocup-logistics.org/rules>
- [146] *ROS.org | Is ROS For Me?* [online] [Zugriff am: 8. Juni 2018]. Verfügbar unter: <http://www.ros.org/is-ros-for-me/>
- [147] *The 3-Clause BSD License | Open Source Initiative* [online]. 8 Juni 2018 [Zugriff am: 8. Juni 2018]. Verfügbar unter: <https://opensource.org/licenses/BSD-3-Clause>
- [148] *License - OpenCV library* [online]. 12 Mai 2018 [Zugriff am: 8. Juni 2018]. Verfügbar unter: <https://opencv.org/license.html>
- [149] BELLIFEMINE, F., G. CAIRE und D.P.A. GREENWOOD. *Developing multi-agent systems with JADE*. Reprinted. Chichester: Wiley, 2008. Wiley series in agent technology. ISBN 9780470057476

- [150] *Jade Site | Java Agent DEvelopment Framework* [online] [Zugriff am: 8. Juni 2018]. Verfügbar unter: <http://jade.tilab.com/>
- [151] *GNU Lesser General Public License version 3 | Open Source Initiative* [online] [Zugriff am: 8. Juni 2018]. Verfügbar unter: <https://opensource.org/licenses/LGPL-3.0>
- [152] WOOLLEY, B.G. *Unified Behavior Framework for Reactive Robot Control in Real-Time Systems*. AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING AND MANAGEMENT
- [153] JOSEPH, L. *Mastering ROS for robotics programming. Design, build, and simulate complex robots using Robot Operating System and master its out-of-the-box functionalities*. Birmingham, UK: Packt Publishing, 2015. Community experience distilled. ISBN 978-1-78355-179-8
- [154] LAGANIÈRE, R. *OpenCV computer vision application programming cookbook. Over 50 recipes to help you build computer vision applications in C++ using the OpenCV library*. 2nd ed. Birmingham, U.K: Packt Pub, 2014. ISBN 9781782161486
- [155] DIACONESCU, E. und C. SPIRLEANU. Communication solution for industrial control applications with multi-agents using OPC servers. In: M. BROJBOIU, Hg. *2012 International Conference on Applied and Theoretical Electricity (ICATE)*. 25 - 27 Oct. 2012, Craiova, Romania. Piscataway, NJ: IEEE, 2012, S. 1-6. ISBN 978-1-4673-1810-5
- [156] DGUV. 68, *Unfallverhütungsvorschrift Flurförderzeuge* [Zugriff am: 8. Juni 2018]. Verfügbar unter: <http://publikationen.dguv.de/dguv/pdf/10002/vorschrift68.pdf>
- [157] BUSCHHAUS, A., S. KRUSEMARK, E. KARLIDAG und J. FRANKE. Universal fine interpolation algorithms for accuracy improvements of industrial robots. In: *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*: IEEE, 18. Oktober 2016 - 20. Oktober 2016, S. 356-362. ISBN 978-1-4673-8818-4
- [158] GRUNER, S., J. PFROMMER und F. PALM. RESTful Industrial Communication With OPC UA [online]. *IEEE Transactions on Industrial Informatics*, 2016, 12(5), 1832-1841. ISSN 1551-3203. Verfügbar unter: doi:10.1109/TII.2016.2530404

- [159] CINTUGLU, M.H., H. MARTIN und O.A. MOHAMMED. An intelligent multi agent framework for active distribution networks based on IEC 61850 and FIPA standards. In: *2015 18th International Conference on Intelligent System Application to Power Systems (ISAP)*. Piscataway, NJ: IEEE, 2015, S. 1-6. ISBN 978-1-5090-0191-0
- [160] MAHNKE, W., S.-H. LEITNER und M. DAMM. *OPC Unified Architecture*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. ISBN 978-3-540-68898-3

