

Entwicklung einer Methode zum Einsatz von Multi-Agent Reinforcement Learning für die virtuelle Verdrahtung im 3D-CAD

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau
des Karlsruher Instituts für Technologie (KIT)

angenommene
Dissertation

von
M. Sc. Tizian Dagner

Tag der mündlichen Prüfung: 10.07.2025

Erste Gutachterin: Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova

Zweite Gutachterin: Prof. Dr.-Ing. Kristin Paetzold-Byhain



Dieses Werk ist lizenziert unter einer Creative Commons Namensnennung -
Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (CC BY-SA 4.0):
<https://creativecommons.org/licenses/by-sa/4.0/deed.de>

Zusammenfassung

Die zunehmende Elektrifizierung und kürzere Lebenszyklen von Produkten erfordern eine höhere Flexibilität in der Produktentwicklung. Unternehmen müssen ihre Planungsprozesse anpassen, um effizienter zu sein, besonders wenn es um die manuelle Definition von Kabelverläufen in begrenzten mechanischen Räumen geht. Zwar unterstützen dreidimensionale Computer-Aided Design (CAD)-Systeme den Prozess, jedoch erfordert die exakte Festlegung von Wegpunkten für elektrische Verbindungen zusätzliche Ressourcen.

Der hohe Freiheitsgrad der virtuellen Verdrahtung im kontinuierlichen dreidimensionalen Raum erschwert jedoch den Einsatz von herkömmlichen Berechnungs- und Automatisierungsmethoden. Zudem führt die iterative Vorgehensweise, kombiniert mit der Einhaltung der vielfältigen Randbedingungen, häufig nur zu begrenzt optimalen Resultaten. Dies wirkt sich vor allem im dynamischen Produktentstehungsprozess aus, der von einer Vielzahl an Iterationszyklen geprägt ist. Eine umfassende Überarbeitung und Anpassung der bestehenden Verdrahtung erfordert angesichts der Komplexität des Optimierungsproblems einen unverhältnismäßigen Zeitaufwand.

Das Ziel der vorliegenden Arbeit besteht darin, eine Methode zu entwickeln, die eine Automatisierung der virtuellen Verdrahtung innerhalb eines 3D-CAD-Systems ermöglicht. Hierzu werden Methoden der Künstlichen Intelligenz (KI) mit Ansätzen zur Kabelbaumerzeugung auf neuartige Weise zusammengeführt. Im initialen methodischen Abschnitt erfolgt die Entwicklung einer Methode zur Konsolidierung der Eingangsinformationen aus einem zweidimensionalen elektrischen und einem dreidimensionalen mechanischen CAD-System. Diese integriert die Zusammenführung der heterogenen Daten und schafft eine konsistente Grundlage für die virtuelle Verdrahtung. Im zweiten Abschnitt der Methode erfolgt die Abstraktion der Problemstellung der virtuellen Verdrahtung zu einem Multi-Agenten-Pfadfindungsproblem, womit die Anwendung der Methoden der KI ermöglicht wird. In einer Zusammenführung der Einzelverbindungen in einem Kabelbaum werden im dritten Abschnitt der Methode unterschiedliche Reinforcement-Learning-Methoden eingesetzt, um die Problemstellung der virtuellen Verdrahtung zu lösen.

Die Validierung der entwickelten Methode wird anhand von drei praxisnahen Verdrahtungsbeispielen durchgeführt. Jede Validierungsinstanz repräsentiert dabei eine andere Anwendungsdomäne. Die Resultate der Experimente zur Kabelbaumgenerierung demonstrieren nicht nur eine signifikante Reduktion der Agentenanzahl und somit der Problemkomplexität, sondern weisen auch auf den hohen Einfluss der Bündelungsstrategien hin. Die Analyse der Anwendung der neuen Multi-Agent Reinforcement Learning-Methoden zeigt, dass die neuen Methoden dazu fähig sind, ganzheitlich optimierte Verdrahtungsergebnisse zu generieren. Diese Resultate sind nicht nur kollisionsfrei, sondern integrieren auch weitere Randbedingungen. Die Kombination der graphbasierten Kabelbaumgenerierung mit den neuen Reinforcement Learning-Methoden stellt somit einen skalierbaren Ansatz im kontinuierlichen Raum des 3D-CAD-Systems dar.

Abstract

Increasing electrification and shorter product lifecycles require greater flexibility in product development. Companies must adapt their design processes to be more efficient, particularly in the manual definition of cable pathways within limited mechanical installation spaces. While 3D Computer-aided design (CAD) systems can assist in this process, the detailed definition of waypoints for electrical connections requires additional time.

However, the high degree of freedom of virtual wiring in continuous three-dimensional space makes it difficult to use conventional calculation and automation methods. In addition, the iterative manual approach, combined with the various boundary conditions, often leads to limited optimal results. This has a particular impact on the dynamic product development process, which is characterized by a high number of iteration cycles. Due to the complexity of the optimization problem, a comprehensive revision and adaptation of the existing wiring requires a disproportionate amount of time.

The aim of this thesis is to develop a method for automating virtual wiring within a 3D CAD system. For this purpose, different methods of artificial intelligence (AI) are combined with approaches of wire harness generation in a novel way. In the first part of the methodology, a method is established to consolidate the input information from a 2D electronic CAD (ECAD) and a 3D mechanical CAD (MCAD) system. This consolidates the heterogeneous data and creates a consistent basis for the virtual wiring harness. In the second part of the method, the virtual wiring problem is abstracted into a multi-agent pathfinding problem to allow the application of AI approaches. In the third part of the methodology, various reinforcement learning methods are used to solve the virtual wiring problem by merging the individual connections in a wire harness.

The validation of the proposed methodology is conducted through an examination of three practical wiring examples, each serving as a representation for distinct application sectors. The results of the wire harness generation experiments not only show a significant reduction in the number of agents and thus in the complexity of the problem, but also point to the significant impact of the bundling strategies. The analysis of the application of the new multi-agent reinforcement learning methods shows that the new methods are able to generate holistically optimized wiring results. These results are not only collision-free, but also integrate additional constraints. The combination of graph-based wire harness generation with the new Reinforcement Learning methods therefore represents a scalable approach in the continuous space of the 3D CAD system.

Schlagworte: *Kabelbaumgenerierung, Multi-Agent Reinforcement Learning, Computer-aided design (CAD)*

Keywords: *Cable Harness Routing Problem, Multi-Agent Reinforcement Learning, Computer-aided design (CAD)*

Danksagung

Die vorliegende Arbeit entstand während meiner Zeit als Doktorand bei der Siemens AG. Der Austausch mit den Kollegen in der Forschungsgruppe *Digital Manufacturing Technologies* und mit den Kollegen in den verschiedenen Werken der Siemens AG hat wesentlich zur Entstehung dieser Arbeit beigetragen. Dafür möchte ich mich an dieser Stelle herzlich bedanken.

Mein besonderer Dank gilt Herrn Dr.-Ing. Peter Robl, dem Leiter der Forschungsgruppe, für das entgegengebrachte Vertrauen und die Möglichkeit, eine Industriepromotion zu verfassen. Die zahlreichen Diskussionen und Anregungen haben meine Perspektive erweitert und mein Wissen vertieft.

Auch bedanke ich mich bei Frau Prof. Dr. Dr.-Ing. Dr. h. c. Jivka Ovtcharova für ihr Interesse und die Übernahme meiner Betreuung. Ihre Expertise und ihre konstruktive Kritik haben mir geholfen, meine Forschungsarbeit zu einem erfolgreichen Abschluss zu bringen.

Des Weiteren danke ich Frau Prof. Dr.-Ing. Kristin Paetzold-Byhain für das Interesse und die Übernahme der Zweitbetreuung.

Abschließend möchte ich meiner Familie meinen aufrichtigen Dank aussprechen. Ihre bedingungslose Unterstützung, Ermutigung und Geduld haben mir die Kraft gegeben, diese anspruchsvolle Zeit zu meistern.

Tizian Dagner
13.05.2025

Inhaltsverzeichnis

Zusammenfassung	i
Abstract	ii
Danksagung	iii
Abbildungsverzeichnis	viii
Tabellenverzeichnis	xi
Abkürzungsverzeichnis	xii
Variablenverzeichnis	xiv
1 Einleitung	1
1.1 Motivation	2
1.2 Problemstellung	3
1.3 Zielsetzung	5
1.4 Aufbau der Arbeit	7
2 Theoretische Grundlagen	8
2.1 Grundlagen der Pfadplanung	9
2.1.1 Formalisierte Definition des Shortest-Path-Problems im dreidimensionalen Raum	9
2.1.2 Erweiterung des Shortest-Path-Problems zu einer Multi-Agenten-Pfadplanung	10
2.1.3 Arten von Kollisionen in der Pfadplanung	12
2.1.4 Erzeugung und Gegenüberstellung von Steiner Tree und Minimum Spanning Tree	13
2.2 Repräsentation und Verwaltung von Produktdaten im Computer-Aided Design (CAD)	15
2.2.1 Basis-Spline-Kurven im 3D-CAD-System	15
2.2.2 Non-Uniform Rational B-Splines (NURBS) im 3D-CAD-System	16
2.2.3 Mesh-Strukturen und Volumenelemente (Voxel) in 3D-CAD-Systemen	17
2.2.4 Quantifizierung geometrischer Ähnlichkeit von dreidimensionalen Geometrien	19
2.2.5 Interdisziplinärer Datenaustausch in der Produktentstehung	20
2.3 Virtuelle Verdrahtung von elektrischen Komponenten im 3D-CAD	22
2.3.1 Intention und Ziele der virtuellen Verdrahtung	22

2.3.2	Aufbau und Inhalte der Eingangsgrößen für die virtuelle Verdrahtung	22
2.3.3	Modellbedingte Einflussfaktoren und Randbedingungen im virtuellen Verdrahtungsprozess	25
2.3.4	Einfluss der Kinematik auf das Kabelrouting	27
2.3.5	Prozessbeschreibung für eine virtuelle Verdrahtung	30
2.3.6	Virtuelle Verdrahtung als 3D-Pfadoptimierung	34
2.4	Grundlagen des maschinellen Lernens und künstlicher neuronaler Netze	35
2.4.1	Definition des Begriffs der künstlichen Intelligenz	35
2.4.2	Aufbau eines künstlichen neuronalen Netzes	36
2.4.3	Methoden des Supervised Learnings	39
2.4.4	Methoden zum Clustern von Datensätzen	40
2.5	Methoden des Reinforcement Learnings	46
2.5.1	Grundlagen und Taxonomie von Reinforcement Learning	46
2.5.2	Trainingsschemata und Kooperationsverhalten im Multi-Agent Reinforcement Learning	49
2.5.3	Kommunikationsmethoden im Multi-Agent Reinforcement Learning	50
2.5.4	Herausforderungen der Methoden des Multi-Agent Reinforcement Learning	52
2.6	Methoden des Transfer Learnings	54
2.6.1	Grundlagen des Transfer Learnings	54
2.6.2	Transfer Learning im Kontext des Reinforcement Learnings	55
2.6.3	Kategorisierung von Transfer Learning-Methoden	56
2.6.4	Intra- und Inter-Agent Transfer Learning	58
2.6.5	Metriken zur Evaluierung des Wissenstransfers	59
2.7	Zusammenfassung der Analyse der theoretischen Grundlagen im Kontext dieser Arbeit	60
3	Stand der Technik in Industrie und Forschung	61
3.1	Virtuelle Verdrahtung im 3D-CAD-System in der Industrie	62
3.2	Stand der Forschung zur Automatisierung der virtuellen Verdrahtung	63
3.2.1	Heuristische und naturinspirierte Methoden zur Pfadoptimierung	64
3.2.2	Lernbasierte Lösungsmethoden zur Pfadoptimierung	65
3.2.3	Repräsentation der dreidimensionalen Geometrie	65
3.3	Abgeleitete Anforderungen an die automatisierte virtuelle Verdrahtung	66
3.4	Fazit aus den Analysen zum aktuellen Stand der Forschung	67
4	Methode zum Einsatz von Multi-Agent Reinforcement Learning für die virtuelle Verdrahtung im 3D-CAD	70
4.1	Anforderungen und Voraussetzungen für die entwickelte Methode zur virtuellen Verdrahtung	71
4.1.1	Einordnung der neuen Methode zur virtuellen Verdrahtung in den Produktentstehungsprozess	71
4.1.2	Anforderungen an die 3D-CAD-Geometrie, die Verbindungsliste und die Kabelstammdaten	73

4.1.3	Abstraktion der virtuellen Verdrahtung als Multi-Agent Pathfin- ding Problem	75
4.1.4	Festlegung der Optimierungsziele	76
4.1.5	Annahmen und Limitationen für die Lösung des 3D-Routingproblems	78
4.2	Datentechnische Integrationsstrategie	79
4.2.1	Zugrunde liegende IT-Systemarchitektur zur Integration der neuen Methoden	79
4.2.2	Zusammenführung von ECAD- und MCAD-Daten mittels Auto- mationML	82
4.2.3	Transformation des 3D-CAD-Modells und domänenspezifische Reduzierung des Zustandsraumes	85
4.2.4	Erweiterung der Routing-Umgebung um Kinematikanforderungen	87
4.2.5	Approximation und Rückführung von Kabelpfaden in das 3D- CAD-System	90
4.3	Virtuelle Verdrahtung als Markov Game	92
4.3.1	Environment Repräsentation der 3D-CAD-Geometrie	92
4.3.2	Repräsentanz als Markov Game	93
4.4	Automatisierte virtuelle Verdrahtung mittels kommunizierenden Multi- Agent Reinforcement Learning	97
4.4.1	Clustering der Einzelverbindungen zur Komplexitätsreduzierung	97
4.4.2	Funktionsweise des Communicating Multi-Agent Twin Delayed DDPG	100
4.4.3	Erweiterung der Observation zur direkten Informationsübertra- gung der Agenten	101
4.4.4	Anwendungsgebiet des kommunizierenden Ansatzes	104
4.5	Skalierbares Multi-Agent Reinforcement Learning mit Transfer Learning	105
4.5.1	Skalierung der Methode auf große Geometrien und hohe Verbin- dungsanzahlen	105
4.5.2	Trainingsframework zur Verbesserung der Generalisierungsfähig- keit auf unterschiedliche Geometrien	111
4.6	Zusammenfassung der neuen Methode zur automatisierten virtuellen Ver- drahtung im 3D-CAD	116
5	Validierung der entwickelten Methoden	117
5.1	Aufbau der Validierung und deren Bewertungskriterien	118
5.2	Definition der Validierungsszenarien und Überführung in Environments	119
5.3	Experimente	124
5.3.1	Erzeugung eines Navigation-Graphs	124
5.3.2	Clustering der Kabelpfade auf Grundlage von Beispielpfaden . .	128
5.3.3	Standardtraining in statischen 3D-CAD-Geometrien	129
5.3.4	Erweiterung der virtuellen Verdrahtung um bewegliche Bauteile	137
5.3.5	Kabelbaumerzeugung und Skalierbarkeit	141
5.3.6	Transfer Learning mit randomisierten Anschlusspunkten	148
5.4	Zusammenfassung und Diskussion der Validierungsergebnisse	155

6	Zusammenfassung und Ausblick	159
6.1	Zusammenfassung der Ergebnisse und Fazit	160
6.2	Ausblick	163
A	Appendix	164
A.1	UML-Sequenzdiagramm	164
A.2	Twin-Delayed Deep Deterministic Policy Gradient	165
A.3	Proximal Policy Optimization	166
A.4	Vergleichsalgorithmen Rapidly-exploring Random Tree und Probablistic Roadmap	166
A.5	Methode des Alpha Wrappings mit Offset	168
A.6	Publikationsliste	169
A.7	Patente	170
	Literatur	172

Abbildungsverzeichnis

1.1	Zusammenhang der drei Hauptprobleme dieser Arbeit	5
1.2	Aufbau der Arbeit	7
2.1	Graphische Darstellung des Shortest-Path-Problems	9
2.2	Gegenüberstellung der Optimierungsziele in der Multi-Agent Pfadplanung	11
2.3	Darstellung von Kollisionstypen in der Multi-Agenten-Pfadplanung . . .	12
2.4	Vergleich zwischen initialem Graphen, Steiner Tree und Minimum Span- ning Tree	14
2.5	Exemplarische Approximierung einer Spline	16
2.6	Darstellung der Transformation eines 3D-CAD-Modells in das Voxel- und das Mesh-Format	17
2.7	Exemplarische Darstellung von kabelspezifischen Einschränkungen in der virtuellen Verdrahtung	26
2.8	Exemplarische Darstellung von möglichen Kollisionsarten in der virtuellen Verdrahtung	27
2.9	Einfluss der Kinematik auf die virtuelle Verdrahtung am Beispiel eines Roboters mit drei Gelenken	28
2.10	Beispiel für die Auswirkung der Kinematik auf die virtuelle Verdrahtung bzw. die Kabellänge	29
2.11	Prozess der virtuellen Verdrahtung im 3D-CAD-System	30
2.12	Beispiele für zwei mögliche Darstellungsformen eines Kabelbaums in der virtuellen Verdrahtung	31
2.13	Arten von Methoden des maschinellen Lernens	35
2.14	Schematische Darstellung zum Aufbau eines Multilayer Perceptron . . .	36
2.15	Prinzipieller Aufbau eines Convolutional Neural Networks	37
2.16	Beispielhafter Aufbau eines Convolutional Filters	38
2.17	Beispielhafte Darstellung der Funktionsweise der MaxPooling-Operation	39
2.18	Agglomeratives und divisives Clustering	42
2.19	Schematische Darstellung von dichte-basierten und partitionierenden Clustering- Methoden	43
2.20	Schematische Darstellung des Markov Decision Process	46
2.21	Gegenüberstellung von Markov Decision Process, Markov Game und Par- tially Observable Markov Game	47
2.22	Trainingsschemata im Multi-Agent Reinforcement Learning	49
2.23	Kommunikationstypen im Multi-Agent Reinforcement Learning	51
2.24	Vergleich zwischen Machine Learning und Transfer Learning	54
2.25	Unterschiedliche Arten von Knowledge-Generalisierung zwischen Tasks	56
2.26	Gegenüberstellung von Intra- und Inter-Agent Transfer Learning	58

3.1	Exemplarische Darstellung der manuellen Pfaddefinition in einem 3D-CAD-System	62
3.2	Exemplarische Darstellung zur Erzeugung eines Routing Networks . . .	63
4.1	Zusammenwirken der entwickelten Methoden mit einem Cable Harness Routing Problem	71
4.2	Erforderliche Qualifizierung eines Anschlusspunktes in einem exemplarischen mechanischen 3D-CAD-Modell	73
4.3	Exemplarische Darstellung der Qualifizierung einer Rotationsbewegung am Beispiel eines Robotergelenks	75
4.4	UML-Klassendiagramm der Methode für die virtuelle Verdrahtung im 3D-CAD-System	80
4.5	Exemplarische Darstellung der AutomationML-Struktur einer Baugruppe	83
4.6	Ableitung des 3D-Routing-Environments aus der 3D-CAD-Geometrie . .	85
4.7	Domänenspezifische Reduzierung des Zustandsraumes als Mesh-Darstellung in zwei Beispielgeometrien	86
4.8	Prozess zur Festlegung von kinematischen Bereichen in einer 3D-CAD-Geometrie	87
4.9	Integration von kinematischen Einflüssen in Bereichen um exemplarische Joints	89
4.10	Schematische Darstellung des Approximationsprozesses für elektrische Verbindungen im 3D-CAD-System	90
4.11	Darstellung eines exemplarischen Environments in drei unterschiedlichen Betrachtungsweisen	92
4.12	Kontinuierlicher Aktionsraum eines Agenten a^i im 3D-Environment . . .	94
4.13	Clustering von Kabelpfaden anhand der räumlichen Similarität in einer Beispielgeometrie	99
4.14	Schaubild eines hybriden Trainingsschemas mit clusterbasiertem CTCE .	100
4.15	Darstellung der Kommunikationseinheit innerhalb eines Clusters aus Agentenperspektive	102
4.16	Beispiel für die Berechnung des erwarteten Richtungsvektors eines Agenten auf Grundlage des Beispielpfades	103
4.17	Grafische Darstellung zur Anwendung der Alpha-Wrap-Methode	106
4.18	Exemplarische Erweiterung der 3D-Mesh-Struktur durch elektrische Anschlusspunkte	106
4.19	Exemplarische Erzeugung einer Subgraphstruktur mit modifizierten Kantengewichten	107
4.20	Exemplarische Berechnung der Bündeldurchmesser in Kabelbäumen . .	109
4.21	Verortung der Kabelbaumerstellung im Prozessablauf der Methode in einer exemplarischen 3D-CAD-Geometrie	111
4.22	Schematische Darstellung der Transfer-Learning-Methode in zwei exemplarischen 3D-CAD-Modellen	112
4.23	Exemplarische Zielpunktrandomisierung in einem Beispiel-Environment	113
4.24	Schematische Darstellung zum Prozess der Quantifizierung der geometrischen Ähnlichkeit von exemplarischen 3D-CAD-Baugruppen	114

5.1	Überblick über die Schritte der Validierung	118
5.2	Validierungsfall 1 zur Verkabelung einer Fahrzeugtür	119
5.3	Environment einer Fahrzeugtür (<i>Validierungsfalls 1</i>)	121
5.4	3D-CAD-Darstellung des Validierungsfalls 2 für einen Wagenkasten eines Zuges	122
5.5	Environment eines Wagenkastens (Validierungsfall 2)	123
5.6	Gegenüberstellung von Navigation-Graphs in Validierungsfall 1 mit unterschiedlichen w_B	125
5.7	Gegenüberstellung von zwei Navigation-Graphs für große Geometrien am Beispiel von Validierungsfall 2	127
5.8	Gegenüberstellung der MARL-Metriken für Validierungsfall 1 ohne Kabelbaumerzeugung	130
5.9	Darstellung der besten Kabelpfade im 3D-CAD-System für Validierungsfall 1 ohne Kabelbaum	132
5.10	Gegenüberstellung der MARL-Metriken für Validierungsfall 2 ohne Kabelbaumerzeugung	134
5.11	Darstellung der identifizierten Kabelpfade im 3D-CAD-System für Validierungsfall 2	136
5.12	Validierungsfall Kinematik für ein Rotationsgelenk	137
5.13	Gegenüberstellung der MARL-Metriken für Validierungsfall Kinematik	138
5.14	Darstellung der elektrischen Verbindungen mittels MARL-Methode für den Validierungsfall Kinematik	140
5.15	Gegenüberstellung der MARL-Metriken für Validierungsfall 1 mit Kabelbaumerzeugung	142
5.16	Darstellung der besten Kabelpfade im 3D-CAD-System für Validierungsfall 1 mit Kabelbaum	144
5.17	Gegenüberstellung der Ergebnisse der MARL-Metriken für Validierungsfall 2 mit Kabelbaumerzeugung	145
5.18	Ergebnisse zur Pfadfindung mit Kabelbaum im 3D-CAD-System für Validierungsfall 2	147
5.19	Gegenüberstellung der Ergebnisse der MARL-Metriken für Validierungsfall 1 mit Zielpunktrandomisierung	149
5.20	Gegenüberstellung der Ergebnisse der MARL-Metriken im Zero-Shot-Transfer in Validierungsfall 1	151
5.21	Gegenüberstellung der Transfer-Learning-Ergebnisse im 3D-CAD-System für Validierungsfall 1 mit Kabelbaum	153
A.1	UML-Sequenzdiagramm für die Anwendung der neuen Methode im 3D-CAD-System	164
A.2	Gegenüberstellung von RRT und PRM in einer exemplarischen 3D-CAD-Umgebung für eine Einzelverbindung	167
A.3	Gegenüberstellung von Meshes auf Basis unterschiedlicher Werte von α_W	168

Tabellenverzeichnis

2.1	Gegenüberstellung von neutralen Datenaustauschformaten	20
2.2	Exemplarische elektrotechnische Eingangsinformationen für den Verdrahtungsprozess	24
3.1	Analyse der Forschungslücke in den relevanten Veröffentlichungen zur virtuellen Verdrahtung	68
4.1	Einflussfaktoren zur Ermittlung des Rewards	95
5.1	Elektrische Anschlusspunkte der Komponenten in der Validierungsgeometrie 1	120
5.2	Auszug der elektrischen Geräte und ihrer Anschlusspunkte für den Validierungsfall 2	122
5.3	Gegenüberstellung der Verbindungslängen im Navigation-Graph für den Validierungsfall 1	126
5.4	Auszug aus der Fréchet-Distanzmatrix für Validierungsfall 1	128
5.5	Numerische Auswertung für ein Standardtraining in Validierungsfall 1	131
5.6	Numerische Auswertung für das Standardtraining in Validierungsfall 2	135
5.7	Numerische Auswertung des Standardtrainings für den Validierungsfall Kinematik	139
5.8	Numerische Auswertung des Standardtrainings für Validierungsfall 1 mit Kabelbaum	143
5.9	Numerische Auswertung im Standardtraining für Validierungsfall 2 mit Kabelbaum	146
5.10	Numerische Auswertung der Ergebnisse des randomisierten Trainings für Validierungsfall 1	150
5.11	Numerische Auswertung der Experimente für Validierungsfall 1 mit Zero-Shot-Transfer	152
A.1	Publikationsliste	169
A.2	Patentanmeldungen	170

Abkürzungsverzeichnis

ACO	Ant Colony Optimization
BMKZ	Betriebsmittelkennziffer
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CoACD	Collision-Aware Approximate Convex Decomposition
COLLADA	COLLABorative Design Activity
CHRP	Cable Harness Routing Problem
CTCE	Centralized Training Centralized Execution
CTDE	Centralized Training Decentralized Execution
DAF	Degree of Action Freedom
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DDPG	Deep Deterministic Policy Gradient
DTDE	Decentralized Training Decentralized Execution
EBOM	Engineering Bill of Materials
ECAD	Electrical CAD
ERP	Enterprise Resource Planning
GJK	Gilbert-Johnson-Keerthi-Distanz-Algorithmus
IGES	Initial Graphics Exchange Specification
JT	Jupiter Tessellation
KI	Künstliche Intelligenz
MAPF	Multi-Agenten Pfadplanung
MARL	Multi-Agent Reinforcement Learning
MBOM	Manufacturing Bill of Materials

MDP	Markov Decision Process
MG	Markov Game
MST	Minimum Spanning Tree
MLP	Multi-Layer Perceptron
ML	Machine Learning
MCAD	Mechanical-CAD
NURBS	Non-Uniform Rational B-Splines
OPTICS	Ordering Points to Identify the Clustering Structure
OBB	Oriented Bounding Box
PDM	Produktdatenmanagement
PMI	Product and Manufacturing Information
POMG	Partially Observable Markov Game
PPO	Proximal Policy Optimization
PRM	Probabilistic Roadmap
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
REST	Representational State Transfer
RRT	Rapidly-exploring Random Tree
SL	Supervised Learning
SPP	Shortest-Path-Problem
STEP	Standard for the Exchange of Product Data
TD3	Twin Delayed DDPG
TL	Transfer Learning
UML	Unified-Modeling-Language
URDF	Universal Robot Data Format
VHACD	Volumetric-Hierarchical Approximate Convex Decomposition

Variablenverzeichnis

A_t^i	Aktion des Agenten i im Zeitschritt t	9
A_{kin}^i	Aktion in der kinematischen Zone	89
$A_{i,j}$	Adjazenzmatrix	114
B_i	Körper i	18
C_n	Cluster n	99
$Conv(f, d, s)$	Convolutional Layer mit f Filtern, d Dimensionen und Stride s	39
D_S	Quelldomäne	54
D_T	Zieldomäne	54
D_{Kabel}	Kabeldurchmesser	25
$D_{i,j}$	Gradmatrix	114
E	Kante in einem Graphen	9
E_α	Kante im Alpha-Wrap-Graph	107
Eps	Anzahl Episoden	129
$Full(n)$	Fully-Connected Layer mit n Neuronen	39
$G'_\alpha(V, E)$	Mittels Anschlusspunkten erweiterter Alpha-Wrap-Graph	107
$G(V, E)$	Graph mit den Knoten V und Kanten E	9
$G_\alpha(V, E)$	Mittels Alpha-Wrap-Methode modifizierter Graph	106
$H \subset G(V, E)$	Kabelbaum als Subset des Graphen $G(V, E)$	77
H_α	Erzeugter Kabelbaum auf dem Alpha-Wrap-Graphen	108
K	Statische Objekte in einem Graphen	9
K_{min}	Minimale Anzahl an Kollisionen in einer Episode	129
L	Laplace-Beltrami-Matrix	114
$L(\tau^i)$	Länge der Trajektorie τ^i	76
M	Dynamische Objekte in einem Graphen	9
M_G	Gesamttransformationsmatrix	30
M_R	Rotationsmatrix	29
M_T	Transformationsmatrix	29
M_{B_i}	Mesh des Körpers i	18
N	Anzahl der Agenten	10
$N_i^k(t)$	Basisfunktion einer B-Spline	15
N_E	Anzahl Agenten auf Edge	109
OOS_{min}	Minimale Anzahl der Überschreitung der zulässigen Oberflächendistanz innerhalb einer Episode	129
$Out(o)$	Output-Layer mit o Neuronen	39
P_S^i	Startposition des Agenten i	9
P_T^i	Zielposition des Agenten i	9
P_B	Verzweigungspunkte Kabelbaum	77

P_i	Kontrollpunkt einer B-Spline	15
R	Rewardfunktion Single Agent Reinforcement Learning	47
R_{total}	Gesamt-Reward einer Episode	49
$T \subset G(V, E)$	Teilbaum des Graphen $G(V, E)$	13
TL_{AP}	Asymptotische Performance Transfer Learning	59
TL_{IP}	Initiale Performance Transfer Learning	59
TL_{TR}	Transfer Ratio Transfer Learning	59
TL_{ToR}	Total Reward Transfer Learning	59
T_S	Lerntask Quelldomäne	54
T_T	Lerntask Zieldomäne	54
V	Knoten in einem Graphen	9
$[x, y, z]$	3D-Koordinate	16
\vec{A}_t	Actionvektor	46
$\vec{\tau}_*^i$	Vorhergesagte Trajektorie des Agenten i	103
\vec{n}	Normalenvektor	88
α	Rotationswinkel	29
α_W	Kontrollparameter α_W zur Definition des Einflussbereiches bei der Erstellung von Wrapper-Meshes	168
β_j	Gelenkvariable	74
δ_W	Offset-Parameter δ_W zur Definition des Abstandes bei der Erstellung von Wrapper-Meshes	168
ϵ	Grenzwert für die Clusterzuweisung	42
γ	Diskontierungsfaktor	47
\hat{A}_t	Advantage Function in einer Proximal Policy Optimization (PPO)	166
λ	Eigenwert	115
\mathcal{A}	Sammlung aller zulässigen Actions	46
$\mathcal{A}^{1...N}$	Gemeinsamer Aktionsraum	47
\mathcal{N}	Punkte eines Clusters	43
O^i	Observation des Agenten i	48
O_{com}	Communication-Observation	102
\mathcal{P}	Partitionen eines Clusters	43
\mathcal{R}	Rewardfunktion Multi-Agent Reinforcement Learning	47
\mathcal{R}_{IQR}	IQR des Rewards in allen Episoden	129
\mathcal{R}_{max}	Maximaler Reward einer Episode	129
\mathcal{R}_{min}	Minimaler Reward einer Episode	129
\mathcal{S}	Sammlung aller zulässigen States	46
\mathcal{T}	Transitionfunktion	46
π	Policy	47
$\pi(S, A)$	Ausführungswahrscheinlichkeit der Action A im State S	47
π^*	Optimale Policy	58
π_{C_n}	Policy des Clusters C_n	100
$\tau = \{\tau^1, \dots, \tau^N\}$	Multi-Agenten Gesamttrajektorie	10
τ^i	Trajektorie des Agenten i	9
θ	Parameterset eines künstlichen neuronalen Netzes	36

$\{j\}$	Koordinatensystem eines Joints	75
$\{l\}$	Koordinatensystem eines Links	74
$a = \{a^1, \dots, a^N\}$	Menge der Agenten	10
a^i	Agent i	9
b_{Kabel}	Biegeparameter eines Kabels	25
d	3D-Translationsvektor	30
d_B	Durchmesser eines Astes im Kabelbaum	109
$d_E(P, Q)$	Euklidische Distanz zwischen den Punkten P und Q	41
$d_F(A, B)$	Fréchet-Distanz zwischen den Kurven A und B	41
d_{avg}	Durchschnittlicher Kabeldurchmesser	109
d_{max}	Maximaler Oberflächenabstand	78
d_{ref}	Abstand der Ebenen zur Begrenzung des kinematischen Bereichs um einen Joint	88
f^d	Zielfunktion eines neuronalen Netzes	39
$f_T(\bullet)$	Lerntask Quelldomäne	54
j	Joint	74
k	Ordnung einer B-Spline	15
l	Link	74
$minPts$	Mindestanzahl Punkte in einem Cluster	44
r_i	Radius eines Kabels	109
r_{min}	Minimaler Biegeradius	25
s	Glättungsfaktor einer Kurve	90
t	Zeitschritt	10
t_x, t_y, t_z	Verschiebewerte nach Achsen in M_T	29
v	Eigenvektor	115
$w(u, v)$	Gewicht einer Kante in einem Graphen	9
w_B	Bündelungsfaktor für die Kabelbaumerzeugung	107
w_i	nicht-negatives Gewicht eines Punktes	16

1 Einleitung

In der vorliegenden Arbeit wird eine Methode zum Einsatz von Multi-Agent Reinforcement Learning (MARL) für die virtuelle Verdrahtung im 3D-CAD entwickelt. Das einleitende Kapitel legt die Motivation in Kapitel 1.1 dar. In den Kapiteln 1.2 und 1.3 wird darauf aufbauend die Problemstellung der zugehörigen Zielsetzung gegenübergestellt. Der Aufbau der Dissertation ist in Kapitel 1.4 erläutert.

1.1 Motivation

Eine Vielzahl von Branchen sieht sich mit der Herausforderung einer zunehmenden Variantenvielfalt und kundenseitig geforderten Produktindividualisierungen konfrontiert [104]. Die zeitgleich sinkenden Losgrößen und innovationsgetrieben kürzeren Produktlebenszyklen erhöhen unweigerlich den Druck auf produzierende Unternehmen, ihre Flexibilität zu steigern. Durch eine Internationalisierung der Märkte müssen in diesem Kontext auch länderspezifische Produktanforderungen berücksichtigt werden [118]. Diese Faktoren resultieren in einem erhöhten Zeitaufwand im Produktentwicklungsprozess, um dem Wandel von der Massenfertigung zur Fertigung mit kleinen Losgrößen gerecht zu werden. Folglich sind neuartige Lösungsmethoden für die Bewältigung des Spektrums an Produktvarianten erforderlich [150]. Der parallel verlaufende Trend zur Automatisierung und die zunehmende Elektrifizierung von Produkten stehen dabei unter wechselseitiger Einflussnahme zur Variantenvielfalt [179, 118, 171]. So weist die exemplarisch aufgeführte Automobilindustrie immer mehr neue, intelligente Anwendungen, wie Assistenzsysteme oder integrierte Unterhaltungsgeräte, auf. Für die notwendige Datenkommunikation sind zusätzliche Sensoren, Mikroprozessoren und andere elektronische Komponenten im Produkt notwendig. Um diese Komponenten miteinander verbinden zu können, ist eine größere Anzahl von Kabeln erforderlich, die in den meist begrenzten mechanischen Bauräumen Platz finden muss. Neben topologischen Optimierungen ist die Berücksichtigung von optimalen Kabelpfaden, zulässigen Füllgraden in kabelführenden Kanälen und weiteren Randbedingungen essenziell für valide Ergebnisse der Verdrahtung [37]. Dabei ist zu beachten, dass die Komplexität der virtuellen Verdrahtung mit zunehmender Anzahl an Kabeln in einem begrenzten Bauraum überproportional steigt [187, 91]. Eine zentrale Anforderung, die sich aus diesen Trends ergibt, ist die Entwicklung von Methoden zur schnellen und einfachen Planung von Kabelpfaden innerhalb des Produktentstehungsprozesses. Dies geschieht heute entweder auf Basis der physikalischen Planung in Prototypen oder virtuell mit Hilfe von 3D-CAD-Systemen. Insbesondere die virtuelle Verdrahtung bietet eine Reihe von Vorteilen. Direktes Feedback und mögliche Anpassungen ohne Zeitverzögerung sind hier beispielhaft zu nennen [140]. Da es derzeit keine industrietauglichen automatisierten Lösungen gibt, die eine virtuelle Kabelverlegung realitätsgetreu abbilden können, handelt es sich zum heutigen Zeitpunkt um ein rein manuelles Verfahren.

Das dabei auftretende Problem der Festlegung von Kabelwegen im dreidimensionalen Raum ist als Optimierungsproblem einzuordnen [209]. Zur effizienten Lösung von Optimierungsproblemen kommen im industriellen Umfeld vermehrt intelligente mathematische Algorithmen zum Einsatz [104]. Die Methoden der Künstliche Intelligenz (KI) haben in den letzten Jahren durch die steigenden Rechenkapazitäten große Verbesserungen erfahren und sind mittlerweile in zahlreichen Anwendungen den menschlichen Fähigkeiten sogar überlegen [157]. Nicht erst seit der medialen Präsenz von einfach nutzbaren KI-Methoden wie ChatGPT besteht auch in der Industrie der Wunsch nach einem breiteren Einsatz intelligenter IT-Lösungen [127]. Ziele sind dabei zum einen die Übernahme repetitiver Tätigkeiten durch KI-Methoden, zum anderen die kontinuierliche Verbesserung der Ergebnisqualität [140]. Die vorhandenen Algorithmen ermöglichen eine ganzheitliche Betrachtung der vorliegenden Problemstellung, was bei der Lösung von Optimierungsproblemen als wesentlicher Vorteil einzuordnen ist. Dabei sind KI-Methoden in der Lage,

mehr Daten mit höherer Geschwindigkeit zu verarbeiten. Es können somit Muster und Nichtlinearitäten aufgedeckt werden, die von Menschen kaum wahrnehmbar sind [144]. Intelligente Algorithmen werden zunehmend auch im industriellen Kontext eingesetzt, um beispielsweise unübersichtliche Störsituationen in der Fertigung ganzheitlich optimal lösen zu können [104]. Die Anwendungen von KI-Methoden, die in eine CAD-Software integriert sind, befinden sich hingegen weiterhin in einem frühen Entwicklungsstadium [140]. Der Fokus liegt primär auf intelligentem Design durch Ähnlichkeits- und Feature-analytik, die den Gestaltungsprozess effizienter und damit ökonomischer macht [156]. Für die Automatisierung der 3D-Kabelverlegung bietet die Nutzung dieser Methoden daher ein großes Potenzial [209].

1.2 Problemstellung

In der virtuellen Verdrahtung von mechanischen Komponenten besteht in gängigen Mechanical-CAD (MCAD)-Systemen bereits heute die Möglichkeit, Kabelpfade im dreidimensionalen Raum festzulegen. Diese Funktionen stützen sich auf die Existenz einer ausreichend qualifizierten dreidimensionalen Geometrie sowie auf verfügbare Informationen zur Schaltplanung und zu den Kabelstammdaten. Zur Festlegung der Kabelpfade ist eine Abarbeitung der Verbindungsliste erforderlich. Die Bestimmung einzelner Wegstreckenpunkte jedes Kabels ist jedoch mit einem hohen zeitlichen und manuellen Aufwand verbunden [170, 209]. Das zugrunde liegende Optimierungsproblem ist als Cable Harness Routing Problem (CHRP) definiert [30]. Für jede elektrische Verbindung sind die einzelnen Koordinaten mit Ausprägung in allen drei Achsen festzulegen, durch welche der Kabelpfad eindeutig bestimmt ist. Im Falle von komplexen Geometrien mit einer Vielzahl an Kabeln und langen zu überbrückenden Distanzen steigt die dafür erforderliche Zeit überproportional [187]. Die Abarbeitung der Verbindungsliste erfolgt dabei iterativ. Es werden die zu planenden Pfade auf Basis der Kabeldicke oder der Kritikalität der Kabel für die Funktionsweise des Endproduktes priorisiert und sortiert. Eine nachfolgende Einzelbetrachtung der Wegstrecke mit zugehörigen Randbedingungen ist jedoch aufgrund der hohen Komplexität und der potenziell auftretenden Wechselwirkungen zu anderen Kabeln anfällig für lokale Optima. Zwar wird für die einzelne Verbindung mitunter ein idealer Pfad identifiziert, die Auswirkungen auf die Optimalität der anderen Kabelwege sind dabei aufgrund der Problemkomplexität allerdings von Menschen kaum zu überblicken. So besteht das Problem, dass die zyklische Abarbeitung der Verbindungsliste in der Mehrzahl aller Fälle lediglich beschränkt optimale Lösungen erreicht [28, 209].

Als datentechnische Grundlage für die virtuelle Verdrahtung dienen neben der mechanischen Geometrieprepräsentanz aus einem MCAD-System insbesondere die elektrischen Planungsdaten aus einem Electrical CAD (ECAD) [162]. Diese beinhalten einerseits Informationen über die jeweils erforderlichen elektrischen Verbindungen, andererseits die Anforderungen an den zu verwendenden Kabeltyp [7, 118]. Im Entwicklungsprozess der virtuellen Verdrahtung treffen somit eine bauräumorientierte mechanische 3D-Entwicklung und eine funktionsorientierte elektrische 2D-Entwicklung aufeinander [118, 65]. Neben der differierenden Beschreibungsdimensionalität lassen sich dabei hohe Anteile an Medienbrüchen zwischen den disziplinspezifischen IT-Systemen identifizieren [41]. Eine asynchrone

Entwicklung von elektrischen Funktionen im ECAD und mechanischen Komponenten im MCAD ist der Kern des Problems einer begrenzten Nachverfolgbarkeit von Änderungen in den jeweils konträren CAD-Systemen. In Ermangelung einer kontinuierlichen Vernetzung zwischen den domänenspezifischen Prozessen kommt es bei Änderungen zu manuellen Nachpflegeaufwänden. Die Folgen sind Mehraufwände durch Planungsabweichungen und auftretende Inkonsistenzen [60, 7]. Die Tatsache, dass es keinen *Single Point of Truth* als datentechnische Ausgangsbasis gibt, hat bisherige Entwicklungen automatisierter Lösungsansätze gehemmt. Eine Vielzahl von Softwareschnittstellen ist notwendig, um die heterogene IT-Landschaft ausreichend zu vernetzen. Dies begünstigt die unzureichende Verfügbarkeit vollständig definierter Randbedingungen als Einflussfaktor auf den Verdrahtungsprozess. Insbesondere die auftretenden Wechselwirkungen der einzelnen Anforderungen erhöhen die Fehleranfälligkeit des Planungsprozesses erheblich [209].

Das Produktportfolio der meisten produzierenden Unternehmen ist durch eine Vielzahl geometrisch ähnlicher Varianten gekennzeichnet [150]. Diese Ähnlichkeit spiegelt sich jedoch nur bedingt in einer Übertragung von Wissen und Erkenntnissen aus vorangegangenen Kabelpfadplanungen in Baugruppen wider. Zwar erfolgt je nach Branche und Stückzahl die Entwicklung eines übergreifenden Kabelbaums als Schnittmenge aller Varianten, der an die funktionalen Besonderheiten der jeweiligen Variante angepasst wird. Ein variantenübergreifender Wissenstransfer ist jedoch nicht zu beobachten [42, 43]. Vielmehr unterliegt das Kabellayout im dreidimensionalen Raum einer variantenspezifischen Individualisierung [7]. Die dabei fehlende Übertragung bereits identifizierter Lösungsmöglichkeiten des CHRP auf kongruente Baugruppen führt dazu, dass das Potenzial von Synergien, Kabelwege ähnlich zu planen, nicht genutzt wird. Stattdessen wird aufgrund des hohen manuellen Anpassungsaufwands eine Kabelpfadplanung meist von Grund auf neu initiiert [44]. Ähnliche Probleme sind auch bei mechanischen Anpassungen im Produktentstehungsprozess zu beobachten. Die Wiederverwendung und Weiternutzung gesammelter Erfahrungswerte hinsichtlich optimaler Lösungen geschieht folglich personenbezogen und damit nur eingeschränkt.

In der Zusammenfassung der dargelegten Problematiken wird in Problem 1 die unzureichende Lösungsfähigkeit der bestehenden Methoden in Bezug auf dreidimensionale Geometrien beschrieben. Die ausbleibende Datenkonsolidierung der disziplinspezifischen CAD-Systeme und die daraus resultierende Dateninkonsistenz stellen das Problem 2 dar. Problem 3 besteht darin, dass ein Transfer des Wissens innerhalb von Modifikationen eines einzelnen CHRP, aber auch mit der Übertragung der Erkenntnisse zwischen unterschiedlichen CHRP nicht stattfindet. Abbildung 1.1 veranschaulicht den Zusammenhang der beschriebenen Problemstellungen dieser Arbeit.