Verzeichnis promotionsbezogener, eigener Publikationen

- [P1] KREITLEIN, S., I. KUPFER, M. SCHOLZ und J. FRANKE. The Least Energy Demand Method as Metric to Describe the Relative Energy Efficiency of a Product based on its Manufacturing. In: J. FRANKE und S. KREITLEIN, Hg. *Energy Efficiency in Strategy of Sustainable Production*. Pfaffikon: Trans Tech Publications Ltd, 2015, S. 19-24
- [P2] KREITLEIN, S., M. SCHOLZ und J. FRANKE. The Automated Evaluation of the Energy Efficiency for Machining Applications based on the Least Energy Demand [online]. *5th Conference on Learning Factories*, 2017, 61, 404-409. ISSN 2212-8271. Verfügbar unter: doi:10.1016/j.procir.2016.11.167
- [P3] KÄSTLE, C., H. FLEISCHMANN, M. SCHOLZ, S. HÄRTER und J. FRANKE. Cyber-Physical Electronics Production. In: S. JESCHKE, C. BRECHER, H. SONG und D.B. RAWAT, Hg. *Industrial Internet of Things. Cybermanufacturing Systems*. Cham: Springer, 2017, S. 47-78. ISBN 978-3-319-42558-0
- [P4] SCHOLZ, M. E|Flow: Energieeffiziente, vielseitige und autonome Transportfahrzeuge für den innerbetrieblichen Materialfluss. In: J. FRANKE, S. KREITLEIN und A. HÖFT, Hg. *Tagungsband zum 2. Green Factory Bavaria Kolloquium 2015. Energieeffiziente Produktion*, 2015, S. 60-66. ISBN 978-3-00-050523-2
- [P5] SCHOLZ, M., T. DONHAUSER und J. FRANKE. Integration eines Multi-Agenten-Systems in eine Ereignisdiskrete Materialflusssimulation. In: T. LOOSE, Hg. *Tagungsband Workshop 2018 ASIM/GI-Fachgruppen. Hochschule Heilbronn, 8. und 9. März 2018*. Wien: ARGESIM Verlag, 2018, 27-32. ISBN 978-3-901608-19-3
- [P6] SCHOLZ, M., S. KREITLEIN und J. FRANKE. E|Flow - Decentralized Computer Architecture and Simulation Models for Sustainable and Resource Efficient Intralogistics. In: J. FRANKE und S. KREITLEIN, Hg. *Energy Efficiency in Strategy of Sustainable Production Vol. II*. Pfaffikon: Trans Tech Publications Ltd, 2017, S. 117-122
- [P7] SCHOLZ, M., S. KREITLEIN, C. LEHMANN, J. BÖHNER, J. FRANKE und R. STEINHILPER. Integrating Intralogistics into

- Resource Efficiency Oriented Learning Factories [online]. *Procedia CIRP*, 2016, 54, 239-244. Verfügbar unter: doi:10.1016/j.procir.2016.05.067
- [P8] SCHOLZ, M., S. OBERSCHACHTSIEK, T. DONHAUSER und J. FRANKE. Software-in-the-loop testbed for multi-agent-systems in a discrete event simulation. Integration of the Java Agent Development Framework into Plant Simulation. In: *2017 IEEE International Symposium on Systems Engineering (ISSE)*, S. 1-6
- [P9] SCHOLZ, M., S. VENKATARAMAN RMASUBRAMANIAN, A. BLANK, S. KREITLEIN und J. FRANKE. E|Flow II - Infrastructural Sensor Concepts to Digitize the Workspace for Sustainable and Resource Efficient Intralogistics [online]. *Applied Mechanics and Materials*, 2017, 871, 97-102. ISSN 1662-7482. Verfügbar unter: doi:10.4028/www.scientific.net/AMM.871.97
- [P10] RAMER, C., J. SEßNER, M. SCHOLZ, Z. XU und J. FRANKE. Fusing low-cost sensor data for localization and mapping of automated guided vehicle fleets in indoor applications. In: IEEE, Hg. *Proceedings of the 2015 IEEE International Conference on Multi-sensor Fusion and Integration for Intelligent Systems (MFI)*, 2015, S. 65-70
- [P11] SCHOLZ, M., J. SEBNER, A. BLANK und J. FRANKE. Implementation of autonomous and semi-autonomous mobile systems to operation rooms: A study of nowadays standardization, general conditions and the concept of a constructive concept [online]. A study of nowadays standardization, general conditions and the concept of a constructive concept, 441-446. Verfügbar unter: doi:10.1109/IECBES.2016.7843489
- [P12] SCHOLZ, M., M. SERNO, J. FRANKE und P. SCHUDERER. A Hybrid Transport Concept for the Material Supply of a Modular Manufacturing Environment. In: *FAIM 2017*, S. 1448-1453
- [P13] SCHOLZ, M., X. ZHANG und J. FRANKE. Implementation of an Intralogistics Routing-Service Basing on Decentralized Workspace Digitization [online]. *Applied Mechanics and Materials*, 2018, 882, 90-95. ISSN 1662-7482. Verfügbar unter: doi:10.4028/www.scientific.net/AMM.882.90
- [P14] SCHOLZ, M., X. ZHANG, S. KREITLEIN und J. FRANKE. Decentralized Intelligence [online]. The Key for an Energy Efficient and Sustainable Intralogistics. *Procedia Manufacturing*, 2018, 21, 679-

685. ISSN 23519789. Verfügbar unter:
doi:10.1016/j.promfg.2018.02.171
- [P15] ZHANG, X., M. SCHOLZ, PARYANTO und J. FRANKE. Adaptive Optimal Multiple Object Tracking Based on Global Cameras for a Decentralized Autonomous Transport System [online]. *Applied Mechanics and Materials*, 2016, 840, 1-7. ISSN 1662-7482. Verfügbar unter: doi:10.4028/www.scientific.net/AMM.840.1
- [P16] SCHOLZ, M. und J. FRANKE. Bitte ein Taxi! Autonomer Materialtransport in der Intralogistik. *Handling*, 2015, (12), 52-53
- [P17] BÖHNER, J., M. SCHOLZ, J. FRANKE und A. SAUER. Integrating digitization technologies into resource efficiency driven industrial learning environments [online]. *Procedia Manufacturing*, 2018, 23, 39-44. ISSN 23519789. Verfügbar unter: doi:10.1016/j.promfg.2018.03.158
- [P18] SCHOLZ, M., M. HUßNÄTTER, S. KREITLEIN und J. FRANKE. E|Flow - From Production Line Concept to a physically and digitally fullmeshed Production Network. In: J. FRANKE und S. KREITLEIN, Hg. *Energy Efficiency in Strategy of Sustainable Production*. Pfaffikon: Trans Tech Publications Ltd, 2015, S. 94-101
- [P19] SCHOLZ, M., S. KOLB, C. KÄSTLE und J. FRANKE. Operation-oriented One-piece-flow Manufacturing. Autonomous and Smart Systems as Enabler for a Full-meshed Production Network. In: *49th CIRP Conference on Manufacturing Systems*, 2016, S. 722-727

Verzeichnis promotionsbezogener studentischer Arbeiten

- [S1] EGEL, E. *Analyse und Bewertung etablierter intralogistischer Materialflusslösungen bezüglich der Herausforderungen und Anforderung aus „Industrie 4.0“*. Projektarbeit. Erlangen, 2014
- [S2] HAUERHOF, A. *Kostenoptimierung im Bereich Intralogistik: Analyse der Integration von Logistikabläufen ortsnaher Zulieferer in interne Logistikstrategien*. Projektarbeit. Erlangen, 2015
- [S3] HUßNÄTTER, M. *Studie zu Fertigungsstrategien der Zukunft vor dem Hintergrund hochflexibler intralogistischer Transportsysteme*. Projektarbeit. Erlangen, 2014
- [S4] LJUBOJA, N. *Studie zu agilen, fraktalen und holonen Fabrikkonzepten und deren Veränderung im Rahmen einer fortschreitenden Digitalisierung*. Masterarbeit. Erlangen, 2018
- [S5] LJUBOJA, N. *Studie zur Veränderung des Geschäftsmodells von Industrieunternehmen induziert durch die Vergabe der Intralogistik an einen Lead Logistics Provider und den Einsatz von autonomen Transportsystemen*. Projektarbeit. Erlangen, 2017
- [S6] OGUZ, S. *Studie zur Auftragsbearbeitung der Zukunft vor dem Hintergrund hochflexibler intralogistischer Transportsysteme*. Bachelorarbeit. Erlangen, 2014
- [S7] PETER, D. *Studie zum Einfluss von hochflexiblen intralogistischen Transportentitäten auf den innerbetrieblichen Materialfluss und der Ausbildung neuer Geschäftsmodelle*. Projektarbeit. Erlangen, 2017
- [S8] ROTH, J. *Studie zur Veränderung der etablierten Fabrikstrukturen und deren Ausprägungsformen vor dem Hintergrund veränderter Kundenbedürfnisse*. Bachelorarbeit. Erlangen, 2018
- [S9] ROTH, S. *Evaluierung möglicher Anwendungsgebiete von Multi-koptern in der Indoorlogistik*. Projektarbeit. Erlangen, 2015
- [S10] ROTH, S. *Studie zur Kombination der innerbetrieblichen Materialfeinversorgung mit autonomen Einheiten unter Verwendung des Routenzugprinzips*. Masterarbeit. Erlangen, 2016
- [S11] EBITSCH, S. *Autonation bei einem hochflexiblen intralogistischen Transportsystem*. Projektarbeit. Erlangen, 2016