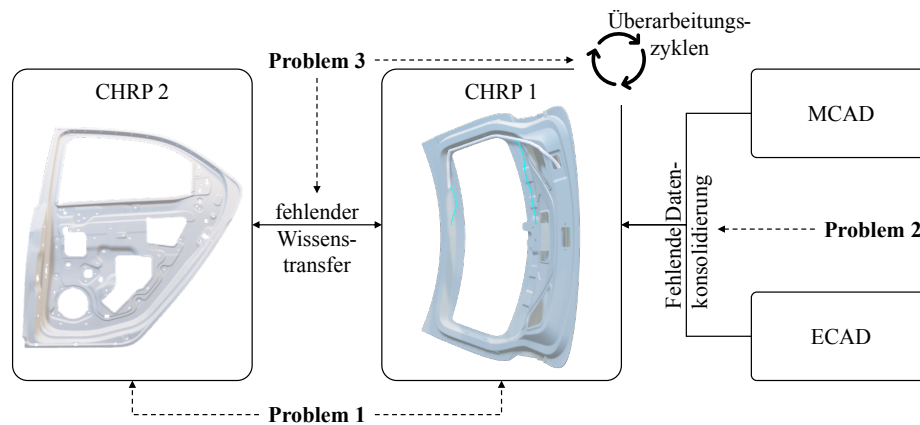


Abbildung 1.1: Zusammenhang der drei Hauptprobleme dieser Arbeit

1.3 Zielsetzung

Gegenstand dieses Abschnitts ist die Darstellung der aus der Problemstellung in Kapitel 1.2 abgeleiteten Forschungsziele. Es werden assoziierte Ziele formuliert, die im Verlauf der Validierung (vgl. Kapitel 5.4) zu überprüfen sind. Diese dienen im Fazit (vgl. Kapitel 6.1) als Grundlage für die Bewertung, ob die gesetzten Ziele in angemessenem Maße erreicht werden.

Das vorrangige Ziel dieser Arbeit ist es, für das CHRP innerhalb eines dreidimensionalen CAD-Raumes optimale oder nahezu optimale Lösungen herbeizuführen. Dieses Optimum ist hier definiert als der kürzeste Summenweg aller zu planenden Einzelverbindungen unter Berücksichtigung der jeweiligen Randbedingungen. Dazu ist der derzeitige manuelle und zeitintensive Pfadplanungsprozess durch einen Automatismus zu substituieren. Die zu entwickelnde Methode muss die Pfadoptimierung ganzheitlich betrachten und dadurch eine global optimale Lösung im vorliegenden CHRP erzeugen. Dafür müssen neben den kabel- auch die umgebungsspezifischen Einschränkungen in die Ergebniserzeugung einfließen.

Hypothese 1: Unter Zuhilfenahme von KI-Methoden ist es möglich, das CHRP in Geometrien unterschiedlicher Komplexitätsstufen zu lösen. Damit substituiert es die manuelle, zeitintensive Arbeitsweise und erreicht durch eine ganzheitliche Problembetrachtung ein globales Optimum bzw. kann eine Lösung erzeugen, die nahe am Optimum liegt.

Das Ziel der vorliegenden Untersuchung ist die Entwicklung einer Methode zur Erzeugung eines durchgängigen Single Point of Truth, der durch die Konsolidierung disziplinspezifischer Informationen eine konsistente Planung der Kabelpfade im virtuellen 3D-CAD-Modell gewährleistet. Dabei ist es ein Subziel, dass eine kontinuierliche Zusammenführung der disziplinspezifischen Ansichten und Anforderungen gewährleistet ist. Somit sollen alle relevanten Eingangsgrößen wie auch die zu berücksichtigenden Randbedingungen für das Lösen eines CHRP zur Verfügung stehen. Hierbei sollen bestehende disziplinspezifische Austauschformate nicht substituiert, sondern es soll ein disziplinübergreifender Kommunikationsstandard etabliert werden. Die Lösung soll sowohl 3D-Daten aus dem MCAD-Umfeld inkludieren als auch die assoziierten strukturierten und tabellarischen Verdrahtungsinformationen anbinden. Dies verfolgt im Kern das Ziel

der Zusammenführung der ECAD- und MCAD-Daten zur Reduzierung der Heterogenität der CAD-Systemlandschaft [37]. Im Zusammenhang damit ist es das Ziel, die Harmonisierung der Datenbasis zur Ermöglichung der automatisierten virtuellen Verdrahtung zu erreichen.

Hypothese 2: Mittels Konsolidierung der mechanischen und elektrischen Daten in einem Single Point of Truth wird ein einheitlicher Datenausgangspunkt geschaffen, auf dem eine KI-Methode aufsetzen kann. Damit sollen sich vielfältige Randbedingungen ohne starres Regelwerk in das CHRP nahtlos integrieren lassen.

Die entwickelte KI-Methode muss befähigt werden, auch Abwandlungen der 3D-Geometrien des Trainingsprozesses sicher zu lösen. Auf diese Weise soll es einerseits möglich sein, das bereits erzeugte Wissen zu transferieren, andererseits aber auch eine variantenübergreifende Standardisierung zu verwirklichen. Die angestrebte Generalisierungsfähigkeit der Methode soll durch Adaption an die neue Umgebung optimale Ergebnisse im Sinne des CHRP innerhalb der definierten Grenzen generieren [75]. Das analoge Ziel gilt für eine nahezu optimale Lösung auch außerhalb dieser Limitationen. Bestehende bekannte Beschränkungen der Generalisierungsfähigkeit von KI-Methoden sollen mit entsprechenden Trainingsschemata überwunden werden. Die damit entstehende Robustheit der Methode gegenüber Veränderungen innerhalb des Produktentstehungsprozesses soll infolgedessen einen vielseitigen und flexiblen Einsatz ermöglichen, der nicht auf den Einsatz in Einzelprodukten limitiert ist.

Hypothese 3: Mit einer dafür konzipierten Trainingsmethode ist es möglich, den Wissenstransfer von KI-Methoden so zu verbessern, dass die bereits erlernten Fähigkeiten gezielt auf ähnliche Geometrien übertragen werden können.

1.4 Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in sechs Kapitel, deren Abfolge und deren Inhalte im Folgenden detailliert werden. Abbildung 1.2 zeigt die Grundstruktur dieser Arbeit.

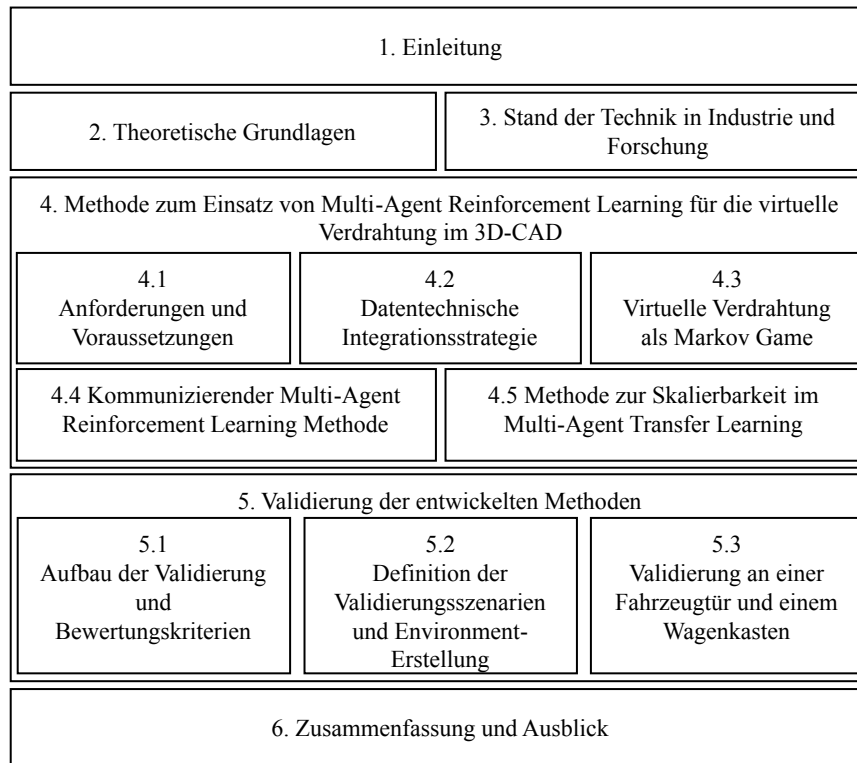


Abbildung 1.2: Aufbau der Arbeit

Die beiden Kapitel 2 und 3 (*Theoretische Grundlagen und Stand der Technik in Industrie und Forschung*) bilden die Basis für die in dieser Arbeit zu entwickelnde Methode. Eingangs zeigt Kapitel 2 die Grundlagen der Pfadoptimierung sowie die derzeitigen Arbeitsmethoden in der virtuellen Verdrahtung von elektrischen Komponenten im 3D-CAD-System. Ausführungen zum maschinellen Lernen komplettieren das Kapitel. In Kapitel 3 (*Stand der Technik in Industrie und Forschung*) sind relevante Ansätze des maschinellen Lernens aufgeführt, die in der Industrie für die Pfadoptimierung im 3D-CAD bereits Anwendung finden. Diese werden um Methoden zum Transfer Learning (TL) zur Übertragung von Wissen auf ähnliche Problemstellungen ergänzt. In Kapitel 4 (*Methode zum Einsatz von Multi-Agent Reinforcement Learning für die virtuelle Verdrahtung im 3D-CAD*) wird eine neue Methode entwickelt, beginnend mit den notwendigen Anforderungen und Voraussetzungen in Kapitel 4.1. In den Unterkapiteln 4.4 bis 4.5 werden die zugrunde liegenden Untermethoden im Kontext der Pfadoptimierung im dreidimensionalen Raum ausgearbeitet. Kapitel 5 (*Validierung*) zeigt die Validierung der entwickelten Methode anhand von zwei unterschiedlichen Geometriebeispielen. Die Arbeit schließt mit Kapitel 6 (*Zusammenfassung und Ausblick*) ab.

2 Theoretische Grundlagen

In diesem Kapitel werden die notwendigen Grundlagen für die im weiteren Verlauf der Arbeit zu entwickelnde Methodik analysiert. Abschnitt 2.1 betrachtet das abstrahierte Problem der Pfadoptimierung für die Kabelwege. Die Abschnitte 2.2 und 2.3 untersuchen die 3D-modellbasierte Arbeitsweise und die zugehörige virtuelle Verdrahtung. In den Abschnitten 2.4 bis 2.6 sind die relevanten Aspekte bestehender KI-Methoden detailliert analysiert. Die Zusammenfassung der notwendigen theoretischen Grundlagen für diese Dissertation befindet sich im abschließenden Abschnitt 2.7.

2.1 Grundlagen der Pfadplanung

Das nachfolgende Kapitel stellt die Grundlagen der Pfadplanung und -optimierung für die Kabelwege vor. Dafür wird in Abschnitt 2.1.1 auf die formalisierte Beschreibung des Problems im dreidimensionalen Raum eingegangen. In Abschnitt 2.1.2 folgt darauf aufbauend die Erweiterung der Problemstellung zur Multi-Agenten-Pfadplanung. Die auftretenden Arten von Kollisionen in diesem Szenario werden in Abschnitt 2.1.3 konkretisiert. Damit wird der Komplexitätsgrad der virtuellen Verdrahtung in Kapitel 2.3 verdeutlicht. Abschnitt 2.1.4 zeigt, wie Teilbäume in einem Graphen berechnet werden können.

2.1.1 Formalisierte Definition des Shortest-Path-Problems im dreidimensionalen Raum

Das Shortest-Path-Problem (SPP) ist ein grundlegendes, NP-schweres Optimierungsproblem in der Informatik und Mathematik [18]. Es bezieht sich auf die Suche nach dem kürzesten Weg zwischen zwei Punkten in einem gewichteten Graphen. Der Graph besteht aus Knoten V (en: *Vertices*) und Kanten E (en: *Edges*). Jedem Kantensegment ist ein numerischer Wert zugeordnet, oft als Gewicht $w(u, v)$ (en: *Weights*) bezeichnet. Dieses Gewicht repräsentiert in der Regel die Kosten, die Entfernung oder die Zeit, die benötigt wird, um von einem Knoten zu einem anderen zu gelangen. Diese Gewichte sind innerhalb des Optimierungsproblems des SPP zu minimieren [168].

Formal ist ein ungerichteter Graph $G(V, E)$ gegeben, wobei V die Menge der Knoten und E die Menge der Kanten ist [49, 152]. Jeder Kante wird ein Gewicht $w(u, v)$ zugewiesen [155]. Die Umgebung kann zudem M dynamische und K statische Objekte enthalten [28]. Das Ziel besteht darin, eine Folge von Knoten innerhalb des Graphen zu identifizieren, die den kürzesten Weg zwischen dem Startknoten P_S^i und dem Zielknoten P_T^i darstellt. In diesem wird die Gesamtsumme der Gewichte der Kanten minimiert [67]. Abbildung 2.1 veranschaulicht den Zusammenhang der Knoten innerhalb eines simplifizierten SPP.

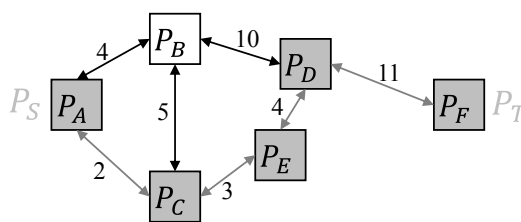


Abbildung 2.1: Graphische Darstellung des Shortest-Path-Problems

Der Knoten P_A (vgl. Abbildung 2.1) ist der Startpunkt P_S und P_F der Zielpunkt P_T des Optimierungsproblems im Graphen $G(V, E)$. Die Zahlen (siehe Abbildung 2.1) an den bidirektionalen Kanten zwischen den jeweiligen Knoten repräsentieren die assoziierten Gewichte $w(u, v)$. Der grau hinterlegte Pfad in Abbildung 2.1 deutet den kürzesten Pfad auf Basis der Kantengewichte an. Die Abfolge der einzelnen Aktionen A_t^i des individuellen Agenten a^i ist in der Gesamttrajektorie τ^i als Sequenz abgebildet. Diese beginnt mit der Startposition P_S und endet folglich mit der Zielposition P_T . Als Optimierungsziel des SPP

ist die Minimierung der erforderlichen Gesamtsumme der Kantengewichte zum Erreichen der Zielposition P_T festgelegt. [160]

2.1.2 Erweiterung des Shortest-Path-Problems zu einer Multi-Agenten-Pfadplanung

Das als Multi-Agenten Pfadplanung (MAPF) (en: *Multi-Agent Pathfinding*) deklarierte Optimierungsproblem erweitert das SPP aus Abschnitt 2.1.1 um weitere Agenten für das Szenario $N > 1$ [155, 109, 160]. In diskretisierten Zeitschritten t kann dabei jeder Agent a^i eine einzelne Aktion A^i in der Umgebung $G(V, E)$ durchführen [160]. Die Existenz weiterer Agenten erhöht somit die Komplexität des Problems, da nicht nur der Zustandsraum exponentiell mit der Anzahl der Agenten wächst, sondern auch ein hindernisfreier Pfad identifiziert werden muss, auf den das Verhalten der anderen Agenten Einfluss nimmt [49, 168].

1. Formalisierte Beschreibung der Multi-Agenten-Pfadplanung

Die MAPF besteht aus einer Menge von Agenten (hier die zu verlegenden Kabel), deren Aufgabe es ist, gemeinsam einen hindernisfreien Pfad in einer definierten Umgebung zu finden. Formalisiert ist das Problem durch eine Menge von N Agenten und K Hindernissen. [160] Entsprechend ist die Menge der Agenten festgelegt als $a = \{a^1, \dots, a^N\}$ [49, 109]. Analog zum SPP ist $G(V, E)$ der Graph, in dem sich die Hindernisse K befinden und somit die Pfadplanung der einzelnen Agenten einschränken. Jeder Agent $a^i \in a$ ist durch das Tupel $\langle P_S^i, P_T^i \rangle$ definiert, in dem die individuellen Start- und Zielpunkte je Agent enthalten sind [109]. Die bidirektionalen Kanten des Graphen können von jedem Hindernis und jedem Agenten belegt werden. Die Bewegung eines Agenten erfolgt durch die unabhängige Aktion A_t^i , die jeder Agent a^i innerhalb eines Zeitschrittes t ausführt. Existiert eine gemeinsame Folge von Aktionen $\tau = \{\tau^1, \dots, \tau^N\}$, die weder Kollisionen noch Deadlock-Szenarien aufweisen, so ist diese als eine gültige Lösung des MAPF einzuordnen [67, 92]. Dies schließt eine Kollisionsfreiheit sowohl auf den Knoten als auch auf den Kanten ein (vgl. Kapitel 2.1.3). Sobald mehrere realisierbare Pfade ausführbar sind, kann das Problem als Optimierungsproblem betrachtet werden. Das Ziel der Optimierung besteht darin, die Pfade zu finden, die eine gegebene Zielfunktion optimieren. [92, 28]

2. Optimierungsziele der Pfadplanung

Während das Optimierungsziel innerhalb der klassischen Pfadplanung in Abschnitt 2.1.1 meist die reine Optimierung der Pfadlänge verfolgt, ist die Zielsetzung im Multi-Agenten-Szenario anwendungsspezifisch. Hierbei wird zwischen drei Fällen differenziert: *Flowtime*, *Makespan* und Zielerreichung unter einer gegebenen *Deadline* [109, 161]. Der Begriff der *Flowtime* bezieht sich auf das Optimierungsziel des MAPF, bei dem die Minimierung der Summe aller Zeitschritte (en: *Sum of Costs (SOC)*) mit $SOC(\tau) = \sum_{i=1}^N |\tau^i|$ aller Agenten N zur Erreichung ihres jeweiligen Zielpunktes im Vordergrund steht (vgl. Abbildung 2.2a) [155]. Die Norm der Trajektorie $|\tau^i|$ repräsentiert hier die Länge des Pfades. Die

Makespan $M(\tau) = \max_{1 \leq i \leq N} |\tau^i|$ hingegen berücksichtigt die Anzahl der erforderlichen Zeitschritte, die der letzte Agent benötigt, um seine jeweilige Zielposition zu erreichen (vgl. Abbildung 2.2b) [168]. Die aufgewendeten Zeitschritte der anderen Agenten sind folglich irrelevant. Seltener werden hingegen *Deadlines* in die Optimierung integriert. Hier besteht das Optimierungsziel darin, in einer fest definierten *Makespan* (Deadline) möglichst viele der Agenten zu ihren Zielpositionen zu bringen [109]. Im Kontext dieser Arbeit hat die *Flowtime* neben der Vollständigkeit aller Kabelpfade Einfluss auf das Optimierungsziel in Kapitel 4.1.4. Die Optimierungen nach der *Flowtime* und der *Makespan* sind in den Abbildungen 2.2a und 2.2b schematisch gegenübergestellt.

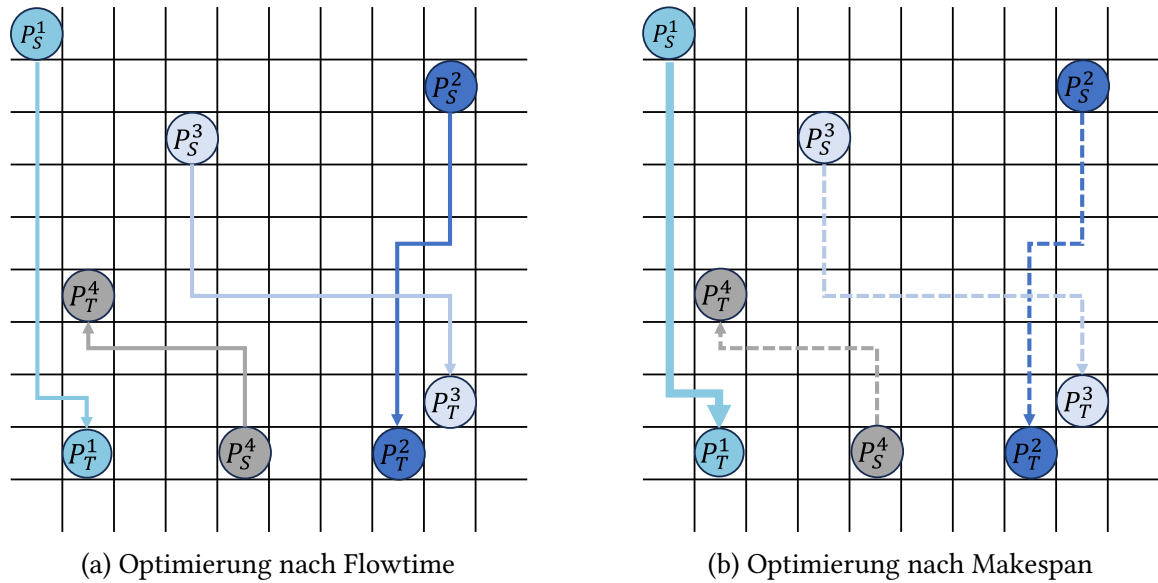


Abbildung 2.2: Gegenüberstellung der Optimierungsziele in der Multi-Agent Pfadplanung i. A. a. [28]

Wie in Abbildung 2.2a dargestellt, fließen in die Optimierung nach *Flowtime* alle Einzelpfade ein. Die aufsummierte Gesamtlänge der Agenten a^1 bis a^4 ergibt in diesem Beispiel eine *Flowtime* von $SOC = 42$. In der Optimierung nach *Makespan* (vgl. Abbildung 2.2b) findet lediglich der Pfad für den Agenten a^4 Berücksichtigung. Dieser stellt in diesem Beispiel mit einer Länge von $M(\tau_4) = 18$ das Maximum aller Einzelpfade dar und wird daher berücksichtigt. Die konkreten Pfadlängen und benötigten Zeitschritte der Agenten a^1 bis a^3 sind in der Optimierung in diesem Szenario (vgl. Abbildung 2.2b) nicht von Bedeutung.

3. Kardinalität und Planungsstrategien zur Pfadoptimierung

In Abhängigkeit von der Kardinalität der Start-Ziel-Kombinationen des MAPF sind drei Hauptszenarien zu unterscheiden. Während die One-to-one-Pfadplanung das Äquivalent zum SPP darstellt, zeigen das One-to-many- und das Many-to-many-Szenario Varianten, in denen sich die einzelnen Agenten innerhalb ihrer Pfadoptimierungen gegenseitig

beeinflussen [184, 152]. Im Gegensatz zur Pfadplanung mit einem einzelnen Agenten (One-to-one-Pfadplanung) besteht in der dynamischen Umgebung mit zusätzlichen Agenten ein Trade-off zwischen optimaler Pfadplanung und effizienter Generierung aller einzelnen Agentenverbindungen. Im Kontext der Planungsstrategien wird der Pfad des proaktiven Agenten durch seine Vorhersage der nächsten Beobachtung beeinflusst. Dieses Verfahren kann nur dann qualitativ hochwertige Lösungen liefern, wenn die Schätzung der Vorhersage zuverlässig ist. Andernfalls kann es zu inakzeptablen Situationen kommen. Reaktive Agenten hingegen korrigieren ihre Trajektorien τ auf der Basis einzelner Beobachtungen. Diese Methode ist sicherer und recheneffizienter, kann aber in komplexen Szenarien zu suboptimalen Trajektorien oder unzureichender Navigation führen. Neben der Anzahl der Agenten und ihrer Reaktivität bei der Planung hat auch ihre Bewegungsfreiheit einen entscheidenden Einfluss auf das Optimierungsziel. Der Grad der Aktionsfreiheit (en: *Degree of Action Freedom (DAF)*) gibt an, wie stark die verfügbaren Aktionen von den möglichen Aktionen abweichen. Während ein hoher DAF nahe dem Wert 1 mit einer hohen Manövrierbarkeit eines Agenten einhergeht, bedeutet ein gegen null tendierender Wert eine geringere Problemkomplexität bei der Lösung des MAPF-Problems. Als Beispiel sind in einem diskreten Raum 27 Aktionen (inklusive Verweilen auf der derzeitigen Position) möglich, die ein Agent anstreben kann. Sind davon lediglich sieben für den Agenten durchführbar, so ergibt sich ein DAF von $\frac{7}{27} = 0,259$. Das Verhältnis von Agenten zum verfügbaren Planungs- und Aktionsraum hat ebenfalls einen Einfluss auf die Komplexität des Problems. Die elementare Restriktion zum Erreichen optimaler Pfade stellt die Existenz von Hindernissen und möglichen Kollisionsarten dar. [28]

2.1.3 Arten von Kollisionen in der Pfadplanung

Im Rahmen der Erweiterung der Pfadplanung um zusätzliche Agenten entstehen neben der möglichen Kollision mit der umliegenden Geometrie weitere Konflikttypen. Das übergeordnete Ziel der MAPF ist dabei die Identifikation einer Lösung, die einen kollisionsfreien Pfad aller Einzelagenten umfasst. Dieser Pfad ist als valide einzuordnen, sobald kein Konflikt zwischen zwei oder mehreren Agenten besteht. In diesem Kontext sind τ^i und τ^j die Trajektorien der beiden Agenten i und j [160]. Die Abbildungen 2.3a bis 2.3e zeigen eine Übersicht potenziell auftretender Konflikte.

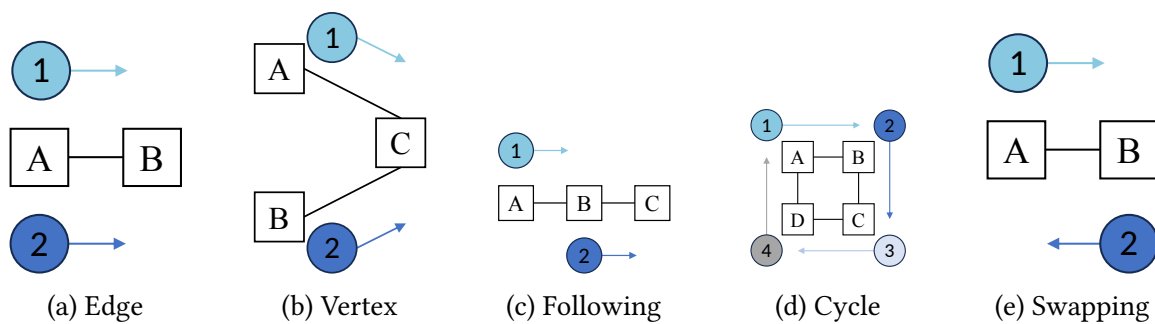


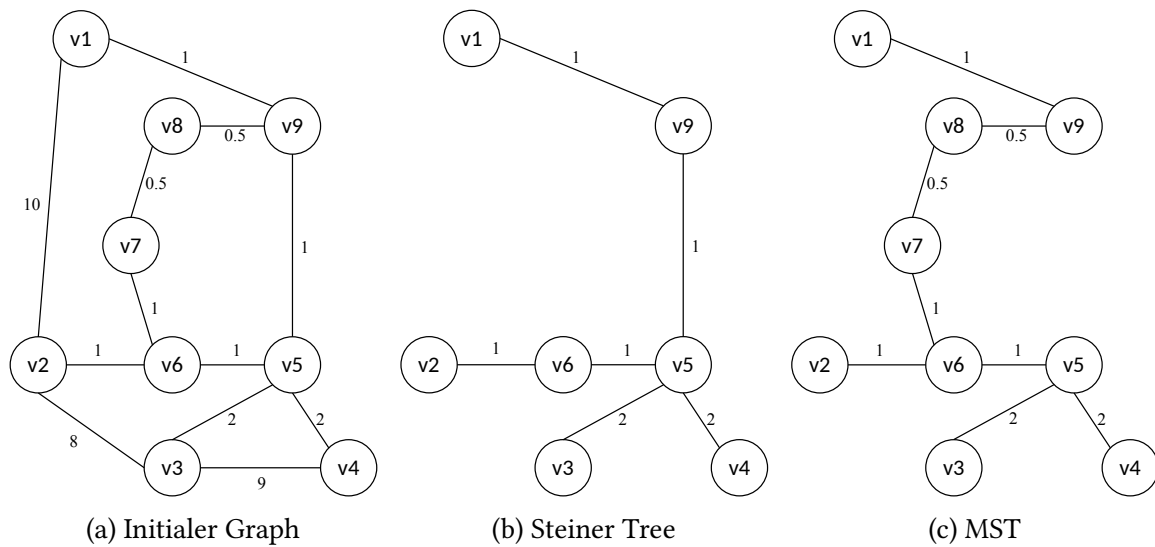
Abbildung 2.3: Darstellung von Kollisionstypen in der Multi-Agenten-Pfadplanung i. A. a. [160]

Einzelne dieser Konflikte (vgl. Abbildung 2.3) sind je nach Anwendungsfall nicht unzulässig. Sie liegen im Rahmen der validen Lösungen. Dies umfasst primär den Swapping- (vgl. Abbildung 2.3e) und den Following-Konflikt (vgl. Abbildung 2.3c) [49]. Im Kontext dieser Arbeit und insbesondere als Teil der neuen Methoden in Kapitel 4 finden hingegen alle fünf Kollisionstypen i. A. a. [160] Anwendung. Die Kollisionstypen sind im Einzelnen:

1. Das Überqueren derselben Kante (u,v) von beiden Agenten wird als *Edge-Konflikt* bezeichnet (vgl. Abbildung 2.3a). Dieser liegt vor, wenn die Agententrajektorien τ^i und τ^j im gleichen Zeitschritt t die gleiche Kante nutzen wollen. Die formale Beschreibung des Konfliktes ist als $\tau_{t+1}^i = \tau_{t+1}^j$ definiert. Er unterliegt der Vorbedingung von $\tau_t^i = \tau_t^j$ und setzt damit die Mehrfachbelegung von Positionen bzw. Knoten im Graphen $G(V, E)$ voraus.
2. Ein *Vertex-Konflikt* (vgl. Abbildung 2.3b) liegt zwischen den beiden Agenten vor, wenn beide beabsichtigen, den gleichen Knoten (en: *Vertex*) innerhalb des gleichen Zeitschrittes zu belegen [168]. Formal liegt dieser Konflikt zwischen τ^i und τ^j vor, sobald ein Zeitschritt t existiert für den $\tau_t^i = \tau_t^j$ gilt [49].
3. Der *Following-Konflikt* tritt auf, sobald ein Agent versucht, die Position einzunehmen, die ein anderer Agent im vorhergehenden Zeitschritt besetzt hat (vgl. Abbildung 2.3c). $\tau_t^i = \tau_{t+1}^j$ beschreibt diesen Konflikt formal.
4. Ein *Cycle-Konflikt* (vgl. Abbildung 2.3d) entsteht zwischen einer Menge von Trajektorien $\tau^i, \tau^{i+1}, \dots, \tau^j$ je Einzelagent, wenn sich jeder Agent im selben Zeitschritt zu einem Knotenpunkt bewegt, der im vorangegangenen Zeitschritt $t - 1$ von einem anderen Agenten bereits besetzt war. So ergibt sich ein rotierendes Zyklusmuster das einen Sonderfall des Following-Konfliktes darstellt.
5. Der *Swapping-Konflikt* (vgl. Abbildung 2.3e) entwickelt sich, sobald beide Agenten die Absicht haben, die Positionen in einem Zeitschritt zu tauschen. Formal ist der Konflikt mit den beiden Bedingungen $\tau_{t+1}^i = \tau_t^j$ und $\tau_{t+1}^j = \tau_t^i$ beschrieben. Die uni- oder bidirektionale Beschaffenheit der Kanten beeinflusst das Auftreten bzw. die Zulässigkeit dieses Konfliktes.

2.1.4 Erzeugung und Gegenüberstellung von Steiner Tree und Minimum Spanning Tree

Ein Minimum Spanning Tree (MST) ist ein Teilbaum $T \subset G(V, E)$, der alle Terminalknoten eines Graphen $G(V, E)$ verbindet, ohne zyklische Verbindungen zu bilden. Er wird unter Verwendung der Edges mit dem geringsten Kantengewicht erzeugt. In der Regel werden für die Erzeugung von MSTs der Prim'sche- oder der Kruskal'sche-Algorithmus angewendet [134, 96]. Beide Algorithmen sind aufgrund ihrer Wirkungsweise in der Anwendung in die Kategorie der gierigen (en: *greedy*) Verfahren einzuordnen. In Abbildung 2.4 werden ein initialer Graph (vgl. Abbildung 2.4a), ein Steiner Tree (vgl. Abbildung 2.4b) und der MST (vgl. Abbildung 2.4c) für beispielhafte Terminalknoten V^1, V^2, V^3, V^4 dargestellt und verglichen.



Abbildungung 2.4: Vergleich zwischen initialem Graphen, Steiner Tree und Minimum Spanning Tree i. A. a. [95]

In Abbildung 2.4a ist ein initialer Graph mit neun Vertices V^i und zwölf Edges dargestellt. Die Generierung eines MSTs (vgl. Abbildung 2.4c) beginnt an einem beliebigen Vertex des Ausgangsgraphen (vgl. Abbildung 2.4a) und die benachbarten Edges werden auf Basis des geringsten Kantengewichtes zum Baum hinzugefügt. Dadurch wird ein Knoten innerhalb des Graphen mit einem anderen Knoten außerhalb des Graphen verbunden, ohne eine zyklische Verbindung zu erzeugen. [134] Dieser Vorgang wird so lange wiederholt, bis alle bzw. alle benötigten Knoten Teil des Baumes sind. Abwandlungen dieses Verfahrens sortieren zunächst alle Edges des Graphen nach ihrem Gewicht und beginnen mit der Erzeugung des MSTs, wobei jeder Knoten einen eigenen Tree darstellt. Der zyklische Ansatz (vgl. Abbildung 2.4c) verbindet dann sukzessive die Subtrees, bis ein ganzheitlicher MST entsteht. [96]

Der Steiner Tree (vgl. Abbildung 2.4b) basiert analog zum MST auf der Konstruktion eines Teilbaumes [57]. Während der MST einen minimalen Baum für alle Scheitelpunkte berechnet (vgl. Abbildung 2.4c), zielt der Steiner Tree (vgl. Abbildung 2.4b) auf einen minimalen Baum für eine gegebene Menge von Knoten ab. Aufgrund des hohen Rechenaufwandes wird der Steiner Tree als *NP-vollständig* klassifiziert und in der Praxis durch Approximationsalgorithmen gelöst. [95]

2.2 Repräsentation und Verwaltung von Produktdaten im Computer-Aided Design (CAD)

Die Analyse und Darstellung von Objekten aus dem 3D-CAD-System ist ein wesentlicher Bestandteil der neuen Methoden in Kapitel 4. Als Grundlage dafür werden in den Kapiteln 2.2.1 und 2.2.2 die Basis-Spline-Kurven und deren Erweiterung in Form von Non-Uniform Rational B-Splines (NURBS)-Kurven im Kontext von 3D-CAD-Systemen beschrieben. Es bestehen vielfältige Möglichkeiten zur speichereffizienten Verarbeitung von 3D-CAD-Modellen. In Kapitel 2.2.3 werden die etablierten Voxel-Methoden analysiert und den Mesh-Strukturen gegenübergestellt. Kapitel 2.2.4 schließt mit der Analyse zur Quantifizierung geometrischer Ähnlichkeiten an und zeigt die Grundlage für die Wissensübertragung zwischen ähnlichen 3D-CAD-Objekten auf. Im abschließenden Kapitel 2.2.5 werden die Standardaustauschformate der am Produktentstehungsprozess beteiligten Disziplinen und IT-Systeme analysiert und deren Heterogenität aufgezeigt.

2.2.1 Basis-Spline-Kurven im 3D-CAD-System

Als Verallgemeinerung von Bézier-Kurven stellen Basis-Spline-Kurven (B-Spline) eine approximierte Kurve dar, die über einen Polygonzug von Stützpunkten definiert ist [171]. Sie sind segmentweise festgelegt und ermöglichen damit eine Definition des Polynomgrades, der unabhängig von der Anzahl der Kontrollpunkte $P_i = [P_{ix}, P_{iy}, P_{iz}]$ ist. Die normalisierten B-Spline-Basisfunktionen $N_i^k(t)$ (vgl. Gleichung 2.1) werden rekursiv nach den Gleichungen 2.2 und 2.3 definiert. Die Kurve $C(t)$ in Gleichung 2.1 wird als gewichtete Summe der Kontrollpunkte berechnet. [171]

$$C(t) = \sum_{i=0}^n N_i^k(t) P_i \quad (2.1)$$

Dabei ist in Gleichung 2.1 die Variable i der Knoten, an dem die jeweilige Funktion $N_i^k(t)$ beginnt. k ist die Ordnung der B-Spline und P_i der zugehörige Stützpunkt der Nummer i . Die Variable t ist ein Kurvenparameter, der Werte zwischen 0 (Anfangspunkt) und 1 (Endpunkt) annehmen kann. Gleichung 2.2 zeigt die Basisfunktion für den Grad $k = 0$. [132, 69]

$$N_i^0(t) = \begin{cases} 1 & \text{wenn } t_i \leq t < t_{i+1} \\ 0 & \text{sonst} \end{cases} \quad (2.2)$$

Gleichung 2.2 hat zur Folge, dass eine Basisfunktion vom Grad $k = 0$ nur in einem bestimmten Intervall aktiv ist und außerhalb dieses Intervalls keinen Einfluss mehr hat. Gleichung 2.3 zeigt das Szenario für $k > 0$ auf [69, 171].

$$N_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} N_i^{k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1}^{k-1}(t) \quad (2.3)$$

Die rekursive Definition erlaubt die Konstruktion von Basisfunktionen höherer Grade aus Basisfunktionen niedrigerer Grade. Der rekursive Ansatz gewährleistet, dass die B-Spline-Basisfunktionen stückweise definiert sind und über ihre Intervalle hinweg kontinuierlich und glatt verlaufen. Abbildung 2.5 zeigt eine exemplarische Darstellung einer B-Spline-Basisfunktion.

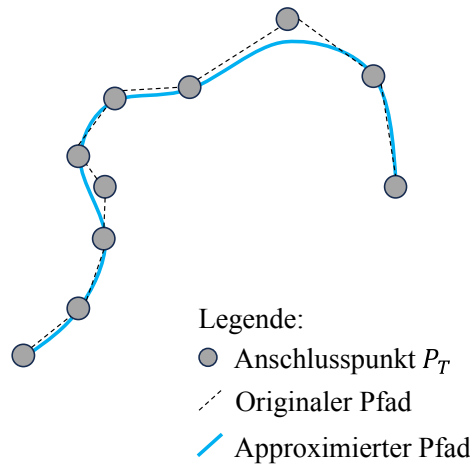


Abbildung 2.5: Exemplarische Approximierung einer Spline

2.2.2 Non-Uniform Rational B-Splines (NURBS) im 3D-CAD-System

NURBS-Kurven haben sich als „Quasistandard“ für die Darstellung von komplexen Kurven in 3D-CAD-Systemen etabliert [16, 172]. Ebenso kommen sie für die Definition von elektrischen Verbindungen im 3D-CAD-System zum Einsatz [172, 200].

Eine NURBS-Kurve mit dem Grad k ist eine stückweise kontinuierliche Funktion mit $n+1$ Stützpunkten P_i . Jeder Stützpunkt wird mit einem nicht negativen Gewicht w_i annotiert [171, 69]. Somit ist jeder Punkt durch seine Koordinaten $[x, y, z]$ und sein Gewicht w_i definiert. Die mathematische Definition einer NURBS-Kurve ist in Gleichung 2.4 gegeben [48].

$$X(t) = \frac{\sum_{i=0}^n w_i \cdot N_i^k(t) \cdot P_i}{\sum_{i=0}^n w_i \cdot N_i^k(t)} \quad (2.4)$$

In Gleichung 2.4 entspricht $N_i^k(t)$ einer normalisierten B-Spline (vgl. Kapitel 2.2.1). Der Index i bezeichnet den Knoten, an dem die jeweilige Funktion $N_i^k(t)$ beginnt. w_i stellt das nicht negative Gewicht dar und ist damit das wesentliche Differenzierungsmerkmal zur B-Spline (vgl. Kapitel 2.2.1). [171]

Werden alle Gewichtungsfaktoren $w_i = 1$ gesetzt, so entsteht wieder eine gewöhnliche, nichtrationale B-Spline, da der Nenner in Gleichung 2.4 den Wert 1 annimmt [69]. Damit kann eine genauere Approximation für spezifische Bereiche erreicht werden [132]. Dies ist die maßgebliche Differenzierung zur B-Spline, da andernfalls eine lokale Anpassung oder Manipulation der Kurven schwierig ist [48].

2.2.3 Mesh-Strukturen und Volumenelemente (Voxel) in 3D-CAD-Systemen

Die Darstellung von 3D-CAD-Modellen beinhaltet vielfältige Möglichkeiten, die sich in ihrer Verarbeitungs- und Speichereffizienz deutlich voneinander unterscheiden. Konkrete Anforderungen an die jeweilige Repräsentanz ergeben sich vor allem aus der Anwendung. So sieht das im MCAD gängig verwendete NeutralfORMAT Standard for the Exchange of Product Data (STEP) nach ISO 10303 als Inhalt neben den physischen auch funktionale Aspekte eines Produktes vor [79]. Da für die Verarbeitung des 3D-CAD-Modells in der neu zu entwickelnden Methode vor allem die geometrische Repräsentanz in einem speicheroptimierten Format im Vordergrund steht (vgl. Kapitel 4.1.2), beinhaltet dieses Kapitel vorrangig die Abbildung einer Geometrie in Gitterstrukturen (en: *Mesh*) und dreidimensionalen Volumenelementen (en: *Voxel*).

Die dreidimensionalen Äquivalente eines Pixels werden als Voxel bezeichnet. Es sind die kleinsten unterscheidbaren Elemente in einem Volumendatensatz und sie stehen repräsentativ für Punkte innerhalb eines dreidimensionalen Raums [18, 71]. Sie entsprechen abstrakten, kubischen 3D-Einheiten mit vordefiniertem Volumen, Position und Attributen [97]. Damit ist es möglich, diskrete Punkte in einer topologisch expliziten und informationsreichen Weise strukturell und speichereffizient darzustellen [142, 194]. Ein Voxel ist durch die drei wesentlichen Attribute Position, Größe und Wert eindeutig beschrieben [113]. Die Position des Voxels beschreibt die Verortung innerhalb des dreidimensionalen Raums [34]. Der Abstand zwischen benachbarten Voxeln ist über die Größe eines einzelnen Voxels bestimmt. Zur Abbildung von Belegungsgrad, Dichte, Farbe oder Materialzusammensetzungen ist einem Voxel ein eindeutiger numerischer Wert zugewiesen [97]. Der 3D-Netz-basierten Darstellung von Voxeln steht die flächenbasierte Repräsentanz von Gittern (en: *Mesh*) gegenüber. Beide dienen als abstrakte und strukturierte Einheiten zur Beschreibung einer 3D-Geometrie [194, 182]. In Abbildung 2.6 wird der Einfluss der Granularität auf die Transformation einer Geometrie in Voxeln und in einem triangulierten Mesh-Format dargestellt und verglichen.

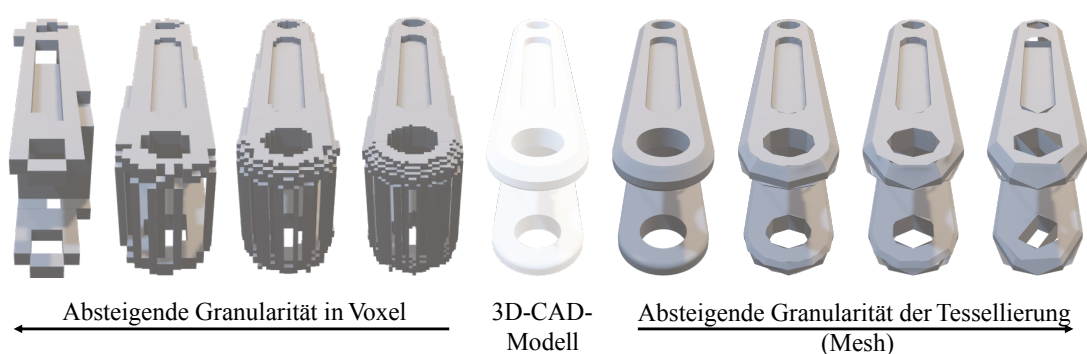


Abbildung 2.6: Darstellung der Transformation eines 3D-CAD-Modells in das Voxel- und das Mesh-Format i. A. a. [182]

In den Grafiken in Abbildung 2.6 ist zu erkennen, dass mit zunehmender Größe eines einzelnen Voxels mitunter ganze Strukturen der ursprünglichen Geometrie verloren gehen. Daraus lässt sich folgern, dass für das Abbilden von dünnwandigen Objekten eine

entsprechend hohe Auflösung notwendig ist [16, 182]. Der Verlust von Strukturen ist im Kontext der Tessellierung (vgl. Abbildung 2.6 rechts) nicht im selben Maß erkennbar. Stattdessen nähert sich die Geometrie zunehmend einer dreiecksgeprägten Darstellung an [171]. In diesem Vergleich ist festzuhalten, dass die Datengröße mit zunehmender Voxelgröße im Vergleich zur Tessellierung stärker sinkt. Es ist zu berücksichtigen, dass für die Darstellung von gekrümmten Oberflächen mittels Voxeln eine kompaktere Speicherung möglich ist als die große Menge an Dreiecken, die zur Annäherung an die Geometrie benötigt werden [148, 65]. Die Ableitung eines hinreichend genauen Voxelmodells aus einem 3D-CAD-Volumenmodell ist jederzeit möglich, jedoch lässt sich dieser Prozess nicht mit ausreichender Genauigkeit revidieren [176]. Unter Anwendung entsprechender Software ist diese Transformation von Polygonflächen in NURBS-Flächen hingegen möglich [71].