- [S12] GMEINER, M. *Konzeptionierung eines idealen autonomen Transportsystems für den Indoor- und Outdoor-Betrieb zur innerbetrieblichen Materialversorgung der BMW Group*. Masterarbeit. Erlangen, 2017
- [S13] HOFMANN, B. *Konzipierung einer zukunftsfähigen Montagestructur am Beispiel der Einbauvorbereitung bei Audi in Ingolstadt*. Masterarbeit. Erlangen, 2017
- [S14] TREMMEL, C. *Studie zum Montagelayout der Zukunft unter dem Einfluss kundenindividueller Produkte und Industrie 4.0*. Projektarbeit. Erlangen, 2014
- [S15] ZUREK, A. *Studie zur Energiespeicherung und -versorgung bei hochflexiblen intralogistischen Transportsystemen*. Bachelorarbeit. Erlangen, 2014
- [S16] BARTH, F. *Studie zum Einfluss hochflexibler intralogistischer Transportentitäten auf das Kennzahlensystem der Logistik und die Ableitung notwendiger Kenngrößen*. Projektarbeit. Erlangen, 2017
- [S17] BUßMANN, M. *Konzeption und Umsetzung einer Auftragsallokations- und Kommunikationsstrategie zwischen Maschinen, Informationssystemen und autonomen Transportentitäten im intralogistischen Bereich*. Masterarbeit. Erlangen, 2016
- [S18] EDMEIER, J. *Anforderungsanalyse und Konzeptionierung der universellen und modularen Schnittstelle eines autonomen Intralogistiksystems zu Produktionsmaschinen*. Masterarbeit. Erlangen, 2016
- [S19] EITH, F. *Konzeptionierung und Implementierung eines Auftragsallokationsalgorithmus für intralogistische Transportroboter auf Basis eines Multi-Agenten-Systems*. Masterarbeit. Erlangen, 2018
- [S20] FOSSLER, M. *Realisierung einer automatischen und dezentralen Auftragsterminierung für ein hochflexibles Intralogistiksystem*. Masterarbeit. Erlangen, 2019
- [S21] HEIDENREICH, R. *Analyse der Übergabepunkte eines Materialflusses von Logistik und Produktion zur Entwicklung einer Methodik der Standardisierung*. Masterarbeit. Erlangen, 2016

- [S22] KRAUS, P. *Entwicklung und Implementierung einer Softwarelayer zur generischen Fähigkeitsbeschreibung von vielseitigen und autonomen Transportfahrzeugen*. Masterarbeit. Erlangen, 2017
- [S23] LIEDEL, J. *Gegenüberstellung, Konzeptionierung und prototypische Umsetzung einer Kommunikationsstruktur zwischen Fertigungsanlagen und autonomen Transportentitäten zur Allokation von Fertigungsaufträgen*. Projektarbeit. Erlangen, 2017
- [S24] TCHOUTA, A. *Umsetzung einer Schnittstelle zwischen einem verteilten, ortsflexiblen Multi-Agenten-Systems zur Behaviour-Engine von vielseitigen autonomen Transportfahrzeugen*. Masterarbeit. Erlangen, 2017
- [S25] ZWINGEL, M. *Konzeptionierung, Implementierung und Validierung einer generischen Fähigkeitsbeschreibung intralogistischer autonomer Transportroboter*. Masterarbeit. Erlangen, 2018
- [S26] ARNET, C. *Optimierung und Integration eines Algorithmus zum Kooperationsbetrieb von autonomen und intelligenten Transportentitäten im intralogistischen Materialfluss*. Bachelorarbeit. Erlangen, 2016
- [S27] BARTSCH, T. *Konzeption und Implementierung einer visuell basierenden Indoor-Ortung für autonome, intralogistische, flurfreie Transportentitäten*. Projektarbeit. Erlangen, 2016
- [S28] BARTSCH, T. *Konzeption, Implementierung und Validierung eines auf Kameras basierenden Indoor-Ortungssystems für autonome Flugroboter*. Masterarbeit. Erlangen, 2016
- [S29] BIGOTT, M. *Konzeptionierung und Implementierung einer optischen Erkennung und Verfolgung dynamischer Hindernisse zur optimierten lokalen Pfadplanung*. Masterarbeit. Erlangen, 2018
- [S30] BIGOTT, M. *Konzeption und Erprobung einer zentralen optischen Lokalisierung zur Generierung eines Weltmodels im Produktionsumfeld*. Bachelorarbeit. Erlangen, 2015
- [S31] BIGOTT, M. *Konzeptionierung und Validierung eines Ansatzes zur Generierung des digitalen Zwillings einer Fertigungsumgebung unter Zuhilfenahme von integrierten optischen Sensoren*. Projektarbeit. Erlangen, 2017

- [S32] COZZIO, C. *Implementierung eines Algorithmus zur Detektion von statischen und dynamischen Hindernissen anhand eines Raspberry Pi*. Bachelorarbeit. Erlangen, 2017
- [S33] DEGEL, M. *Konzeptionierung und Implementierung einer Serviceebene zur Bereitstellung eines digitalen Weltmodells für die globale Bahnplanung eines Fahrerlosen Transportsystems*. Masterarbeit. Erlangen, 2018
- [S34] DEGEL, M. *Konzeptionierung und Implementierung eines globalen Pfadplaners zur Generierung eines digitalen Weltmodells für autonome Transportentitäten auf Basis von Infrastruktursensordaten*. Projektarbeit. Erlangen, 2017
- [S35] DEGEL, MARTIN. *Evaluierung von Bahnplanungsalgorithmen auf Basis eines Weltmodells*. Bachelorarbeit. Erlangen, 2015
- [S36] DIETRICH, F. *Steigerung der Robustheit einer innerbetrieblichen Szenenerkennung durch Deckenkameras gegenüber Störeinflüssen*. Masterarbeit. Erlangen, 2016
- [S37] EDMEIER, J. *Gegenüberstellung und Bewertung von verschiedenen Betrachtungswinkeln zur Szenenerkennung*. Projektarbeit. Erlangen, 2016
- [S38] GIESE, P. *Steigerung der Stabilität des digitalen Abbilds einer Fertigungsumgebung gegenüber Störfaktoren*. Bachelorarbeit. Erlangen, 2017
- [S39] GRÜNDEL, L. *Konzeptionierung und Implementierung einer dezentralen Bahnplanung auf Basis eines globalen multiquery Pfads*. Projektarbeit. Erlangen, 2017
- [S40] LANGENMANTEL, C. *Konzeptionierung und Umsetzung einer webbasierten Benutzerschnittstelle zur Überwachung und Interaktion mit einer servicebasierten, intralogistischen Softwareebene*. Masterarbeit. Erlangen, 2018
- [S41] MEUSEL, K. *Konzeption und Umsetzung eines Kooperationsbetriebs innerhalb eines Schwarmbetriebs autonomer Entitäten*. Bachelorarbeit. Erlangen, 2015
- [S42] SCHEINER, L. *Evaluation der Absolutgenauigkeit einer deckenbasierten Objekterkennung im medizinischen Umfeld*. Bachelorarbeit. Erlangen, 2015