Die Darstellung einer dreidimensionalen Geometrie als Mesh-Objekt ist eine der grundlegenden Vorgehensweisen bei der Erzeugung von Neutralformaten [65]. Es erweitert dabei einen gewichteten Graphen aus Scheitelpunkten (en: *Vertices*) und Kanten (en: *Edges*) auf datentechnischer Ebene, indem es Flächen (en: *Faces*) hinzufügt, um die Oberfläche des Objekts zu bilden (vgl. Kapitel 2.1) [16, 58]. Dafür setzt der Prozess eine lücken- und überlappungsfreie Modellierung der Ausgangsgeometrie voraus [176]. Ein Scheitelpunkt (en: *Vertex*) beschreibt im Kontext der Mesh-Methode eine Position im dreidimensionalen Raum, die auch Informationen der Oberflächennormale, Farbe oder Texturkoordinate beinhalten kann. Die Edges verbinden bestehende Vertices und bilden damit die Oberfläche des dreidimensionalen Objektes ab. [65]

Die Identifikation von minimalen Distanzen zwischen zwei Körpern in Mesh-Strukturen ist Bestandteil der neuen Methode in Kapitel 4. Für die Bestimmung des kürzesten Abstands zwischen den Oberflächen zweier Körper (B_i) B_1 und B_2 , die durch die Meshes (M_{B_i}) M_{B_1} und M_{B_2} repräsentiert werden, ist der minimale Abstand zwischen einem beliebigen Paar von Elementen zu bestimmen. Diese Elemente können Vertices, Edges und Faces der Meshes M_{B_1} und M_{B_2} sein. Dieser Prozess umfasst mehrere Schritte, bei denen die Abstände Vertex-Vertex, Vertex-Edge, Edge-Edge sowie Vertex-Face berücksichtigt werden. Der Mindestabstand zwischen zwei Körpern ergibt sich aus dem Minimum dieser Abstände [16]. Im Folgenden werden vier unterschiedliche Abstände betrachtet [16]:

1. Die Distanz Vertex-Vertex ist durch die euklidische Distanz zwischen zwei Punkten im dreidimensionalen Raum beschrieben.
2. Der Abstand zwischen Vertex und Edge kann als senkrechter Abstand vom Punkt zur Kante oder zu einem ihrer Endpunkte gemessen werden, wenn die Scheitelpunktprojektion außerhalb des Kantensegments liegt.
3. Die gemeinsame Normale zweier Liniensegmente ist der minimale Abstand in der Edge-Edge-Distanz.
4. Der Abstand zwischen Vertex und Face wird als der senkrechte Abstand zwischen dem Scheitelpunkt und der Ebene gemessen, die mit der Oberfläche zusammenfällt. Liegt die Scheitelpunktprojektion außerhalb der Fläche, so berücksichtigt die Berechnung die Abstände zum Scheitelpunkt der Fläche.

Da in der praktischen Anwendung die zeitgleiche Minimierung aller Distanzen (1-4) sehr rechenintensiv ausfallen kann, sind räumliche Datenstrukturen wie Bounding-Volume-Hierarchien erforderlich, um die Anzahl der Abstandsberechnungen zu reduzieren. Sie eliminieren schnell die Elemente des Netzes, die weit voneinander entfernt sind. [94]

Ein recheneffizienterer Ansatz basiert auf der Verwendung von konvexen Meshes als Kollisions-Shape. Hierfür werden recheneffiziente Methoden, wie der Gilbert-Johnson-Keerthi-Distanz-Algorithmus (GJK) eingesetzt, der die minimale Distanz zweier konvexer Mengen berechnet. Diese Vorgehensweise ist insofern vorteilhaft, als dass sie einen schnellen und effizienten Umgang mit komplexen Formen zulässt. Die hohe Recheneffizienz resultiert aus einer isolierten Betrachtung der Existenz von Kollisionen, nicht aus einer Spezifikation der Art von Kollisionen. [56, 173, 186]

2.2.4 Quantifizierung geometrischer Ähnlichkeit von dreidimensionalen Geometrien

Die Grundlage für einen Transfer von KI-Methoden zwischen 3D-CAD-Geometrien ist eine Ähnlichkeit der geometrischen Eingangsgrößen (vgl. Kapitel 2.6). Zu diesem Zweck besteht die Notwendigkeit, die geometrische Ähnlichkeit von 3D-CAD-Modellen zu quantifizieren. Die existierenden Methoden lassen sich in vier relevante Teilkategorien untergliedern [81, 26]:

1. Methoden, die auf *globalen Features* basieren, nutzen globale Eigenschaften des 3D-CAD-Modells wie Momente, Invarianten, Fourier-Deskriptoren und geometrische Verhältnisse zur Ermittlung der Ähnlichkeit zweier 3D-CAD-Geometrien [125]. Insbesondere lassen sich damit starre Transformationen der 3D-CAD-Geometrie wie Rotationen oder Skalierungen identifizieren [181].
2. In *graphbasierten* Methoden wird hingegen die Topologie des 3D-CAD-Modells im ersten Schritt in eine Graphendarstellung überführt. Eine Messung der Ähnlichkeit stützt sich dann auf die Quantifizierung der Similarität dieser Graphen.
3. Bei *histogrammbasierten* Methoden erfolgt die Erzeugung einer Zufallsauswahl von Oberflächenpunkten als datentechnisches Hilfsmittel sowie die Extraktion der geometrischen Charakteristika aus den 3D-Punkten. Die erkannten Merkmale werden in Form von Histogrammen oder Verteilungen organisiert, welche die Häufigkeit ihres Auftretens darstellen [116]. Die Ähnlichkeit lässt sich durch einen Vergleich der Histogramme unter Zuhilfenahme einer Abstandsfunktion ermitteln [77].
4. Die Gruppierung der Methoden zur Ähnlichkeitsbemessung auf Grundlage von *3D-Objekterkennung* bezieht die Methoden von Convolutional Neural Network (CNN) ein (vgl. Kapitel 2.4.2), um Strukturen innerhalb der Geometrien zu erkennen [198, 181]. Sie erfordern jedoch eine große Menge an Trainingsdaten [181].

Die Methoden der Kategorien *produktinformations-* und *fertigungsmerkmalsbasiert* werden mangels Relevanz für diese Arbeit nicht näher betrachtet. Ergänzend zu den aufgeführten Verfahren ist die Übertragung von 2D-Methoden auf das 3D-CAD-Modell zu nennen.

Diese haben ihren Ursprung in der Verarbeitung von Bildern, bei der die Ähnlichkeit in der Helligkeit, im Kontrast und in der Struktur abgeglichen wird [185]. Je nach Eingangsdimension der Berechnung sind diese Methoden auch auf dreidimensionale Objekte adaptierbar.

2.2.5 Interdisziplinärer Datenaustausch in der Produktentstehung

Ein effizienter Datenaustausch zwischen verschiedenen CAD-Quellen ohne Informationsverlust ist im Produktentwicklungsprozess ein entscheidender Faktor für eine konsistente und durchgängige Planung [162]. Die Betrachtung des Datenaustausches von elektrotechnischen und mechanischen Daten des MCAD und ECAD steht im Fokus der neuen Methode in Kapitel 4.2.2. Insbesondere dafür wird dieser Datenaustausch zwischen ECAD- und 3D-MCAD-Systemen analysiert. Die disziplinspezifischen Autorensysteme zeichnen sich durch sehr heterogene Standardaustauschformate aus [39]. Dies ist eng an die Planungsdimensionalität des jeweiligen Fachbereichs geknüpft. Während das mechanische Engineering durch eine Produktentwicklung im 3D-CAD-System gekennzeichnet ist, dominiert in der elektrotechnischen Planung der 2D-Schaltplan im ECAD-System. Er basiert auf einer grafischen 2D-Struktur, die zeigt, welche elektrischen Geräte mit Strom- oder Datenkabeln verbunden sind. [65]

Von praktischer Relevanz sind im industriellen Kontext von 3D-CAD-Systemen im mechanischen Engineering die Standardformate Initial Graphics Exchange Specification (IGES) ([8]), Jupiter Tessellation (JT) ([80]) und STEP ([79]) [171, 65]. Dem stehen die Formate DXF (Drawing Exchange Format) und DWG (Drawing) in den elektrotechnischen Planungssystemen gegenüber. In Tabelle 2.1 werden die Inhalte und Möglichkeiten der jeweiligen Datenformate gegenübergestellt. Ergänzt wird der Vergleich durch das Format AutomationML, ein XML-basiertes, objektorientiertes Austauschformat zwischen Engineering-Systemen [60, 39].

Tabelle 2.1: Gegenüberstellung von neutralen Datenaustauschformaten

Format-name	Anwendungs-domäne (primär)	Daten-struktur	Verarbeitung von 3D-Geometrie	Erhalt der Baugruppen-struktur	Integration von elektrischen Komponenten	Integration von Kinematik	Funktionalitäten für ECAD-MCAD-Integration
STEP	Mechanik	ISO 10303, textbasiert	Ja	Ja	Teilweise	Ja	Mittel
IGES	Mechanik	ASCII, textbasiert	Ja	Ja	Nein	Nein	Niedrig
JT	Mechanik	ISO 14306, Binär, XML	Ja	Ja	Nein	Ja	Niedrig
DXF	Elektrik	ASCII, textbasiert	Nein	Nein	Teilweise	Nein	Niedrig
DWG	Elektrik	Binär, proprietär	Ja	Nein	Teilweise	Nein	Mittel
AML	unabhängig	IEC 62714, XML	Ja	Ja,	Ja	Ja, über URDF / COLLADA	Hoch

In Tabelle 2.1 ist die Kurzform STEP nach ISO 10303-242 als Äquivalent zu STEP242 zu sehen, AutomationML wird mit AML abgekürzt. Die Tabelle 2.1 gibt neben der Anwendungsdomäne auch an, welche Datenstruktur im jeweiligen Datenaustauschformat

verwendet wird. Ergänzend ist die Information enthalten, inwieweit die Kriterien der *Verarbeitung von 3D-CAD-Geometrien* bis hin zur *Integration von Kinematik* erfüllt sind. Die Spalte *Funktionalitäten für ECAD-MCAD-Integration* fasst die Eignung des Formats für die neue Methode in Kapitel 4.2.2 zusammen. Anhand der aufgezeigten Kriterien ist eine nähere Betrachtung der Formate STEP und AutomationML sinnvoll, da diese den höchsten Erfüllungsgrad der Anforderungen aufweisen. [176, 71, 65, 171]

Als nähere Spezifikation der Zusammenführung von ECAD- und 3D-MCAD-Daten ist zudem die Art der Datenkollaboration festzulegen. Hierbei ist zwischen einer Offline- und einer Online-Kollaboration, sowie einer Kopplung mit und ohne Produktdatenmanagement (PDM)-System zu unterscheiden. Von besonderem Interesse für die neue Methode in Kapitel 4.2.2 ist die Online-Kollaboration mit assoziierten PDM-Systemen, da sowohl ECAD- als auch MCAD-Systeme bereits heute jeweils meist in deren PDM-Systeme integriert sind. Die Anforderung an die neue Methode ist daher ein webbasierter Service, der die benötigten Daten referenziert, anstatt eine redundante Datenhaltung in einem neuen IT-System zu erzeugen. [162]

2.3 Virtuelle Verdrahtung von elektrischen Komponenten im 3D-CAD

Für die Weiterentwicklung bestehender Methoden ist es notwendig, den zugrunde liegenden Prozess der virtuellen Verdrahtung im 3D-CAD genauer zu analysieren. Abschnitt 2.3.1 beschreibt die Intention der virtuellen Verdrahtung und die verfügbaren Methoden dieses Prozesses. Die dafür vorgesehenen und üblicherweise verfügbaren Eingangsgrößen sind in Abschnitt 2.3.2 detailliert erläutert. Einen signifikanten Einfluss haben die Randbedingungen, die in Abschnitt 2.3.3 dargelegt werden. Darüber hinaus wird in diesem Abschnitt betrachtet, wie die Randbedingungen die Validität der Ergebnisse beschränken. Als spezieller Einflussfaktor wird in Abschnitt 2.3.4 die Kinematik von kabelführenden Bauteilen und beweglichen Störgeometrien analysiert. Die Eingangsinformationen und die Randbedingungen fließen in den Gesamtprozess ein, über den Abschnitt 2.3.5 einen Überblick bereitstellt. Zudem werden hier die einzelnen Teilprozesse ausgeführt. Abschnitt 2.3.6 schließt das Kapitel mit der Abstraktion der virtuellen Verdrahtung als 3D-Pfadoptimierung in einem CHRP ab.

2.3.1 Intention und Ziele der virtuellen Verdrahtung

Aufgrund des hohen zeitlichen Aufwands, den der Prozess in der virtuellen Produktentwicklung einnimmt, muss dem Subprozess der virtuellen Verdrahtung eine entsprechende Relevanz für das Gesamtergebnis beigemessen werden [90]. Die Intention der virtuellen Verdrahtung untergliedert sich dabei in zwei Bereiche. Einerseits dient sie der Validierung, ob die vorangegangenen elektrischen und mechanischen Planungen sich als realisierbar und kombinierbar erweisen. Diese Validität ist im Wesentlichen durch die Bauraumbeschränkung geometrisch restriktiert, aber auch durch elektrotechnische Anforderungen eingeschränkt [91]. Insbesondere innerhalb enger mechanischer Bauräume ist ein hoher Detaillierungsgrad zur Sicherstellung der Montierbarkeit erforderlich [73]. Dieser Nachweis ist von einer detaillierten wie auch exakten Planung der Kabelpfade abhängig und verschärft daher auch die Problematik eines hohen Zeitaufwands (vgl. Kapitel 1.2). Andererseits wird durch die virtuelle Verdrahtung die Ermittlung der benötigten Kabellängen für Bestellprozesse und die physische Verdrahtung in der Montage ermöglicht [90]. Dies geht mit dem Ziel einher, sicherheitsbedingte Überlängen und damit unnötigen Materialbedarf der Kabel zu reduzieren [200]. Das Hauptziel lässt sich folglich mit der präventiven Planung der Realisierung innerhalb der mechanischen 3D-CAD-Geometrie und der Ermittlung der notwendigen Kabellängen zusammenfassen [37]. Diese beiden Informationen stellen den erwarteten Output des Prozesses der virtuellen Verdrahtung dar [45].

2.3.2 Aufbau und Inhalte der Eingangsgrößen für die virtuelle Verdrahtung

Für eine virtuelle Verdrahtung im 3D-CAD-System sind drei wesentliche Eingangsinformationen erforderlich. Neben der geometrischen Repräsentanz des zu verkabelnden Produktes aus dem MCAD nutzt der Prozess die Geräteliste und die elektrischen Verbindungen aus

dem ECAD [209]. Diese werden um Merkmalsinformationen der verwendeten Kabeltypen ergänzt. Die Merkmale des expliziten Kabeldurchmessers und der jeweils zulässige Biegeradius sind hier exemplarisch zu nennen. Optional stehen z. B. Designspezifikationen als Eingangsgröße zur Verfügung, die neben physikalischen auch unternehmensspezifische Randbedingungen beschreiben.

Das **3D-MCAD-Modell** ist der erste wesentliche Bestandteil der Eingangsinformationen. Es stellt die geometrische Repräsentanz der zu verdrahtenden Objekte dar. Gleichzeitig enthält es alle mechanischen und elektrischen CAD-Objekte in einer dreidimensionalen Darstellung. Dies beinhaltet neben der Hüllgeometrie auch Merkmalswerte zum verwendeten Material (Werkstoff), dessen Masse und weitere Metadaten. Das zugrunde liegende Datenmodell reicht von grundlegenden geometrischen Primitiven wie Würfel, Zylinder und Kegel bis hin zu parametrischen Kurven, Flächen und Volumen [65, 171]. Eine übergeordnete Struktur auf Bauteilebene fasst die geometrischen Körper zu einem MCAD-Einzelteil zusammen. Untergliedert in funktionale Baugruppen und Unterbaugruppen werden die Einzelteile in der 100 %-Engineering Bill of Materials (EBOM) zusammengefasst. Die Mehrfachverwendung von Einzelteilen in der EBOM reduziert die Datenmenge und -komplexität, erfordert aber eine eindeutige Identifizierbarkeit der Teile in der CAD-Struktur. Das MCAD-3D-Modell beinhaltet zudem die Beziehung zwischen Einzelteilen und Baugruppen. Dazu gehören neben möglichen Montageinformationen auch Randbedingungen und dynamische Informationen in Form von Product and Manufacturing Information (PMI) [143]. Kinematische Ketten ermöglichen die Simulation des dynamischen Zusammenwirkens von Bauteilen und -gruppen und bilden gleichzeitig die Datenbasis für die Teilmethode in Kapitel 4.2.4 [162, 65]. Für die Anwendung des 3D-CAD-Modells in der virtuellen Verdrahtung ist darüber hinaus die Festlegung der notwendigen Konnektoren erforderlich. Diese legen den exakten Punkt und Richtungsvektor fest, in dem das Kabel den Konnektor verlässt bzw. in diesen eintritt (vgl. Kapitel 4.1.2). [45]

Die **elektrische Verbindungsliste** aus dem ECAD bildet für den Prozess der virtuellen Verdrahtung die zweite Eingangsgröße. Sie beschreibt im ECAD-System, welche Verbindungen notwendig sind, um die Funktionalität des Produktes herzustellen [200]. Diese tabellarischen oder XML-basierten Informationen umfassen die Summe aller Verbindungselemente sowie deren Verbindungspunkte. Jedes Verbindungselement besteht aus der jeweiligen Betriebsmittelkennziffer (BMKZ) der Start- und Zielkomponenten, deren jeweiligem Anschlussidentifikator und dem einzusetzenden Kabeltyp. Außerdem enthält es eine eindeutige Kombination aus Kabeltyp und Durchmesser. Eine exemplarische Auflistung in Form von Verbindungslisten ist in Tabelle 2.2 dargestellt.

Tabelle 2.2: Exemplarische elektrotechnische Eingangsinformationen für den Verdrahtungsprozess

(a) Exemplarische Darstellung einer elektrischen Verbindungsliste für ein Produkt

Verbindungsliste Produkt 5822						
Betriebsmittelkennziffer Start (BMKZ)	Anschluss-Pin Start	Betriebsmittelkennziffer Ziel (BMKZ)	Anschluss-Pin Ziel	Kabeltyp	Kabeldurchmesser in mm ²	Farbe
-F20	1.3	-H1	2.2	H07Z1-K	2.5	schwarz
-XC31_104	2.1	-XQ0_24	1.1	HX4GAF	1.5	blau
...

(b) Exemplarische Darstellung von Kabelstammdaten

Kabelstammdaten				
Kabeltyp	Kabeldurchmesser in mm ²	Zulässiger Biegeradius	Temperaturbereich	Farbe
H07Z1-K	2.5	5x Außendurchmesser	-40 °C bis +70 °C	schwarz
HX4GAF	1.5	4x Außendurchmesser	-40 °C bis +120 °C	blau
...

Für die Verarbeitung der Daten aus Tabelle 2.2a im MCAD ist ein Zusammenhang zwischen den BMKZ der Start- und Zielkomponente sowie dem 3D-CAD-Modell essenziell. Gleiches ist auch für die Anschlusspins notwendig, um eine exakte Kabellänge in der virtuellen Verdrahtung ermitteln zu können. Dieser eindeutige Zusammenhang ist eine der wesentlichen Grundlagen für die Synchronisierung von ECAD- und MCAD-Daten (vgl. Kapitel 2.2.5). Ausschließlich über diese Methode ist es möglich, die elektrotechnischen Merkmale in den mechanischen Planungsprozess zu integrieren. Diese Integration zeigt sich maßgeblich in der virtuellen Verdrahtung, da eine große Anzahl von Randbedingungen (vgl. Kapitel 2.3.3) durch elektrotechnische Gegebenheiten definiert ist. [162]

Die assoziierten **Kabelstammdaten** der in der Verbindungsliste vorkommenden Kabeltypen (vgl. Tabelle 2.2b) sind entweder direkt in das ECAD-System eingepflegt und damit Bestandteil der Verbindungsliste oder es werden vom Lieferanten bereitgestellte Datenblätter oder XML-Dateien im Enterprise Resource Planning (ERP)- oder in einem PDM-System in einem Materialstammsatz hinterlegt. Die tabellarische Information der Kabelstammdaten (vgl. Tabelle 2.2b) zu einem Kabeltyp umfasst neben einem eindeutigen Identifikator z. B. auch die Merkmalswerte wie das Gewicht und den Preis pro laufenden Meter, den Durchmesser, den minimalen Biegeradius des Kabels und die Farbe [45]. Sie dienen als beschreibende Parameter innerhalb der Verbindungsliste und ergeben sich aus dem elektrotechnischen Anwendungsfall. Mit weiteren Angaben wie dem Werkstoff, den Ummantelungsinformationen und den Angaben zum Aufbau des Leiters sind sie beliebig erweiterbar. Tabelle 2.2b zeigt einen Ausschnitt einer Liste von Kabelstammdaten.

Exemplarisch sind hier die minimale und die maximale Einsatztemperatur als ergänzende Merkmale aufgeführt.

Für die Beschreibung der einzuhaltenden **Randbedingungen** für die virtuelle Verdrahtung besteht kein etabliertes, einheitliches Format. Daher basieren derartige Beschreibungen meist auf einem Konglomerat aus Freitexten, firmen- und branchenindividuellen Standards und Informationen aus rechtlichen Verordnungen. Randbedingungen (z. B. die Kollisionsfreiheit - vgl. Kapitel 2.3.3), die sich bereits direkt aus der Geometrie ergeben, sind häufig nicht explizit formuliert. Zumeist stützt sich der Prozess der virtuellen Verdrahtung auf die Routine und Praxiserfahrung des mit der Aufgabe betrauten Mitarbeiters. Diesem obliegt die Verantwortung, die Einhaltung der Rahmenbedingungen sicherzustellen. [42, 37]

2.3.3 Modellbedingte Einflussfaktoren und Randbedingungen im virtuellen Verdrahtungsprozess

Eine Vielzahl von Randbedingungen für valide Lösungen eines Verdrahtungsergebnisses ist bereits durch den Bauraum festgelegt. Diese Randbedingungen umfassen eine Reihe an kabel- und umgebungsspezifischen Charakteristika. Eine zentrale Anforderung ist die Einhaltung der individuellen Biegeradien [37]. Der minimale Biegeradius r_{min} eines Kabels ist bedeutend, um Beschädigungen des Kabels während oder nach der Verlegung zu vermeiden [11]. Er wird berechnet, indem der Kabeldurchmesser D_{Kabel} mit einem spezifischen Biegeparameter b_{Kabel} für den Kabeltyp multipliziert wird [200]. Abbildung 2.7a zeigt ein Beispiel für die Unterschreitung des Mindestbiegeradius und die daraus resultierende Schädigung der elektrischen Verbindung. Abbildung 2.7b veranschaulicht die Wechselwirkung von Kabeln mit unterschiedlichen Spannungstypen. Zwischen den verschiedenen Kabel- und Spannungstypen entstehen Verletzungen der einzuhaltenden Mindestabstände in Bezug auf elektromagnetische Verträglichkeit [37]. Diese haben physikalische Auswirkungen, die eine ordnungsgemäße Funktion des Kabels einschränken. Als Beispiel lässt sich hier die kapazitive Kopplung anführen, die aufgrund der unterschiedlichen elektrischen Potentiale zwischen den Hochspannungs- und den Signalleitungen auftritt.

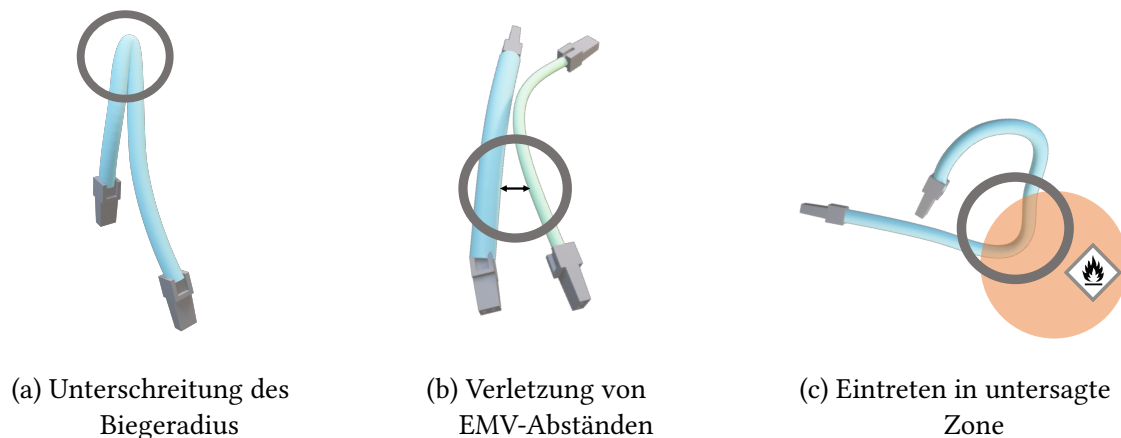
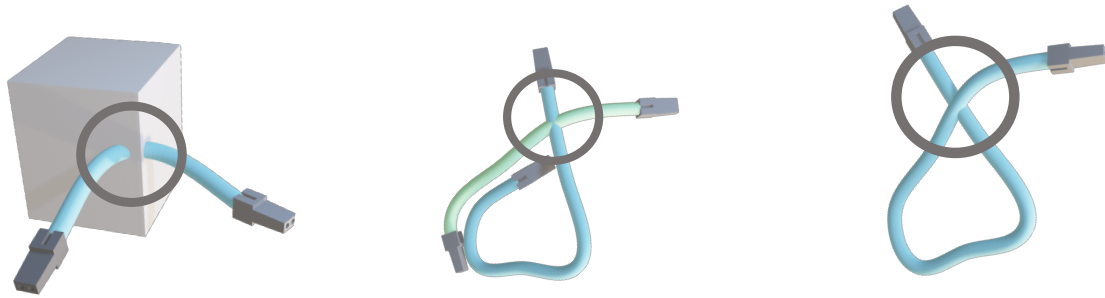


Abbildung 2.7: Exemplarische Darstellung von kabelspezifischen Einschränkungen in der virtuellen Verdrahtung

Die dritte kabelspezifische Randbedingung betrifft die Durchquerung von Sperrzonen im dreidimensionalen Raum (siehe Abbildung 2.7c). Dies kann durch eine potenzielle Entflammbarkeit des Bereiches oder andere Störgrößen begründet sein. Bestimmte Bereiche innerhalb des MCAD-3D-Modells können als gefährlich oder für andere Disziplinen vorbehalten definiert sein [187]. Als Gefahrenbereiche gelten im Rahmen der Verdrahtung insbesondere jene, die Einfluss auf die Materialbeschaffenheit und auf die Funktionsweise der elektrischen Verbindung haben oder sich signifikant auf deren Lebensdauer auswirken. Extremtemperaturen, Entflammbarkeit von Material oder Vibration sind hier exemplarisch zu nennen. Es existieren aber auch Bereiche, die explizit für die Verkabelung vorgesehen sind. Die von Entwicklern vorgesehenen Zonen sind so als bevorzugte Pfadabschnitte festgelegt. Je nach Anwendungsfall kann hier eine Differenzierung bis auf Kabeltypebene erfolgen. [45, 200]

Umgebungsspezifische Randbedingungen legen hingegen geometrische Einschränkungen des Bewegungsraumes der Kabel fest [209]. Während des Prozesses der virtuellen Verdrahtung treten diverse Kollisionen mit der mechanischen Geometrie (Einzelteile und Baugruppen), anderen Kabeln oder dem Kabel selbst auf. Sie werden unter anderem durch die Kollisionstypen innerhalb der MAPF definiert (vgl. Kapitel 2.1.3). Diese Kollisionsarten müssen bei der Pfadfindung vermieden werden, um ein valides Gesamtergebnis zu erreichen. Abbildung 2.8 zeigt Beispiele für die jeweiligen Kollisionsarten auf.

In Abbildung 2.8a ist die Überschneidung des Kabelpfades mit einem Festkörper der MCAD-Konstruktion zu sehen. Die Durchdringung eines Festkörpers in der virtuellen Verdrahtung im physischen Pendant ist nicht realisierbar und hat Planungsabweichungen zur Folge. Die Überschneidung von zwei unterschiedlichen Kabelpfaden ist in Abbildung 2.8b dargestellt. Dieser Kollisionstyp entspricht dem Vertex-Konflikt aus Kapitel 2.1.3. Den Sonderfall einer Kabel-Kabel-Kollision zeigt Abbildung 2.8c mit einer Eigenkollision eines Kabels, das sich selbst kreuzt.



(a) Komponenten-Kollision

(b) Kabel-Kabel-Kollision

(c) Kabel-Eigenkollision

Abbildung 2.8: Exemplarische Darstellung von möglichen Kollisionsarten in der virtuellen Verdrahtung

Diese wesentlichen Einflussfaktoren sind als Randbedingungen in der neuen Methode in Kapitel 4.2.3 zu berücksichtigen, um ein valides Gesamtergebnis zu gewährleisten. Zugleich repräsentieren die dargestellten Randbedingungen aus den Abbildungen 2.7 und 2.8 die Hauptgründe für einen nicht validen Kabelpfad in einer manuellen Festlegung der Wegstreckenpunkte einer elektrischen Verbindung [130].

2.3.4 Einfluss der Kinematik auf das Kabelrouting

Als spezieller Einflussfaktor auf die Validität der virtuellen Verdrahtung ist die Kinematik einzustufen. Im Gegensatz zur Mehrzahl der Randbedingungen aus Kapitel 2.3.3 führt hier die Ausführung der mechanischen Konstruktion zu einer besonderen Form der Komponenten-Kollision (vgl. Abbildung 2.8a). Diese kann auch zu einer Unterschreitung des kabelspezifischen Biegeradius führen (vgl. Abbildung 2.7a). Basierend auf den zugrunde liegenden modellbedingten Einflussfaktoren erfolgt eine Untergliederung der Kinematikauswirkungen in leitungsführende Geometrien und dynamische Störgeometrien [200]. In Abbildung 2.9 werden beispielhaft die beiden Fälle differenziert dargestellt und schematisch wird die Folge der Rotation um einen Winkel von 49,8 Grad veranschaulicht.

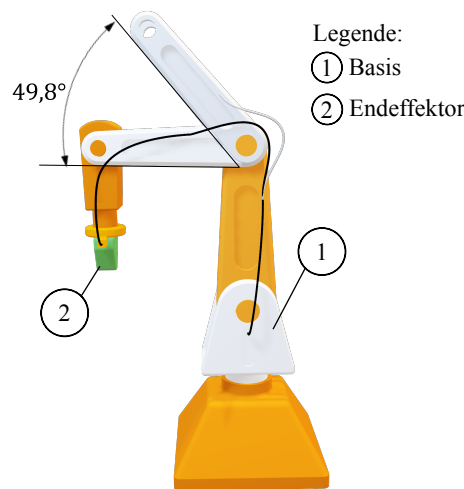


Abbildung 2.9: Einfluss der Kinematik auf die virtuelle Verdrahtung am Beispiel eines Roboters mit drei Gelenken

In Abbildung 2.9 ist ein Roboterarm mit drei Gelenken dargestellt, bei dem zwei elektrische Komponenten verbaut sind. Es ist festgelegt, dass eine elektrische Verbindung zwischen der Komponente an der Basis und der Komponente am Endeffektor herzustellen ist. In diesem Beispiel ist das mechanische Verbindungsstück als kabelführende Geometrie einzuordnen. Die beiden Extrempositionen des Gelenks ergeben sich aus dem Öffnungswinkel von 0 Grad zu 49,8 Grad. Die Beweglichkeit des Bauteils ist nach ISO 16792 mit einem separaten Konstruktionsmodell ausgeführt [36]. Die zweite Extremposition ist weiß dargestellt, um sie abzugrenzen. Die Auswirkung der jeweiligen Extremposition auf das Kabel kann sich in drei Ausprägungen zeigen. Erstens kann die Gesamtkabellänge nicht ausreichend sein, um den mechanisch geplanten Einstellwinkel zu ermöglichen. Zweitens können die Biegeradien an den Anschlussstellen überschritten werden, was zu Beschädigungen und Fehlfunktionen führen kann (vgl. Abschnitt 2.3.3). Drittens kann eine mechanische Beschädigung durch die rotierenden Bauteile auftreten, sofern die elektrische Verbindung deren Bewegungsraum kreuzt. Abbildung 2.10 veranschaulicht diesen limitierenden Faktor am Beispiel eines Kabels, das entlang eines Drehgelenks (en: *Revolute Joint*) verläuft.



Abbildung 2.10: Beispiel für die Auswirkung der Kinematik auf die virtuelle Verdrahtung bzw. die Kabellänge

Die in Abbildung 2.9 dargestellte kinematische Anforderung der mechanischen Baugruppe verdeutlicht die Notwendigkeit einer entsprechend größeren Kabellänge zwischen den jeweiligen Befestigungspunkten (en: *Clamps*) zur Realisierung der maximalen Joint-Ausprägung (vgl. Abbildung 2.10b). Diese Zusatzlänge bedeutet im Umkehrschluss einen Kabeldurchhang (en: *Slack*) in der neutralen Gelenkposition (vgl. Abbildung 2.10a). Die Einhaltung dieser Randbedingung, die aus der Konstruktion und der Kinematik hervorgeht, wird als Bestandteil der neuen Methode in Kapitel 4.2.4 berücksichtigt.

Eine Berücksichtigung der Bauteilbeweglichkeit als Störgeometrie oder als kabelführendes Bauteil findet zum derzeitigen Stand innerhalb der Veröffentlichungen zur virtuellen Verdrahtung noch keine Anwendung bzw. es finden sich keine Veröffentlichungen dazu (vgl. Kapitel 3). Vielmehr beruht die Integration dieser Randbedingung auf dem Fachwissen und den Vorerfahrungen des jeweiligen Elektrokonstruktors. Die mechanische 3D-Geometrie ist meist als statische Baugruppe in einer Neutralposition dargestellt (vgl. Kapitel 2.3.2). Eine Validitätsprüfung des CHRP-Ergebnisses ist Teil eines nachgelagerten Simulationsprozesses oder leitet sich aus den Erkenntnissen aus dem Prototypenbau ab. Die Informationen über den Bewegungsraum der Einzelkomponenten und deren kinematische Ketten liegen üblicherweise bereits zum Zeitpunkt der virtuellen Verdrahtung vor. Joints schränken die Bewegungsmöglichkeiten über sogenannte kinematische Bindungen ein und können als starre Gelenke, Führungen, Lagerungen oder Regeleinrichtungen ausgeprägt sein [23, 171]. Die wesentlichen Joint-Typen und ihre Freiheitsgrade sind in der Literatur ausführlich beschrieben [107]. Der notwendige Informationsgehalt für die Unterscheidung nach Bewegungen ohne Rotation (Translationsbewegungen) und Drehbewegungen (Rotationsbewegungen) zeigt sich in den zugehörigen Matrizen M_T und M_R (vgl. Matrizen 2.5) [171].

$$M_T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}, M_R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \quad (2.5)$$

In den Matrizen 2.5 sind t_x, t_y, t_z in M_T die Verschiebungswerte in der x-, y- und z-Achse der Koordinaten. Der Winkel α repräsentiert den Rotationswinkel als Bogenmaß

in der Matrix M_R für eine Rotation um die x-Achse. Die Gesamttransformationsmatrix M_G beschreibt die Kombination von Rotation und Translation mit $M_G = M_R \cdot M_T$ in Form einer 4×4 -Matrix. In Kapitel 4 findet die gängige verkürzte Schreibweise $T(A, B) = [R_\alpha | d]$ Anwendung, welche die Transformation von Bauteil A zu Bauteil B über die Rotationsmatrix R_α mit dem Winkel α und dem 3D-Translationsvektor d beschreibt. Die entwickelte Methode in Kapitel 4.2.4 nutzt die Integration von Rotation und Translation in offenen kinematischen Ketten als Einflussfaktor auf die virtuelle Verdrahtung. [107]

2.3.5 Prozessbeschreibung für eine virtuelle Verdrahtung

Der Teilprozess der virtuellen Verdrahtung ist ein besonders iterativer und repetitiver Bestandteil des gesamten Produktentstehungsprozesses. Aufbauend auf den Eingangsgrößen, die in Abschnitt 2.3.2 beschrieben sind, wird in Abbildung 2.11 der grundlegende Ablauf der virtuellen Verdrahtung im 3D-CAD-System dargestellt. [37, 45, 200]

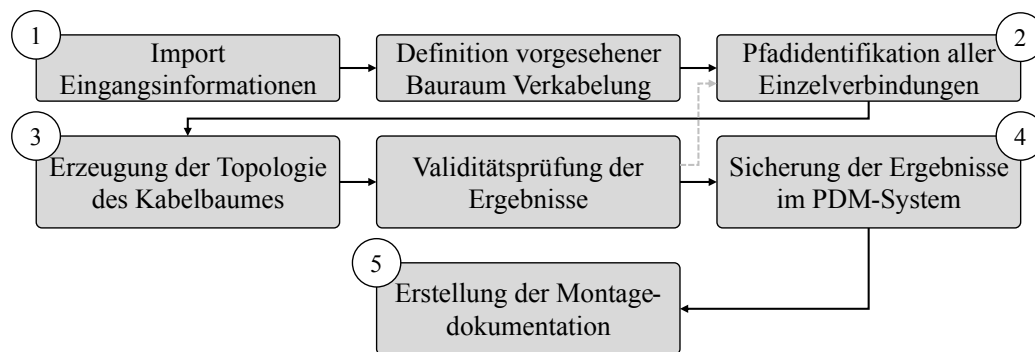


Abbildung 2.11: Prozess der virtuellen Verdrahtung im 3D-CAD-System i. A. a. [37] und [200]

Der Prozess beginnt mit dem Import der notwendigen Daten aller elektrischen und nicht elektrischen Komponenten in das 3D-CAD-System. Diese werden um Informationen zu den erforderlichen elektrischen Verbindungen ergänzt (vgl. Kapitel 2.3.2). Im 3D-CAD-Modell erfolgt je nach Anwendungsfall auch die Vordefinition einer Bauraumreservierung in der mechanischen Geometrie. Den wesentlichen Bestandteil des Prozesses bildet die Pfadfestlegung unter Berücksichtigung aller relevanten Randbedingungen (vgl. Kapitel 2.3.3) [41]. Dies geschieht durch die Verwendung einer Logik zur Definition der Einzelpfade, um die Topologie des Kabelbaums zu erzeugen. Dem folgen eine Validitätsprüfung des Zwischenergebnisses und eine damit zusammenhängende Überarbeitungsschleife. Die identifizierten Verbindungen finden sich in Form von Wegstreckenpunkten einzelner B-Splines im MCAD wieder (vgl. Kapitel 2.2.1). Die Sicherung bzw. Speicherung der Ergebnisse des virtuellen Verdrahtungsprozesses z. B. in einem PDM-System bildet den nächsten Schritt. Mit der Erstellung der erforderlichen Dokumentation für die Vorkonfektionierung und -montage der Kabel schließt der Prozess ab [37]. Nachfolgend werden die jeweiligen Prozessschritte im Detail beschrieben (vgl. Unterpunkte 1. bis 6.)

1. Import und Verarbeitung der Eingangsgrößen

Die in Kapitel 2.3.2 beschriebenen Daten liegen meist in heterogenen Datenformaten vor und werden innerhalb des ersten Prozessschrittes in das MCAD-System importiert. Da die Standardformate der jeweiligen Disziplinen nicht kompatibel sind (vgl. Kapitel 2.2.5), besteht ein Medienbruch [41]. Eine Zusammenführung der Informationen aus den einzelnen Autorensystemen ist zur Schaffung einer konsistenten Datenbasis erforderlich. Die Harmonisierung der Daten ist davon abhängig, ob die Referenzidentifikatoren der 3D-CAD-Geometrie mit dem elektrischen Schema übereinstimmen [45]. Inkonsistente oder fehlende Daten müssen identifiziert und bereinigt bzw. ergänzt werden. Erst dann ist das Erstellen eines validen Ausgangsmodells im 3D-CAD-System möglich. In Abhängigkeit vom verkabelnden Objekt bzw. Produkt oder von der Branche entspricht die Zusammenfassung der Einzelverbindungen in Bündel der gängigen Arbeitspraxis. Folglich findet die virtuelle Verdrahtung nicht mehr auf Einzelverbindungs-, sondern auf Bündelebene statt. Diese Zusammenführung und die damit einhergehende Simplifizierung der virtuellen Verdrahtung lässt sich in der Erstellung eines Kabelbaumes zusammenfassen [206]. In den Abbildungen 2.12a und 2.12b wird diese Herangehensweise exemplarisch im 3D-CAD-Modell und auf schematischer Ebene mit zugehöriger Terminologie dargestellt (vgl. Kapitel 2.1.4).

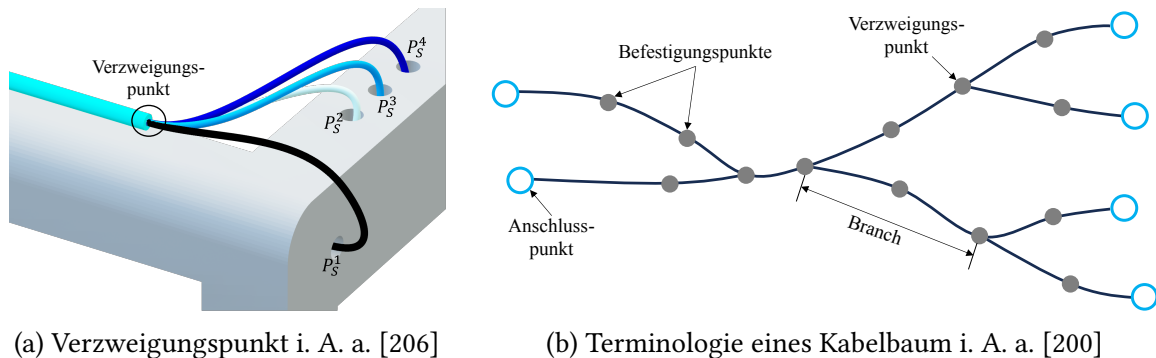


Abbildung 2.12: Beispiele für zwei mögliche Darstellungsformen eines Kabelbaums in der virtuellen Verdrahtung

Zur Verbindung der jeweiligen Anschlusspunkte innerhalb des 3D-CAD-Modells bestehen die Herausforderungen im Kabelbaumdesign darin, die Befestigungs- und Verzweigungspunkte (en: *Branching Point*) festzulegen [42]. Der Bereich zwischen den Verzweigungspunkten wird hierbei als Ast (en: *Branch*) bezeichnet (vgl. Abbildung 2.12b). In der exemplarischen Darstellung in Abbildung 2.12a verlaufen die Einzelverbindungen 1 bis 4 in einem Verzweigungspunkt zusammen. Ab diesem Zeitpunkt findet für sie keine individuelle Betrachtung mehr statt, sondern sie sammeln sich im entsprechenden Branch für die Pfadfindung durch die dreidimensionale Geometrie [45]. Diese Vorgehensweise der Bündelbetrachtung reduziert zwar einerseits die Problemkomplexität und den manuellen Arbeitsaufwand, führt aber andererseits zu einer Reduzierung der Planungsgranularität [73, 200]. Eine Differenzierung nach der Position einer Verbindung innerhalb des Bündels ist damit nicht mehr möglich. Die Festlegung der Branching-Punkte, mitsamt der Bün-

delung von Einzelverbindungen, bildet den nahtlosen Übergang in den Teilprozess der Pfadidentifikation. [206]

2. Pfadidentifikation aller Einzelverbindungen

Die Identifikation der Wegstreckenpunkte innerhalb der mechanischen 3D-CAD-Geometrie ist die zentrale Aufgabe bei der virtuellen Verdrahtung und erfordert einen hohen zeitlichen Aufwand [91]. Dabei muss für jede elektrische Verbindung ein gültiger Pfad bestimmt werden, der alle relevanten Randbedingungen berücksichtigt (vgl. Kapitel 2.3.3) [37]. Neben den physikalischen Beschränkungen sind in der Regel domänenspezifische Vorschriften oder firmeninterne Richtlinien zu beachten. Je nach Produktcharakteristik wird dies mit der Festlegung von Fixpunkten an der mechanischen Geometrie für Einzelverbindungen oder Kabelbündel realisiert. Die Komplexität der Identifikation der Fixpunkte hängt stark von den Parametern der Verbindungsanzahl, ihren wechselseitigen Beeinflussungen und den geometrischen Restriktionen ab. Die freie Geometrie wird durch die Identifikation des zulässigen Verlegeraumes beschränkt [148]. Dieser Raum ist sowohl durch die MCAD-Geometrie als auch durch die produktspezifischen Eigenschaften (vgl. Kapitel 2.3.3) beschränkt. In der Regel erfolgt die Pfadplanung zunächst für die Verbindungen mit den größten Kabeldurchmessern und den weitreichendsten Restriktionen der zulässigen Biegeradien. Mehrere Korrekturzyklen sind unerlässlich, um ein gültiges Gesamtlayout zu erreichen, das durch eine iterative und mitarbeiterindividuelle Arbeitsweise geprägt ist [187]. Je nach mechanischen, strukturellen oder positionsabhängigen Faktoren können weitere Anpassungen erforderlich sein, die bis zu einer vollständigen Neuplanung während der Pfadidentifikation reichen können. Im laufenden Prozess besteht zusätzlich der Einfluss durch eine kontinuierliche Weiterentwicklung der Eingangsinformationen. So kann aufgrund von mechanischen Anpassungen ein virtuelles Verdrahtungsergebnis bei seiner Fertigstellung bereits überholt sein [43].

3. Übertragung der Pfade in ein 3D-CAD-System und Erzeugung der Topologie

Nach der manuellen Festlegung der Wegstreckenpunkte im MCAD erfolgt die möglichst realitätsgetreue manuelle Abbildung der elektrischen Verbindungen im MCAD-System. Dies umfasst neben der Transformation der manuell festgelegten B-Spline-Kurven in 3D-CAD-Objekte mit einem kabelspezifischen Durchmesser (vgl. Kapitel 2.2.1) auch die Auswahl der zu wählenden Farbe. Ergänzungen durch anzubringende Stecker, Kabelschlänche oder andere Hilfsgeometrien bilden weitere Bestandteile des vierten Prozessschrittes in Abbildung 2.11 [170]. Ziel ist die Erzeugung eines möglichst realitätsnahen digitalen Mock-ups, das Aufschluss über die Realisierbarkeit des elektrischen Designs in der MCAD-Geometrie und die dafür notwendigen Kabellängen gibt [200].

4. Verwaltung und Sicherung der Verdrahtungsergebnisse im Produktdatenmanagement-System

Die Sicherung und die anschließende Möglichkeit zur Versionierung der Ergebnisse des virtuellen Verdrahtungsprozesses erfolgt aus mechanischer Perspektive z. B. in einem

PDM-System [176]. Als Bestandteil der übergeordneten Gesamtbaugruppe oder als eigenständiger Part der Baugruppe fließt die Topologie in den Speicherprozess der virtuellen Produktentwicklung ein [5]. Aufgrund der hohen Wechselwirkungen zur mechanischen Konstruktion und wegen der mitunter täglichen Änderungen ist vor allem die Verwaltung bei Überarbeitungen entscheidend [159, 171]. Für die Sicherstellung der Datenkonsistenz zu den jeweiligen mechanischen Entwicklungsständen und den Produktvarianten findet darüber hinaus für die Verdrahtung selbst eine Versionierung des jeweiligen Entwicklungsstands statt [44]. Den Kabelpfaden werden dabei die zugehörigen Kabelisolierungen, -befestigungen und Hilfsgeometrien zugeordnet, die der Montage und der Fixierung dienen [43]. In Abhängigkeit von der Anwendungsdomäne wird die Verdrahtung auch einzelnen mechanischen Funktionsbaugruppen zugeordnet oder orientiert sich an der Strukturierung der Manufacturing Bill of Materials (MBOM) des jeweiligen Produktes für die Montage. Dies ist eng verknüpft mit der nachfolgenden Bereitstellung der Daten auf dem Shopfloor. Aus datentechnischer Perspektive werden die Informationen zu Kabeltyp, Leitungslänge und Aderendhülsen im jeweiligen Format für eine Kabelvorkonfektionierung gesichert und anhand des Freigabeprozesses an vollautomatische Kabelverarbeitungsmaschinen auf den Fertigungs- und Montageflächen übertragen.

5. Erstellung der Montagedokumentation

Im Prozess der virtuellen Verdrahtung ist auch das Bereitstellen der Informationen für die Montage der Einzelverbindungen und Kabelbünde zu berücksichtigen. Hierfür besteht kein etablierter Standard, jedoch haben die existenten Arbeitsweisen gemein, dass sie von zweidimensionalen Daten aus dem ECAD geprägt sind. Da alle Kabelbaumdaten, einschließlich der kompletten Stückliste (BOM), in einem konsistenten ECAD-Modell zusammengefasst sind, können sie für nachgelagerte Prozesse in geeigneten Dateiformaten exportiert werden [45]. Während sich in Teilen der mechanischen Fertigung und Montage zunehmend eine 3D-modellbasierte Arbeitsweise etabliert, ist eine Weitergabe von Tabellen und Bildschirmaufnahmen (en: *Screenshots*) in der elektrischen Montage üblich [7]. Der Informationsgehalt beschränkt sich zumeist auf die BMKZ der Start- und Zielkomponenten, die Merkmale der zu verlegenden Kabeltypen sowie die 2D-Schaltpläne [45]. Eine Ergänzung von abstrahierten Pfadstrecken über die Nennung von Zwischenkomponenten wie Kabelkanälen existiert vereinzelt, entspricht jedoch nicht der Norm. In der mechanischen Fertigungs- und Montagedokumentation mehrten sich Ansätze zur Bereitstellung von dreidimensionalen geometrischen Darstellungen zu den zu erfüllenden Arbeitsschritten je Arbeitsplatz. Eine gängige Bereitstellung der Daten für eine Vormontage der Verdrahtung ist die Umwandlung der Kabelbünde in eine zweidimensionale Kabelbaumdarstellung (2D-Zeichnung), auch Nagelbrettzeichnung genannt (vgl. Abbildung 2.12b) [170, 206]. Sie gibt wieder, wie jedes Bündel und jede Einzelader verläuft und angeschlossen ist. Der Zusammenführungspunkt der einzelnen Äste des Kabelbaumes entspricht auch hier dem Branching-Punkt (vgl. Abbildung 2.12b). Ausgehend davon erfolgen die Vorkonfektionierung der Einzelverbindungen und Bündel sowie deren physischer Einbau in die mechanische Konstruktion. Ein Rückfluss der Informationen zur Validität und Realisierbarkeit des virtuellen Verdrahtungsergebnisses aus der Montage erfolgt lediglich sporadisch. Eine Ausnahme bildet der Fall, wenn ein Prototyp existiert, aus dem die Daten

explizit ermittelt werden. Dies hat zur Folge, dass das reale Produkt und die zugehörige virtuelle Planung mitunter stark divergieren [172, 130].

2.3.6 Virtuelle Verdrahtung als 3D-Pfadoptimierung

Durch die Darstellung der Start- und Zielpunkte elektrischer Verbindungen als dreidimensionale Punkte mit Koordinatensystemen und die Betrachtung der mechanischen Geometrie als Bewegungsraum wird das CHRP auf eine höhere Abstraktionsebene gehoben [206]. So lässt sich dieses Problem analog zum Pipe-Routing als eine 3D-Pfadoptimierung beschreiben (vgl. Kapitel 2.1.1) [200, 187, 37]. Aufgrund der Planung von mehreren Kabeln mit wechselseitigem Einfluss liegt eine MAPF vor (vgl. Kapitel 2.1.2). Dabei ist ein elektrischer Anschlusspunkt einer einzelnen Verbindung mit dem Startpunkt P_S bzw. Zielpunkt P_T eines formalisierten Shortest-Path-Problem gleichzusetzen (vgl. Kapitel 2.1) [90]. Innerhalb des mechanischen Bewegungsraumes ergeben sich auf Grundlage der Hindernisse mögliche weitere, diskrete Positionen als Knoten eines Graphen [37]. Die Kantengewichte basieren hier beispielsweise auf der gewichteten, euklidischen Distanz zwischen zwei konkreten Positionen. Im Falle einer Betrachtung des Bewegungsraumes als Voxel (vgl. Kapitel 2.2.3) ist der Aktionsraum eines Agenten zwar ebenfalls durch die Hindernisse beschränkt, jedoch häufig feingranularer unterteilt [138, 97]. Dies hat zur Folge, dass eine exaktere Pfadplanung möglich ist, die jedoch stärker von den 3D-CAD-Objekten im Raum losgelöst ist. Unter der Prämisse einer hohen Granularität ist mit einem starken Anstieg der Berechnungsdauer in einer zellenbasierten Umgebung zu rechnen [90]. Durch den hohen Komplexitätsgrad der MAPF im dreidimensionalen Raum bestehen vermehrt Lösungsansätze unter Anwendung von Methoden des Machine Learning (ML). Das als *NP-schwer* eingestufte Problem wird von diesen Methoden nach einem gegebenen Optimierungsziel approximiert. Eine exakte, optimale Lösung kann lediglich in kleinen Szenarien berechnet werden. Aufgrund der Beschaffenheit des Anwendungsgebietes bietet sich mitunter die Integration eines Kabelbaumes in die Pfadoptimierung an (vgl. Kapitel 2.3.5), um die Pfade zu einem Kabelbaum zu aggregieren [45]. Im Kontext der neuen Routing-Methode in Kapitel 4.1.1 wird dieser Ansatz für die Skalierbarkeit der Lösung eingeführt.

2.4 Grundlagen des maschinellen Lernens und künstlicher neuronaler Netze

Aufgrund der vielfältigen Definitionen der Begrifflichkeit künstliche Intelligenz (KI) ordnet dieses Kapitel die Begrifflichkeit der Künstlichen Intelligenz in Abschnitt 2.4.1 ein. Die zugehörigen Lernmethoden des ML werden in diesem Kapitel ebenfalls analysiert. Abschnitt 2.4.2 detailliert den Aufbau künstlicher neuronaler Netze, die als Grundlage für Methoden des Supervised Learnings in Abschnitt 2.4.3 dienen. Methoden zur Gruppierung von Datensätzen werden in Abschnitt 2.4.4 beschrieben.

2.4.1 Definition des Begriffs der künstlichen Intelligenz

Der Terminus der *Künstlichen Intelligenz* steht repräsentativ als Sammelbegriff für die Teilbereiche des ML. Ergänzt werden diese um die Ausprägungen des Natural Language Processing, Teilbereiche der Robotik und der Verarbeitung visueller Daten [86]. Im Rahmen dieser Arbeit liegt der Fokus auf dem Anwendungsgebiet des ML. Abbildung 2.13 untergliedert die Begrifflichkeit des ML nach der gängigsten Klassifizierung in Supervised Learning (SL), Unsupervised Learning und Reinforcement Learning (RL) [86, 87]. Farblich hervorgehoben sind die relevanten Teilbereiche und deren Anwendung zur Entwicklung der neuen Methoden in den Kapiteln 4.4.2 und 4.5.1.

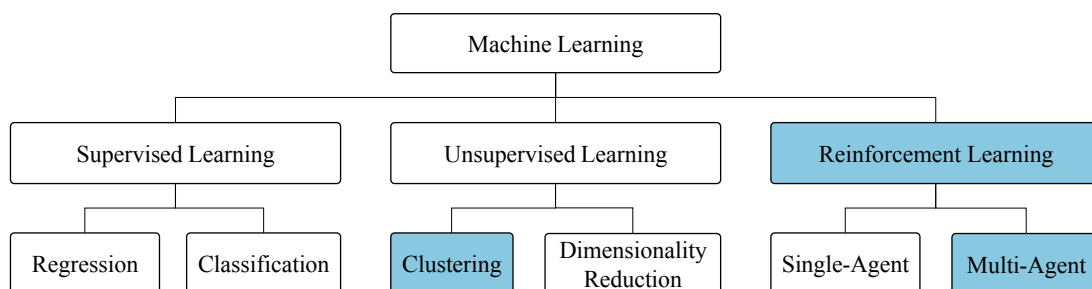


Abbildung 2.13: Arten von Methoden des maschinellen Lernens i. A. a [86]

Die ML-Methode des RL wird aufgrund der Relevanz für die neue Methode in Kapitel 2.5 genauer analysiert. Im Anwendungsfall des CHRP kommt ein Multi-Agenten-Ansatz zum Einsatz, der die Skalierbarkeit, Anpassungsfähigkeit und Generalisierbarkeit der neuen Methode sicherstellt (vgl. Kapitel 4.5) [38, 104]. In der neuen Methode in Kapitel 4.4.1 und in der Anwendung der KI-Methode für den Kabelbaum finden darüber hinaus Clustering-Methoden aus dem Bereich des Unsupervised Learnings Anwendung [10]. Daher sind auch diese methodischen Bestandteile in Abbildung 2.13 als besonders relevant hervorgehoben. Die methodischen Gebiete des Supervised Learnings sind im Rahmen dieser Arbeit weniger bedeutend, da ein Lernprozess auf Grundlage historischer Daten an deren Verfügbarkeit und Qualität geknüpft ist [131, 59, 86]. Gerade im Zusammenhang der virtuellen Verdrahtung von Neuentwicklungen ist dies als signifikante Einschränkung einzustufen.

2.4.2 Aufbau eines künstlichen neuronalen Netzes

In der Literatur finden in ML-Ansätzen gängigerweise künstliche neuronale Netze Anwendung. Diese spiegeln einen Paradigmenwechsel hin zum Deep Learning wider, der mit der Lösung nicht trivialer Problemstellungen einhergeht [123, 38]. Sie ähneln der Funktionsweise des menschlichen Nervensystems und vernetzen zur Verarbeitung von Eingabewerten Neuronen untereinander [3]. Das Netz dient der Approximation von Funktionen unter Verwendung des Parametersets θ [157]. In der neuen Methode in Kapitel 4.5.1 sind hierzu vor allem die Untertypen des Multi-Layer Perceptron (MLP) und des CNN relevant [59, 124].

1. Multilayer Perceptron (MLP)

Das MLP ist durch die Aneinanderreihung von mehreren Neuronenschichten (en: *Layer*) charakterisiert, welche die eingegebenen Daten verarbeiten [86]. Abbildung 2.14 veranschaulicht den strukturellen Aufbau dieses Netztyps schematisch.

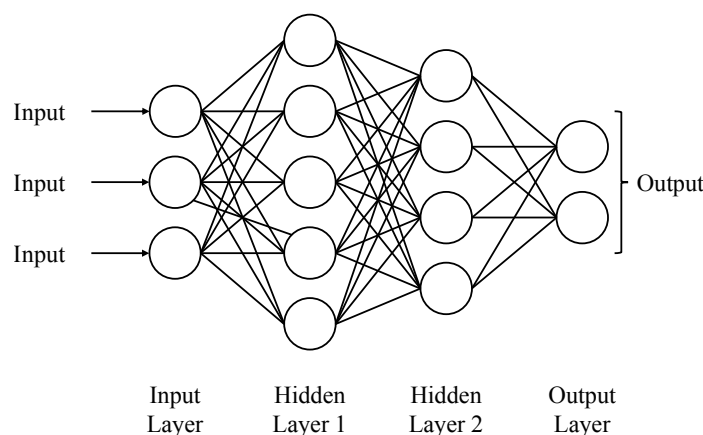


Abbildung 2.14: Schematische Darstellung zum Aufbau eines Multilayer Perceptron i. A. a. [87]

Die grundlegenden Bausteine eines MLP sind die drei unterschiedlichen Arten von Layern, aus denen dieser Typ eines neuronalen Netzes besteht [123]. Die Übergabe der Eingangsinformationen fließt in die Input-Ebene (en: *Input Layer*) ein. Der Einfluss dieser Ebene auf den Lernprozess ist als gering einzustufen. Die Aufgabe der Schicht liegt lediglich darin, den entsprechenden Input an die nachfolgenden Layer weiterzugeben [50]. Eine Veränderung der Daten findet an dieser Stelle nicht statt [3]. Die mögliche Ausprägung der Eingabe kann vielfältig sein und neben Skalaren, Vektoren oder Matrizen auch als Tensor beschaffen sein [59]. Essenziell ist lediglich, dass die Anzahl der Input-Neuronen mit jener der Eingangsinformationen übereinstimmt [50, 104].

Die darauffolgenden Layer werden als verborgene Schichten (en: *Hidden Layer*) bezeichnet, da keine Interaktion mit Funktionen oder Daten außerhalb des neuronalen Netzes besteht [38, 3]. Die Hidden Layer ermöglichen eine hierarchisch fortgesetzte Transformation der Eingabemuster [59]. Damit lernen diese Layer, die Eingangsgröße auf eine

Art zu verarbeiten, dass der Ausgabewert des MLP einfacher vorhergesagt werden kann [99]. Sie lassen sich als eine Projektion der Eingabemuster in einen hochdimensionalen Vektorraum beschreiben - analog zu einem *Embedding* in der Sprachverarbeitung [87, 59]. Wenn die Verteilung der Inputdaten nichtlinear und schlecht konditioniert ist, kann eine Transformation in diesen höherdimensionalen Raum helfen, die Gesamtleistung des MLP zu verbessern [50, 38]. Mit steigender Dimensionalität der Hidden Layer ist jedoch auch der Trainingsprozess komplizierter und zeitintensiver, weshalb sorgfältig zwischen der zusätzlichen Komplexität und der Leistungsverbesserung abzuwägen ist [87]. Zwar ist bewiesen, dass aus mathematischer Perspektive ein einzelnes Hidden Layer ausreichend ist, um beliebige Zielfunktionen vorherzusagen, dennoch zeigt sich in der Praxis die Notwendigkeit mehrerer Schichten für die Lösung nicht trivialer Probleme [104].

Die abschließende Ausgabeschicht (en: *Output Layer*) bündelt und transformiert die für das Ergebnis relevanten Daten [59]. Sie nutzt üblicherweise eine Aktivierungsfunktion zur Durchführung einer Klassifizierung oder Regression (vgl. Kapitel 2.4.3) und stellt dem Nutzer das Ergebnis bereit [123, 3]. Beispiele für die genannte Aktivierungsfunktion sind Sigmoid oder Softmax [38, 87]. Die Herangehensweise zum Trainieren der einzelnen Netzwerkparameter θ zwischen der zufälligen Initialisierung und der laufenden Abweichungsreduzierung (en: *Loss*) ist in der Literatur ausführlich beschrieben [38, 3, 86].

2. Convolutional Neural Network (CNN)

Ein CNN stellt eine spezialisierte Form eines künstlichen neuronalen Netzwerks dar, das für die Verarbeitung und Analyse grafischer 2D- oder 3D-Daten ausgelegt ist [182, 83]. Der grundlegende Aufbau eines CNN ist in Abbildung 2.15 dargestellt. Dieser teilt sich in den Bereich der Feature Extraction und die anschließende Classification auf [87].

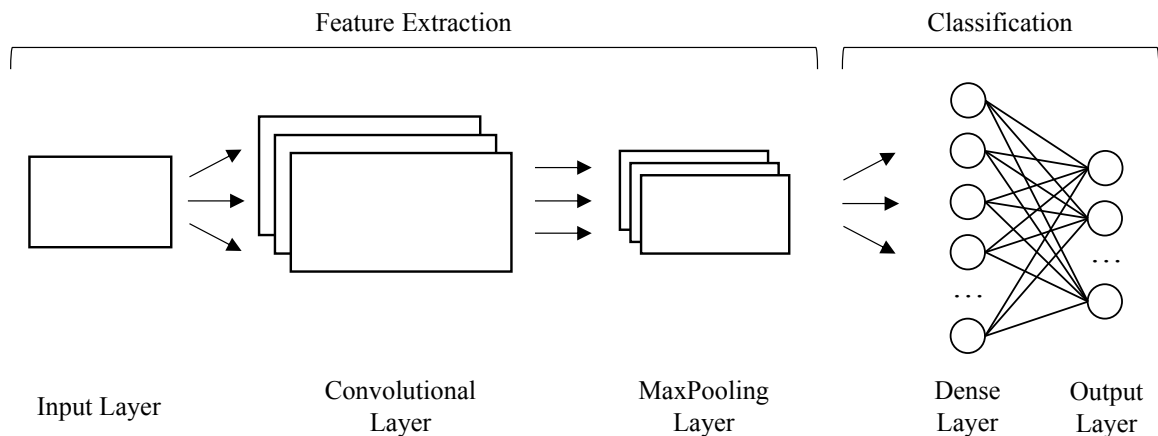


Abbildung 2.15: Prinzipieller Aufbau eines Convolutional Neural Networks

Als Teil der Feature-Extraction erfassen die Convolutional und Pooling Layer räumliche Beziehungen und Hierarchien von Merkmalen. Die Dense Layer lernen die nichtlinearen Kombinationen von Merkmalen. Daher können Aufgaben wie Bildklassifikation, 2D- oder 3D-Objekterkennung oder Bewegungserkennung mit CNN durchgeführt werden [113, 158]. Im Allgemeinen besteht ein CNN aus einem Input-Layer, einer beliebigen Anzahl

von Convolutional, Pooling und Dense Layern und einem Output Layer (vgl. Abbildung 2.15). Das Input-Layer, die Dense-Layer und das Output Layer entsprechen den Schichten analog zu einem MLP. [100]

Eine Faltungsschicht (en: *Convolutional Layer*) besteht aus einer Gruppe von trainierbaren Filtern oder Kernels [124, 122]. Jeder Filter entspricht dabei einer Matrix von Gewichten. Die räumliche Dimension eines Kernels ist in der Regel kleiner als die der Eingabe, um die Dimensionalität des Inputs zu reduzieren [158]. Die Tiefe der Schichten ist in der Regel äquivalent zur Eingabe. Im Kontext der Beschreibung einer einzelnen Schicht des CNN beschreibt die Tiefe keine räumliche Information, sondern die Anzahl der Kanäle (en: *Channel*) [3]. So hat z. B. ein RGB-Bild eine Tiefe von 3, die repräsentativ für die Farbkanäle der Primärfarben steht [38]. Die Breite und Höhe repräsentieren die räumliche Dimension. Der Filter bewegt sich folglich über die Eingabedaten und führt dabei elementweise Multiplikationen und Summierungen durch [29]. Abbildung 2.16 veranschaulicht diesen Prozess am Beispiel einer 5×5 -Matrix als Eingangsgröße [124].

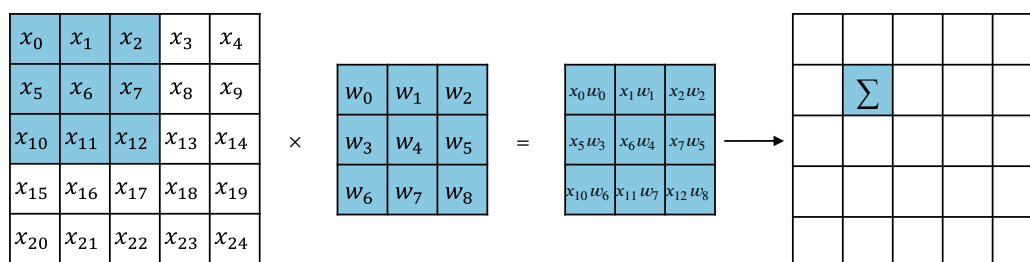


Abbildung 2.16: Beispielhafter Aufbau eines Convolutional Filters i. A. a. [124]

Das Ergebnis der Faltung eines Filters über die gesamte Eingabe ist eine Merkmalskarte (en: *Feature Map*) [122, 100]. Jede Feature-Map repräsentiert spezifische Merkmale oder Muster in der Eingabe [158]. Die Anzahl der Filter in einem Convolutional Layer bestimmt die Anzahl der verschiedenen Feature-Maps, die eine jeweilige Schicht ausgibt. Die räumlichen Dimensionen werden durch die Größe, den Schritt und das Padding (en: *Size, Stride and Padding*) der Faltungsoperation festgelegt [175, 38, 100]. Ähnlich wie bei Dense-Schichten finden nichtlineare Aktivierungsfunktionen Anwendung, um Nichtlinearität in die Operation einzubringen [123]. Während des Trainingsprozesses lernt das Netzwerk mittels *Backpropagation* die Gewichte der Filter, um relevante Merkmale zu extrahieren [99].

Im Anschluss an Convolutional Layer finden sich in vielen methodischen Beschreibungen Pooling-Schichten (vgl. Abbildung 2.15) [87, 100]. Deren Zweck besteht darin, die räumliche Ausdehnung der Feature-Maps zu reduzieren und gleichzeitig die wichtigsten Informationen beizubehalten [195]. Das MaxPooling Layer ist das am häufigsten verwendete Pooling Layer [158, 87]. Hierbei wird jede Feature-Map in kleine Regionen unterteilt und der höchste Wert jeder Region in der Ausgabe-Feature-Map beibehalten [124, 3]. Der Prozess des MaxPoolings ist in Abbildung 2.17 exemplarisch dargestellt.

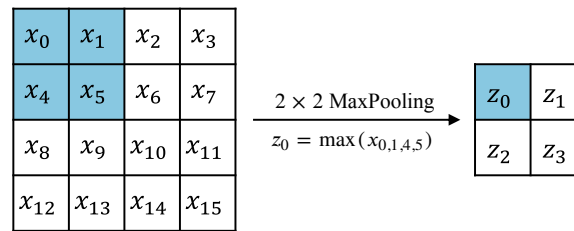


Abbildung 2.17: Beispielhafte Darstellung der Funktionsweise der MaxPooling-Operation

Das Pooling-Verfahren wird unter Beibehaltung der Anzahl an Feature-Maps auf jede Feature-Map unabhängig angewendet [99]. Das Downsampling reduziert dabei die Anzahl der Parameter und der Berechnungen im Netz, erhöht somit die Effizienz und verhindert eine Überanpassung (en: *Overfitting*) [123, 59]. Varianten des Pooling Layers extrahieren nicht den maximalen Wert innerhalb des Kernels, sondern beispielsweise den Durchschnittswert [182, 100]. Diese Abwandlung wird als Average-Pooling bezeichnet [38, 156].

Diese Architektur eines CNN weist den Vorteil auf, dass sie effizient mit größeren Eingaben operieren kann und dabei eine angemessene Anzahl von Parametern im Vergleich zu einem MLP beibehält [124]. Dies ist der Fall, da CNNs eine Reihe von Gewichtungen nutzen, die als Filter dargestellt und auf verschiedene räumliche Orte der Eingabe angewendet werden [100]. Ein MLP hingegen verwendet individuelle Gewichte für jede Verbindung des neuronalen Netzes (vgl. Abbildung 2.14). Eine Eignung von 3D-CNN für die Verarbeitung von dreidimensionalen Körpern zeigt sich exemplarisch in der Echtzeiterkennung von Objekten unter Anwendung der Voxel-Methode [113, 182]. Nachfolgend beschreibt die vereinfachte Syntax den Aufbau der jeweiligen CNN-Architekturbausteine. Die Funktion $Conv(f, d, s)$ nutzt f Filter der Dimensionen d mit einem Stride s . Die analoge Darstellung für Pooling Layer wird mit $Pool(m)$ simplifiziert, in der m den Bereich angibt. $Full(n)$ beschreibt ein Fully Connected Layer mit n Output-Neuronen [113]. Die in Abbildung 2.15 dargestellte CNN-Struktur lässt sich folglich auch exemplarisch mit der Syntax $In(x, y) - Conv(f, d, s) - MaxPool(m) - Full(n) - Out(o)$ vollständig beschreiben [113].

2.4.3 Methoden des Supervised Learnings

Das überwachte Lernen (en: *Supervised Learning*) ist die am weitesten verbreitete Teildisziplin des ML [59]. Diese Methode basiert auf der Prämisse, dass mit Hilfe einer hinreichenden Menge von *gelabelten* Trainingsdaten eine Funktion f approximiert werden kann, die der Zielfunktion f^d nahekommt oder mit ihr übereinstimmt [86, 104]. Diese Methode dient dem Zweck der Vorhersage von Werten Y auf der Basis von Eingangsinformationen X . Der Zielwert Y (en: *Label*) muss dabei stets definiert und verfügbar sein. Das SL ist das Lernparadigma, bei dem die Netzwerkparameter θ so optimiert werden, dass die Differenz zwischen der Zielausgabe und der berechneten Ausgabe minimiert wird [86]. Beim SL werden im Wesentlichen zwei Teilbereiche unterschieden, die sich auf die Architektur der angewendeten neuronalen Netze auswirken - Klassifikation (en: *Classification*) und Regression (en: *Regression*) [17].

Die binäre **Klassifikation** (en: *Binary Classification*) ist neben der Mehrfachklassifizierung (en: *Multiple Classification*) eine von zwei artverwandten Klassifikationsarten für $Y \in \{0, 1, \dots, c - 1\}$ [38]. Dabei ist c die Anzahl der Klassen [17]. Die Binary Classification zeichnet sich dadurch aus, dass die Entscheidung lediglich für $c = 2$ zu treffen ist [3]. Die **Mehrfachklassifizierung** hingegen erweitert die Anzahl der möglichen Klassen auf $c > 2$ [64]. Die jeweilige Anwendung des Klassifikationstypus zeigt sich auch in der Anzahl der Output-Neuronen des neuronalen Netzes. So befinden sich in einer Binärklassifizierung zwei Ausgangsneuronen im Output-Layer, während die Anzahl der Neuronen in der letzten Schicht der Mehrfachklassifizierung $c > 2$ ist [86].

Die **Regression** dient dagegen der Vorhersage einer kontinuierlichen Größe auf Basis der vorliegenden Eingangsgrößen X [59]. Dabei ist Y keine Menge diskreter Labels, sondern ein Wertebereich reeller Zahlen [50]. Es besteht die Unterscheidung zwischen univariater und multivariater Regression. Die univariate Regression dient der Vorhersage eines einzigen Wertes, wogegen die multivariate Regression mehr als einen Ausgabewert schätzt [86]. Insbesondere die multivariate Regression ist für die neuen Methoden in Kapitel 4 von Relevanz, da ein Richtungsvektor im kontinuierlichen dreidimensionalen Raum durch das künstliche neuronale Netz bestimmt werden soll (vgl. Kapitel 4.3.2).

Der Lernprozess im SL ist in zwei Phasen untergliedert. Während das Anlernen auf der Grundlage eines zur Verfügung stehenden Datensatzes einer Induktion entspricht, handelt es sich bei der Anwendung um eine Deduktion [38, 104]. Die Methode des Supervised Learnings steht im Kontrast zum Unsupervised Learning, das ohne die Existenz einer Labelzuordnung für die Gruppierung von Datensätzen auskommt [17, 64].

2.4.4 Methoden zum Clustern von Datensätzen

Die Grundintention der Anwendung von Cluster-Methoden besteht für die neue Methode in Kapitel 4 in der Reduzierung der Problemkomplexität. So sind die Kabelpfade mit räumlich ähnlichen Verläufen zu gruppieren (vgl. Kapitel 4.4.1). Unter dieser Voraussetzung beschreibt Abschnitt 2.4.4.1 das übergeordnete Ziel einer Clusteranalyse. In Abschnitt 2.4.4.2 folgt die Bestimmung der Ähnlichkeit von Kurven im dreidimensionalen Raum. Eine Unterscheidung zwischen hierarchischem und dichtebasiertem Clustering wird in Abschnitt 2.4.4.3 erläutert. Darauf bauen die Abschnitte 2.4.4.4 und 2.4.4.5 zur Analyse und zur Auswahl eines geeigneten Algorithmus sowie der optimalen Clusteranzahl auf.

2.4.4.1 Ziel der Clusteranalyse

Die Intention der Anwendung eines Clusterings als Teil der Methoden des Unsupervised Learnings (vgl. Abbildung 2.13) besteht in der Identifikation von Ähnlichkeiten oder Distanzen zur Erzeugung von Gruppierungen, beispielsweise von Kabeln [4, 66]. Der Ansatz fundiert auf der Prämisse, dass Daten, die einem gemeinsamen Cluster angehören, einen hohen Grad an Ähnlichkeit aufweisen [193, 17]. Die Vorgehensweise lässt sich in vier wesentliche Schritte unterteilen, die dem Ziel der Gruppierungsidentifikation dienen [10, 104]:

1. Bestimmung der Ähnlichkeit von Datenobjekten (vgl. Kapitel 2.4.4.2)
2. Gegenüberstellung der Clustering-Methoden (vgl. Kapitel 2.4.4.3)
3. Auswahl des geeigneten Clustering-Algorithmus (vgl. Kapitel 2.4.4.4)
4. Bestimmung der optimalen Clusteranzahl (vgl. Kapitel 2.4.4.5)

2.4.4.2 Bestimmung der Ähnlichkeit von Datenobjekten

Der erste Schritt der Clusteranalyse besteht in der Bestimmung der Ähnlichkeit zwischen den Datenobjekten, um eine Distanzmatrix zu erzeugen. Da für die Entwicklung der neuen Methode (vgl. Kapitel 4.5.1 und 4.4.1) neben dreidimensionalen Punkten auch Kurven im dreidimensionalen Raum auf ihre Ähnlichkeit hin untersucht werden, steht neben der Verwendung der euklidischen Distanz die Betrachtung der Fréchet-Distanz im Fokus [108, 20]. In der euklidischen Distanz $d_E(P, Q)$ zwischen den Punkten P und Q stehen $p_i, q_i \in \mathbb{R}$ für die Koordinaten der Punkte P und Q in der i -ten Dimension $i \in \{1, 2, 3\}$ (vgl. Gleichung 2.6) [191, 46].

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (2.6)$$

Die Fréchet-Distanz hat sich als ein wirkungsvolles Maß für die Bestimmung der Ähnlichkeit zwischen zwei Kurven im dreidimensionalen Raum etabliert. Sie ermöglicht einen umfassenden Vergleich der geometrischen Formen, indem sie die kontinuierliche Sequenz von Punkten entlang der Kurven erfasst und somit die Gesamtheit ihrer räumlichen Trajektorien berücksichtigt. Im Gegensatz zu einfacheren Distanzmaßen, die oft lediglich punktweise Unterschiede einbeziehen, ist die Fréchet-Distanz ein präziseres Maß dafür. Die Fréchet-Distanz $d_F(A, B)$ ist definiert als ein Ähnlichkeitsmaß zwischen zwei Kurven A und B . Sie berechnet sich nach der Gleichung 2.7. [19]

$$d_F(A, B) = \inf_f \max_{t \in [0,1]} \|A(t) - B(f(t))\| \quad (2.7)$$

In Gleichung 2.7 steht $d_F(A, B)$ für die Fréchet-Distanz zwischen den Kurven A und B . Die Funktion $f : [0, 1] \rightarrow [0, 1]$ repräsentiert eine stetige und monoton steigende Abbildung von dem Parameterintervall der einen Kurve A auf das Parameterintervall der anderen Kurve B . Der Term \inf_f bezeichnet das Infimum über alle kontinuierlichen und monotonen Abbildungen f . Das Maximum $\max_{t \in [0,1]}$ über alle t im Intervall $[0, 1]$ und $\|A(t) - B(f(t))\|$ misst die euklidische Distanz zwischen den Punkten auf den Kurven A und B zu einem gegebenen Zeitpunkt t . [20]

2.4.4.3 Hierarchische, dichtebasierte und partitionierende Clusteringverfahren

Die Taxonomie der Clusteringverfahren folgt der gängigen Unterteilung in die Überkategorien der hierarchischen und der partitionierenden Methoden [46]. Aufgrund der Relevanz

für die neue Methode in den Kapiteln 4.4.1 und 4.5.1 wird das dichtebasierte Vorgehen gesondert betrachtet [66].

Die Methoden des hierarchischen Clusterings lassen sich wiederum in zwei distanzbasierte Unterkategorien unterteilen. Auf Grundlage der gebildeten Zerlegung eines Datensatzes kann ein Algorithmus entweder als *agglomerativ* oder als *divisiv* eingestuft werden [108]. Beide Unterkategorien stellen eine konträre Herangehensweise dar, die implizit auf der Verwendung eines Dendrogramms basiert [191, 108, 46]. Während im divisiven Verfahren zunächst alle Partitionen einem Cluster zugehörig sind und diese dann schrittweise in immer kleinere Cluster untergliedert werden, nähert sich das agglomerative Verfahren dem Ergebnis über einen Bottom-up-Ansatz an [66, 3]. So bildet in der agglomerativen Methode zunächst jedes Datenobjekt ein eigenes Cluster, die dann sukzessive zu größeren Clustern zusammengefasst werden [10]. Grundlage für die Zusammenfassung ist in beiden Unterkategorien, dass Objekte mit einer geringen Distanz bzw. hohen Ähnlichkeit zusammengehören [17]. Beide Verfahren sind in ihrer Herangehensweise sehr strikt, da gebildete Cluster unveränderlich sind und einzelne Objekte nicht mehr vertauscht werden können. Stattdessen verfeinern (*divisiv*) bzw. verallgemeinern (*agglomerativ*) sie ihre Clusterstruktur nur noch [191]. Dabei ist festzuhalten, dass die Anzahl der Cluster nicht a priori definiert sein muss. Eine Gegenüberstellung der beiden hierarchischen Methoden zeigen die Abbildungen 2.18a und 2.18b.

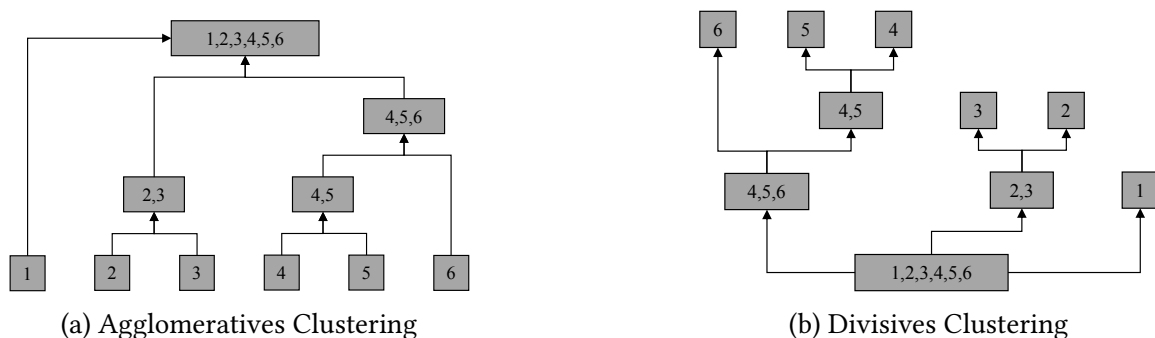


Abbildung 2.18: Agglomeratives und divisives Clustering

*Dichte*basierte Methoden (vgl. Abbildung 2.19) zeichnen sich hingegen dadurch aus, dass sie die Objekte nach einer Dichte in einem multidimensionalen Raum unterteilen [66]. Dabei sind die Gruppierungen durch Gebiete mit einer geringeren Dichte getrennt [145, 135]. Das Verfahren orientiert sich an der Überschreitung eines vordefinierten Grenzwertes ϵ . Dieser gibt den maximalen Abstand eines Objektes zum Clusterzentrum (en: *Core Point*) an. Überschreitet ein Datenobjekt den Wert ϵ zu allen benachbarten Datenobjekten, so gilt dieser als Teil des Rauschens (en: *Noise*) [178, 146].

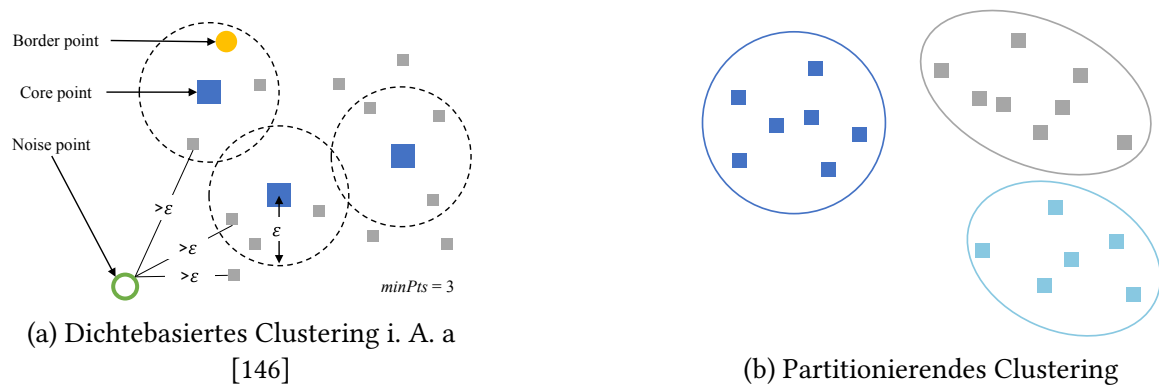


Abbildung 2.19: Schematische Darstellung von dichtebasierten und partitionierenden Clustering-Methoden

Die allgemeine Idee der dichtebasierten Algorithmen besteht darin, ein bestimmtes Cluster so lange mit Datenobjekten zu vergrößern, bis die Dichte (Anzahl der Datenpunkte) in der Nachbarschaft einen zu definierenden Schwellenwert überschreitet [66, 46]. Im Gegensatz zu den meisten anderen Clustering-Algorithmen können dichtebasierte Clustering-Algorithmen zur Erstellung von Clustern mit beliebiger Form verwendet werden [108]. Häufig angewendete Algorithmen dieser Gruppierungsmethoden sind Density-Based Spatial Clustering of Applications with Noise (DBSCAN) und Ordering Points to Identify the Clustering Structure (OPTICS) [193]. Als wesentliche Vorteile dieser Methoden zeigen sich eine hohe Resistenz gegenüber *Outlier* oder *Noise* sowie die Möglichkeit zur Verarbeitung von nichtlinearen Clustern unterschiedlicher Größe und Form [46, 146]. Vor allem die nicht notwendige Vordefinition der Clusteranzahl ist in der praktischen Anwendung ein positiver Aspekt. Dem steht die Herausforderung der Ermittlung von optimalen Werten für den Radius ϵ und $minPts$ bei unbekannten Daten gegenüber [139, 191].

Das partitionierende Clusteringverfahren (vgl. Abbildung 2.19b) unterteilt einen vorliegenden Datensatz aus N Punkten in \mathcal{P} ($N \geq \mathcal{P}$) Partitionen, von denen jede ein Cluster darstellt [66, 108]. Abbildung 2.19b veranschaulicht diesen Prozess. Jede Gruppierung unterliegt dabei den Anforderungen von mindestens einem Datenobjekt je Cluster und der Zugehörigkeit eines Datenobjektes zu ausschließlich einem Cluster [85, 66]. Die zugrunde liegende Idee besteht darin, eine iterative Verschiebungstechnik zu verwenden, die versucht, die Partitionierung zu verbessern, indem Objekte von einer Gruppe in eine andere verschoben werden, bis ein entsprechendes Abbruchkriterium erreicht ist [193, 46]. Ein Alleinstellungsmerkmal der *partitionierenden* Verfahren ist die hohe Variabilität. So können bei den partitionierenden Methoden alle Elemente beliebig von einem Cluster in ein anderes verschoben werden, was eine signifikante Einschränkung der hierarchischen Verfahren darstellt [10]. Dem gegenüber steht die notwendige Definition der Clusteranzahl \mathcal{P} [46]. Als bekanntester Vertreter dieser Methoden ist der k-means-Algorithmus zu nennen [108].

2.4.4.4 Auswahl des Clustering-Algorithmus

Aufgrund der stark divergierenden Ansätze hängt die Auswahl eines geeigneten Clustering-Algorithmus neben der Beschaffenheit der Daten, auch von den zu definierenden Eingangsgrößen ab. Während bei den hierarchischen Verfahren keine A-priori-Definition der Clusteranzahl K notwendig ist, sind die Cluster in ihrer Datenzuordnung unveränderlich [10]. Im Gegensatz dazu erfordern partitionierende Verfahren eine Vordefinition der Clusteranzahl \mathcal{P} , weisen aber eine damit einhergehende Variabilität im laufenden Prozess auf [10]. Das dichtebasierte Clustering erfordert neben dem Distanzschwellwert ϵ eine Mindestanzahl von Datenobjekten pro Cluster *minPts* [193].

Unter Nutzung der Ergebnisse der durchgeführten Analyse ergibt sich für die neue Methode in Kapitel 4 die Verwendung von zwei Unterkategorien. So wird für die Identifikation ähnlicher Pfadverläufe in der 3D-CAD-Geometrie ein dichtebasierter Ansatz angewendet (vgl. Kapitel 4.4.1). Da für die eingesetzte Kommunikationsmethode (vgl. Kapitel 2.5.3) eine feststehende Mindestanzahl von Agenten erforderlich ist, entspricht dies der Eingangsgröße *minPts*. Die Bestimmung eines optimalen Wertes ϵ bleibt in diesem Kontext als Herausforderung bestehen [193]. Der zweite Anwendungsfall für einen Clustering-Algorithmus ist die Erstellung eines Kabelbaumes für eine skalierbare Methode (vgl. Kapitel 4.5.1). Hier wird eine Gruppierung von Kabelpfaden auf Grundlage ihrer Start- und Zielpunktkombinationen vorgenommen. Diese fließen in die Bestimmung der Bündelungspunkte innerhalb des dreidimensionalen Raumes ein (vgl. Kapitel 2.3.5). Ausschlaggebendes Kriterium ist die Variabilität im Clusteringprozess, durch die sich Datenobjekte flexibel zwischen Gruppierungen verschieben lassen [193]. Die datentechnische Notwendigkeit einer vordefinierten Clusteranzahl K deckt sich dabei mit der domänenspezifischen Definition einer Bündelanzahl und wird daher in der neuen Methode in Kapitel 4.5.1 nicht als Einschränkung angesehen [46].

2.4.4.5 Bestimmung der optimalen Clusteranzahl

Bei der Identifikation einer optimalen Anzahl von Clustern ist zwischen der Homogenität der Datenpunkte innerhalb eines Clusters und der Handhabbarkeit der Anzahl der Cluster abzuwägen [108, 104]. Unter diesem Gesichtspunkt quantifizieren *interne*, *externe* und *relative* Validierungsindizes die Güte von Clustering-Algorithmen und deren Parameterkonfigurationen [46].

Interne Methoden bewerten in diesem Zusammenhang die Qualität eines Clusterings auf Grundlage der Daten selbst, unabhängig von externen Referenzdaten. Sie basieren auf den inhärenten Merkmalen der Daten und bewerten die Cluster nach Homogenität, Separation und Dichte. Zu den gängigsten internen Validierungsindizes gehören der Davies-Bouldin-Index, der Dunn-Index und die Inertia. Diese sind in der Literatur ausführlich beschrieben [114, 103, 191]. Der Silhouette-Score, eine weit verbreitete Metrik, misst die Kohäsion und die Separation von Clustern [141, 46]. Er wird berechnet, indem die durchschnittliche Entfernung eines Objekts zu den anderen Objekten desselben Clusters mit der durchschnittlichen Entfernung zu den Objekten des nächstgelegenen Clusters verglichen wird [108]. Ein hoher Silhouetten-Score signalisiert eine klare Abgrenzung

zwischen den Clustern, wobei die Objekte einem Cluster deutlicher zugeordnet sind als anderen Clustern [141].

In der Evaluierung von Clustering-Algorithmen nehmen **externe Validierungsindizes** eine entscheidende Rolle ein, indem sie die Übereinstimmung zwischen den im Clustering erzeugten Gruppierungen und einem vordefinierten Satz von Referenzklassifikationen messen [46]. Diese Indizes sind besonders wertvoll in Szenarien, in denen eine *Ground Truth* bekannt ist [66]. Sie ermöglichen die Quantifizierung der Effektivität von Clustering-Methoden im Hinblick auf ihre Fähigkeit, vorgegebene Klassifikationen zu reproduzieren [191]. Gängige Vertreter sind hier der Adjusted Rand Index und die Normalized Mutual Information [178].