- [S43] SCHILLER, K. von. *Entwicklung und Umsetzung einer kostengünstigen Indoor-Lokalisierung zur initialen Ortung von autonomen Entitäten*. Bachelorarbeit. Erlangen, 2016
- [S44] SONGSONG, P. *Implementierung einer lokalen Pfadplanung für ein autonomes und mobiles Transportsystem*. Masterarbeit. Erlangen, 2018
- [S45] WEINHARDT, J. *Konzeptionierung und Applikation einer laserbasierten on-board Ortung für einen autonomen, intralogistischen Flugroboter und deren aktiv geregelten Lagerung*. Bachelorarbeit. Erlangen, 2016
- [S46] WOLF, T. *Konzeption und Umsetzung einer Indoornavigation für den Einsatz von Multikoptern im intralogistischen Bereich*. Bachelorarbeit. Erlangen, 2016
- [S47] WURM, S. *Studie zur Interaktion und Kommunikation autonomer Einheiten im Schwarmbetrieb*. Projektarbeit. Erlangen, 2014
- [S48] ZHOU, Y. *Konzeption einer dezentralen und kostengünstigen Szenenerkennung zur Digitalisierung eines Arbeitsumfeldes mit Deckenkameras*. Bachelorarbeit. Erlangen, 2016
- [S49] ZWINGEL, M. *Konzeption und Auslegung einer fehlersicheren Kollisionsabsicherung mit Hilfe von kostengünstigen Nahfeldsensoren und Mikrocontrollern*. Bachelorarbeit. Erlangen, 2015
- [S50] DAN, X. *Auswahl und Bewertung von wirtschaftlichen Einsatzmöglichkeiten autonomer Transportfahrzeuge in einer variantenreichen Leiterkartenfertigung*. Masterarbeit. Erlangen, 2018
- [S51] HAASE, E. *Konzeptionierung und Implementierung des Leergut- und Fertigteilmanagements eines intralogistischen Materialflusses mit autonomen, cyberphysischen Transportentitäten*. Projektarbeit. Erlangen, 2016
- [S52] HAASE, E. *Definition und Implementierung des Energie- und Lademangements einer Flotte aus autonomen Transportsystemen in der Intralogistik*. Masterarbeit. Erlangen, 2017
- [S53] HUßNÄTTER, M. *Konzeption und Erprobung einer Materialflusslösung mit autonomen, cyber-physischen Transporteinheiten im Bereich Automotive*. Masterarbeit. Erlangen, 2015

- [S54] JÄHNERT, C. *Konzeptionierung und Simulation des Restmengenmanagement eines intralogistischen Materialflusses mit autonomen, cyber-physischen Transportentitäten*. Projektarbeit. Erlangen, 2016
- [S55] JÄHNERT, C. *Konzeption und Evaluation eines flexiblen, cyber-physisch Fertigungssystems in der mehrstufigen Elektronikproduktion*. Masterarbeit. Erlangen, 2017
- [S56] KASPAR, K. *Methode zum Informationsaustauschs zwischen einer ereignisdiskreten Simulation und einem Multi-Agenten-Systems auf Basis von OPC UA*. Masterarbeit. Erlangen, 2018
- [S57] KOLB, S. *Konzeptionierung und Simulation einer verrichtungsorientierten Elektronikproduktion im One-Piece-Flow*. Masterarbeit. Erlangen, 2016
- [S58] OBERSCHACHTSIEK, S. *Entwicklung und Integration eines Multi-Agenten-Systems in eine ereignisdiskrete Materialflusssimulation*. Masterarbeit. Erlangen, 2016
- [S59] WIRTH, P. *Konzeption und Simulation unterschiedlicher Fertigungsabläufe und Lötovenkonzepte bei einer verrichtungsorientierten Elektronikproduktion*. Bachelorarbeit. Erlangen, 2016
- [S60] WISSMANN, M. *Empirische Ermittlung des Energiebedarfs eines Fahrerlosen Transportfahrzeugs und Einbindung der Ergebnisse in eine Simulation des Transportsystems*. Bachelorarbeit. Erlangen, 2015
- [S61] ACHTZEHN, M. *Konzeptionierung, mechatronischer Aufbau und Erprobung eines prototypischen Flugroboters für die intralogistische Materialversorgung*. Diplomarbeit. Erlangen-Nürnberg, 2016
- [S62] BOLTYSCHEW, V. *Konzipierung, Design und Auslegung einer vertikalen Drehachse für ein kleinskaliges intralogistisches Transportsystem mit einer modularen Maschinenschnittstelle*. Projektarbeit. Erlangen, 2017
- [S63] FENG, X. *Konzeptionierung, Umsetzung und Implementierung einer Visualisierungs-Applikation von intralogistischen Services für Android-Endgeräte*. Masterarbeit. Erlangen, 2018
- [S64] FOSSLER, M. *Konzeption und Umsetzung eines autonomen Materialtransportsystems zur flexiblen Anlagenvernetzung in der Elektronikproduktion*. Projektarbeit. Erlangen, 2018