Die Methoden der **relativen Validierung** sehen vor, die Ergebnisse verschiedener Clustering-Methoden zu vergleichen, um das beste Resultat zu erzielen. Dabei wird nicht auf Informationen aus externen Quellen zurückgegriffen. Ein Ansatz besteht darin, Silhouette-Scores oder andere Metriken über mehrere Clustering-Methoden hinweg zu verwenden, um die optimale Lösungsmethode zu identifizieren. [46]

2.5 Methoden des Reinforcement Learnings

In Abschnitt 2.5.1 werden die relevanten Grundlagen des Single-Agent RL und MARL erläutert, die einen integralen Bestandteil der neuen Methode in Kapitel 4 bilden. Die relevanten Trainingsschemata und Ausprägungen von kooperativem Verhalten im MARL sind in Abschnitt 2.5.2 beschrieben. In Abschnitt 2.5.3 finden sich mögliche Gestaltungsformen von Kommunikation im MARL. Die auftretenden Herausforderungen im MARL und mögliche Strategien, diesen entgegenzuwirken werden in Abschnitt 2.5.4 dargestellt.

2.5.1 Grundlagen und Taxonomie von Reinforcement Learning

Als Teilbereich des ML (vgl. Abbildung 2.13) besteht die Kernaufgabe des RL in der Lösung von sequentiellen Aufgabenstellungen [53]. Zu diesem Zweck interagiert ein Entscheidungsträger (en: *Agent*) in einer Umgebung (en: *Environment*) mit unbekannter Dynamik [207]. Der Agent befindet sich innerhalb der Umgebung und ist zugleich ein Teil von ihr [105]. Über die Interaktion mit dem Environment zieht der Agent Rückschlüsse auf sein eigenes Handeln, das durch eine numerische Belohnung (en: *Reward*) optimiert wird [164]. In mehreren Iterationen kann der Agent so seine Verhaltensstrategie (en: *Policy*) adaptieren und verbessern. Abbildung 2.20 zeigt das Zusammenwirken zwischen Agent und Environment im Sinne eines Markov Decision Process (MDP) [164].

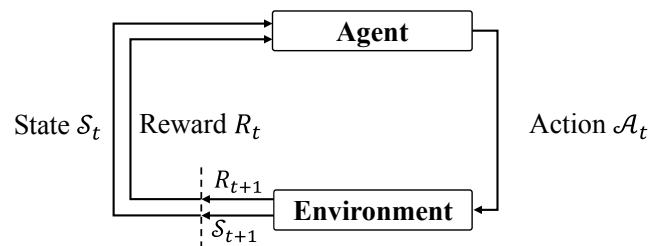


Abbildung 2.20: Schematische Darstellung des Markov Decision Process i. A. a. [164] und [38]

Das Environment in der RL-Methode ist mathematisch als MDP definiert [164]. Dieser ist durch das Tupel $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma \rangle$ formalisiert [53, 61]. Dabei gilt:

- \mathcal{S} ist eine Sammlung aller zulässigen Environment-Zustände. $S_t \in \mathcal{S}$ repräsentiert den Zustand im aktuellen Zeitschritt t [40].
- \mathcal{A} ist das Set aller zulässigen Aktionen (Aktionsraum), die einem Agenten a^i für einen gegebenen Zustand zur Verfügung stehen. $A_t \in \mathcal{A}$ ist folglich die Aktion, die der Agent zum entsprechenden Zeitpunkt t ausführt [105]. Mit dieser Aktion erreicht der Agent a den nachfolgenden Zustand S_{t+1} . Je nach Anwendungsfall kann der Aktionsraum \mathcal{A} durch diskrete Schritte oder eine Sammlung von Richtungsvektoren \vec{A}_t beschrieben sein.
- \mathcal{T} repräsentiert die Transition-Funktion. Diese ist definiert als $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$.

- R ist die Belohnungsfunktion (en: *Reward*) $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathcal{R}$. Dabei ist \mathcal{R} ein kontinuierliches Set an möglichen Rewards in einem Bereich von $R_{max} \in \mathbb{R}^+$.
- $\gamma \in [0, 1)$ repräsentiert den sogenannten Diskontierungsfaktor (en: *Discount factor*). Dieser bestimmt eine Gewichtung zwischen unmittelbarem und zukünftigem Reward [204, 154].

Der MDP sieht vor, dass das Ziel des Agenten darin besteht, so zu handeln, dass die erwartete Leistung langfristig eine unbekannte Transition-Funktion maximiert [61]. Daher lernt der Agent eine Policy π , die sein Verhalten innerhalb des Environments entsprechend steuert [38]. Dabei setzt der MDP voraus, dass die Wahrscheinlichkeit, einen nachfolgenden Zustand S_{t+1} zu erreichen, ausschließlich vom Zustand S_t abhängt. Die Vorgängerzustände S_{t-1} nehmen keinen Einfluss auf die Entscheidungsfindung [87]. Die Policy π selbst kann für zwei unterschiedliche Szenarien unterschiedlich ausgeprägt sein. Im deterministischen Fall ist sie beschrieben durch $\pi(S) : \mathcal{S} \rightarrow \mathcal{A}$. Für den stochastischen Fall hingegen ist die Policy $\pi(S, A) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, wobei $\pi(S, A)$ die Wahrscheinlichkeit beschreibt, dass eine Action A im State S ausgewählt wird. [53, 40]

Im Gegensatz zu einem MDP ist in einem Markov Game (MG) die Beteiligung mehrerer Agenten $a^i \in N$ zu berücksichtigen [106, 38]. Dies setzt neben der Erweiterung des Aktionsraumes \mathcal{A} zu einem gemeinsamen Aktionsraum $\mathcal{A}^{1...N}$ die Anpassung der Reward-Funktion voraus [128]. $\mathcal{R} = \{\mathcal{R}^1, \dots, \mathcal{R}^N\}$ ist die Sammlung aller individuellen Rewards der Einzelagenten [61]. Letztlich kann durch die Anpassung der Transition-Funktion analog zum MDP das MG durch das Tupel $\langle N, \mathcal{S}, \mathcal{A}^{1...N}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ beschrieben werden [121]. Abbildung 2.21 stellt die Konzepte des MDP und des MG gegenüber und erweitert den Vergleich um das Partially Observable Markov Game (POMG) [204, 28].

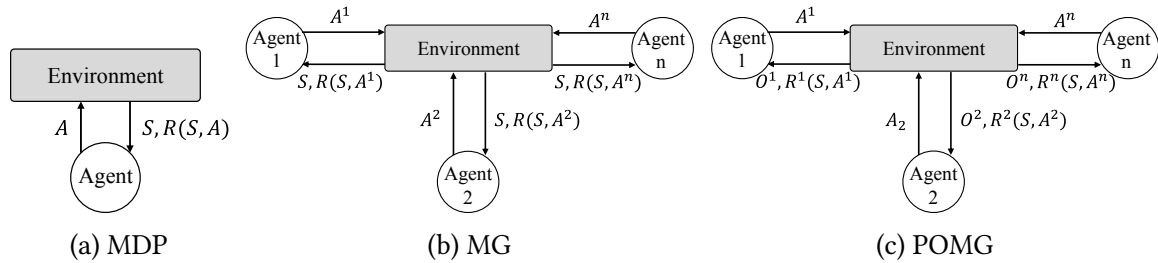


Abbildung 2.21: Gegenüberstellung von Markov Decision Process, Markov Game und Partially Observable Markov Game i.A.a [28]

In einem MG erweitert sich der MDP um mehrere Agenten [102, 128]. Die Anzahl der Agenten, die in einem gemeinsamen Environment interagieren ist mit $N = \{1, \dots, N\}$ für $N > 1$ definiert [61]. Zugleich erweitert sich der Aktionsraum \mathcal{A} zu einem gemeinsamen Aktionsraum (en: *Joint Action Space*) für $\mathcal{A}^{1...N} = \mathcal{A}^1 \times \mathcal{A}^2, \dots, \mathcal{A}^N$ [121]. Der Reward wird durch ein Set von individuellen Reward-Funktionen je Agent repräsentiert, wodurch sich das beschriebene Tupel zu $\langle N, \mathcal{S}, \mathcal{A}^{1...N}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ ändert (vgl. Abbildung 2.21b) [106]. Alle Agenten interagieren mit dem gemeinsamen Environment und führen für jeden diskreten Zeitschritt t eine gemeinsame Aktion $A^{1...N}$ aus, damit keine sequentiellen Abhängigkeiten entstehen [189]. Dafür erhalten sie einen entsprechenden Reward sowie den

daraus resultierenden neuen State S_{t+1} [38]. Innerhalb des MG existieren unterschiedliche Ausprägungen des kooperativen und des kompetitiven Lernverhaltens [21, 38, 61]. Diese spiegeln sich vor allem in der Reward-Funktion wider. Der vollständig kooperative Ansatz (en: *fully cooperative*) zeigt in der Mehrzahl der Multi-Agenten-Ansätze einen höheren Gesamtnutzen [38]. Er unterliegt der Prämisse, dass alle Agenten den gleichen Reward $R = R^i = \dots = R^N$ für die Transition erhalten [128]. In einem Umfeld mit gleichmäßig geteiltem Reward sind die Agenten zur Kooperation motiviert, um das Scheitern eines einzelnen Agenten zu verhindern [61]. Dies dient dazu, den Gesamtnutzen aller Agenten zu maximieren. Daraus folgt, dass ein individueller Agent einen höheren Nutzen im Sinne des Rewards aus der Kooperation ziehen muss, anstatt ein nicht-kooperierendes Verhalten zu erlernen. Der gänzlich konträre, kompetitive Ansatz (en: *fully competitive*) zeichnet sich hingegen durch das Nullsummen-MG (en: *Zero-Sum-MG*) aus [40]. Dies bedeutet, dass die Summe aller Rewards für eine Transition gleich Null ist ($R = \sum_{i=1}^N R^i(S_t, \mathcal{A}, S_{t+1}) = 0$). Die kompetitiv agierenden Agenten sind darauf ausgelegt, ihre individuelle Belohnung zu maximieren, während die Belohnung der anderen partizipierenden Agenten zeitgleich minimiert wird [38]. Dementsprechend ist bei einem Zero-Sum-MG jeder Agent dazu ermutigt, sich gegenüber anderen Agenten durchzusetzen [61]. In den gemischten Ansätzen (en: *Mixed Setting*) ist keine eindeutige Zuordnung zu einem kooperativen oder kompetitiven Lernverhalten möglich [40].

In den meisten Fällen fehlt den Agenten jedoch die direkte Beobachtung des zugrunde liegenden States für eine optimale Action-Auswahl (vgl. Abbildung 2.21c) [28]. Demzufolge ist ein Mapping von Beobachtungshistorien (Observations \mathcal{O}) oder Glaubenszuständen (Belief States) auf Actions erforderlich, was zusätzliche Herausforderungen hervorruft [204]. Diese partielle Beobachtung (en: *Partially Observability*) des Environments findet sich im POMG wieder [110]. Dabei ist es einem einzelnen Agenten nicht möglich, den ganzheitlichen State des Environments zu erfassen, stattdessen erfasst er lediglich eine Observation $\mathcal{O}^i \subset \mathcal{S}$ [61]. Das beschreibende Tupel verändert sich im Vergleich zum MG zu $\langle N, \mathcal{O}^{1\dots N}, \mathcal{S}, \mathcal{A}^{1\dots N}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ und schließt damit den Joint Observation-Space ein. Dieser fasst alle einzelnen Observations der Agenten zusammen ($\mathcal{O}^{1\dots N} = \mathcal{O}^1 \times \dots \times \mathcal{O}^N$) [110]. Analog dazu ändert sich die Transition-Funktion, die entsprechend auf der Observation aufbaut [28]. \mathcal{S} repräsentiert im Kontext des POMG den globalen, jedoch nicht erfassten Zustand des Environments [61]. Die unterschiedlichen Grade der Kooperation gelten analog zum MG. Weitere MARL-Frameworks und Abwandlungen des MDP, wie der Partially Observable MDP, sind in der Literatur beschrieben [204].

2.5.2 Trainingsschemata und Kooperationsverhalten im Multi-Agent Reinforcement Learning

In einem Multi-Agenten-System existieren unterschiedliche Trainings- und Ausführungsschemata für die Entwicklung und den Einsatz der Policies π aller Agenten. Die Schemata lassen sich in drei übergeordnete Kategorien unterteilen, die nachfolgend detailliert beschrieben und analysiert werden [61]. Abbildung 2.22 zeigt eine Übersicht über die Trainingsschemata und stellt diese dem Single-Agent-Ansatz aus Kapitel 2.5.1 gegenüber.

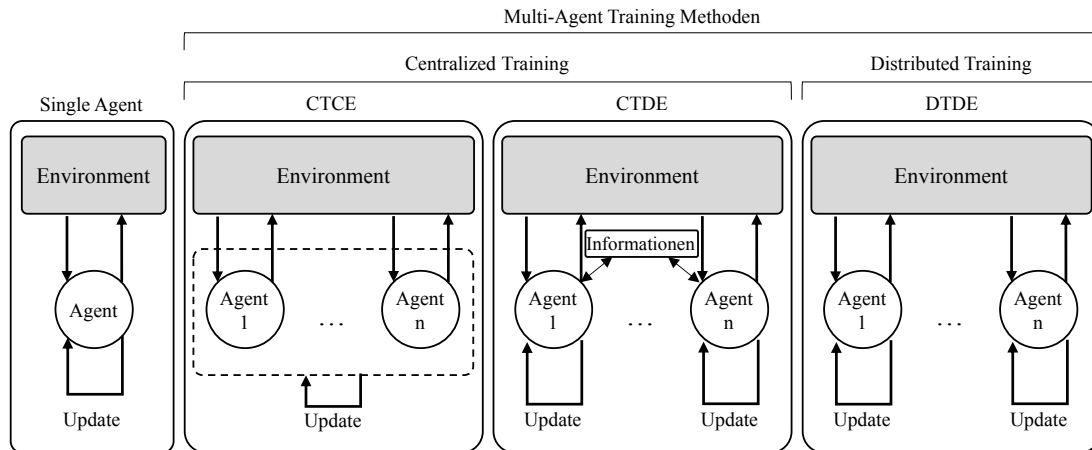


Abbildung 2.22: Trainingsschemata im Multi-Agent Reinforcement Learning i. A. a. [61]

Das Schema des *Centralized Training Centralized Execution (CTCE)* in Abbildung 2.22 nimmt als Rahmenbedingung eine vollständige Kenntnis über alle anderen Agenten während der Trainingszeit an. Es beschreibt eine zentrale Ausführungseinheit mit einer gemeinsamen Policy $\pi : \mathcal{O} \rightarrow P(\mathbf{A})$, welche die Sammlung verteilter Observations \mathcal{O} auf einen Satz von Verteilungen über individuelle Aktionen abbildet [61]. Dabei wird in einer Vielzahl von Anwendungen ein uneingeschränkter und unmittelbarer Austausch von Informationen angenommen [40]. Damit entspricht das CTCE-Paradigma dem direkten Einsatz einer Single-Agent-Trainingsmethode auf Multi-Agenten-Probleme [15, 208]. Jedoch heißt dies auch, dass der State-Action-Space exponentiell mit der Anzahl der Agenten anwächst. Eine wesentliche Einschränkung des CTCE stellt das Phänomen des *Lazy Agent* dar. Es tritt auf, wenn ein Agent a^i eine gute Policy π^i erlernt, ein zweiter Agent a^{i+1} jedoch weniger Anreiz hat, eine gute Policy π^{i+1} zu erlernen, da seine Handlungen den ersten Agenten behindern können [163]. Dies führt zu einem geringeren Gesamt-Reward R_{total} . [61]

In der Anwendung des *Centralized Training Decentralized Execution (CTDE)* in Abbildung 2.22 hat jeder Agent i eine individuelle Policy $\pi^i : \mathcal{O}^i \rightarrow P(\mathbf{A}^i)$, die eine lokale Observation auf eine Verteilung individueller Aktionen abbildet [61]. Während der Trainingszeit stehen zusätzliche Informationen zur Verfügung, die zur Testzeit verworfen werden. Dieses Paradigma findet vielfältigen Einsatz in MARL-Problemstellungen [52]. Die Agenten nutzen gemeinsame Formen der Kommunikation (siehe Abschnitt 2.5.3), um schon während des Trainings Daten auszutauschen. Durch die gemeinsame Nutzung von Informationen kann der Trainingsprozess vereinfacht und potenziell verkürzt werden [40]. Dies wirkt

sich insbesondere auf die direkte Interaktion der Agenten aus. Die Lerngeschwindigkeit ist im Vergleich zu unabhängig trainierten Agenten bei dieser Methode höher, das bedeutet, die erforderliche Lernzeit ist kürzer [51]. Mit Blick auf die nichtstationäre Natur (*en: Non-Stationary*) des Environments ist der Informationsaustausch der Agenten eine Möglichkeit, diese Environment-Beschaffenheit zu umgehen. Mittels Verfügbarkeit zusätzlicher Daten über die ausgewählten Aktionen aller Agenten während des Trainings ist es zudem möglich, die Folgen von Aktionen den einzelnen Agenten zuzuordnen. [61]

Bei der Verwendung des *Decentralized Training Decentralized Execution (DTDE)* (vgl. Abbildung 2.22) als Schema im MARL folgt jeder Agent a^i seiner individuellen Policy $\pi^i : O^i \rightarrow P(A^i)$ [40, 61]. Diese bildet analog zum CTDE eine lokale Observation O^i auf eine Verteilung individueller Aktionen ab. Es findet bei dieser Methode jedoch keine Informationsübertragung zwischen den Agenten statt. Folglich lernt jeder Agent a^i seine Policy π^i unabhängig von anderen Agenten. Dies korreliert mit der wesentlichen Einschränkung des DTDE-Paradigmas. Sie spiegelt sich aus Sicht des Agenten in der Non-Stationarity der Umgebung wider [106]. Ursächlich für die Non-Stationarity ist, dass der Akteur keinen Zugriff auf das Wissen der anderen Akteure hat und keine gemeinsame Handlung erkennbar ist. Darüber hinaus zeigt die Analyse, dass verteilte Trainingsverfahren aufgrund der zusätzlichen Stichprobenkomplexität, die dem Lernproblem hinzugefügt wird, sich mit der Anzahl der Agenten schlecht skalieren lassen. Verteilte Methoden weisen im Vergleich zu Policies, die mit einem zentralisierten Trainingsparadigma trainiert werden, eine geringere Leistung auf [63, 106]. Eine langsamere Lerngeschwindigkeit von Actor-Critic-Methoden in verteilten Lernverfahren ist in ähnlicher Weise nachgewiesen [51].

Aufgrund bestehender Einschränkungen im Kontext der Skalierbarkeit des DTDE-Paradigmas sowie der Notwendigkeit von Kooperation und Wissensübertragung finden im Rahmen der methodischen Entwicklung in den Kapiteln 4.4.2 und 4.5.2 die Trainings-schemata CTCE und CTDE Anwendung. Mit Blick auf das angestrebte Optimierungsziel der minimierten Gesamtkabellänge fließt in die Methode in Kapitel 4.4 der vollständig kooperative Ansatz ein.

2.5.3 Kommunikationsmethoden im Multi-Agent Reinforcement Learning

Zur Strukturierung der existierenden Ansätze zur Kommunikation im MARL gibt es eine Vielzahl von möglichen Gruppierungen der Kommunikationsmethoden. Diese Arbeit konzentriert sich auf die Unterteilung nach den Adressaten der zu empfangenden Informationen, weitere Klassifizierungen sind in der Literatur definiert [208]. Die Kommunikationsansätze im MARL lassen sich abhängig von den Adressaten der übermittelten Informationen in drei Kategorien einteilen. Sie sind als Unterscheidung zwischen den CTDE-Methoden (siehe Abschnitt 2.5.2) einzugliedern [208]. Ihre Anwendung wird vor allem im Kontext von POMGs untersucht, in denen ein kooperatives Ziel der Agenten besteht, die Agenten jedoch nur Ausschnitte des Environments wahrnehmen (vgl. Kapitel 2.5.1) [126]. Die erste Kategorie wird als **Broadcasting**-Szenario bezeichnet, in dem alle Agenten die gesendeten Nachrichten erhalten. Die zweite Kategorie ist die **zielgerichtete** (*en: Target*) Kommunikation, die Aufmerksamkeitsmechanismen verwendet, um die Empfänger der Nachrichten auszuwählen. Die dritte Kategorie ist die **Netzwerk-kommunikation** (*en: Network*). Agenten kommunizieren nur mit Agenten in ihrer unmittelbaren

Umgebung und nicht mit der Gesamtheit aller Agenten. Abbildung 2.23 zeigt eine abstrakte Darstellung dieser Klassifizierung [61].

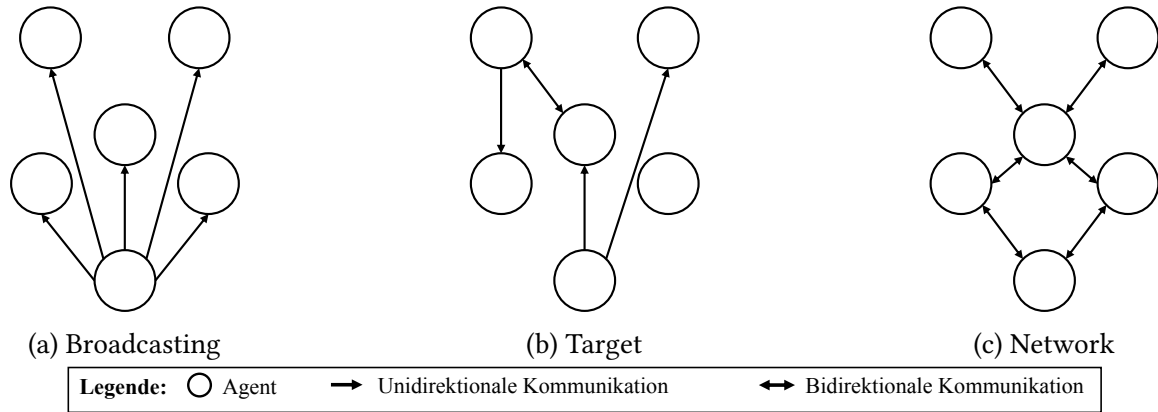


Abbildung 2.23: Kommunikationstypen im Multi-Agent Reinforcement Learning i. A. a. [61]

Der erste Typ des **Broadcastings** (vgl. Abbildung 2.23a) umfasst alle Ansätze, die eine vollumfängliche Kommunikation vorsehen [61]. In dieser Kategorie ist jedes Paar von Agenten $a^i, a^j \in N$ miteinander verbunden und die Nachrichten werden über das Broadcast-Verfahren übermittelt [208]. Dieser Ansatz ermöglicht es jedem Agenten, Informationen gleichzeitig an alle anderen Agenten zu übertragen [89, 63]. Dies ist insbesondere in vollständig kooperativen POMGs nützlich, in denen die Zielsetzung darin besteht, die gemeinsame Aufgabe zu optimieren (vgl. Kapitel 2.5.1) [1]. Dabei steht das Erlernen, welche Informationen zu übermitteln sind, im Fokus dieser Methoden. Indem die Agenten sich aneignen, welche Inhalte sie untereinander austauschen sollen, können sie ihre Handlungen besser koordinieren und die Gesamtleistung verbessern [52].

Bei der **Target**-Kommunikation (vgl. Abbildung 2.23b) hingegen entscheidet jeder Agent a^i aktiv und individuell, ob mit einem anderen Agenten a^{i+1} kommuniziert wird [208]. Die Agenten lernen dabei, Nachrichten dynamisch und selektiv zu übermitteln, sodass sie nur dann miteinander kommunizieren, wenn es erforderlich ist [82]. Weiterentwicklungen dieser Methoden bestimmen nicht nur, mit wem eine Informationsübermittlung stattfinden soll, sondern auch, welcher Inhalt in einem Partially-Observable Environment kommuniziert werden soll [33]. Häufig basiert die Target-Kommunikation auf der Differenzierung nach der Wahrnehmung der sich in der Umgebung befindenden Agenten. Umliegende Agenten können als solche definiert werden, die sich innerhalb einer vordefinierten Distanz zum betrachteten Agenten befinden [208]. Aufgrund der einzuhaltenden konstanten Observation-Größe ist eine Limitation der kommunizierenden Agenten in der Target-Kommunikation gesetzt [83]. Insbesondere in großen Environments mit einer hohen Anzahl von Agenten lernen diese so, welche Informationen für die Entscheidungsfindung am nützlichsten sind [61].

Die **Network**-Kommunikation (vgl. Abbildung 2.23c) basiert auf einem bidirektionalen Informationsaustausch mit der unmittelbaren Umgebung [119, 203]. Die Agenten des MG agieren dezentral auf der Grundlage lokaler Observations sowie empfangener Nachrichten

benachbarter Agenten [61]. Diese Art der Kommunikation stützt sich in den entsprechenden Integrationen häufig auf einen Graphen. Innerhalb dieses Graphen finden sich Anwendungen wieder, die in einem Actor-Critic (vgl. Kapitel 2.5.1) die Informationen des Critic zur Förderung der globalen Optimalität teilen [205]. Analysen zeigen, dass weitergehende Methoden die Übertragungseffizienz von Informationen unter Nutzung der Network-Kommunikation erhöhen [27]. Im Vergleich zu Broadcasting- und Target-Kommunikation zielt diese Informationsübertragung auf eine Übertragungseffizienz und die selektive Übermittlung von Informationen ab. Diese Anforderungen ergeben sich meist aus dem spezifischen Anwendungsfall sowie einer limitierten Datenübertragungskapazität. [27]

2.5.4 Herausforderungen der Methoden des Multi-Agent Reinforcement Learning

Den beschriebenen Vorteilen des Multi-Agenten-Ansatzes stehen vielfältige Herausforderungen gegenüber, die ihren Ursprung vor allem im Paradigmenwechsel hin zu einem MARL haben. Die erste Problemstellung ergibt sich aus der Non-Stationarity des Environments. Jeder Agent $a^i \in N$ ist nicht nur mit einer sich verändernden Umgebung konfrontiert, sondern wird auch von den sich ändernden und anpassenden Strategien π^{i+1} anderer Agenten $i + 1$ beeinflusst [208]. Diese dynamischen Veränderungen wirken sich auf die Bildung der Policy π^i aus [163]. Ein Problem besteht darin, dass sich die Strategie jedes Agenten mit fortschreitendem Training ändert und die Umgebung aus der Sicht eines einzelnen Agenten nicht mehr stationär ist. Dies lässt sich nicht durch Änderungen in der Strategie des Agenten selbst erklären [106]. Folglich gilt die Annahme eines idealisierten MG aus der Perspektive eines einzelnen Agenten a^i nicht mehr [61]. Eine Methode, um diese Problematik zu lösen, ist die Integration von Kommunikationsstrategien (vgl. Kapitel 2.5.3).

Eine damit eng verbundene Herausforderung in einem MARL ist das Problem des Beitrages zum Reward (en: *Credit Assignment Problem*). Bei einem kooperativen Ziel aller Agenten ist der einzelne Agent nach Erhalt eines gemeinsamen Rewards nicht in der Lage, den eigenen Beitrag zum Erfolg oder Misserfolg des Kollektivs zu ermitteln [189]. Diese Problemstellung kommt bei den CTCE-Ansätzen (vgl. Kapitel 2.5.2) zum Tragen, wenn eine Gemeinschaftsaktion (en: *Joint Action*) eine Differenzierung des Reward-Beitrages erschwert oder gar gänzlich unterbindet [51, 121].

Die Problemstellung des katastrophalen Vergessens (en: *Catastrophic Forgetting*) bezieht sich auf ein Phänomen im Lernprozess eines Agenten. Es tritt besonders dann auf, wenn die Agenten sequenziell auf mehrere Aufgabenstellungen angelernet werden [93, 38]. Künstliche neuronale Netze (vgl. Kapitel 2.4.2) haben Schwierigkeiten mit kontinuierlichen, inkrementellen Lernaufgaben. Das Wissen über vorherige Aufgaben kann abrupt verloren gehen, wenn neue Informationen für eine aktuelle Aufgabe aufgenommen werden [93]. Die Vermeidung dieses Phänomens stellt gleichzeitig eine der größten Herausforderungen des TL dar (vgl. Kapitel 2.6) [210]. Sie geht einher mit der Notwendigkeit einer großen Datenmenge für einen stabilen Lernprozess, was besonders im Zusammenhang mit Daten physisch existierender Agenten eine Einschränkung darstellen kann [3].

Die Abbildbarkeit eines Fully-Observable MG ist eine Problemstellung, die mit einer zunehmenden Environment-Größe oder einer hohen Anzahl an Agenten einhergeht. Aufgrund dessen zeigen Analysen der Literatur Ansätze für die Überwindung dieser Einschränkung durch die Verwendung eines POMG (vgl. Abbildung 2.21c in Kapitel 2.5.1) [189, 38]. Da der Agent den Zustand des Environments nicht direkt beobachten kann, muss er seine Entscheidungen unter Unsicherheit über den wahren Zustand des Environments treffen. Durch Interaktion mit dem Environment und das Sammeln von Beobachtungen kann der Agent jedoch seine Entscheidungsunsicherheit reduzieren. Eine Konsequenz dieses Verhaltens besteht darin, dass ein optimales Verhalten Aktionen beinhalten kann, die nur ausgeführt werden, weil sie die Einschätzung des Agenten über den aktuellen Zustand verbessern [61]. Mit dieser Einschränkung geht eine potenzielle Restriktion der Skalierbarkeit einer MARL-Methode einher. In der Anwendung zeigen sich dezentralisierte Ansätze mit der Übertragung von möglichst wenig Informationen zwischen den Agenten (vgl. Kapitel 2.5.3) [117].

Aus Basis der analysierten MARL-Methoden (vgl. Kapitel 2.5.2) und der Gegenüberstellung mit den bestehenden Herausforderungen innerhalb dieser KI-Methode ergibt sich ein ganzheitliches Bild zum Einsatz in der neuen Methode in Kapitel 4. Neben dem großen Potenzial, ein globales Optimum für das NP-schwere MAPF-Problem zu finden, ist insbesondere die selbstständige Entwicklung einer Lösungsstrategie (Policy π) hervorzuheben. Ohne die Abhängigkeit von historischen, gelabelten Lerndaten oder feststehenden Regelwerken finden die RL-Methoden eine valide Lösung als kooperative Einheit aller Agenten. Zugleich müssen für die beschriebenen Herausforderungen innerhalb der neuen Methode Lösungen gefunden werden, um das volle Potenzial des MARL in der virtuellen Verdrahtung ausschöpfen zu können.

2.6 Methoden des Transfer Learnings

In diesem Abschnitt wird die Methode des TL, auch Knowledge Transfer genannt, im MARL beschrieben. Zunächst widmet sich der Abschnitt 2.6.1 dem grundlegenden Konzept und stellt dieses dem traditionellen ML gegenüber. In Abschnitt 2.6.2 wird dann der Einfluss des TL im Kontext des RL detailliert analysiert. Eine Klassifizierung von TL-Methoden erfolgt in den Abschnitten 2.6.3 und 2.6.4. Zur Einordnung des Lernerfolgs des TL finden etablierte Metriken Anwendung, die in Abschnitt 2.6.5 veranschaulicht werden.

2.6.1 Grundlagen des Transfer Learnings

Neben der Generalisierungsfähigkeit von KI-Methoden (siehe Kapitel 2.4.1) ist die Weiterverwendung von bereits erzeugtem Wissen ein wesentlicher Aspekt der Nutzung von Methoden des ML. Die Methode des TL hat hierbei ihren Fokus auf der Übertragung von Wissen auf neue Aufgabenstellungen im Zusammenhang mit künstlichen neuronalen Netzen. Der Terminus des TL dient lediglich als Oberbegriff für das grundlegende Konzept. [68]

Definition: Bei einer Quelldomäne D_S und einem Lerntask T_S , einer Zieldomäne D_T und dem zugehörigen Lerntask T_T zielt TL darauf ab, das Lernen der Zielvorhersagefunktion $f_T(\bullet)$ in D_T unter Verwendung des erzeugten Wissens aus D_S und T_S zu unterstützen. Dabei gilt $D_S \neq D_T$ und $T_S \neq T_T$. [129]

Mit dieser Definition grenzt sich das TL von der Herangehensweise des klassischen ML-Ansatzes ab. Vielmehr motiviert es die Fragestellung, wann welcher Informationsgehalt auf welche Art und Weise übertragen werden soll, um einen neuen Lerntask effizienter zu bewältigen [210]. Das traditionelle ML und die Methode des TL sind in Abbildung 2.24 auf einer konzeptionellen Ebene gegenübergestellt [129].

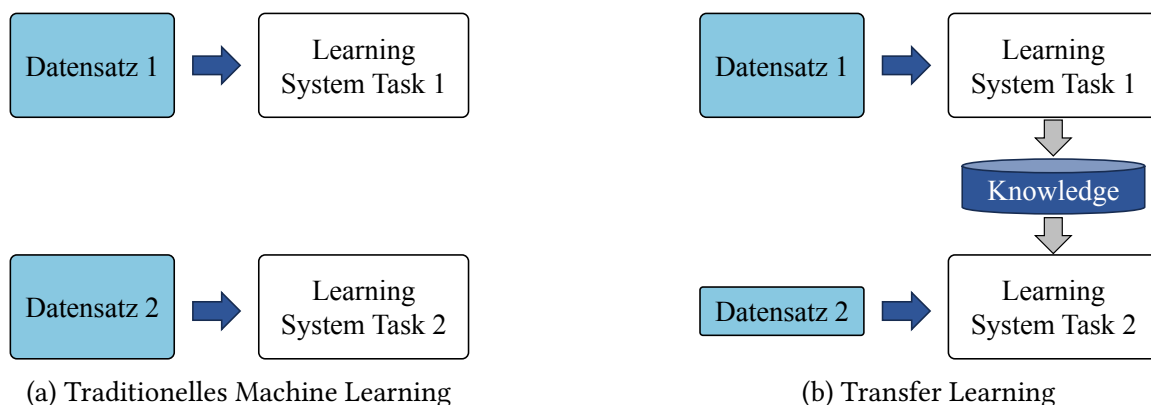


Abbildung 2.24: Vergleich zwischen Machine Learning und Transfer Learning i.A.a. [129]

In diesem Zusammenhang zeigt Abbildung 2.24a die Kardinalität von 1 : 1 zwischen dem Datensatz und der zu lernenden Aufgabe. Es ist ersichtlich, dass für jede neue Aufgabe ein Datensatz in entsprechender Größe notwendig ist. Eine Übertragung von bereits erlerntem Wissen von einem Datensatz auf den anderen ist nicht gegeben. Das traditionelle ML

entspricht damit einem isolierten, aufgabenbezogenen Lernen [129]. Wissen wird nicht behalten oder akkumuliert, das Lernen selbst erfolgt ohne Berücksichtigung früherer Kenntnisse, die aus anderen Aufgaben resultieren. Das TL (vgl. Abbildung 2.24b) hingegen stützt sich beim Erlernen einer neuen Aufgabe auf Erkenntnisse aus zuvor erlernten Aufgaben [75]. Der Lernprozess ist schneller, genauer und kann eine vergleichbare Optimalität mit weniger Trainingsdaten erreichen [31]. Diese Eigenschaft ist im Zusammenhang mit begrenzter Datenverfügbarkeit oder Zeitlimitationen als positiv zu bewerten. Die erzeugte Erfahrung (en: *Knowledge*) liegt im Zusammenhang mit neuronalen Netzen nicht in Form einer Wissensdatenbank, sondern als implizites Wissen innerhalb des neuronalen Netzes vor (vgl. Abbildung 2.24b). [210]

2.6.2 Transfer Learning im Kontext des Reinforcement Learnings

Im Single-Agent-RL findet ein Wissenstransfer exklusiv zwischen Policies π unterschiedlicher Domänen statt. Somit gilt nachfolgend $D_S \neq D_T$ (vgl. Kapitel 2.6.1). Die Übertragung gesammelter Informationen zwischen den Agenten in einer gemeinsamen Domäne setzt die Anwendung des MARL voraus (vgl. Kapitel 2.6.4) [32]. Eine Domäne beschreibt in diesem Zusammenhang einen MDP oder ein spezifisches MG [31]. Eine Abweichung zwischen Quell- und Zieldomäne kann dabei in allen Bestandteilen eines MDP oder MG auftreten. So können sowohl abweichende Action-Spaces \mathcal{A} als auch eine Modifikation der Reward-Funktion R die Domänen differenzieren. [210]

Für den Transfer im RL ist darüber hinaus zwischen *Zero-Shot*, *Few-Shot* und dem *Sample-Efficient* Transfer zu differenzieren [210]. Der *Zero-Shot*-Transfer zeichnet sich dadurch aus, dass ein antrainierter Agent direkt auf eine Zieldomäne D_T anwendbar wird, ohne dass eine Trainingsinteraktion erforderlich ist [55]. Die erlernte Policy π bleibt folglich unverändert. Im *Few-Shot-Transfer* ist hingegen vorgesehen, dass der Agent wenige Interaktionen mit der Zieldomäne D_T durchführt, um seine bestehende Policy π auf die konkrete Problemstellung zu adaptieren [183]. Dieser Prozess ermöglicht eine erhöhte Lerngeschwindigkeit des RL [68]. Der *Sample-efficient* Transfer ist anwendbar, wenn der Agent vom TL profitiert, indem er bessere Ergebnisse erzielt, als wenn er die neue Aufgabe T_T von Grund auf neu erlernt. Die angewendete Differenzierung bedient die Teilfragestellung des TL, wie sample-efficient die TL-Methode ist. [75]

Die Analyse zeigt, dass eine Methode zur Verbesserung eines Transfers von Wissen die Domänenrandomisierung (en: *Domain Randomization*) ist [74]. Diese Methode beschreibt eine Technik zur Steigerung der Verallgemeinerungsfähigkeit von trainierten Strategien in modifizierten Domänen [115]. Neben der zu wählenden Randomisierungsvariablen sind deren Wertebereiche zu bestimmen [177]. Ein häufiger Anwendungsfall für die Domain Randomization ist die Übertragung von Policies π aus der Simulation in die Realität, der sogenannte Sim-to-Real-Transfer [177]. Aufgrund der eingeschränkten Möglichkeit, alle physikalischen Gegebenheiten einer echten Situation im Environment abzubilden, kann der Agent damit dennoch adäquate Ergebnisse im realen Einsatz erzielen. Von besonderer Bedeutung für die neue Methode in Kapitel 4.5 ist die laufende Modifikation der Objekte innerhalb des dreidimensionalen Raumes. Dieser Ansatz unterbindet das Overfitting (vgl. Kapitel 2.5.4) auf eine spezifische Problemstellung und dient dem allgemeinen Erlernen einer Lösungsstrategie π [169]. Die Begrifflichkeit der Domain Randomization ist dabei

von der Methode des Meta-RL abzugrenzen. Während die Domain Randomization darauf abzielt, einen artverwandten Task innerhalb von modifizierten Environments zu erlernen, besteht die Intention des Meta-RL im Erlernen von mehreren MDPs [12]. Meta-RL konzentriert sich folglich auf die Entwicklung von Lernstrategien für eine schnelle Anpassung auf neue Aufgaben. Das TL im RL hingegen ermöglicht die Übertragung von Wissen und Fähigkeiten auf verwandte Aufgaben [74].

2.6.3 Kategorisierung von Transfer Learning-Methoden

Die gängige Unterscheidung der TL-Methoden orientiert sich an den zugehörigen drei wesentlichen Fragestellungen - dem Informationsgehalt, der Art sowie dem Zeitpunkt des Transfers. Auf Grundlage dessen unterscheidet das Unterkapitel 2.6.3.1 nach übertragenem Wissen, während der nachfolgende Abschnitt 2.6.3.2 nach der Aufgaben- und der Domänensimilarität differenziert.

2.6.3.1 Differenzierung nach zu übertragendem Wissen im Transfer Learning

Beim RL mit mehreren Aufgaben (*en: Multi-Task*) stellt prozedurales Wissen dynamische Informationen (z. B. Proben) und deklaratives Wissen statische Informationen (z.B. Wertefunktion) dar. Auf Basis dieser Differenzierung lassen sich verschiedene Arten der Knowledge-Generalisierung zwischen den Tasks unterscheiden. In Abbildung 2.25 werden diese in einen direkten Transfer (vgl. Abbildung 2.25a), den indirekten Transfer (vgl. Abbildung 2.25b) und einen sequenziellen Transfer von Wissen (vgl. Abbildung 2.25c) unterteilt. [183]

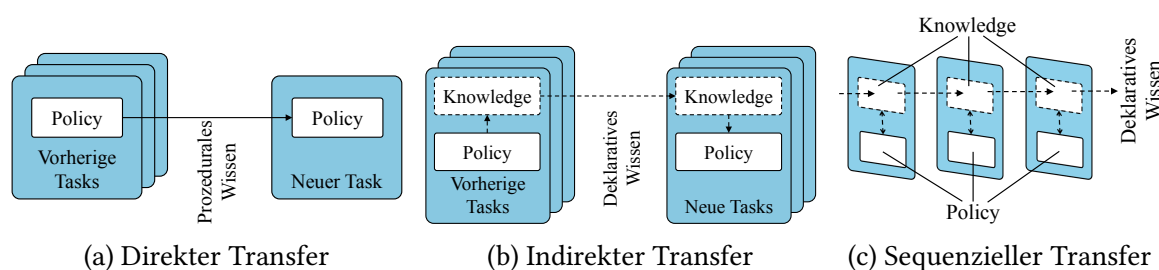


Abbildung 2.25: Unterschiedliche Arten von Knowledge-Generalisierung zwischen Tasks i. A. a. [183]

Wie in Abbildung 2.25 dargestellt, besteht der Unterschied zwischen der direkten und der indirekten Übertragung in der Fragestellung, ob der Übertragungsprozess auf prozeduralem oder deklarativem Wissen aufbaut (vgl. Abbildungen 2.25a und 2.25b).

Bei einem **direkten Transfer** (vgl. Abbildung 2.25a) besteht die Wissensübermittlung darin, dass ein Modell, das in einer Reihe von vorherigen Tasks angelernt wird, direkt auf den neuen Task angewendet werden kann, ohne das ML-Modell weiter anzupassen. Dieser Ansatz ist vorteilhaft, wenn T_S und T_T ähnliche Merkmale oder ähnliche Verteilungen aufweisen. In der Anwendung von RL bedeutet direkter Transfer, dass die spezifischen Erfahrungen, wie States \mathcal{S} , Actions \mathcal{A} und Rewards \mathcal{R} , oder die aus diesen Erfahrungen abgeleitete Policy π direkt in der neuen Umgebung Anwendung finden [183]. Es erfolgt

keine wesentliche Umstrukturierung oder Generalisierung des Wissens. Exemplarisch ist das Lernen aus zuvor generierten Demonstrationen (en: *Learning from Demonstrations*) an State-Action-Reward-Tupeln zu nennen, bei dem die Agenten die Verhaltensweisen von Experten über Beispiele direkt nachahmen. Hierbei kann der Agent in der Zieldomäne T_T aus dem Verhalten von Experten lernen, um seine Exploration effizienter zu gestalten [210]. Die dabei generierten Tupel $\langle S_t, \mathcal{A}, \mathcal{R}, S_{t+1} \rangle$ dienen als Eingangsinformation und finden sich je nach angewandeter RL-Methode im Replay-Buffer des Agenten wieder [13, 31]. Der neue Agent entnimmt eine Reihe von Interaktionen aus dem Buffer und trainiert seine Policy π , noch bevor er das Environment initial erkundet [22]. Weitere Ansätze des direkten Transfers sind das *Sample Weighting* und das *Reward Shaping*. Diese Methoden sind für die neue Methode in Kapitel 4.5 von untergeordneter Bedeutung und werden in der Literatur ausführlich beschrieben. [183]

Im **indirekten Transfer** (vgl. Abbildung 2.25b) erfolgt keine Übertragung von explizitem Wissen in Form von State-Action-Reward-Tupeln. Stattdessen extrahiert der indirekte Transfer zunächst aus dem prozeduralen das deklarative Wissen [183]. Dieser Prozess wird Knowledge Induction genannt. Das gewonnene Wissen findet dann in der neuen Aufgabenstellung T_T Anwendung. Es basiert auf der Annahme, dass eine erlernte Fähigkeit oder Erkenntnis in der Ausgangsdomäne D_S universell genug ist, um auch in der ähnlichen Zieldomäne D_T nützlich zu sein [183]. In einer speziellen Ausprägung besteht der Transfer von einem größeren, komplexeren Modell auf ein kleineres, effizientes Modell, welches das Ausgangsmodell in seiner Verhaltensweise imitieren soll. Diese Vorgehensweise geht mit einer deutlich reduzierten Modellkomplexität und schnellerer Ausführungszeit einher. Damit trägt die entsprechende *Knowledge Distillation* dazu bei, den Lernprozess in der Zieldomäne D_T positiv zu beeinflussen. [98]

Der **sequenzielle Transfer** (vgl. Abbildung 2.25c) ist durch das kontinuierliche Weiterlernen über mehrere Tasks gekennzeichnet und stellt eine Sonderform des *indirekten Transfers* dar. Die Methode wird in der Literatur mit dem Begriff des *Lifelong Learnings* gleichgesetzt [13]. Zunächst findet ein Trainingsprozess der Policy π in einer initialen Aufgabenstellung T_{T_0} statt. Auf Grundlage dessen erfolgt die Übertragung des deklarativen Wissens auf einen nachfolgenden Task T_{T_1} . Dieser Prozess wird danach stetig für weitere Tasks $T_{T_1 \dots N}$ wiederholt, in denen der Agent (MDP) oder die Agenten (MG) auf dem bisher erlernten Wissen aufbauen [183]. Dieser Ansatz ist besonders effektiv in Szenarien zu nutzen, in denen es eine natürliche Progression oder eine Sequenz von Aufgaben gibt, die in ihrer Beschaffenheit aufeinander aufsetzen [183].

2.6.3.2 Differenzierung des Transfer Learning nach Aufgaben- und Domänensimilarität

Eine Unterscheidung gemäß der Similarität von Task und Domäne ist in der Kategorisierung von TL-Methoden ebenfalls möglich. Hier erfolgt eine Kategorisierung in induktives (en: **Inductive**), transduktives (en: **Transductive**) und unüberwachtes (en: **Unsupervised**) TL [192]. Die Konzepte sind gleichermaßen für das SL (vgl. Kapitel 2.4.3) und RL (vgl. Kapitel 2.5) anwendbar.

Das **Inductive TL** zeichnet sich durch eine Wissensübertragung aus, die abweichende Aufgabenstellungen beinhaltet [192]. So gilt $T_S \neq T_T$, unabhängig davon, ob eine Similarität der Domänen D_S und D_T besteht oder nicht [129]. Das vorrangige Ziel besteht bei

dieser Methode in der Beschleunigung der Lerngeschwindigkeit und der Steigerung der Gesamtleistung der Agenten, indem auf bereits erlerntes Wissen aus ähnlichen Aufgaben zurückgegriffen wird [75]. Dies entspricht der Übertragung einer vortrainierten Policy π und einer entsprechenden Modifikation für die Zielaufgabe T_T (vgl. Kapitel 2.6.1).

Dem steht das **Transductive TL** gegenüber. Dieses sieht vor, dass die zu lösende Aufgabe dieselbe bleibt ($T_S = T_T$), während sich jedoch die Environments in ihren Eigenschaften und Dynamiken unterscheiden ($D_S \neq D_T$) [188]. Der Schwerpunkt dieses methodischen Ansatzes liegt auf der Anpassung der Agenten, ohne einen neuen Trainingsprozess zu benötigen. Die existente Policy π bleibt damit im neuen Environment bestehen. [129]

Die Kategorie des **Unsupervised TL** ähnelt dem Inductive TL [129]. Dabei unterscheiden sich die Ausgangs- und Zielaufgaben, diese weisen aber Ähnlichkeiten zueinander auf. Das Unsupervised TL berücksichtigt das Szenario der Übertragung von Wissen von D_S auf D_T , ohne dass annotierte Daten oder explizite Rewards in der Zielumgebung zur Verfügung stehen. [188, 75]

2.6.4 Intra- und Inter-Agent Transfer Learning

Unter Berücksichtigung des allgemeinen Konzeptes des TL (vgl. Abbildung 2.24) sind in Abbildung 2.26 die Konzepte des Intra- und Inter-Agent TL im MARL gegenübergestellt.

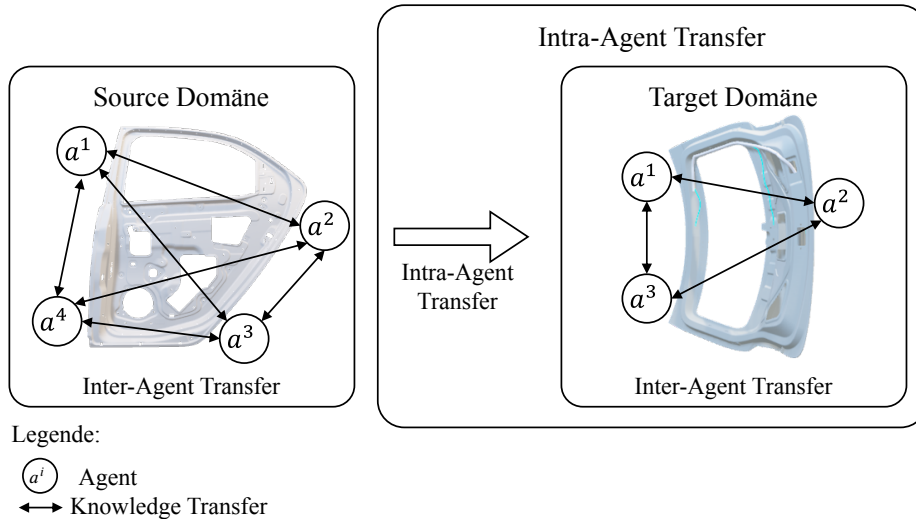


Abbildung 2.26: Gegenüberstellung von Intra- und Inter-Agent Transfer Learning

In beiden Szenarien, die in Abbildung 2.26 dargestellt sind, besteht der Grundgedanke in der Weiternutzung von bereits erlerntem Wissen zur Lösung eines definierten MG. Beide Domänen beinhalten eine spezifische Menge von Agenten $a \in N$. Während D_S und D_T individuell betrachtet einen Inter-Agent Transfer darstellen, ist die Übertragung zwischen den Domänen ein Intra-Agent Transfer. Innerhalb eines MGs entspricht der Austausch von erzeugtem Wissen in expliziter (vgl. Kapitel 2.5.3) oder impliziter Form über das Training einer gemeinsamen Policy π (vgl. CTCE in Kapitel 2.5.2) dem Inter-Agent TL [32]. Da mehrere Agenten $a \in N$ mit dem Environment interagieren, können die gesammelten Tupel $\langle S_t, \mathcal{A}, \mathcal{R}, S_{t+1} \rangle$ aller Agenten zum Erreichen der optimalen Policy π^*

einen kollektiven Mehrwert schaffen. In der Konstellation eines MARL geschieht dieser Schritt idealerweise zur Laufzeit [165]. In einem DTDE kann das Inter-Agent TL mit dem beschriebenen Learning from Demonstrations (vgl. Kapitel 2.6.3.1) gleichgesetzt werden [31]. Intra-Agent TL hingegen ist die Übertragung von erzeugtem Wissen eines Agenten auf eine neue Aufgabe T_T bzw. die Wiederverwendung von erzeugtem Wissen in einer neuen Domäne D_T [31, 32]. Es ist damit mit der Wissensübertragung aus Kapitel 2.6.3.1 vergleichbar.

2.6.5 Metriken zur Evaluierung des Wissenstransfers

Die Festlegung der Evaluationsmetriken ist als eine der zentralen Herausforderungen des TL deklariert [166]. Ein mögliches Ziel des Transfers von Wissen besteht darin, die Gesamtzeit für das Erlernen einer komplexen Aufgabe zu verkürzen. In diesem Fall ist ein Gesamtzeitszenario, das explizit die zum Erlernen der ursprünglichen Aufgabe(n) benötigte Zeit berücksichtigt, am besten geeignet. Andererseits ist ein zweites sinnvolles Ziel des TL die effektive Wiederverwendung von früher erworbenem Wissen in einer neuen Aufgabe. In diesem Fall ist ein Zielaufgabenszenario sinnvoll, das nur die für das Erlernen der Zielaufgabe benötigte Zeit einbezieht. Es ist auch zu berücksichtigen, dass ein negativer Transfer zwischen Agenten oder Environments stattfinden kann [211]. Dieser äußert sich in einer zusätzlichen Zeitaufwendung zum Erreichen der Zielsetzung [22]. Als besonders zielführend für eine Quantifizierung des konkreten Transfers erweisen sich die Metriken 1 bis 5 [31, 210]:

1. Die initiale Performance TL_{IP} eines Agenten in einer Zieldomäne D_T , verbessert durch den Transfer aus dem Source Task T_S , ist mit der *Jumpstart Performance* umschrieben.
2. Als *Asymptotic Performance* TL_{AP} ist die bestmögliche Performance, die ein Agent nach dem Transfer erreicht, definiert.
3. Die auch als Accumulated Rewards bezeichnete Metrik des *Total Reward* TL_{ToR} umfasst die Fläche unter der Lernkurve und setzt dabei die Werte in D_T des TL in Relation zu einem neu angestoßenen Lernprozess.
4. Das Verhältnis zwischen der Asymptotic Performance eines Agenten mit TL und einem Agenten ohne TL ist mit der *Transfer Ration* TL_{TR} beschrieben.
5. Die *Time to Threshold* TL_{tT} bezieht sich auf die Lernzeit in Iterationen bis zum Erreichen eines vordefinierten Schwellwertes

Die genannten Parameter basieren in diesem Zusammenhang vorrangig auf dem Lernprozess des Agenten in der Target-Domäne D_T [31].

2.7 Zusammenfassung der Analyse der theoretischen Grundlagen im Kontext dieser Arbeit

Mit der Erweiterung des als NP-schwer deklarierten Optimierungsproblems eines SPP um zusätzliche Agenten entsteht ein MAPF (vgl. Kapitel 2.1.2). Durch die Hinzunahme des zusätzlichen Einflussfaktors weiterer Agenten ergeben sich neben abweichenden Optimierungszielen auch divergierende Planungsstrategien, welche die Problemstellung des CHRP als Optimierungsproblem adäquat beschreiben (vgl. Kapitel 2.3.6). Neben der Nutzung der mechanischen 3D-CAD-Baugruppe (vgl. Kapitel 2.2.3) wird die Repräsentation der im 3D-CAD-System als B-Splines modellierten elektrischen Verbindungen (vgl. Kapitel 2.2.1) als ein wesentlicher Einflussfaktor angesehen. Aufgrund der Notwendigkeit einer feingranularen Abbildung beliebiger 3D-CAD-Geometrien in der MARL-Umgebung (vgl. Kapitel 2.5.1) und der effizienten Berechnung von Kollisionen (vgl. Kapitel 2.1.3) zeigen die Mesh-basierten Verfahren Vorteile gegenüber den Voxel-basierten Methoden (vgl. Kapitel 2.2.3). Grundlage hierfür ist ein disziplinübergreifendes Datenformat, das die 2D-basierte elektrotechnische Planung und die 3D-CAD-Modelle aus dem MCAD-System konsolidieren kann (vgl. Kapitel 2.3.2 und 2.2.5). Das XML-basierte, strukturierte AutomationML-Format erfüllt diese Anforderungen in ausreichendem Maße und inkludiert über die Referenzierung auf COLLABorative Design Activity (COLLADA)-Dateien die geometrische Repräsentation des 3D-CAD-Modells (vgl. Kapitel 2.2.5). Darüber hinaus kann es neben den kabel- und geometriespezifischen Randbedingungen (vgl. Kapitel 2.3.3) auch den separaten Einflussfaktor der Kinematik in standardisierter Form übertragen (vgl. Kapitel 2.3.4).

Für eine Substitution bestehender Prozessschritte der virtuellen Verdrahtung (vgl. Kapitel 2.3.5) bilden die MARL-Methoden in Kombination mit TL-Methoden einen wichtigen Baustein. Während die MARL-Methoden in einem kooperativen Ansatz das gemeinsame Ziel der Flowtime-Optimierung (vgl. Kapitel 2.1.2) in einem MG (vgl. Kapitel 2.5.1) abbilden können, ermöglichen die TL-Methoden die Transferierbarkeit des Wissens auf ähnliche Geometrien (vgl. Kapitel 2.6.2 und 2.2.4). So können durch die Anwendung von TL-Methoden die Agenten sowohl innerhalb eines Environments, als auch Task-übergreifend voneinander profitieren (vgl. Kapitel 2.6.4) und damit bestehende MARL-Herausforderungen lösen (vgl. Kapitel 2.5.4). Zur Gewährleistung der Skalierbarkeit der neuen Methode und zur Realisierung einer domänenspezifischen Verwendung von Kabelbäumen werden sowohl Clustering- als auch Subgraph-Identifikationsmethoden eingesetzt (vgl. Kapitel 2.4.4 und 2.1.4). Beide Methoden führen zu einer Reduzierung der Komplexität von MAPF-Problemen. Während die Clustering-Methoden auf die Anwendung räumlich spezifizierter Policies abzielen (vgl. Kapitel 2.4.4.2), bündeln die Methoden zur Identifikation von Teilbäumen mehrere Agentenpfade (vgl. Kapitel 2.1.4). Die Analysen der theoretischen Grundlagen lassen den Schluss zu, dass die Kombination aus MARL und TL den bestehenden Einschränkungen im CHRP entgegenwirkt. Die Symbiose aus Methoden zur Zusammenführung von Eingangsdaten aus disziplinspezifischen IT-Systemen, der Abstraktion des CHRP als MAPF, dem Zusammenwirken von MARL und TL mit Methoden zur Reduzierung der Problemkomplexität bildet daher das Grundgerüst der neuen Methode in Kapitel 4.

3 Stand der Technik in Industrie und Forschung

Dieses Kapitel vermittelt einen Überblick zur Analyse über den derzeitigen Stand der Technik in Industrie und Forschung im Bereich der virtuellen Verdrahtung. Der Fokus liegt dabei auf den Ansätzen zur Automatisierung der virtuellen Verdrahtung im dreidimensionalen Raum. Abschnitt 3.1 geht insbesondere auf den industriellen Kontext ein und zeigt auf, welche Methoden derzeit in den Unternehmen Anwendung finden. Davon ausgehend werden in Kapitel 3.2 sowohl der Stand der Forschung in der Anwendungsdomäne der 3D-CAD-Systeme als auch die entwickelten Methoden zum abstrahierten Problem der Pfadoptimierung analysiert. Daraus leiten sich die entsprechenden Anforderungen an einen automatisierten virtuellen Verdrahtungsprozess ab, die in Abschnitt 3.3 detailliert sind. Die untersuchten Veröffentlichungen werden in Abschnitt 3.4 im Kontext der Problemstellung aus Kapitel 1.2 sowie der Ziele (vgl. Kapitel 1.3) und Anforderungen aus Abschnitt 3.3 eingeordnet und es wird die bestehende Forschungslücke aufgezeigt.

3.1 Virtuelle Verdrahtung im 3D-CAD-System in der Industrie

Wie in Kapitel 1.1 beschrieben, ist die manuelle Definition von Wegstreckenpunkten die derzeit dominierende Vorgehensweise zur virtuellen Verdrahtung innerhalb von 3D-CAD-Systemen im industriellen Umfeld. Sie folgt den aufgezeigten Prozessschritten in Abbildung 2.11 und inkludiert eine händische Festlegung von Stützpunkten einer B-Spline im dreidimensionalen Raum (vgl. Kapitel 2.2.1). Abbildung 3.1 zeigt einen exemplarischen Prozessausschnitt der manuellen Arbeitsweise am Beispiel des 3D-CAD-Systems Siemens NX in der Version 2412.

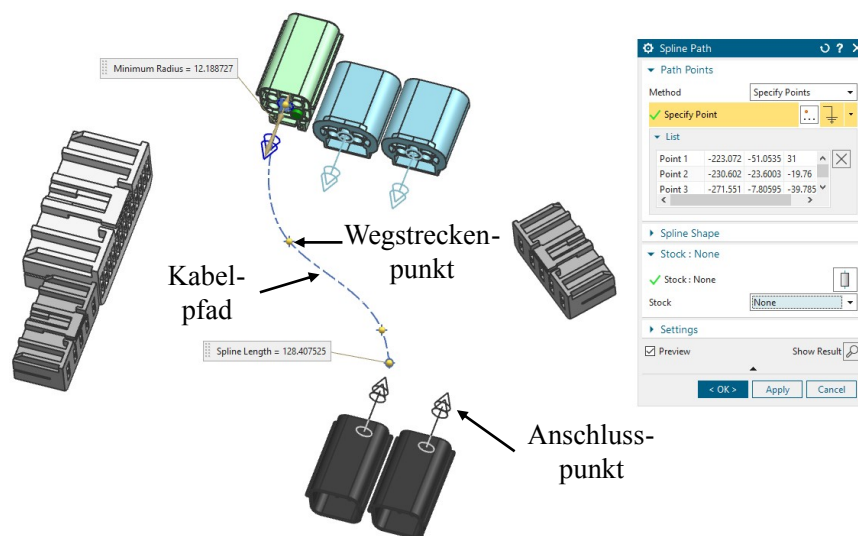


Abbildung 3.1: Exemplarische Darstellung der manuellen Pfaddefinition in einem 3D-CAD-System

Im Rahmen dieser Tätigkeit (vgl. Abbildung 3.1) legt der Anwender die Wegstreckenpunkte manuell im 3D-CAD-System mit dem Ziel fest, die jeweiligen Anschlusspunkte zu einem Kabelpfad zu verbinden. Der Prozess ist in Kapitel 2.3.5 im Detail beschrieben. Der vorzufindende Automatisierungsgrad ist daher als gering einzustufen. Andere existente Automatisierungslösungen in 3D-CAD-Systemen setzen auf die manuelle Vordefinition eines Netzwerkes. Abbildung 3.2 veranschaulicht den Prozess anhand einer exemplarischen Steckergruppierung in einem 3D-CAD-System.

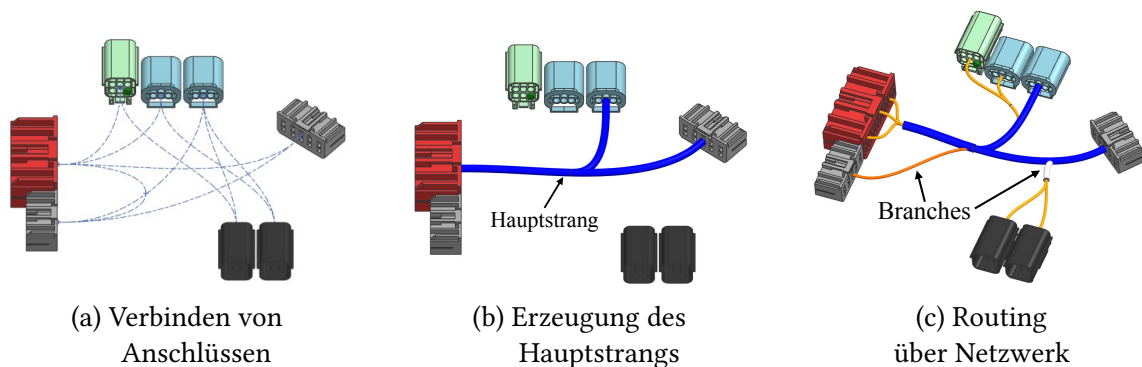


Abbildung 3.2: Exemplarische Darstellung zur Erzeugung eines Routing Networks i. A. a. [200]

Im initialen Schritt (vgl. Abbildung 3.2a) ist eine manuelle Festlegung von möglichen Pfaden erforderlich, die auf dem Aufbau einer logischen Referenz zwischen zwei Anschlusspunkten basiert. Diese manuelle Festlegung dient der Vermeidung von Kollisionen (vgl. Kapitel 2.3.3) und zeigt gleichzeitig auf, welche Leitungsführungen vorgesehen bzw. geplant sind. Das entstehende Netzwerk muss dabei alle enthaltenen elektrischen Anschlusspunkte der Verbindungsliste abdecken (vgl. Kapitel 2.3.2). Im darauf aufbauenden zweiten Schritt (vgl. Abbildung 3.2b) erfolgt die Erzeugung des Hauptstrangs. In diesem Schritt generiert das 3D-CAD-System die Topologie jener Pfadabschnitte des Netzwerkes, die vorgibt, über welche die meisten Einzelverbindungen verlaufen. Der zugehörige Durchmesser des entstehenden Bündels wird über eine zylindrische Form angenähert. Im letzten Schritt der Abfolge (vgl. Abbildung 3.2c) werden die verbliebenen Einzelverbindungen als *Branches* ergänzt (vgl. Kapitel 2.3.5). Sie vervollständigen den Kabelbaum und bilden damit die eingangs bereitgestellte elektrische Verbindungsliste in ihrer Gesamtheit ab. Als wesentliche Einschränkung ist hier die Notwendigkeit der manuellen Festlegung möglicher Pfade zu nennen. Folglich stellt diese Form des Automatismus vielmehr eine Verbesserung und Konsolidierung bestehender Pfadplanungen als eine holistische Optimierung und Automatisierung in einem 3D-CAD-System dar.

Weitere etablierte Automatisierungslösungen in der industriellen Anwendung sind in den unterschiedlichen 3D-CAD-Systemen zum Zeitpunkt dieser Arbeit nicht bekannt bzw. von den industriellen Anwendern nicht publiziert.

3.2 Stand der Forschung zur Automatisierung der virtuellen Verdrahtung

Die existenten Forschungsansätze zeichnen sich durch eine große Vielfalt an entwickelten Methoden mit unterschiedlichen Anwendungsschwerpunkten aus. Sie zeigen auf, wie das CHRP oder artverwandte Problemstellungen des MAPF auf unterschiedliche Art und Weise gelöst werden können. Im Rahmen der vorliegenden Analyse werden die untersuchten Methoden in die Kategorien „nicht-lernende“ (vgl. Kapitel 3.2.1) und „lernende“ Methoden (vgl. Kapitel 3.2.2) unterteilt. Ergänzt ist diese Segmentierung durch die Ge-

genüberstellung unterschiedlicher Environment-Repräsentationen (vgl. Kapitel 3.2.3) und deren Einflussnahme auf den Lernprozess.

3.2.1 Heuristische und naturinspirierte Methoden zur Pfadoptimierung

Die heuristischen Methoden zur Approximierung eines optimalen Pfades werden von Methoden geprägt, die den A*-Algorithmus anwenden. Die Funktionsweise des A*-Algorithmus in einer diskreten 3D-Umgebung basiert auf der Suche des kürzesten Weges von einem Startpunkt P_S zu einem Zielpunkt P_T (vgl. Kapitel 2.1). Zu diesem Zweck wird eine Prioritätswarteschlange eingesetzt, in welcher die Knoten anhand ihrer geschätzten Gesamtkosten bewertet werden. Die tatsächlichen Kosten des bisher zurückgelegten Weges werden mit einer Heuristik kombiniert, welche die geschätzten Kosten vom aktuellen Knoten zum Zielknoten angibt. Auf diese Weise ist es möglich, den optimalen Pfad zu finden. Die bestehenden Methoden im Kontext des CHRP setzen dabei auf eine iterative Abarbeitung der elektrischen Verbindungen. [37, 45, 82, 120, 209, 72, 149]

Die Methoden der Particle Swarm Optimization (PSO) und der Ant Colony Optimization (ACO) machen sich die naturinspirierte Schwarmintelligenz zunutze, um einen optimalen Pfad im dreidimensionalen Raum zu identifizieren. Die PSO verwendet eine Gruppe von Partikeln, die sich im zulässigen Suchraum bewegen [28]. Dabei aktualisiert jeder Partikel seine Position basierend auf der eigenen Erfahrung sowie der besten bekannten Position des gesamten Schwarms [201]. Die Bewegung der Partikel wird durch ihre Geschwindigkeit gesteuert, die von persönlichen und globalen Bestwerten beeinflusst wird [9, 14]. Dadurch ergibt sich eine Balance zwischen Exploration und Exploitation des Suchraums, sodass das Optimum effizient identifiziert werden kann [202].

Die ACO ist ein algorithmischer Ansatz, der von der natürlichen Fähigkeit von Ameisen inspiriert ist, den kürzesten Weg zwischen ihrem Nest und einer Nahrungsquelle zu finden [28]. In der ACO simulieren künstliche Ameisen den Prozess, indem sie beim Bewegen durch den Suchraum Pheromonspuren hinterlassen, die für andere Ameisen als Leitsignal dienen. Diese Pheromone verdunsten mit der Zeit, wodurch kürzere und effizientere Wege stärker hervorgehoben werden, da sie häufiger genutzt und mit mehr Pheromonen angereichert werden [14]. Durch diesen Mechanismus der positiven Rückkopplung und der kontinuierlichen Pfadbewertung können die Ameisenkolonien effizient optimale oder nahezu optimale Lösungen finden. [138, 137]

Im Zusammenhang mit naturinspirierten Methoden zeigen sich vermehrt Verfahren unter Anwendung von genetischen Algorithmen [90, 206, 199, 84]. Diese Form von Algorithmen ist von der natürlichen Evolution inspiriert und nutzt Prinzipien natürlicher Selektion, Mutation und Rekombination, um optimale Lösungen in einem Suchraum zu finden [90]. Begonnen wird mit einer Population von potenziellen Lösungen, die als Chromosomen bezeichnet werden [206]. Die Bewertung der Chromosomen erfolgt anhand einer Fitnessfunktion, wobei die besten ausgewählt werden, um die nächste Generation zu erzeugen [199]. Durch Kreuzung und Mutation entstehen neue Lösungen, die den Suchraum erkunden. Über mehrere Generationen hinweg verbessert sich die Population, indem sie sich an die besten Lösungen anpasst [84].

Die bestehenden Grenzen der Anwendung von heuristischen und naturinspirierten Verfahren wie dem A*-Algorithmus zeigen sich insbesondere bei hohen Wechselwirkungen

zwischen den einzelnen Agenten und einer großen Anzahl von Agenten [28]. Obgleich naturinspirierte und metaheuristische Ansätze den Suchraum nach optimalen Lösungen sehr gut erkunden können, sind sie bei einer signifikanten Zunahme der Problemdimensionen (z. B. durch eine höhere Agentenanzahl N) mit einer erhöhten Rechenlast verbunden [78].

3.2.2 Lernbasierte Lösungsmethoden zur Pfadoptimierung

Den heuristischen und naturinspirierten Methoden aus Kapitel 3.2.1 werden in diesem Abschnitt die lernenden Methoden gegenübergestellt. Mittels ihrer Anwendung wird die Identifikation einer Strategie zur Lösung des vorliegenden MAPF auf Basis der Methoden des ML verfolgt (vgl. Kapitel 2.4). Insbesondere MARL-Methoden werden im Folgenden genauer analysiert. In der Teilkategorie der Value-based Methoden (vgl. Kapitel 2.5) bewegen sich die Agenten zumeist in einem diskreten, Grid-basierten Environment [110, 111, 25, 35, 180, 197]. Wenige Ausnahmen nutzen einen kontinuierlichen Raum für die Actions [35]. Varianten und Abwandlungen eines Deep-Q-Networks bilden dabei die Mehrheit der Lösungsalgorithmen in den untersuchten Methoden.

Die Policy-based Methoden (vgl. Kapitel 2.5) bieten ein breiteres Spektrum an Lösungsalgorithmen und werden konträr zu den Value-based Methoden vor allem im kontinuierlichen Raum eingesetzt [190, 70, 47, 24, 136, 62, 153]. Vor allem Methoden für das MAPF mit einer hohen Agentenanzahl wenden hierbei ein diskretes Environment an [147]. Sie sind geprägt von der Formalisierung als POMG und nutzen somit als Input für das künstliche neuronale Netz eine Observation O statt eines ganzen State S (vgl. Kapitel 2.5.2) [190, 70, 47, 24, 136, 62, 147, 153]. Ausnahmen in dieser Kategorie der Lösungsalgorithmen wenden ein Fully-observable Environment an [70, 76]. Auffällig ist eine Reduzierung des DAF auf 0.185 bzw. 0.333, was den Handlungsspielraum der einzelnen Agenten $a^i \forall i \in \{1, \dots, N\}$ deutlich einschränkt (vgl. Kapitel 2.1.2). Zudem finden die meisten dieser Methoden lediglich in Problemstellungen im zweidimensionalen Raum Anwendung [147, 47, 24]. Weiterführende Gegenüberstellungen von MAPF-Methoden nach anderen Kriterien sind in der Literatur zu finden, werden jedoch im Kontext dieser Arbeit nicht näher beleuchtet [14, 6].

3.2.3 Repräsentation der dreidimensionalen Geometrie

Insbesondere die Gestaltung des Environments und die damit zusammenhängenden Auswirkungen auf den Lernprozess sind ein elementarer Einflussfaktor auf die automatisierte virtuelle Verdrahtung in einem 3D-CAD-System. Neben den in Kapitel 2.2.3 analysierten räumlichen Darstellungen in Form eines Mesh oder eines Voxels ist die Überführung der 3D-CAD-Geometrie in ein konvexes Kollisionsobjekt von Interesse (vgl. Kapitel 2.2.3). Aufgrund der besonders performanten Berechnung von Kollisionen ist ein konvexes Objekt für eine effiziente Pfadsuche im kontinuierlichen Raum sehr geeignet (vgl. Kapitel 2.2.3).

In der Mehrzahl der analysierten Methoden ist das Environment zur Pfadplanung als diskrete räumliche Darstellung festgelegt [209, 91]. Diese Darstellungsformen lassen sich weiter in graph- und zellgestützte Repräsentationen differenzieren. Insbesondere graph-basierte Darstellungsformen (en: *Road Maps*) von 3D-CAD-Geometrien finden sich in zahlreichen Anwendungsfällen wieder [14]. Dies ist darauf zurückzuführen, dass etablierte Pfadplanungsalgorithmen wie der A*-Algorithmus hier in recheneffizienten Verfahren gute

Ergebnisse erzielen [200, 209, 112]. Eine damit einhergehende Reduktion der Problemdimensionalität führt zu einer Beschleunigung des Gesamtprozesses. Die direkte Extraktion der geometrischen Daten aus dem 3D-CAD-native-Objekt als Mesh (vgl. Kapitel 2.2.3) vereinfacht zudem den Gesamtprozess [174, 45, 202]. Dem steht die zellbasierte Beschreibung (en: *Navigation Mesh*) der 3D-CAD-Geometrie gegenüber [14]. Die kontinuierliche Form wird in ein gleichmäßiges, überschneidungsfreies Raster (en: *Grid*) bzw. in Zellen unterteilt. Sie kann analog zur Beschreibung eines Voxels betrachtet werden [91, 90, 149] (vgl. Kapitel 2.2.3). Zudem existieren diverse Mischformen, die beide Environment-Repräsentationen zusammenführen, um die jeweiligen Einschränkungen zu reduzieren [120]. Weitergehende Forschung untersucht den Einfluss einer Reduzierung des Zustandsraumes bzw. die Belohnung für die Nutzung von spezifischen Bereichen innerhalb der 3D-CAD-Geometrie [9].

Die Darstellungsform von 3D-CAD-Geometrien als konvexe Objekte stützt sich im Stand der Technik auf die Methoden der Volumetric-Hierarchical Approximate Convex Decomposition (VHACD) und der Collision-Aware Approximate Convex Decomposition (CoACD). Beide Ansätze sehen die Zerlegung eines gegebenen 3D-Modells in eine Menge konvexer Teile vor. Sie unterscheiden sich jedoch in der Zerlegungsabfolge. Während die VHACD-Methode eine voxelbasierte Darstellung des Modells verwendet, um die dreidimensionale Geometrie in kleine Würfel zu zerteilen, nutzt die CoACD-Methode kovarianzbasierte Techniken. Diese Approximierungsmethoden bieten eine Näherungslösung, welche die Form des ursprünglichen 3D-Modells bestmöglich mit konvexen Teilen annähert. [167, 186]

Die effiziente Berechnung von Kollisionen im kontinuierlichen dreidimensionalen Raum macht die konvexen Darstellungsformen zu einem besonders geeigneten Verfahren für die Verarbeitung innerhalb eines 3D-MARL-Systems. Daher findet diese Form der 3D-CAD-Repräsentanz Berücksichtigung in der neuen Methode in Kapitel 4.3.1.

3.3 Abgeleitete Anforderungen an die automatisierte virtuelle Verdrahtung

Aus der analysierten Literatur ergeben sich neben dem Forschungsbedarf (vgl. Kapitel 3.4) auch die wesentlichen Anforderungen an eine neue Methode zur Lösung des CHRP. Analog zur Auflistung von Einflussfaktoren auf die virtuelle Verdrahtung in Kapitel 2.3.3 sind diese in zwei Bereiche kategorisierbar. Die Anforderungen an den Kabelpfad als Bestandteil eines MAPF ergeben sich im Wesentlichen aus den physikalischen Eigenschaften der strom- und signalleitenden Objekte. So motivieren die bestehenden Ansätze neben der Einhaltung von Mindestbiegeradien auch die Kollisionsfreiheit der Kabel mit sich selbst, anderen elektrischen Verbindungen und mechanischen Bauteilen (vgl. Kapitel 2.3.3). Im Bereich der Verdrahtungsumgebung sind die Faktoren der ausreichenden Darstellungsgenauigkeit sowie der Speichereffizienz und der Bereitstellung von Kollisionsberechnungen entscheidend für das Erreichen eines validen Gesamtergebnisses. Aus prozessualer Sicht spielen die verfügbaren Eingangsgrößen und deren Konsolidierung eine ebenso relevante Rolle wie die Informationsbereitstellung der Ergebnisse des CHRP für Folgeprozesse (vgl.

Kapitel 2.3.5). Die aufgeführten Anforderungen (vgl. Punkte 1. bis 5.) dienen als Grundlage für die Bewertung der bestehenden Ansätze sowie der neuen Methode:

1. **Branching:** Eine neue Methode zur automatisierten Lösung eines CHRP (vgl. Kapitel 2.1.4 und 2.3.5) muss nicht nur dazu in der Lage sein, für einzelne Agenten einen optimalen Pfad durch das Environment zu identifizieren (vgl. Kapitel 2.1), sondern diese auch bestmöglich bündeln können. Damit können Anwendungsfälle unterschiedlicher Domänen weitreichend abgedeckt werden.
2. **Biegeradius:** Zur Sicherstellung eines validen Gesamtergebnisses der virtuellen Verdrahtung (vgl. Kapitel 2.3.3) ist es erforderlich, grundlegende Randbedingungen einzuhalten. Die Einhaltung des minimalen Biegeradius je Kabeltyp wird neben der Kollisionsfreiheit dabei als elementarer Faktor betrachtet.
3. **Dynamische Bauteile:** Der Einflussfaktor von Kinematik innerhalb eines CHRP repräsentiert eine spezielle Form der Randbedingung. Neben der datentechnischen Verfügbarkeit der notwendigen Informationen benötigt ein Automatismus die Fähigkeit, den vollständigen Bewegungsraum der mechanischen 3D-CAD-Baugruppe abzudecken (vgl. Kapitel 2.3.4)
4. **Minimale Pfadlänge:** Als Optimierungsziel (vgl. Kapitel 2.1.2) ist für das CHRP die minimale Flowtime gesetzt, da sie eine ganzheitliche Minimierung aller Einzelverbindungen anstrebt. Für ein optimales Ergebnis steht die holistische Problembetrachtung anstelle einer individuellen Pfadoptimierung im Fokus.
5. **Wissenstransfer (TL):** Insbesondere im Kontext von Produktvarianten (vgl. Kapitel 1.2) ist eine Übertragung von bereits erzeugtem Wissen als vorteilhaft anzusehen. Das Erlernen und die Weiterverwendung einer generierten Policy π ist die Intention des TL und bricht damit die bestehende Kardinalität zwischen Lerntask T_T und Zieldomäne D_T auf (vgl. Kapitel 2.6.2)

3.4 Fazit aus den Analysen zum aktuellen Stand der Forschung

Zur effizienten Lösung eines CHRP oder artverwandter MAPF-Probleme stehen verschiedene wissenschaftliche Methoden zur Verfügung. Tabelle 3.1 veranschaulicht den bestehenden Forschungsbedarf auf Basis der analysierten Literatur in den Kapiteln 3.1 und 3.2. Dabei impliziert ein +-Zeichen, dass die Quelle die entsprechende Anforderung (vgl. Kapitel 3.3) vollständig erfüllt. Eine partielle Erfüllung wird durch das Symbol (+) angezeigt. Ist eine Anforderung nicht erfüllt oder nicht Teil der Veröffentlichung, so ist die zugehörige Zelle in Tabelle 3.1 leer. Die Spalte VA repräsentiert die neuen Inhalte der Methodenentwicklung in der vorliegenden Arbeit. Die Bewertungskriterien sind in die Kategorien *Lösungsmethode* und *Anforderungen zur virtuellen Verdrahtung* unterteilt.

Die durchgeführte Gegenüberstellung der Analyseergebnisse in Tabelle 3.1 zeigt bei den angewendeten Lösungsmethoden einen starken Fokus auf iterative Ansätze in diskreten Environments [209]. Innerhalb dieser Darstellungsformen von 3D-CAD-Modellen existierten unterschiedliche Pfadsuchalgorithmen, die optimale Lösungen approximieren können

Tabelle 3.1: Analyse der Forschungslücke in den relevanten Veröffentlichungen zur virtuellen Verdrahtung

Merkmale	[37]	[88]	[209]	[202]	[120]	[91]	[35]	VA
Lösungsmethode								
Single-Agent / iterativ	+	+	+		+	+		
Multi-Agent				(+)			+	+
Diskreter Raum	+	+	+	+		+		
Kontinuierlicher Raum					+		+	+
Kooperatives Verhalten	(+)	+	(+)	(+)	(+)		+	+
Anforderungen zur virtuellen Verdrahtung								
Branching	+	+	+	+				+
Biegeradius	+	(+)	+	+	+	(+)		+
Minimale Flowtime	+	+	+	+	+	+	(+)	+
Wissenstransfer	(+)							+
Dynamische Bauteile							(+)	(+)

(vgl. Kapitel 3.2) [160, 168]. Die Verwendung von kooperativen Multi-Agent-Methoden, insbesondere in kontinuierlichen Lösungsräumen, ist aufgrund des nahezu unendlich großen State-Spaces S im Kontext der virtuellen Verdrahtung bisher wenig erforscht. Der Lösungsraum wird mit einer zunehmenden Agentenanzahl N beinahe exponentiell größer, was eine noch komplexere Lösbarkeit des NP-schweren MAPF zur Folge hat (vgl. Kapitel 2.1.2) [6]. Jedoch wirkt sich die Nutzung einer kooperativen Methode in kontinuierlichen Environments positiv auf die Genauigkeit der entstehenden Lösung aus, was die Rückführung der Ergebnisse in das 3D-CAD-System simplifiziert (vgl. Kapitel 2.3.5). Zudem ermöglicht dies eine Generierung von validen Gesamtergebnissen.

In der Betrachtung der Anforderungen an die virtuelle Verdrahtung im 3D-CAD-System (vgl. Kapitel 3.1) lässt sich ein bestehender Fokus auf die Integration von Branching und die Einhaltung von Biegeradien erkennen (vgl. Kapitel 2.1.4, 2.3.3 und 2.3.5). Die Verwendung von Branching-Methoden steht in engem Zusammenhang mit der adressierten Anwendungsdomäne in mehreren Veröffentlichungen [209, 91, 88]. Die Einhaltung des Biegeradius ist hingegen als wesentliche Randbedingung nahezu in allen Publikationen inkludiert [37]. Darüber hinaus motivieren alle genauer analysierten Methoden die Erzeugung von minimalen Gesamtpfadlängen als Optimierungsziel [202]. Keine Veröffentlichung integriert in das CHRP den Wissenstransfer zwischen artverwandten 3D-CAD-Geometrien oder die Berücksichtigung von beweglichen Baugruppen [44].

Die resultierenden Forschungsbedarfe bzw. -lücken können folglich in Form eines Multi-Agent-Ansatzes in einem kontinuierlichen Environment zusammengefasst werden. Ein kooperatives Verhalten (vgl. Kapitel 2.5.2) ist dabei in der Lage, das Ziel der minimalen Gesamtkabellänge zu unterstützen. Die bisher fehlende Integration eines Wissenstransfers auf artverwandte CHRP (vgl. Kapitel 2.6) macht den Forschungsbedarf ebenso deutlich wie das bisher nicht erfolgte Berücksichtigen beweglicher Bauteile (vgl. Kapitel 2.3.4). Resultierend aus dieser Gegenüberstellung liegt der Fokus der neuen Methode in Kapitel 4 auf der Schließung der analysierten Lücken (vgl. Tabelle 3.1 - Spalte VA). Dies wird

kombiniert mit den Anforderungen aus Kapitel 3.3. Eine Übersicht über die eigenen Publikationen, welche die in diesem Kontext aufgezeigten Forschungsbedarfe adressieren, findet sich in Tabelle A.1.

4 Methode zum Einsatz von Multi-Agent Reinforcement Learning für die virtuelle Verdrahtung im 3D-CAD

Im vorliegenden Kapitel wird die neue Methode zur automatisierten virtuellen Verdrahtung im 3D-CAD beschrieben. Abschnitt 4.1 verwendet dafür die identifizierten Anforderungen aus Kapitel 3.3 und kombiniert diese mit den neuen methodischen Voraussetzungen (vgl. Kapitel 3.3). Die datentechnische Integrationsstrategie und eine damit einhergehende Vorverarbeitung der Eingangsinformation werden in Abschnitt 4.2 entwickelt und detailliert beschrieben. Eine Formulierung der industriellen Problemstellung der virtuellen Verdrahtung im 3D-CAD-System als MG erfolgt in Abschnitt 4.3. Die Abschnitte 4.4 und 4.5 führen aus, wie sich kooperativ kommunizierende MARL-Algorithmen und 3D-CNNs einsetzen lassen, um die Multi-Agent-Pfadoptimierung im kontinuierlichen dreidimensionalen Raum durchzuführen. Eine Zusammenfassung der neuen Methode zur automatisierten virtuellen Verdrahtung erfolgt in Abschnitt 4.6.

4.1 Anforderungen und Voraussetzungen für die entwickelte Methode zur virtuellen Verdrahtung

In diesem Kapitel erfolgt zunächst eine Einordnung der neuen Methode in den Prozess der virtuellen Verdrahtung in Abschnitt 4.1.1. Die dafür essenzielle Qualifizierung der 3D-CAD-Geometrie und die Anforderungen an die Verbindungsliste sind in Abschnitt 4.1.2 beschrieben. Eine Abstraktion der virtuellen Verdrahtung als Pfadoptimierungsproblem ist Inhalt des Abschnitts 4.1.3. Die Festlegung des zu erreichenden Optimierungsziels der konzipierten Methode erfolgt in Abschnitt 4.1.4. Das Erreichen dieser Optimalität unterliegt den Annahmen für die Lösung des CHRP, die in Abschnitt 4.1.5 aufgeführt werden.

4.1.1 Einordnung der neuen Methode zur virtuellen Verdrahtung in den Produktentstehungsprozess

Wie bereits in Kapitel 1.2 detailliert beschrieben, stellt die virtuelle Verdrahtung im 3D-CAD-System eine der zeitintensivsten und repetitivsten Phasen der virtuellen Produktentwicklung dar (vgl. Kapitel 2.3.5). Aufgrund der Vielzahl von Randbedingungen (vgl. Kapitel 2.3.3) ist es für den Menschen nur schwer möglich, die darin enthaltenen Wechselwirkungen zu überblicken. Die neuen Methoden dieser Arbeit fügen sich in den bestehenden Prozess ein und ergänzen bzw. substituieren den Prozessschritt der Pfadfestlegung im dreidimensionalen Raum (vgl. Kapitel 2.3.5). Durch den Einsatz der neuen Methoden kann die Definition der Wegstreckenpunkte automatisiert und der bisherige Zeit- und Arbeitsaufwand signifikant reduziert werden. Darüber hinaus offeriert die Anwendung der entwickelten MARL-Methoden die Möglichkeit, ein globales Optimum (vgl. Kapitel 4.1.4) zu erreichen. Abbildung 4.1 verortet die entwickelten Methoden im Kontext eines CHRP im Produktentstehungsprozess. Dies dient der schematischen Einordnung der neuen MARL-Methoden und ihres Zusammenwirkens mit dem 3D-CAD-Modell im CHRP1 sowie der Übertragung des Wissens auf das CHRP2.

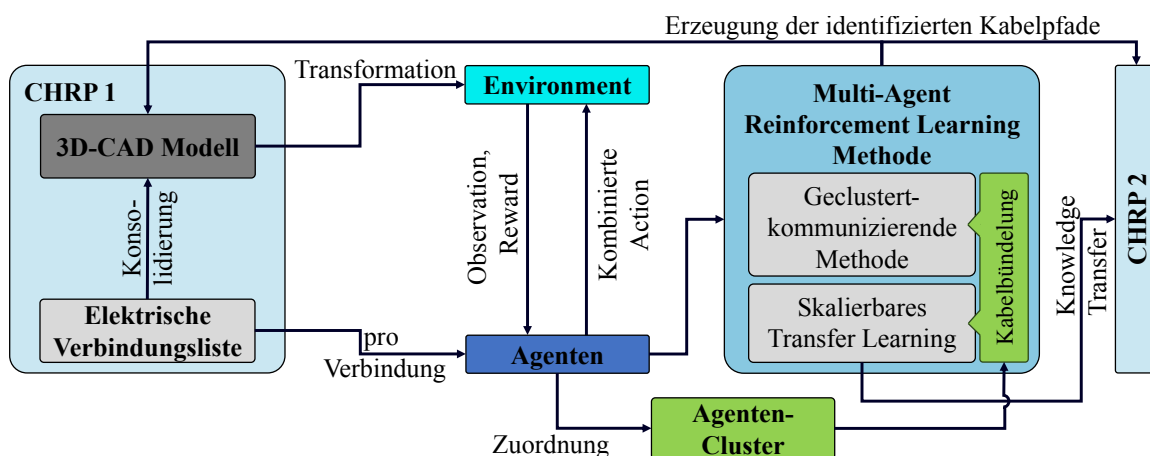


Abbildung 4.1: Zusammenwirken der entwickelten Methoden mit einem Cable Harness Routing Problem

Durch die konkrete Verortung der neuen Methoden im Gesamtprozess der virtuellen Produktentwicklung (vgl. Kapitel 2.3.5) ergeben sich neben den Voraussetzungen für die Outputgrößen auch die Anforderungen an die Verfügbarkeit der Eingangsgrößen zum entsprechenden Zeitpunkt. Dem CHRP1 liegen neben dem 3D-CAD-Modell die Verbindungsliste und deren Merkmale zugrunde (vgl. Kapitel 2.3.2). Das 3D-CAD-Modell wird für die Anwendung des MAPF speichereffizient transformiert (siehe Kapitel 4.2.3) und damit vom konkreten 3D-CAD-System entkoppelt. Analog dazu existiert für jede elektrische Verbindung ein zugehöriger Agent gemäß dem Konzept des MARL (vgl. Kapitel 2.5.2). Dieser Agent beinhaltet neben den Informationen zur Start- und Zielkoordinate auch Merkmale, die den assoziierten Kabeltyp festlegen. Mit dieser datentechnischen Zwischenschicht wird aus dem CHRP ein MAPF-Problem, das mit Methoden des MARL lösbar ist (vgl. Kapitel 2.1.2 und 2.5.1). Im Zusammenwirken mit dem Kollektiv aller Agenten integrieren die entwickelten Lösungsmethoden die Anwendung von kommunizierenden Methoden (vgl. Kapitel 2.5.3) und eine Methode für das TL (vgl. Kapitel 2.6.1). Beide Methoden verfolgen das Ziel der Kabelpfaderzeugung direkt im mechanischen 3D-CAD-Modell. Als optionaler Baustein ergänzt die Kabelbündelung beide Methoden. Basierend auf einem Verfahren zur Subgrapherzeugung (vgl. Kapitel 2.1.4) werden die Agenten zu einem Kabelbaum gruppiert und tragen somit zur Skalierbarkeit der Lösung bei.