- [S65] HAUERHOF, A. *Optimierung eines intralogistischen Wareneingangsprozesses für Recyclingmaterial von einem OEM zum Zulieferer*. Masterarbeit. Erlangen, 2015
- [S66] HENGSTMANN, A. *Konzeptionierung, Umsetzung und ökonomische Bewertung einer vollautomatisierten und autonomen Teilebereitstellung im Bereich Automotive*. Masterarbeit. Erlangen, 2017
- [S67] KIEßLICH, S. *Entwicklung, mechatronische Auslegung und Umsetzung eines kostengünstigen 3D Laserscanner zur Kartierung und Ortung auf Basis eines 2D-LiDAR*. Projektarbeit. Erlangen, 2017
- [S68] KIEßLING, H. *Interaktion eines Multi-Agenten-System mit einer ereignisdiskreten Simulation über eine OPC UA Schnittstelle*. Projektarbeit. Erlangen, 2019
- [S69] KONRAD, T. *Implementierung und Umsetzung einer dezentralen Kommunikation zwischen autonomen Entitäten auf Basis von Nahfeldfunktechnologien*. Projektarbeit. Erlangen, 2017
- [S70] KORANCHIE, D. *Integration eines autonomen kommerziellen Transportroboters in eine intralogistische Versuchsumgebung mit Weltmodell- und Pfadplanungsdiensten*. Masterarbeit. Erlangen, 2018
- [S71] KORANCHIE, D. *Mechatronischer Aufbau und Erstinbetriebnahme eines ortsflexiblen Roboters zum Transport von Kleinladungsträgern im Nebenwertstrom*. Projektarbeit. Erlangen, 2017
- [S72] KORTHALS, M. *Konzeptionierung und Implementierung einer Datenschnittstelle zwischen einer intralogistischen Serviceschicht und einer Webanwendung*. Masterarbeit. Erlangen, 2018
- [S73] MALTER, A. *Umsetzung und Implementierung einer Hintergrundsubtraktion und der Bewegungserfassung von Objekten mit Hilfe eines optischen Infrastruktursensors und eines FPGA*. Masterarbeit. Erlangen, 2018
- [S74] RAMASUBRAMANIAN, S.V. *Digitalization of a working environment to detect moving obstacles with embedded infrastructural sensors combined with a FPGA*. Masterarbeit. Erlangen, 2017
- [S75] SCHWEISS, P. *Anforderungsanalyse und Konstruktion eines autonomen, mobilen und vielseitigen Transportfahrzeugs auf Basis eines Werkstückträgers*. Projektarbeit. Erlangen, 2017

- [S76] SEIDL, C. *Konzeptionierung eines fahrerlosen Schleppfahrzeuges für den Indoor- und Outdoor-Betrieb im Bereich Intralogistik*. Projektarbeit. Erlangen, 2015
- [S77] STÄBLER, M. *Entwicklung und Absicherung von Konzepten für einen staplerlosen Rohbau mit einem Fahrerlosen Transportsystemen (FTS)*. Masterarbeit. Erlangen, 2014
- [S78] STROBEL, J. *Planung und Durchführung der mechatronischen Inbetriebnahme einer autonomen Roboterplattform zur variablen Verknüpfung von Produktionsmaschinen in bestehenden Fertigungsanlagen*. Masterarbeit. Erlangen, 2018
- [S79] STROBEL, J. *Anforderungsanalyse und Konstruktion einer mobilen und modularen Roboterplattform für die autonome Bauteilversorgung der Fertigungsanlagen einer hochflexiblen Elektronikproduktion*. Projektarbeit. Erlangen, 2017
- [S80] WALDENFELS, R. von. *Konzeptionierung und Umsetzung einer Ladezustandsüberwachung und Tiefentladungssicherung von Lithium-Ionen-Akkumulatoren von autonomen intralogistischen Transportfahrzeugen*. Projektarbeit. Erlangen, 2018
- [S81] WORACK, M. *Anforderungsanalyse und prototypische Umsetzung einer mobilen Roboterplattform für die automatisierte Versorgung der Bauelemente in der Elektronikproduktion*. Bachelorarbeit. Erlangen, 2016
- [S82] ZWINGEL, M. *Konzeption und mechatronische Implementierung einer aktiv kompensierenden Rotationsachse für ein autonomes Transportfahrzeug*. Projektarbeit. Erlangen, 2017

Reihenübersicht

wird vom Verlag eingefügt