Im speziellen Fall der TL-Methode wird das Zusammenwirken von MARL-Lösungsmethode und CHRP1 um ein weiteres, artverwandtes CHRP2 ergänzt (vgl. Abbildung 4.1). Dieses differenziert sich entweder durch die Anzahl der elektrischen Verbindungen - respektive Agenten - oder durch eine geometrische Varianz (vgl. Kapitel 2.2.4). Auch die Variation der zu berücksichtigenden Randbedingungen (vgl. Kapitel 2.3.3) kann die Definition der CHRP als Varianten voneinander beeinflussen. Im Sinne der Weiterverwendung von bereits generiertem Wissen kann so der vortrainierte MARL-Algorithmus auf das neue CHRP2 angewendet werden. In Bezug auf den Knowledge Transfer (vgl. Kapitel 2.6.1), dargestellt in Abbildung 4.1, ist im Zusammenhang mit der geometrischen Ähnlichkeit zwischen zwei Szenarien zu unterscheiden. Im Falle einer sehr hohen geometrischen Similarität wird der Zero-Shot-Transfer angewendet (vgl. Kapitel 2.6.2), der ohne jegliche Adaption auf die Zieldomäne einsetzbar ist. Liegt die Ähnlichkeit hingegen unter einem zuvor spezifizierten Schwellwert, so wird ein Few-Shot-Transfer (vgl. Kapitel 2.2.4) durchgeführt. Innerhalb weniger Trainingsiterationen kann sich die MARL-Methode somit an die Modifikationen der Zieldomäne anpassen (vgl. Kapitel 2.6.2).

Überschreitet die Verbindungsliste eine bestimmte Anzahl an Einzelverbindungen, können die isoliert betrachteten MARL-Lösungsmethoden das CHRP nicht mehr effizient lösen. Je nach Anwendungsdomäne besteht zudem eine bevorzugte Planung von Verbindungen in Kabelbündeln, was auch zu einer Reduzierung der MAPF-Komplexität beiträgt (vgl. Kapitel 2.3.2). Zu diesem Zweck werden die einzelnen Agenten in mehrere Cluster gruppiert (vgl. Kapitel 2.4.4). Die Datenbasis dafür sind die Anschlusspunkte im dreidimensionalen Raum und die Kabelmerkmale der Einzelverbindungen. Auf Grundlage dessen trägt die Integration eines Navigation-Graphs (vgl. Kapitel 2.3.5) in beiden Submethoden dazu bei, das CHRP auf vereinfachte Art und Weise zu lösen. Gleichzeitig ermöglicht diese Methode die Integration einer hohen Anzahl an elektrischen Verbindungen in groß dimensionierten 3D-MCAD-Geometrien.

4.1.2 Anforderungen an die 3D-CAD-Geometrie, die Verbindungsliste und die Kabelstammdaten

Für die Anwendung einer automatisierten virtuellen Verdrahtung bestehen diverse Anforderungen an die mechanischen und elektrischen Eingangsdaten. Die datentechnische Grundlage eines CHRP bildet neben der 3D-CAD-Geometrie aus dem MCAD-System eine elektrische Verbindungsliste mit Kabelinformationen aus dem ECAD-System (vgl. Kapitel 2.3.5). In Abhängigkeit von der Anwendungsdomäne muss ebenfalls die Kinematik in der Pfadsuche berücksichtigt werden (vgl. Kapitel 2.3.4 und 4.2.4). Auch hier bestehen Anforderungen an die Verfügbarkeit und die Beschaffenheit der Eingangsdaten.

Der Qualifizierungsgrad der 3D-CAD-Geometrie umfasst vor allem die Verfügbarkeit von elektrischen Anschlusspunkten mit der zugehörigen Benennung. Die Komponentenbezeichnung muss mit dem Identifikator innerhalb der ECAD-Daten übereinstimmen, um die Daten für die virtuelle Verdrahtung in harmonisierter Form nutzen zu können (vgl. Kapitel 2.3.2 und 4.2.2). Innerhalb des Volumenmodells ist festzulegen, an welcher Stelle elektrische Geräte platziert und deren jeweilige Anschlusspunkte positioniert sind. Diese werden neben einer eindeutigen Bezeichnung pro Komponente um einen Richtungsvektor ergänzt, der zur Sicherstellung des kabelspezifischen Biegeradius am Komponenteneintritt dient [45]. Die Anschlusspunkte besitzen zusätzlich die Information, ob es sich nur um eine singuläre Anschlussmöglichkeit handelt oder ob der Kabeleintritt mehrfach belegt werden darf. Dies wirkt sich auf die Übertragung der Information in das RL-Environment aus (vgl. Kapitel 4.2.3 und 4.3.1). Im Falle einer beweglichen Baugruppe oder Komponente muss darüber hinaus die Zusammengehörigkeit der betroffenen Einzelteile festgelegt sein. Abbildung 4.2 zeigt den notwendigen Qualifizierungsgrad eines Anschlusspunktes am Beispiel eines Leuchtmelders in Anlehnung an den ECLASS-Standard. Dieser Anschlusspunkt des Beispielbauteils deckt alle erforderlichen Anforderungen ab und dient nachfolgend als Grundlage für die Qualifizierung aller zur Anwendung kommenden elektrischen Einzelteile innerhalb des 3D-MCAD-Systems.

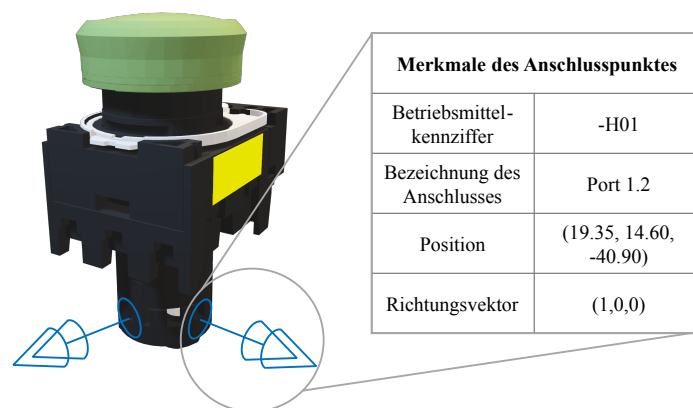


Abbildung 4.2: Erforderliche Qualifizierung eines Anschlusspunktes in einem exemplarischen mechanischen 3D-CAD-Modell

Neben der Verfügbarkeit von Anschlusspunkten muss das 3D-CAD-Modell mit Informationen zu nicht zugänglichen Zonen angereichert sein (vgl. Abbildung 2.7c, Kapitel 2.3.3). Diese schränken den Zugang zu verbotenen Zonen für alle oder eine Teilmenge der Agenten (Kabel) ein. Die zugehörigen Daten können als Bestandteil der Komponentenmerkmale oder in Form einer maschinenlesbaren Annotation (PMI) vorhanden sein. Sie stellen eine beispielhafte Integration von Randbedingungen dar und haben kein standardisiertes Format (vgl. Kapitel 2.3.2). Lediglich ihre maschinenlesbare Ausprägung ist eine elementare Anforderung für die nahtlose Integration und Nutzung im Gesamtprozess.

Eine essenzielle Anforderung an die Eingangsdaten für elektrische Verbindungen besteht darin, dass die Bezeichnungen der Quell- und Zielkomponenten sowie die Bezeichnung ihre Anschlusspunkte mit den geometrischen Äquivalenten übereinstimmen (vgl. Tabelle 2.2a in Kapitel 2.3.2). Andernfalls ist eine automatisierte Zusammenführung der mechanischen und elektrischen Daten aufgrund von Inkonsistenzen nicht möglich. Die tabellarischen oder XML-basierten Informationen zu den jeweiligen Einzelverbindungen sind um deren Kabeltyp, Durchmesser und minimalen Biegeradius zu ergänzen. Diese Informationen gewährleisten die Gültigkeit des Verdrahtungsergebnisses sowie eine realitätsgetreue Kollisionsfreiheit (vgl. Kapitel 2.3.3). Werden ungeschirmte Leitungstypen verwendet, so wirken sich die EMV-Merkmale zusätzlich auf die Pfadoptimierung in Form einer wechselseitigen Beeinflussung der einzelnen Agenten (Kabel) aus (vgl. Kapitel 2.3.3).

Zur Nutzung der kinematischen Informationen einer speziellen Geometrie (vgl. Kapitel 4.2.4) ist eine entsprechende Festlegung der Beziehung zwischen den Einzelkomponenten (Links) notwendig. Diese umfasst neben den beteiligten Bauteilen die Lage, den Typ sowie die minimalen und maximalen Positionen des Gelenks (vgl. Kapitel 2.3.4). Für eine standardisierte und maschinenlesbare Festlegung des Joints sind spezifische Anforderungen je Joint-Typ erforderlich. Jeder verwendete Link hat ein zugehöriges Koordinatensystem $\{l\}$ und ist mit einem Joint j assoziiert. Dieser verbindet den Link l mit dem vorherigen Link $l - 1$. Daraus folgt die Benennung des Eltern-Links (en: *Parent Link*, $l - 1$) und der Kinder-Links (en: *Child Link*, l). Die Relativbewegung der beiden Links zueinander wird durch die Gelenkvariable (oder Translationsvariable) β_j charakterisiert. Sie beschreibt die Rotation um die Gelenkachse im Falle von Revolute Joints bzw. die Translation entlang der Gelenkachse im Falle von Prismatic Joints. Eine Ausnahme bildet hier der Basislink, der keinen assoziierten Joint hat. Darüber hinaus wird festgelegt, dass für jeden Joint j die y-Achse des assoziierten Koordinatensystems der Rotationsachse entspricht und die z-Achse sich parallel zum Richtungsvektor der Oriented Bounding Box (OBB) des Links l ausrichtet. Die x-Achse ergänzt das rechtshändige Koordinatensystem. Abbildung 4.3 zeigt exemplarisch die Qualifikation eines Revolute Joints am Beispiel eines Robotergelenks.

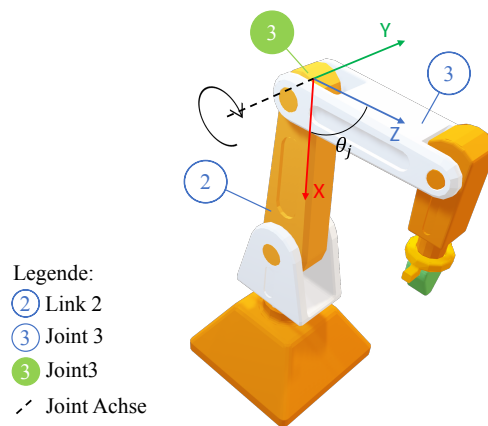


Abbildung 4.3: Exemplarische Darstellung der Qualifizierung einer Rotationsbewegung am Beispiel eines Robotergelenks

Das exemplarisch erzeugte Koordinatensystem $\{j\}$ des Joints $j = 3$ fließt als Festlegung der Beziehung zwischen zwei Einzelteilen $l = 2$ und $l = 3$ in die Erweiterung der Routing-Umgebung um Kinematikbedingungen ein (vgl. Kapitel 4.2.4). Sie bilden die Basis für die Identifikation von kinematischen Bereichen, welche die Validität des Verdrahtungsergebnisses durch die bewegliche Beschaffenheit der Geometrie beschränken (vgl. Kapitel 2.3.4).

4.1.3 Abstraktion der virtuellen Verdrahtung als Multi-Agent Pathfinding Problem

Zur Erreichung des Ziels der Anwendung von KI-Methoden (vgl. Kapitel 2.4.1) für die virtuelle Verdrahtung im 3D-CAD-System ist eine Abstraktion der Wegstreckendefinition für Kabelpfade notwendig (vgl. Kapitel 2.3.6). Die Abstraktion des CHRP als eine Multi-Agenten-Pfadoptimierung eröffnet die Möglichkeit, etablierte Methoden aus diesem Forschungsfeld anzuwenden und diese an bestimmten Stellen maßgeblich zu erweitern (vgl. Kapitel 3.2). Im Folgenden sind die Anschlusskoordinaten der elektrischen Verbindung als äquivalent zu den Start- und Zielpunkten P_S^i und P_T^i des Agenten a^i , $i \in \{1, \dots, N\}$ festgelegt. Mit der Tessellierung des mechanischen 3D-CAD-Modells und der zugehörigen Erzeugung eines Kollisionsobjektes (vgl. Kapitel 4.3.2) wird eine Analogie zur formalisierten Definition der Pfadoptimierung im dreidimensionalen Raum gezogen (vgl. 2.1.1). Die Punkte $P(x, y, z)$, die eine spezifische Position innerhalb der 3D-CAD-Geometrie definieren, repräsentieren in dieser Betrachtungsweise die Knoten V . Der euklidische Abstand zwischen zwei Punkten entspricht dem Gewicht der Kantensegmente. Daraus ergeben sich die entsprechenden Arten von Kollisionen, die in der MAPF weiterhin Bestand haben (vgl. Kapitel 2.1.3). Besonders der *Vertex*-Konflikt (vgl. Abbildung 2.3b) sowie der *Swapping*-Konflikt (vgl. Abbildung 2.3e) müssen betrachtet werden. Aufgrund der Beschaffenheit der Kabel als Festkörper und deren dauerhafter Besetzung einer Position im dreidimensionalen Raum

entfallen die verbleibenden Konflikttypen (vgl. Kapitel 2.1.3). Folglich wird das CHRP als MAPF im kontinuierlichen dreidimensionalen Raum betrachtet. Dieses unterliegt den Randbedingungen, die in Kapitel 2.3.3 beschrieben sind, und dient dem Erreichen der gesetzten Optimierungsziele (vgl. Kapitel 4.1.4).

4.1.4 Festlegung der Optimierungsziele

Wie in den Kapiteln 2.1.2 und 2.6.5 dargestellt, gibt es mehrere Optimierungsziele, die im dargestellten Pfadoptimierungsproblem erreichbar sein sollen. Es sind grundsätzlich die Optimierungsziele der Flowtime, der Makespan und die Zielerreichung unter einer gegebenen Deadline einsetzbar (vgl. Kapitel 2.1.2). Die Optimierung nach Makespan führt lediglich zu einer Minimierung der Kabellänge des längsten Kabels. Somit kann die Erreichung der kürzesten Gesamtkabellänge nicht gewährleistet werden. Die Flowtime hingegen dient der Minimierung der Summe von Zeitschritten aller Agenten. Dies repräsentiert bestmöglich den Anwendungsfall der minimierten Gesamttrajektorie $L(\tau^i)$ gemäß MAPF. Eine Zielerreichung unter gegebener Deadline ist lediglich dann relevant, sofern eine physikalische Restriktion für eine Maximallänge einer elektrischen Verbindung besteht.

Aufgrund des einzubeziehenden Einflusses aller Einzelpfade in die Optimierung wird die Anwendung der Flowtime als Optimierungsziel für die nachfolgende methodische Entwicklung und die zugehörige Validierung festgelegt (vgl. Kapitel 4.3). Dies hat Auswirkungen auf die Gestaltung des Rewards für die Submethoden in den Abschnitten 4.4 und 4.5. An dieser Stelle wird darauf hingewiesen, dass in den realen Anwendungsfällen der virtuellen Verdrahtung auch der betriebswirtschaftliche Einfluss, z. B. die Kosten pro laufenden Meter je Kabeltyp, im Optimierungsziel berücksichtigt werden müssen. Die Gleichung 4.1 beschreibt das Optimierungsziel für die Gesamtkabellänge im Kontext der neuen Methode.

$$\min \sum_{i=1}^N L(\tau^i) \quad (4.1)$$

$$\text{u. d. N.} \quad P^i(t=0) = P_S^i \quad \wedge \quad P^i(t=T) = P_T^i, \quad \forall i \in \{1, \dots, N\} \quad (4.2)$$

$$P^i(t) \neq P^j(t), \quad \forall i \neq j, \forall t \in T \quad (4.3)$$

Dabei beschreibt Gleichung 4.1 das übergeordnete Optimierungsziel der minimierten Gesamtkabellänge. Die Variable τ^i repräsentiert die einzelne Trajektorie des Agenten i , $L(\tau^i)$ die Gesamtlänge dieser Trajektorie. Die minimierte Summe aller Einzeltrajektorien $\tau^i \forall i \in \{1, \dots, N\}$ ist analog zum Ziel der Flowtime-Minimierung anzusehen (vgl. Kapitel 2.1.2). In der Nebenbedingung 4.2 gibt $P^i(t=0) = P_S^i$ an, dass die Position des Agenten i im initialen Zeitschritt t der Startposition entspricht. Diese Bedingung gilt analog für die Zielposition im Zeitschritt $t = T$ für alle Agenten. Darüber hinaus beschreibt $P^i(t) \neq P^j(t)$ die Kollisionsfreiheit der Pfadplanung als weitere notwendige Nebenbedingung des Optimierungsproblems. Speziell für die neue Methode zum Knowledge Transfer in Kapitel 4.5 erweitert sich das angenommene Optimierungsziel um die Metriken des TL aus Kapitel

2.6.5. Diese stehen in Wechselwirkung mit dem im MAPF definierten Ziel der Flowtime-Minimierung. Von besonderer Relevanz sind die Metriken Jumpstart und Transfer Ratio (vgl. Kapitel 2.6.5). Der Jumpstart spielt für die Bewertung der initialen Performance eines Agenten in einem neuen CHRP eine entscheidende Rolle (vgl. Kapitel 2.6.5). Diese Metrik gibt Aufschluss darüber, ob die transferierte Policy π aus der Source-Domäne D_S in der Lage ist, das CHRP der Zieldomäne D_T in ausreichender Güte zu lösen. Im Kontext der Differenzierung zwischen der Notwendigkeit eines Few-Shot-Lernprozesses gegenüber einem Zero-Shot-Transfer ist dies ein elementarer Indikator (vgl. Kapitel 2.6.3). Die Transfer Ratio zeigt an, ob der Transfer überhaupt positiven Einfluss auf die neue Lösung des CHRP in der Zieldomäne D_T hat. Sofern der Lernprozess ohne TL ein besseres Gesamtergebnis im Sinne des Rewards \mathcal{R} aufweist, ist der Transfer als negativ zu bewerten. Die Quantifizierung des Verhältnisses zwischen den beiden Asymptotic Performances (vgl. Gleichung 4.4) ist dafür ein wesentlicher Indikator und findet daher Anwendung in der Validierung der entwickelten Methode in Kapitel 5.

$$TL_{TR} = \frac{\mathcal{R}_T(D_T)}{\mathcal{R}_B(D_T)} \quad (4.4)$$

In Gleichung 4.4 repräsentiert $\mathcal{R}_T(D_T)$ den Gesamt-Reward in der Anwendung der TL-Methode und setzt diesen ins Verhältnis zum Gesamt-Reward ohne TL $\mathcal{R}_B(D_T)$. Folglich sind der Lernprozess wie auch die Auswahl der zu transferierenden Policy π_{D_S} auf Basis der TL-Metriken durchzuführen (vgl. Kapitel 2.6.5). Unter Berücksichtigung der zu optimierenden Flowtime ergibt sich eine Pareto-Optimalität zwischen den Metriken der Multi-Agenten-Pfadoptimierung und den Metriken des TL (vgl. Kapitel 2.6.5).

Für die Bereitstellung eines optimierten Kabelbaums (vgl. Kapitel 2.3.5) besteht ein weiteres, entkoppeltes Optimierungsziel. Innerhalb dessen werden die optimalen Verzweigungspunkte der Einzelverbindungen gesucht, die zwei Endpunkte P_T^i miteinander verbinden. Der resultierende Kabelbaum (en: *Harness*) H ist ein Subgraph des Gesamtgraphen $G(V, E)$ (vgl. Kapitel 2.1.4). Unter Berücksichtigung einer Menge von Startpunkten P_S und Endpunkten P_T der Einzelverbindungen, eines Kabelbaumes $H \subset G(V, E)$ der alle Start- und Endpunkte verbinden soll, und eines Gesamtgraphen $G(V, E)$ beschreibt Gleichung 4.5 das angestrebte Minimierungsziel.

$$\min_H \left(\sum_{u,v \in E_H} d(u,v) \right) \quad (4.5)$$

$$\text{u. d. N.} \quad H \subset G(V, E) \wedge P_B \cap (P_S \cup P_T) = \emptyset \quad (4.6)$$

Gleichzeitig gelten die Nebenbedingungen der Konnektivität, der Verzweigungspunkt-Bedingung und die Minimierung der Gesamtlänge des Subgraphen H , bestehend aus den Edges E_H und Vertices V_H des Graphen. Der Subgraph $H \subset G(V, E)$ muss alle Start- und Zielpunkte $P_S \cup P_T$ verbinden. Darüber hinaus dürfen die Verzweigungspunkte P_B keine Start- oder Zielpunkte sein, also $P_B \cap (P_S \cup P_T) = \emptyset$. In Gleichung 4.5 beschreibt V die

Menge aller Knoten des Graphen $G(V, E)$ und P_B die Menge aller Verzweigungspunkte. Die Variable d beschreibt die Distanz zwischen zwei Punkten. Diese Minimierung ist Bestandteil der Skalierbarkeit der neuen Methode in Kapitel 4.5.1. Sie ist damit eine Anpassung des Steiner Trees mit einer klaren Restriktion auf die Auswahl der Verzweigungspunkte P_B (vgl. Kapitel 2.1.4).

4.1.5 Annahmen und Limitationen für die Lösung des 3D-Routingproblems

Die Anwendung der neuen Methoden erfordert die Erfüllung von Annahmen, die den Lösungsraum des CHRP einschränken und die Validität der Ergebnisse sicherstellen. Die grundlegende Annahme für die Lösung eines CHRP mit den neuen Methoden ist die Existenz einer validen Lösung für die vorliegende Problemstellung aller Einzelverbindungen und ihres Kollektivs. Andernfalls ist es mittels MARL-Methode nicht möglich, die gesetzten Optimierungsziele aus Kapitel 4.1.4 zu erreichen. Eine erfolgreiche Verwendung einer MARL-Methode ergibt sich zudem nur für ein Set von elektrischen Verbindungen für $N \geq 2$ (vgl. Kapitel 2.5.2).

Für die Lösungsidentifikation stehen jedem Kabel, unter Berücksichtigung des kabelspezifischen Biegeradius (vgl. Kapitel 2.3.3), alle Freiheitsgrade für dessen Bewegung im dreidimensionalen Raum zu. Dies umfasst neben Aktionen innerhalb von Ebenen auch das Verfolgen von Diagonalverbindungen. Daraus ergibt sich, dass eine explizite Position im dreidimensionalen Raum lediglich von einem Kabel eingenommen werden kann. Versuchen zwei Kabel gleichzeitig, eine neue gleiche Position einzunehmen, so bleiben beide auf der vorherigen Position stehen, sofern es keine Form der Priorisierung einzelner Verbindungen gibt. Zur Vermeidung dieser Situation ist es einer elektrischen Verbindung innerhalb der Pfadplanung erlaubt, in einem Schritt die Position nicht zu verändern. Dies dient dem reaktiven Verhalten auf die Bewegungsabfolge anderer Beteiligter des Pfadplanungsproblems und hat einen DAF von 1 zur Folge (vgl. Kapitel 2.1.2). Hat ein Kabel seinen Zielpunkt P_T erreicht, kann es weder weitere Aktionen ausführen noch automatische Korrekturen an der bisher gewählten Aktionsabfolge τ^i durchführen.

Im Hinblick auf die Geometrie und die elektrischen Eingangsgrößen besteht neben den in Kapitel 4.1.2 beschriebenen Anforderungen die Einschränkung des Bewegungsraumes. Sie ergibt sich aus einem bestehenden maximalen Abstand von Verbindungen zur Oberfläche der transformierten 3D-CAD-Geometrie oder implizit durch die physikalische Beschaffenheit der Kabel. Der maximale Oberflächenabstand d_{max} ist durch die Notwendigkeit einer Kabelbefestigung gegeben, da in diversen Anwendungsdomänen eine Norm für einen zulässigen Abstand zwischen zwei Befestigungspunkten an der mechanischen Geometrie besteht. Im Kontext der Integration von Kinematik in die neue Methode (vgl. Kapitel 4.2.4) ist für jeden Joint eine maximale und minimale Ausprägung gegeben. Insbesondere in Rotationen unterbindet diese Information die Notwendigkeit von Wicklungen oder Schleifkontakten zwischen zwei Bauteilen. Unter Berücksichtigung der angegebenen Annahmen ist eine Identifikation von validen Kabelpfaden und der damit einhergehenden Lösung des CHRP mit der neuen Methode gegeben (vgl. Kapitel 4.2.5).

4.2 Datentechnische Integrationsstrategie

Auf Grundlage des bestehenden Prozesses zur virtuellen Verdrahtung aus Kapitel 2.3.5 ist in Abschnitt 4.2.1 die nahtlose Integration der neuen Methoden in die gängige IT-Systemarchitektur dargestellt. In Abschnitt 4.2.2 liegt der Fokus auf der Zusammenführung der relevanten Eingangsdaten für die automatisierte Verdrahtung unter Anwendung des Datenformats AutomationML. Die Vorverarbeitung der mechanischen 3D-CAD-Geometrie ist in Abschnitt 4.2.3 beschrieben. Die Erweiterung dieses Zustandsraumes um die Einflussnahme und die Integration von kinematischen Informationen zeigt Abschnitt 4.2.4. Die Rückführung der identifizierten Kabelpfade in das 3D-MCAD-System in Abschnitt 4.2.5 schließt das Kapitel ab.

4.2.1 Zugrunde liegende IT-Systemarchitektur zur Integration der neuen Methoden

Die Systemarchitektur zur Integration der neuen Methoden für eine virtuelle Verdrahtung veranschaulicht den Einsatz von kooperativ kommunizierenden Methoden auf Basis eines Pfadclusterings (vgl. Kapitel 4.4.1 und 4.4) sowie die Verwendung einer TL-Methode mit Metainformationen (vgl. Kapitel 4.5). Die einzelnen Komponenten der Systemarchitektur sind modular aufgebaut und kommunizieren über Representational State Transfer (REST)-Schnittstellen miteinander. Dies dient dem Ziel einer hohen Kohäsion und einer losen Kopplung im Sinne des Softwareprinzips der Single Responsibility. So können einzelne Klassen oder ganze Softwarekomponenten wie der Lösungsalgorithmus oder das Environment ohne Abhängigkeiten innerhalb der Lösungssoftware ausgetauscht werden. Ein zugehöriges Unified-Modeling-Language (UML)-Sequenzdiagramm für den Prozess ist in Abbildung A.1 im Anhang dargestellt.

Abbildung 4.4 zeigt die einzelnen IT-Systemkomponenten und ihr Zusammenwirken in einem UML-Klassendiagramm. Die Einfärbung der vier Hauptkomponenten *Environment*, *Agenten*, *Lösungsalgorithmen* und *CAD-Systeme* ist analog zur methodischen Übersicht in Abbildung 4.1 gehalten.

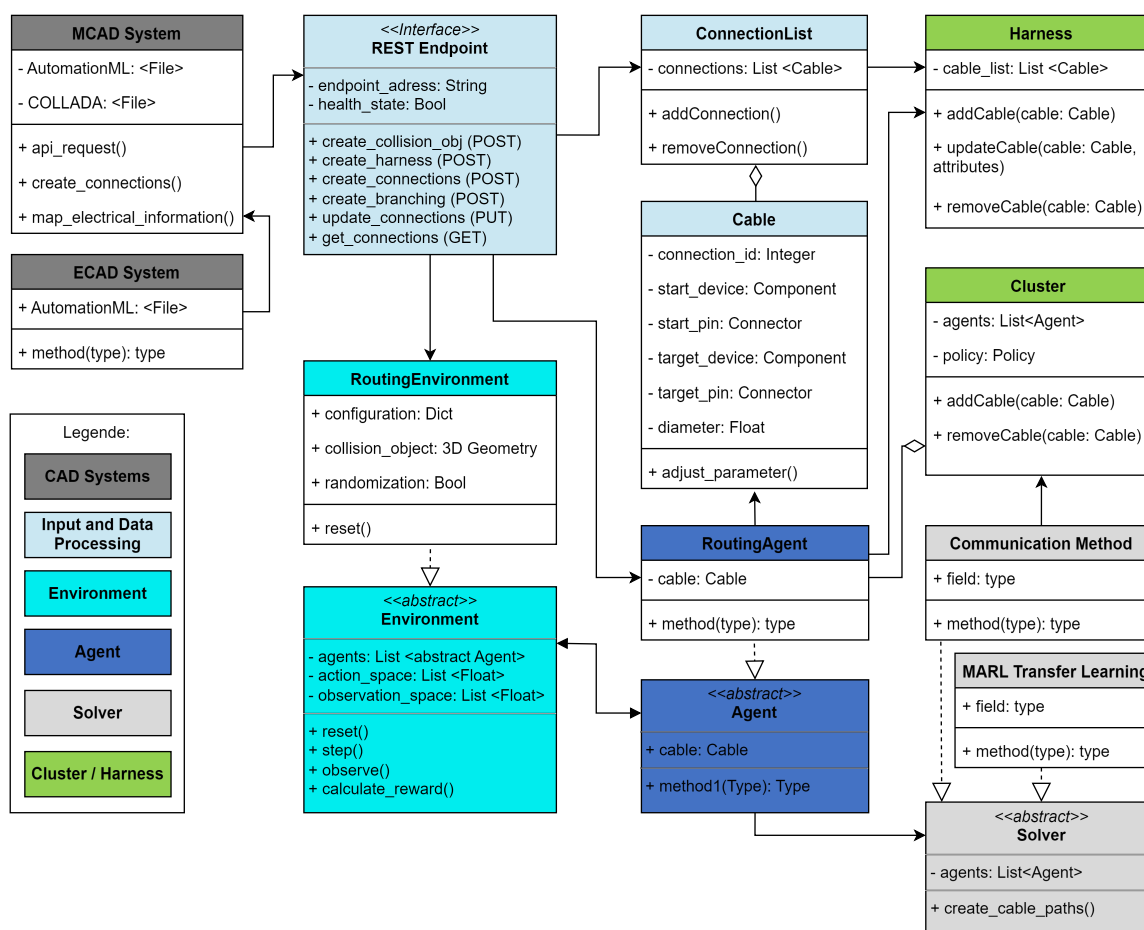


Abbildung 4.4: UML-Klassendiagramm der Methode für die virtuelle Verdrahtung im 3D-CAD-System

Die Beschreibung der einzelnen Klassen und ihres Funktionsumfangs in Abbildung 4.4 folgt dem Datenfluss und ihrer Initialisierung. Ausgehend von den Autorensystemen im MCAD und ECAD bildet eine *REST*-Schnittstelle den Kommunikationsbaustein zwischen der entwickelten Methode und den bestehenden IT-Systemen. Die Architektur bietet damit die Möglichkeit einer unabhängigen Nutzung von 3D-CAD-Systemen und der Lösungsmethode zur Pfadfindung und -optimierung. Die zentrale Aufgabe der Interface-Klasse liegt neben der Bereitstellung der REST-Endpunkte und der Transformation der Geometriedaten insbesondere in der Bereitstellung der erforderlichen Informationen aus der Verbindungsliste und den Kabeltypen für die interne Weiterverarbeitung. Dies erfordert die Verfügbarkeit der MCAD- und ECAD-Daten in Form einer AutomationML-Datei als disziplinunabhängiges Datenaustauschformat (vgl. Kapitel 2.2.5). Die konsolidierten Daten der beteiligten Disziplinen werden in einer angereicherten Verbindungsliste (*Connection-List*) bereitgestellt. Diese erweitert die elektrotechnischen Informationen (vgl. Kapitel 2.3.2) um deren geometrische Zuordnung in Form von 3D-Koordinaten. Damit liegt das CHRP ab diesem Zeitpunkt als vollständig definiertes MAPF vor (vgl. Kapitel 2.1.2). Neben der Verarbeitung der Eingangsinformationen dient der REST-Endpoint der Rückführung der identifizierten optimierten Kabelpfade und deren realitätsgetreuer Erzeugung im MCAD-

System. Für die interne Orchestrierung der Daten stoßen die jeweiligen API-Endpunkte zudem die Erzeugung von *Agent*, *Environment* und einer konkreten Ausprägung der *Solver*-Klasse an. Diese Bausteine bilden gemeinsam den Kern des MARL-Problems (vgl. Kapitel 2.5).

Das *RoutingEnvironment* (siehe Abbildung 4.4) erbt von einer übergeordneten abstrakten Klasse, welche die notwendigen Grundfunktionen für das MG beschreibt. Diese Entkopplung dient der Modularisierung und Individualisierbarkeit einer konkreten Ausprägung eines *RoutingEnvironments*. Jedes Environment muss in der Lage sein, die Grundfunktionen *reset()*, *step()*, *observe()* und *calculate reward()* abzubilden, um einen Lernprozess im MARL zu befähigen. Aus dem *Environment* heraus werden zudem das 3D-Mesh-Objekt und das Kollisionsobjekt erzeugt, aus dem das 3D-*Routing Environment* primär besteht. Diese repräsentieren die 3D-MCAD-Geometrie für die Pfadoptimierung, deren Komponenten um die Anschlusspunkte angereichert sind (vgl. Abbildung 4.2). Die einzelnen Teilbereiche der 3D-Geometrie enthalten im Wesentlichen Informationen über deren Belegung (vgl. Kapitel 4.2.3 und 4.3.1). In einem Factory-Pattern instanziiert das Environment Objekte vom Typ *Agent* und interagiert mit diesen (vgl. Kapitel 2.5).

Eine Instanz der Klasse *Agent* steht für eine elektrische Verbindung aus der Verbindungsliste und enthält neben dem Startpunkt (P_S) und dem Zielpunkt (P_T) auch kabelspezifische Informationen. Zur Laufzeit sind dies die aktuelle Position $P(x, y, z)$, der einzuhaltende Biegeradius r_{min} und die bisher zurückgelegte Trajektorie τ_t^i des Agenten i zum Zeitpunkt t . Abhängig von der verwendeten Methode bildet eine Gruppe von Agenten ein Cluster im Sinne einer Aggregation (vgl. Kapitel 4.4.1) oder einen Strang eines Kabelbaumes. Sie modifizieren in einer Betrachtung als Kabelbündel die Kardinalität zwischen Agent und Kabel von $1 : 1$ zu $1 : n$ (vgl. Kapitel 2.3.5). Folglich stellt zu diesem Zeitpunkt ein Agent das Äquivalent für mehrere Einzelverbindungen dar.

Neben der Verknüpfung zum Environment besteht für die *Agenten*-Klasse eine Verbindung zu einer abstrakten *Solver*-Klasse. Die abstrakte Ausprägung ermöglicht eine einfache Integration zusätzlicher Lösungsalgorithmen und erhöht die Modularisierbarkeit des Gesamtkonzepts. So erben die genutzten Algorithmen der Kapitel 4.4 und 4.5 von der abstrakten Klasse *Solver*. Sie bildet damit ein Strategy-Pattern, das die modulare Erweiterung um zusätzliche abgeleitete Klassen zulässt. Ihre Aufgabe besteht in der Entscheidungsfindung über die Aktionen der Agenten $i \in \{1, \dots, N\}$ im Zeitschritt t zur Erzeugung der jeweiligen Gesamttrajektorien τ^i . Treten die Einzelverbindungen als Teil eines Kabelbündels auf, so bestimmt die *Solver*-Klasse analog dazu die durchzuführende Aktion des Konvoluts.

Für eine Kabelbündelbetrachtung können die Agenten entsprechenden Clustern zugeordnet sein (vgl. Kapitel 2.4.4). In Form einer Aggregation gruppiert ein Objekt der *Cluster*-Klasse mehrere Einzelagenten und somit deren elektrische Einzelverbindungen. Dieses Objekt kann dann für alle erbenden *Solver*-Klassen für eine skalierbare Lösung eingesetzt werden. Primär wird das Cluster jedoch in der Communication-Methode verwendet.

Die Rückgabe der Ergebnisse in das 3D-CAD-System aller Solver, respektive aller Agenten, des Pfadoptimierungsprozesses erfolgt wiederum über den REST-Response. Aufgrund der hohen Anzahl von Wegpunkten in der Umgebung und der begrenzten Übertragungsmöglichkeit in das 3D-CAD-System ist eine Approximation des optimierten Kabelpfades

notwendig. Die Pfade werden im 3D-CAD-System automatisch in Form von B-Splines (vgl. Kapitel 2.2.1 und 2.2.2) dargestellt. Daher werden die einzelnen Punkte des Routingenvironments durch einen entsprechenden B-Spline approximiert und übertragen (vgl. Kapitel 4.2.5). Die Ergebnisse stehen dem Mitarbeiter zur manuellen Nachbearbeitung und zur Erstellung der notwendigen Montageinformationen (vgl. Kapitel 2.3.5) direkt im 3D-MCAD-System zur Verfügung (vgl. Abbildung A.1).

Eine explizite Klasse zur Realisierung des kinematischen Verhaltens der Geometrie ist nicht Bestandteil des UML-Klassendiagramms (vgl. Abbildung 4.4). Da dieses Verhalten lediglich ein Funktionsbaustein innerhalb des Environments ist, handelt es sich vielmehr um eine darin enthaltene Funktionalität. Die spezifische Ausprägung der Kinematik ergibt sich aus den Eingangsinformationen des MCAD-Systems (vgl. Kapitel 2.3.2).

4.2.2 Zusammenführung von ECAD- und MCAD-Daten mittels AutomationML

Der initial notwendige Schritt für die Lösung des CHRP ist die Konsolidierung der Eingangsdaten aus dem ECAD- und dem MCAD-System zur Erzeugung einer synchronisierten Datenbasis. Zwar bestehen nach derzeitigem Stand sowohl für die mechanischen Daten als auch für die elektrischen Daten jeweils Standardaustauschformate, wie STEP für MCAD-Systeme (vgl. Kapitel 2.2.5), diese sind jedoch in einer disziplinübergreifenden Betrachtung stark heterogen. Als disziplinübergreifendes Austauschformat ist AutomationML klassifiziert und damit gut geeignet, um die Anforderungen zu erfüllen (vgl. Kapitel 2.2.5). Es dient der Zusammenführung von Daten aus unterschiedlichen disziplinspezifischen IT-Systemen und hat seinen Ursprung im Anlagenbau. COLLADA, das assoziierte Datenformat für die Repräsentation von dreidimensionalen Geometrien, ist ein offenes Austauschformat und basiert auf einer XML-Struktur. Damit kann es unter anderem Mesh-Strukturen darstellen und austauschen (vgl. Kapitel 2.2.3).

Die neue Methode zur virtuellen Verdrahtung sieht als Konsolidierungsbasis das 3D-MCAD-System vor. Hier liegen alle dreidimensionalen Objekte in einer konsolidierten CAD-Struktur vor, die neben den mechanischen auch die elektrischen 3D-CAD-Komponenten umfasst. Die Mehrheit der 3D-MCAD-Systeme nutzt für den Schritt der Integration von ECAD-Daten in das MCAD-System einen standardisierten Import von Verbindungsdaten oder die manuelle Zuordnung attributiver elektrotechnischer Daten. Basierend auf einer einheitlichen Benennung der Komponenten in Form einer BMKZ und den definierten Anschlusspunkten (vgl. Kapitel 4.1.2) lassen sich die elektrotechnischen und die mechanischen 3D-CAD-Daten initial konsolidieren. Damit steht die Information zur Verfügung, um identifizieren zu können, welche Anschlusspunkte im 3D-CAD-System über eine elektrische Verbindung verknüpft werden sollen und welche Merkmale der verwendete Kabeltyp aufweist. Für eine weiterführende Verarbeitung ist jedoch die Überführung beider disziplinspezifischer Informationen in ein gemeinsames Format notwendig, um eine Unabhängigkeit zwischen der neuen Methode und dem 3D-MCAD-System zu schaffen (vgl. Kapitel 4.2).

Abbildung 4.5 zeigt den strukturellen Aufbau einer AutomationML-Datei an einem Beispiel. Nachfolgend sind in den Punkten 1 bis 6 die einzelnen Bestandteile des übergeordneten **<CAEXFile>** in der AutomationML-Version 2.0 beschrieben. Die Auflistung detailliert den abgebildeten Inhalt in Abbildung 4.5 und die zugehörige datentechnische

Quelle aus den ECAD- bzw. MCAD-Daten. Jeder Bestandteil ist über einen universell eindeutigen Bezeichner (en: *Universally Unique Identifier (UUID)*) in alphanumerischer Form referenzierbar.



Abbildung 4.5: Exemplarische Darstellung der AutomationML-Struktur einer Baugruppe

Die Elemente in der AutomationML-Datei sind:

1. **<InternalElement>** Gesamtbaugruppe: Das übergeordnete *InternalElement* beschreibt die gesamte Baugruppe. Es orientiert sich an der mechanischen Stückliste bzw. der 3D-MCAD-Struktur. Eine analoge Struktur besteht auch für existente Unterbaugruppen.
2. **<Attribute>** Gesamtbaugruppe: Der erste *Attribute*-Abschnitt beschreibt die Position und die räumliche Rotation der Baugruppe über einen 3D-Punkt und einen Rotationsvektor bestehend aus float-Werten. Auch diese Informationen entstammen dem 3D-MCAD-System.
3. **<InternalElement>** Einzelteil: Alle Einzelteile der (Gesamt-)Baugruppe sind unter dem obersten Stücklistenknoten als eigenständiges *InternalElement* mit Position und Rotationsvektor aufgeführt.
4. **<ExternalInterface>** Verweis: Das *ExternalInterface* vom Typ *COLLADAInterface* referenziert auf eine COLLADA-Datei, die mindestens eine visuelle Repräsentanz des Objektes beinhaltet. Zu diesem Zweck werden die 3D-CAD-Objekte in eine Mesh-Struktur überführt (vgl. Kapitel 4.2.3), um sie in die ganzheitliche AutomationML-Struktur einfließen zu lassen. So bleiben trotz der Vereinfachung der dreidimensionalen Bauteile (vgl. Kapitel 2.2.3) auch ergänzende Merkmale, wie der Werkstoff und die Farbzuzuweisung, erhalten. Insbesondere der Erhalt und die Übermittlung von kinematischen Informationen (vgl. Kapitel 2.3.4) innerhalb der COLLADA-Datei sind ein entscheidender Faktor für die Integration von beweglichen Bauteilen in die neue Methode (vgl. Kapitel 4.2.4). Darüber hinaus erlaubt die externe Referenzierung auch die Substitution der COLLADA-Datei durch andere standardisierte Dateiformate, z. B. Universal Robot Data Format (URDF)-Dateien.

5. **<ExternalInterface>** Elektrischer Anschluss: Ein *ExternalInterface* vom Typ *SignalInterface* beschreibt einen spezifischen elektrischen Anschluss innerhalb eines *InternalElements*. Neben der Position ist auch hier ein Rotationsvektor angegeben, der die Einhaltung des Biegeradius ab dem Verbindungsbeginn sicherstellt (vgl. Kapitel 2.3.3). Die zugehörigen Daten entstehen aus dem Mapping von BMKZ aus dem ECAD-System und der Benennung des elektrischen Ports innerhalb der 3D-MCAD-Geometrie (vgl. Kapitel 4.2.2). Die Benennungen sind ein wesentlicher Aspekt für die Zusammenführung der Daten.
6. **<InternalLink>** Elektrische Verbindung: Die einzelnen elektrischen Verbindungen sind als *InternalLink* innerhalb der AutomationML-Struktur der übergeordneten Baugruppe abgebildet. Neben der Referenzierung auf die jeweiligen Anschlusspunkte über die UUID und den elektrischen Port von *RefPartnerSideA* und *RefPartnerSideB* geben sie die zugehörigen Beschreibungsmerkmale zum Kabeldurchmesser und -typ an.

Das gewählte Datenformat ermöglicht durch die Verwendung von AutomationML insbesondere einen hohen Grad an Standardisierung (vgl. Kapitel 2.2.5). Es dient zur disziplinübergreifenden Datenkonsolidierung und schafft für den Prozess der automatisierten virtuellen Verdrahtung eine einheitliche Eingangsgröße. Die damit verbundene Entkopplung von den CAD-Systemen unterstützt das grundlegende Gesamtkonzept der datentechnischen Integrationsstrategie (vgl. Kapitel 4.2). So eröffnet sich die Möglichkeit für die Schaffung eines Single Point of Truth als konsistente Planungsgrundlage (vgl. Kapitel 1.2 und 1.3), der bestehende disziplinspezifische Standards ergänzt (vgl. Kapitel 2.2.5).

4.2.3 Transformation des 3D-CAD-Modells und domänenspezifische Reduzierung des Zustandsraumes

Aufbauend auf der Bereitstellung der Eingangsdaten in Form einer AutomationML-Struktur ist die Erzeugung des Environments eine wesentliche Grundlage für die Abstraktion des CHRP zu einem MAPF. Beim Transformationsprozess hin zu einem MARL-Environment durchläuft die 3D-CAD-Geometrie einen mehrstufigen Prozess, der insbesondere eine domänenspezifische Reduzierung des Zustandsraumes erfordert. Abbildung 4.6 veranschaulicht die Zwischenschritte an einer exemplarischen Geometrie.

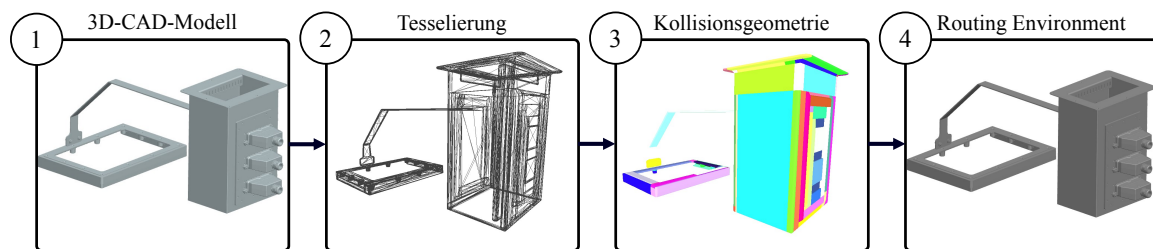


Abbildung 4.6: Ableitung des 3D-Routing-Environments aus der 3D-CAD-Geometrie

Im initialen Schritt (siehe Abbildung 4.6 (1)) erfolgt die Ausleitung der 3D-CAD-Geometrie aus dem 3D-MCAD-System in Form einer COLLADA-Datei. Bei diesem Ausleitungsprozess wird die geometrische Repräsentanz der Baugruppe tesselliert (vgl. Abbildung 4.6 (2) und Kapitel 4.2.2). Dabei wird die genaue mathematische Beschreibung der Flächen des 3D-CAD-Objekts durch B-Splines (vgl. Kapitel 2.2.1) in mehrere polygonale Darstellungen konvertiert. Unter Einflussnahme von Winkel-, Kanten- und Flächentoleranz entsteht eine spezifizierte Granularität der Triangulation (vgl. Abbildung 2.6). Eine höhere Genauigkeit führt zu mehr und kleineren Dreiecken, was die geometrischen Details besser wiedergibt, aber auch die resultierende Datenmenge erhöht. Zum Erhalt unterschiedlicher Detaillierungsstufen der 3D-CAD-Objekte innerhalb einer Baugruppe erfolgt eine iterative Transformation in tessellierte Strukturen für alle Einzelteile. Dies ermöglicht gleichzeitig eine Differenzierung des Detaillierungsgrades bzw. die Vereinfachung der Bauteile. So werden z. B. modellierte Schrauben abstrahierter trianguliert als kabelführende Bauteile (vgl. Kapitel 2.3.5).

Der zweite Zustand (vgl. Abbildung 4.6 (2)) zeigt die tessellierte 3D-Geometrie des exemplarischen Bauteils. Es dient sowohl als datentechnische Grundlage für die Berechnung der Kollisionsgeometrie als auch zur Erstellung des Kabelbaums in Form eines Graphen (vgl. Kapitel 4.5.1).

Im dritten Teilmodell (vgl. Abbildung 4.6 (3)) wird die Kollisionsgeometrie selbst abgebildet. Im Gegensatz zu existierenden anderen Methoden (vgl. Kapitel 3), die Diskretisierungsfehler oder suboptimale Zerlegungen erzeugen, nutzt das in der hier angewendeten CoACD-Methode (vgl. Kapitel 3.2.3) integrierte Verfahren eine mehrstufige Baumsuche in Form einer Monte Carlo Tree Search. Damit wird eine global optimierte Zerlegung der vorgegebenen 3D-Mesh-Struktur erreicht. Darüber hinaus werden hier konvexe Hüllen durch eine präzisere Mesh-Dekomposition anstelle einer bisher üblichen temporären Voxelgenerierung erzeugt (vgl. Kapitel 2.2.3 und 3.2.3). Die CoACD-Methode verwendet dazu

eine *Divide and Conquer-Strategie*, um das Mesh rekursiv zu zerlegen. Dabei wird für jede Komponente, deren Konkavität über einem Schwellwert liegt, eine geeignete Schnittebene gesucht (vgl. Kapitel 3.2.3). Der Konkavität einer jeden Komponente liegt die maximale Hausdorff-Distanz zwischen ihr und ihrer konvexen Hülle zugrunde (vgl. Kapitel 3.2.3). Das entstandene Kollisionsobjekt (vgl. Abbildung 4.6 (3)) bildet die Datengrundlage für die Reduzierung des Zustandsraums. Abbildung 4.7 verdeutlicht anhand von zwei exemplarischen 3D-CAD-Geometrien die Reduzierung des Zustandsraumes der MARL-Agenten in Abhängigkeit von den geometrischen Abmessungen und der Anwendungsdomäne. Das Mesh in Abbildung 4.7 zeigt jeweils die äußere Begrenzung des Zustandsraumes (Environment), die kolorierten Geometrien die innenliegende Kollisionsgeometrie. Die Farbgebung in der CoACD-Darstellung folgt keiner definierten Logik und ist somit willkürlich gewählt (vgl. Kapitel 3.2.3).

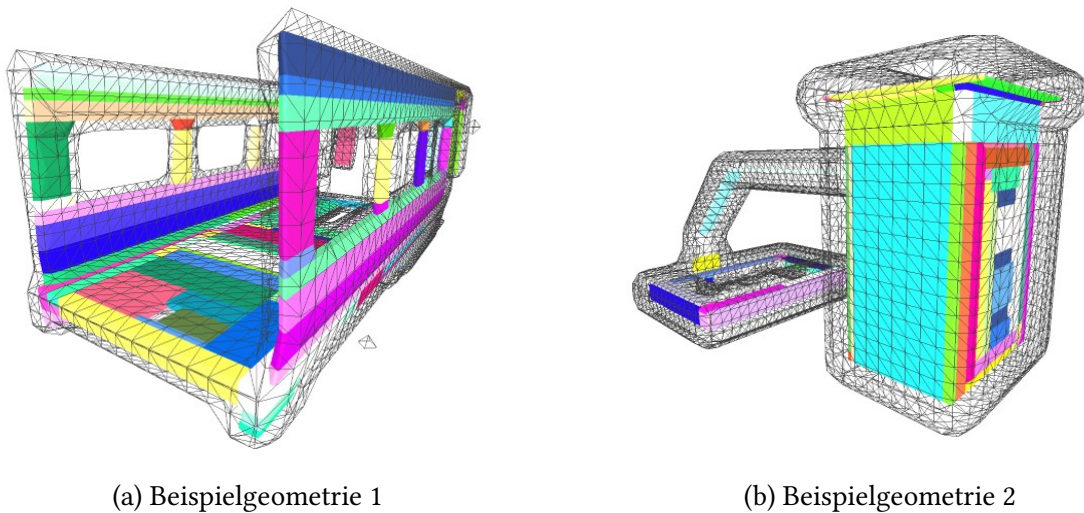


Abbildung 4.7: Domänenspezifische Reduzierung des Zustandsraumes als Mesh-Darstellung in zwei Beispielgeometrien

Da für die Pfadidentifikation im MAPF weniger der Festkörper selbst, sondern vielmehr der zur Verfügung stehende Bewegungsraum entscheidend ist (vgl. Kapitel 2.1.2), folgt der nächste Berechnungsschritt von der Kollisionsgeometrie zur Zustandsraumreduzierung. Auf Basis der erzeugten Kollisionsgeometrie wird eine umliegende Hülle berechnet (vgl. Abbildung 4.7), die alle Agenten $a^i \forall i \in \{1, \dots, N\}$ darin beschränkt, Pfade mit einem zu großen Abstand zum 3D-Objekt zu erzeugen. Die Problemstellung ändert sich damit von einer **hindernisfokussierten** zu einer **bewegungsraumfokussierten** Betrachtungsweise. Für diesen Betrachtungswechsel ist eine effiziente Datenhaltung erforderlich, da innenliegende Strukturen der Geometrie nicht mehr redundant gespeichert werden. Die Reduzierung des Zustandsraumes mit dem Ziel einer Bewegungsraumfokussierung ist vom domänenspezifischen Parameter d_{max} abhängig. Dieser gibt an, welchen maximalen Abstand die elektrischen Verbindungen von der Oberfläche der Geometrie aufweisen dürfen. Dies ist z. B. durch rechtliche Regularien oder durch die physikalischen Eigenschaften der Kabel als biegeschlafe Teile erforderlich (vgl. Kapitel 2.3.3).

Im abschließenden Teilmodell (vgl. Abbildung 4.6 (4)) ist die visuelle Repräsentanz des Routing-Environments (vgl. Abbildung 4.11a) dargestellt. Dieses vereint die unterschiedlichen Teilmodelle (vgl. Abbildung 4.6 (2) und (3)) in sich, beinhaltet aber auch alle datentechnischen Informationen aus der AutomationML-Struktur. Somit existieren in diesem Modell neben der Kollisionsgeometrie die elektrischen Anschlusspunkte mit zugewiesenen Richtungsvektoren (vgl. Kapitel 4.2.2).

4.2.4 Erweiterung der Routing-Umgebung um Kinematikanforderungen

Für die Berücksichtigung und Integration der Kinematik einer 3D-CAD-Baugruppe in die neue Methode ist eine Ergänzung der Eingangsinformationen notwendig (vgl. Kapitel 4.1.2). Dies umfasst neben der Positionierung eines Joints auch dessen Ausprägung und die im 3D-CAD-Modell vorgesehenen Bewegungsrestriktionen (vgl. Kapitel 2.3.4). Aufbauend auf existenten standardisierten Datenformaten können hierfür COLLADA- oder URDF-Dateien den erforderlichen Informationsgehalt in standardisierter Syntax bereitstellen (vgl. Kapitel 2.2.5). Beide Formate beschreiben die Beziehung zwischen einem Parent- und seinem Child-Link, die durch den entsprechenden Joint verbunden sind. Der Parent-Link entspricht dabei dem Bauteil, das strukturell näher an der Basis der Baugruppe liegt. Ist der Joint in seiner Bewegungsfreiheit nicht limitiert, so sind keine weiteren Angaben zu den Limitationen angegeben und er ist ausschließlich über seine Position, den Typ und die Gesamttransformationsmatrix M_G festgelegt (vgl. Gleichung 2.5). Im Falle einer eingeschränkten Bewegung sind hingegen die oberen und unteren Grenzen der betroffenen Achsen zu limitieren. Auf Grundlage dieser Informationen erfolgt die Vorverarbeitung der 3D-MCAD-Geometrie. Dieser Verarbeitungsprozess ist in Abbildung 4.8 visualisiert.

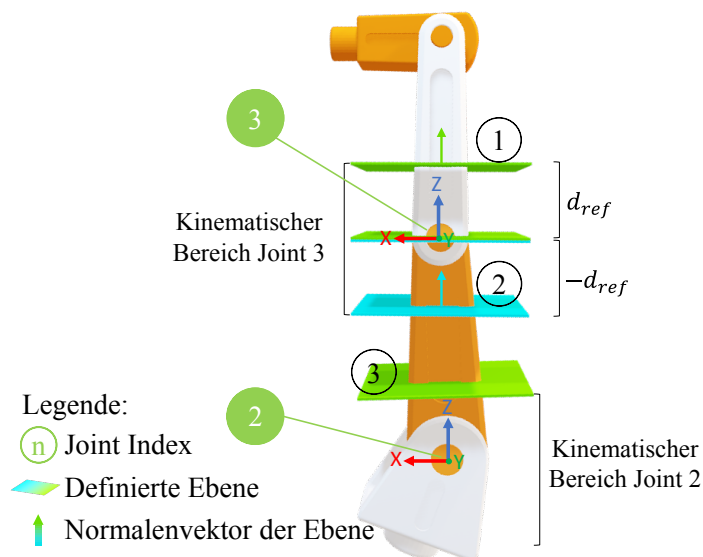


Abbildung 4.8: Prozess zur Festlegung von kinematischen Bereichen in einer 3D-CAD-Geometrie

Ein kartesisches Koordinatensystem (vgl. Abbildung 4.8, Joint (3)) dient als erforderliche Qualifizierung der Joints innerhalb der mechanischen 3D-CAD-Geometrie (vgl. Kapitel 4.1.2). Mit Hilfe dieser Koordinatensysteme ist es möglich, einen umliegenden, kinematischen Bereich zu definieren, in dem die statische Route eines Kabelpfades durch die Beweglichkeit der Baugruppe beeinflusst wird. Im initialen Schritt ist die Baugruppe in einen Zustand zu bringen, in dem alle Joints eine maximale Ausprägung der Rotation aufweisen. Diese ist unabhängig von der zulässigen minimal oder maximal definierten Auslenkung (vgl. Joint (3), Abbildung 4.8). In diesem Fall entspricht der Winkel zwischen den z-Achsen der Einzelteile exakt 0 Grad. Ausgehend von der z-Achse des Koordinatensystems des entsprechenden Joints entsteht mit dem Abstand d_{ref} eine erste Ebene (vgl. Abbildung 4.8 (1)). Der Normalenvektor \vec{n} ergibt sich aus einer Verlängerung des Richtungsvektors entlang der z-Achse des Joint-Koordinatensystems mit einem Abstand von d_{ref} . Die entsprechende Ebene ist mit dem Child-Link des Joints assoziiert und in dessen Koordinatensystem verankert (vgl. Kapitel 4.1.2). Der analoge Prozessschritt erfolgt für die zweite Ebene im Koordinatensystem des Parent-Links (vgl. Abbildung 4.8 (2)). Auch hier orientiert sich die Positionierung der Ebene an der z-Achse des zugehörigen Joint-Koordinatensystems, jedoch diesmal mit einer Verschiebung um $-d_{ref}$. Der Raum zwischen den beiden neu erzeugten Ebenen wird im Nachgang als **kinematischer Bereich** bzw. kinematische Zone deklariert. Weitere Joints und Ebenen innerhalb der 3D-CAD-Geometrie (vgl. Abbildung 4.8 (3)) schränken die Verschiebung der kinematischen Ebenen ein, da eine überlappungsfreie Beschaffenheit der einzelnen kinematischen Zonen zu gewährleisten ist.

Einen wesentlichen Einfluss auf die Validität der Pfadplanung unter Berücksichtigung der Kinematik hat die Positionierung der Links im Startzustand zueinander (vgl. Kapitel 2.3.4). Das 3D-CAD-Modell ist in die Neutralposition aller Joints zu setzen. Abbildung 4.9 zeigt die vorgenommenen Qualifizierungsschritte zur Integration der kinematischen Bereiche in das 3D-Routing-Environment. Darüber hinaus stellt sie die Auswirkungen der Link-Positionierung auf den Kabelpfad gegenüber.

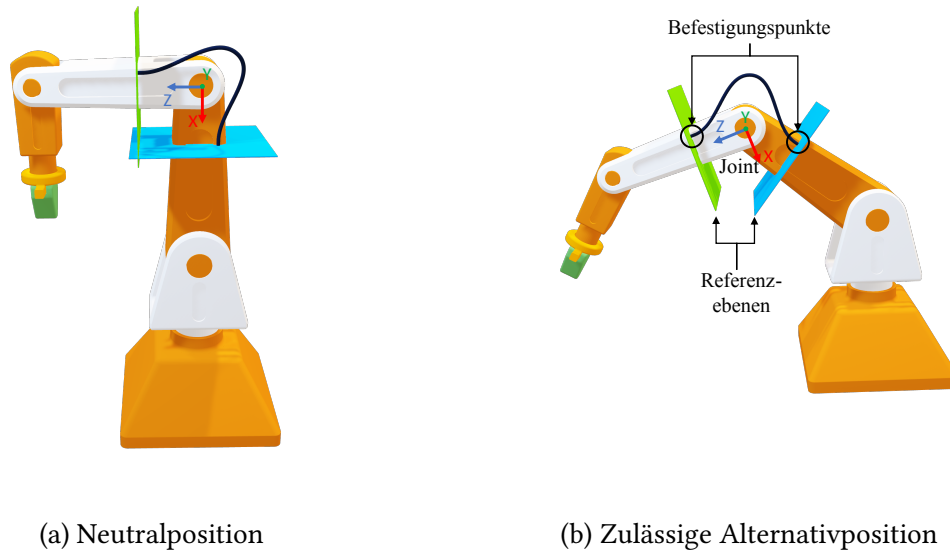


Abbildung 4.9: Integration von kinematischen Einflüssen in Bereichen um exemplarische Joints

Abbildung 4.9a zeigt die Neutralposition des exemplarischen Roboterarms und Abbildung 4.9b eine weitere mögliche Position innerhalb des zulässigen Bewegungsraumes. Anhand des Rotations-Joints ergeben sich die zugehörigen Referenzebenen. Im Schnittpunkt des identifizierten Kabelpfades mit den Ebenen liegt der vorgesehene Befestigungspunkt und somit der Übergang von statischer zu kinematischer Zone. Die Ermittlung der dazwischen liegenden Kabelpfade stellt einen Subprozess der MAPF dar (vgl. Kapitel 2.3.6), der nahtlos in den MARL-Lernprozess (vgl. Kapitel 2.5.1) integriert ist. Sobald ein Agent $a^i, i \in \{1, \dots, N\}$ die zweite Referenzebene schneidet, wird dieser Subprozess angestoßen. Diese Vorgehensweise forciert den Agenten in einer globalen Betrachtungsweise dazu, auch die Schnittpunkte mit den Ebenen so zu optimieren, dass eine minimale Kabellänge in der kinematischen Zone entsteht.

Als Eingangsgrößen für den Subprozess dienen neben der Geometrie und den Schnittpunkten des Kabels mit den Ebenen die Freiheitsgrade hinsichtlich der Joints und der assoziierten Links. Die erforderliche Zusatzlänge des Kabels in der kinematischen Zone entspricht damit dem Reward R^i des Agenten i für die Aktion A_{kin}^i . Die Aktion A_{kin}^i ist entkoppelt vom eigentlichen Aktionsraum A . Stattdessen nimmt der Agent die kinematische Zone als eine einzige Aktion wahr, deren Reward äquivalent zur notwendigen Zusatzlänge durch den kinematischen Einfluss ist.

4.2.5 Approximation und Rückführung von Kabelpfaden in das 3D-CAD-System

Die neuen Methoden, die in den Kapiteln 4.4 und 4.5 detailliert beschrieben werden, zielen darauf ab, feingranulare Wegstreckenpunkte zu erzeugen, die den Kabelpfad im kontinuierlichen dreidimensionalen Raum repräsentieren. Aufgrund der Beschaffenheit von Kabeln als biegeschlaffe Teile (vgl. Kapitel 2.3.3) und deren Darstellung als B-Splines im 3D-MCAD-System (vgl. Kapitel 2.2.1) ist eine entsprechende Transformation der Ergebnisdaten aus der MARL-Methode notwendig. Insbesondere durch die datentechnische Entkopplung der neuen Methoden von den CAD-Systemen besteht die Notwendigkeit einer Rückführung der Resultate der MARL-Methode. Als Zielsystem zur Darstellung der Ergebnisse ist das 3D-MCAD-System vorgesehen. Dort sollen die elektrischen Verbindungen innerhalb der mechanischen 3D-CAD-Baugruppe realitätsgetreu abgebildet werden, um neben der Kabellängenermittlung eine Realisierbarkeitsprüfung der Planung vornehmen zu können (vgl. Kapitel 2.3.5).

Die Lösungsalgorithmen, welche die Solver-Klasse (vgl. Kapitel 4.2.1) konkretisieren, generieren für jede elektrische Verbindung eine Liste an Wegstreckenpunkten. Jeder Punkt dieser Kabelstrecken ist durch eine Koordinate mit Werten in der x-, y- und z-Achse definiert (vgl. Kapitel 4.1.4). Abhängig von der Schrittgranularität im Action-Space A^i kann der euklidische Abstand zwischen den Punkten im Millimeterbereich liegen (vgl. Kapitel 4.3.2). Gängige 3D-MCAD-Systeme sind jedoch nicht in der Lage, eine solche Auflösung bei einer höheren Anzahl von elektrischen Verbindungen effizient darzustellen. Daher werden die Wegstreckenpunkte nicht mit einer B-Spline interpoliert, sondern als solche approximiert (vgl. Kapitel 2.2.1). Der definierte Glättungsfaktor (en: *Smoothing*) s wird im Prozess eines Least-Squares-Fitting verwendet (vgl. Kapitel 2.2.1). Abbildung 4.10 zeigt die drei Ausprägungen exemplarisch an einer elektrischen Verbindung.

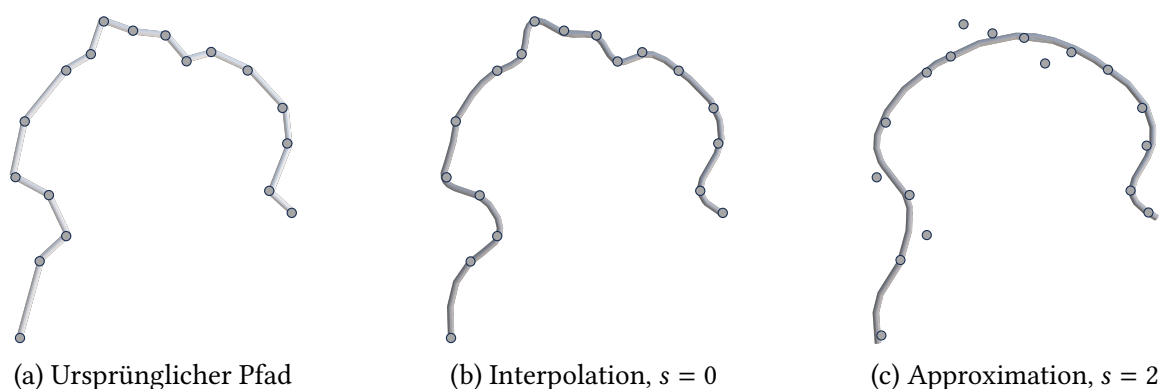


Abbildung 4.10: Schematische Darstellung des Approximationsprozesses für elektrische Verbindungen im 3D-CAD-System

Während der Originalpfad als Ergebnis der MARL-Methoden (vgl. Kapitel 4.4 und 4.5) durch geradlinige Verbindungen zwischen den Wegstreckenpunkten abgebildet wird, werden diese Punkte bei der Interpolation durch eine B-Spline verbunden (vgl. Abbildung 4.10b). Bei der verwendeten Approximation (vgl. Abbildung 4.10c) werden die ursprünglichen Wegstreckenpunkte mit Hilfe des Smoothing-Faktors s angenähert. Dies hat zur

Folge, dass nicht mehr alle mit der MARL-Methode geplanten Punkte auf dem Kabelpfad liegen. Eine entsprechend hohe Punktfrequenz ist jedoch nicht notwendig, da der Kabelpfad bereits über einen Bruchteil der Punkte exakt definiert ist. Aus datentechnischer Sicht stellt diese Liste von 3D-Punkten je Agent $a^i \forall i \in \{1, \dots, N\}$ aus der Approximation den Rückgabewert der REST-Schnittstelle der *DataService*-Klasse dar (vgl. Kapitel 4.2). Als Resultat des Approximationsprozesses und der Übermittlung der Informationen an das 3D-MCAD-System steht am Ende des Prozesses eine generierte Topologie mit kabelspezifischen Durchmessern und Farbdarstellungen für eine manuelle Prüfung, Korrektur oder Anpassung zur Verfügung. Die anschließende Speicherung der Ergebnisse im PDM-System unterliegt der Versionierung des jeweiligen Produktentwicklungsstandes (vgl. Kapitel 2.3.5).

4.3 Virtuelle Verdrahtung als Markov Game

Die Definition des vorliegenden CHRP als MG ist der integrale Bestandteil dieses Kapitels. Zu diesem Zweck wird in Kapitel 4.3.1 die Repräsentation des Environments nach den Erfordernissen des MARL festgelegt (vgl. Kapitel 2.5). Darauf aufbauend zeigt Kapitel 4.3.2, wie das CHRP in einem MG abgebildet werden kann. Dies umfasst neben dem zulässigen Aktionsraum auch die Reward-Funktion zur Erreichung des Optimierungsziels aus Kapitel 4.1.4.

4.3.1 Environment Repräsentation der 3D-CAD-Geometrie

Für die Verwendung eines 3D-CAD-Modells als Environment im Sinne des MG ist im initialen Schritt die Überführung der Geometrie in ein vereinfachtes dreidimensionales Objekt notwendig (vgl. Kapitel 4.2.3). Es gibt dabei drei Betrachtungsweisen, eine visuelle (vgl. Abbildung 4.11a), eine kollisionsorientierte (vgl. Abbildung 4.11b) und eine Mesh-basierte (vgl. Abbildung 4.11c).

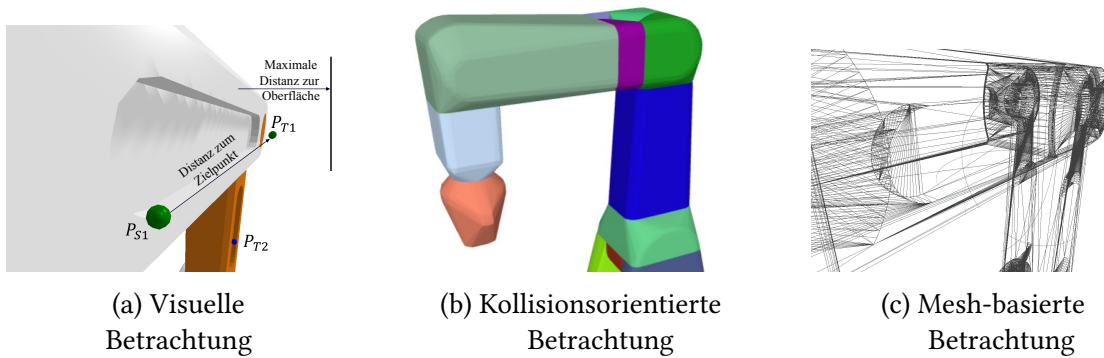


Abbildung 4.11: Darstellung eines exemplarischen Environments in drei unterschiedlichen Betrachtungsweisen

Die visuelle Betrachtung (vgl. Abbildung 4.11a) zeigt eine exemplarische Start-Ziel-Kombination in einer 3D-CAD-Geometrie. Dabei ist neben der Strecke zur Zielposition P_{T1} auch die zulässige Maximaldistanz zur Oberfläche d_{max} des Objektes dargestellt (vgl. Kapitel 4.2.3). Die Unterscheidung nach P_S und P_T je Agent dient dem effizienteren Lernprozess und der verbesserten Transferierbarkeit der antrainierten Policy π (vgl. Kapitel 4.5).

Inbesondere die Beschaffenheit als Festkörper erfordert die Kollisionsfreiheit des zu planenden Kabelpfades (vgl. Abbildung 4.11b und Kapitel 2.3.3). Damit verbunden ist eine stetige Prüfung von Schnittpunkten zwischen den Kabelpfaden und dem 3D-Objekt selbst (vgl. Kapitel 2.2.3). Diese Kontrolle stellt sicher, dass die Gerade zwischen den Punkten P_{t-1}^i und P_t^i keine Fläche des Objekts durchdringt. Für eine effiziente Berechnung kommen dafür konvexe Kollisionsobjekte unter Verwendung von CoACD zum Einsatz (vgl. Kapitel 2.2.3 und 3.2.3), die den Einsatz von Distanzalgorithmien wie dem Gilbert-Johnson-Keerthi-Distanz-Algorithmus erforderlich machen. Damit ist es möglich, komplexere Formen zu verwenden, ohne eine Reduzierung der Recheneffizienz oder erhöhte Speicheraufwände in Kauf zu nehmen (vgl. Kapitel 2.2.3).

Eine Mesh-basierte Betrachtung stellt die Edges und Vertices der tesselierten Geometrie in den Vordergrund (vgl. Abbildung 4.11c). Sie sind insbesondere für die Erzeugung des Navigationsgraphs (en: *Navigation Graph*) als Bestandteil der Kabelbaumerstellung erforderlich (vgl. Kapitel 4.5.1). Da innerhalb eines Graphen die Pfadsuchalgorithmen effizient angewendet werden können (vgl. Kapitel 2.1), bietet dies eine zielführende Datenbasis für die folgenden Berechnungsschritte. Entgegen der Betrachtung in Abbildung 4.11b steht nicht die Kollision mit dem Objekt im Fokus von Abbildung 4.11c, sondern der kürzeste Pfad zwischen zwei Knoten im Graphen (vgl. Kapitel 2.1).

Die Kombination der drei Darstellungsformen stellt die Basis für die Anwendung einer MARL-Methode in einem MG dar. Die Methode wird um die domänenspezifische Reduzierung des Zustandsraumes (vgl. Kapitel 4.2.3) und die kinematischen Informationen (vgl. Kapitel 4.2.4) ergänzt.

4.3.2 Repräsentanz als Markov Game

Das CHRP wird für die neue Methode als POMG angesehen, das durch das Tupel $\langle N, O^{1...N}, S, A^{1...N}, \mathcal{T}, \mathcal{R}, \gamma \rangle$ beschrieben ist (vgl. Kapitel 2.5.2). Die einzelnen Bestandteile sind in den folgenden Punkten 1 bis 7 festgelegt:

1. **Anzahl an Agenten N :** Jede der elektrischen Verbindungen ist innerhalb des Routing-Environments aus Abschnitt 4.3.1 durch einen eigenen Agenten repräsentiert. Die Anzahl der Agenten variiert in Abhängigkeit von den zu planenden elektrischen Verbindungen aus der Verbindungsliste (vgl. Kapitel 2.3.2). Im Rahmen der Anwendung einer Methode zur Kabelbaumerzeugung ist es möglich, dass ein einzelner Agent für die Repräsentation mehrerer elektrischer Verbindungen zuständig ist.
2. **Zustandsraum S :** Der Zustandsraum S des Environments ist ein kontinuierlicher dreidimensionaler Raum, der durch die Bounding-Box des 3D-CAD-Modells beschränkt ist (vgl. Abschnitt 4.3.1). Der Bereich, in dem sich das zu verdrahtende Objekt befindet, ist nicht zugänglich. Er kann von einem Agenten aufgrund von Kollisionen mit den Objektoberflächen (vgl. Kapitel 2.2.3) nicht erreicht werden. Der zulässige Bewegungsraum ist darüber hinaus durch die maximale Distanz zur Oberfläche d_{max} (vgl. Kapitel 4.2.3) eingeschränkt. Jeder Agent nimmt folglich in einem Zeitschritt t eine spezifische Position innerhalb des kontinuierlichen Raumes zwischen der Oberfläche des 3D-CAD-Modells und der Bewegungsraumbegrenzung ein. Diese ist mit einem Edge des SPP gleichzusetzen (vgl. Kapitel 2.1). Aufgrund der Beschaffenheit des MG als POMG ist es einem einzelnen Agenten nicht möglich, den ganzheitlichen State des Environments zu erfassen.
3. **Observation $O^{1...N}$:** Der Joint-Observation-Space als Teilmenge des tatsächlichen State-Spaces $O^{1...N} \subset S$ stellt die für den Agenten a^i wahrnehmbaren Informationen dar. Er enthält neben der Position des Agenten P_t^i im aktuellen Zeitschritt t den Richtungsvektor \vec{P}_T^i zu seinem Zielpunkt P_T^i , die derzeit minimale Distanz zur Oberfläche sowie zum nächstgelegenen anderen Agenten. Damit ist es dem Agenten möglich,

eine relative räumliche Positionierung zu identifizieren, ohne mit anderen Objekten des MG zu kollidieren.

4. **Gemeinsamer Aktionsraum** $A^{1...N}$: Jeder Agent innerhalb des MG hat die Möglichkeit, sich aus seiner aktuellen Position heraus in eine der umliegenden Positionen (in allen drei Achsen) zu bewegen. Der agentenindividuelle Aktionsraum wird mit einem dreidimensionalen Vektor im Wertebereich von -1 bis 1 für alle drei Achsen definiert. Die Aktion A^i wird mit der Position des Agenten zum Zeitpunkt t verrechnet, um die Position des Agenten zum nachfolgenden Zeitschritt $t + 1$ zu erhalten. Formal ist der Aktionsraum A^i durch die Gleichung 4.8 beschrieben. Abbildung 4.12 veranschaulicht den Aktionsraum eines Agenten visuell.

$$A = \{A^1, A^2, \dots, A^n\} \quad (4.7)$$

$$A^i = \{[a_x^i, a_y^i, a_z^i] | a_x^i, a_y^i, a_z^i \in [-1, 1]\} \quad (4.8)$$

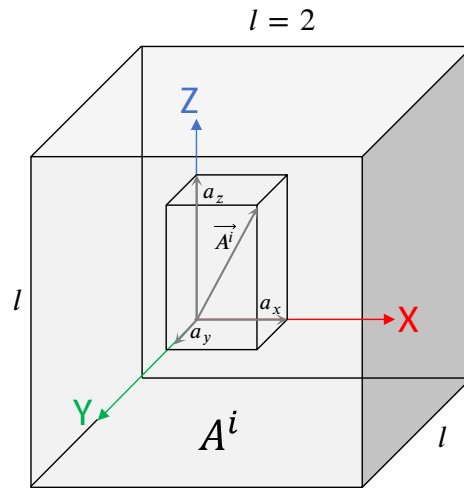


Abbildung 4.12: Kontinuierlicher Aktionsraum im 3D-Environment

Der Aktionsraum A^i ist für jeden Agenten individuell und repräsentiert ein Set an konkreten Ausprägungen des Richtungsvektors $\vec{A}^i = [\vec{a}_x^i, \vec{a}_y^i, \vec{a}_z^i]$ im dreidimensionalen Raum (vgl. Abbildung 4.12). Dieser ist für alle Agenten $a^i \forall i \in \{1, \dots, N\}$ auf einen Wertebereich von $[-1, 1]$ je Achse limitiert (vgl. Gleichung 4.8). Konträr zu existierenden Methoden der Pfadoptimierung im dreidimensionalen Raum (vgl. Kapitel 3.2.1 und 3.2.2) sind alle Freiheitsgrade gegeben, um den Agenten einen größeren Aktionsraum zu schaffen (vgl. Kapitel 2.1.1 und 3.2.2). Der DAF entspricht damit dem Wert 1 (vgl. Kapitel 2.1.2).

5. **Transition Function** \mathcal{T} : Die Transition Function ist für einen einzelnen Agenten deterministisch und durch die Gleichung 4.9 festgelegt. In diesem Zusammenhang steht

der Wert von $E = 0$ für einen freien Raum, während $E = 1$ den bereits okkupierten Raum deklariert.

$$T(s_t, a_t, s_{t+1}) = P(s_{t+1}|s_t, a_t) = \begin{cases} 1, E = 0 \\ 0, E = 1 \end{cases} \quad (4.9)$$

Nach Gleichung 4.9 ist die Wahrscheinlichkeit (en: *Probability*) 1, wenn die Aktion eines Agenten darin besteht, sich in eine Position zu bewegen, die entweder ein Teil des freien Bewegungsraumes $E = 0$ oder seine Zielposition P_T ist. Andernfalls ist die Wahrscheinlichkeit gleich null. Bei einer gemeinsamen Aktion können mehrere Agenten versuchen, auf dieselbe Position zuzugreifen, oder sich überschneidende Richtungsvektoren aufweisen. Sie repräsentieren dabei einen Edge- bzw. Vertex-Konflikt (vgl. Kapitel 2.3). In diesem Fall findet keine Transition statt. Die Transition Function (siehe Gleichung 4.10) ist folglich nicht deterministisch (vgl. Kapitel 2.5).

$$T(s_t, a_t^1, a_t^2, \dots, a_t^n, s_{t+1}) = P(s_{t+1}|s_t, a_t^1, a_t^2, \dots, a_t^n) = \begin{cases} 1, E(a_t^1, a_t^2, \dots, a_t^n) < 2 \\ 0, E(a_t^1, a_t^2, \dots, a_t^n) \geq 2 \end{cases} \quad (4.10)$$

In der Gleichung 4.10 steht $E(a_t^1, a_t^2, \dots, a_t^n) \geq 2$ dafür, dass mindestens zwei Agenten im Zeitschritt t mit ihren assoziierten Aktionen a_t^i versuchen, die gleiche Position in der Umgebung zu beanspruchen. Hieraus lässt sich schließen, dass es in der *Joint Action* A zu einem Konflikt kommt. Eine gleichzeitige Beanspruchung desselben Ortes durch mehrere Agenten führt zu einem Edge-Konflikt (vgl. Kapitel 2.1.3).

6. **Reward-Funktion \mathcal{R} :** Zur Erreichung der Optimierungsziele (vgl. Kapitel 4.1.4) werden die Agenten unter Verwendung eines Rewards R trainiert. Diesen erhält jeder Agent a^i für den Übergang vom Zeitschritt $t - 1$ zu t . Der Reward setzt sich je nach Transition aus mehreren Bestandteilen zusammen, die ein bestimmtes Verhalten fördern oder unterbinden sollen (vgl. Kapitel 2.5.1). Die Errechnung des Rewards für einen individuellen Agenten a^i erfolgt wie in Tabelle 4.1 dargestellt.

Tabelle 4.1: Einflussfaktoren zur Ermittlung des Rewards

Aktion	Bedingung	Reward
Zielvektor	$\vec{A}^i \cdot \vec{P}_T^i$	$\vec{A}^i \cdot \vec{P}_T^i \in [-1, 1]$
Agent Collision	$E(P_t) = 1$	-1.0
Finish	$P_t = P_T$	+100.0

Für jede Aktion A^i , die ein Agent a^i ausführt, erfolgt ein Abgleich zwischen dem gewählten Richtungsvektor der Aktion \vec{A}^i und dem Richtungsvektor zur Zielposition \vec{P}_T^i (vgl. Tabelle 4.1). Dies dient neben der Minimierung der erforderlichen Gesamtdistanz auch der Nutzung des direktesten möglichen Weges innerhalb des

Environments. Gleichzeitig ist dieser Reward Bestandteil für eine Anpassung zur Skalierung der neuen Methode in Kapitel 4.5.1. Der assoziierte Reward ist positiv oder negativ, je nachdem, wie ähnlich die beiden Vektoren zueinander sind. Unter der Verwendung des Skalarprodukts ergibt sich ein Wert zwischen -1 und 1 , der dem jeweiligen Agenten zugeordnet wird. Im Falle einer Kollision mit einem Hindernis, dem Startpunkt eines anderen Agenten oder einem anderen Agenten wird der betroffene Agent mit dem Wert -1 bestraft. Während der Annäherung an die Zielposition P_T erreicht der Agent die nähere Umgebung, wenn er einen Bereich um den Zielpunkt betritt. Sinkt der Abstand zu P_T unter die Konstante $d(P_T)$, wird dem Agenten die Zielposition zugewiesen. Dieser Schritt ist notwendig, da aufgrund des kontinuierlichen Raums kein exaktes Erreichen von P_T zu erwarten ist. Damit hat der Agent a^i die entsprechende Zielposition P_T erreicht und wird mit $+100$ belohnt. Da mehrere Szenarien aus Tab. 4.1 parallel auftreten können, werden die Belohnungen der einzelnen Aktionen addiert. Sie bilden damit den numerischen Reward \mathcal{R}^i des Agenten a^i ab.

7. **Discount-Faktor γ :** Der Diskontierungsfaktor (en: *Discount Factor*) γ ist als Hyperparameter zu betrachten, der in Abhängigkeit vom konkreten Environment zu identifizieren ist. Ein höheres Gamma nahe 1 drückt aus, dass die Agenten langfristige Belohnungen in Betracht ziehen und langfristige Strategien entwickeln. Ein niedrigeres Gamma nahe 0 bedeutet, dass die Agenten kurzfristige Belohnungen bevorzugen und kurzfristige Strategien verfolgen. Häufig dient das Testen einer diskreten Menge möglicher Werte für γ zur Identifikation eines nahezu optimalen Wertes.

4.4 Automatisierte virtuelle Verdrahtung mittels kommunizierenden Multi-Agent Reinforcement Learning

In einem kooperativ agierenden Ansatz werden in diesem Kapitel bestehende Kommunikationsmethoden aus Abschnitt 2.5.3 angewendet, um das CHRP effizient zu lösen. Dafür beschreibt Abschnitt 4.4.1, wie die Einzelverbindungen für eine Reduzierung der Problemkomplexität (en: *Problem-Decomposition*) gruppiert werden können. Hier werden bestehende Lösungsansätze aus Kapitel 2.4.4 angewendet. Abschnitt 4.4.2 baut darauf einen untergliederten kommunizierenden Twin Delayed DDPG (TD3) auf, der die entstandenen Cluster einbezieht. Diese Methode überführt ein Konzept aus dem diskreten in den kontinuierlichen Raum (vgl. Kapitel 3.4 und Tabelle A.1). Abschnitt 4.4.3 erläutert in diesem Kontext die Informationsübertragung über die Observation der jeweiligen Agenten. Der letzte Abschnitt 4.4.4 zeigt das Anwendungsgebiet der entwickelten Methode auf.

4.4.1 Clustering der Einzelverbindungen zur Komplexitätsreduzierung

Um die verschiedenen Kabel in einem bestimmten Environment effizient bündeln zu können, ist es von Bedeutung, vorab eine ungefähre Vorstellung vom Verlauf der einzelnen Kabel zu erhalten. Dies kann erreicht werden, indem die elektrischen Verbindungen im Voraus grob verlegt werden. Der Verlauf der Kabel muss nicht optimal sein, aber einen hohen Grad an Exaktheit aufweisen. Eine vorherige manuelle Durchführung dieses Schrittes ist aufgrund des zeitlichen Aufwands nicht zu rechtfertigen (vgl. Kapitel 1.2 und 2.3.5). Die neue Methode nutzt dafür die Erzeugung von Beispielpfaden über heuristische Methoden (vgl. Kapitel 4.4.1.1). Über eine definierte Distanzmatrix (vgl. Kapitel 4.4.1.2) können die Einzelverbindungen Clustern zugewiesen werden (vgl. Kapitel 4.4.1.3)

4.4.1.1 Generierung von Beispielpfaden für die Grobpfadplanung

Wie in Kapitel 3.2 beschrieben, sind Pfadfindungsalgorithmen ein probates Mittel, um die Anforderung der Grobplanung zu erfüllen. Da für eine Bündelung weder alle Randbedingungen (vgl. Kapitel 2.3.3) noch die Optimalität des Ergebnisses im Sinne des gesetzten Optimierungsziels (vgl. Kapitel 4.1.4) zwingend erforderlich sind, ist eine Bewertung hinsichtlich der zeitlichen Effizienz und des Integrationsaufwands zulässig. Von einem heuristischen Pfadplanungsalgorithmus wird erwartet, dass mit limitiertem Zeitaufwand eine angemessene Ergebnisgenauigkeit erreicht werden kann (vgl. Kapitel 3.2.1). Dies ergibt sich aus der Analyse der bisherigen Einsätze dieser Methoden in bestehenden iterativen Ansätzen zur Pfadoptimierung (vgl. Kapitel 3.4). Für eine Verwendung sind die erforderlichen Eingangsgrößen durch die Überführung der Inputdaten für den Verdrahtungsprozess (vgl. Kapitel 4.2) bereits gegeben. So finden sich die notwendigen Start- und Zielkoordinaten (P_S^i, P_T^i) aller Kabel bzw. Agenten und der zulässige Bewegungsraum in der Routing-Umgebung wieder (vgl. Kapitel 4.2.3). Die Ausgabe des Algorithmus ist ein durch dreidimensionale Wegpunkte definierter Pfad für jeden eigenständigen Agenten ohne die Betrachtung von Wechselwirkungen mit anderen Agenten im Szenario $N \geq 2$. Die Erzeugung der Beispielpfade erfolgt in diesem Schritt iterativ pro Kabel anhand der Verbindungsliste (vgl. Kapitel 2.3.2).

4.4.1.2 Festlegung der Distanzmatrix für die Grobpfadplanung

Die räumliche Trennung der einzelnen Kabelpfade hat einen signifikanten Einfluss auf die potenziell auftretenden Wechselwirkungen zwischen den Einzelverbindungen (vgl. Kapitel 2.3.3). Daher ist es notwendig, auf Basis der Pfadverläufe im dreidimensionalen Raum Cluster zu erstellen (vgl. Kapitel 2.4.4). Dies dient der Reduzierung der Problemkomplexität auf Grundlage der gegenseitigen Einflussnahme der Trajektorien τ^i und τ^{i+1} . Die Gruppierung der Kabel in verschiedene Cluster erfolgt auf Grundlage der erstellten Beispielpfade (vgl. Kapitel 4.4.1.1). In ähnlichen geometrischen Bereichen verlaufende elektrische Verbindungen sollen dabei in einem Cluster zusammengefasst werden. Um diesen Prozess durchführen zu können, ist die Auswahl einer geeigneten Distanzmetrik und eines geeigneten Clustering-Algorithmus erforderlich (vgl. Kapitel 2.4.4.1 und 2.4.4.2).

Die generierten Pfade werden in Form von Listen mit den Koordinaten aller Wegstreckpunkte pro Kabel dokumentiert und die Pfade haben unterschiedliche Längen. Um mit dieser Art von Daten weiterarbeiten zu können, sind nur Distanzmetriken relevant, die in der Lage sind, Kurven zu verarbeiten. Kapitel 2.4.4.2 führt drei mögliche Distanzmetriken für Kurven ein: Hausdorff, Fréchet und eine flächenbezogene Metrik. Nach Abwägung der verschiedenen Möglichkeiten, den Abstand zwischen den Pfaden zu messen, wird das Abstandsmaß nach Fréchet gewählt. Diese Entscheidung basiert auf mehreren Faktoren. Die Fréchet-Distanz berücksichtigt im Gegensatz zur Hausdorff-Distanz oder zur Flächendistanz die Form der Pfade (vgl. Kapitel 2.4.4). Diese Eigenschaft ist beim Vergleich unterschiedlich langer Pfade mit ähnlicher Form besonders effektiv. Außerdem ist die Fréchet-Distanz robust gegenüber Rauschen oder kleinen Störungen in den Pfaden, was bei der Erzeugung von Beispielpfaden ohne Optimierung auftreten kann (vgl. Kapitel 3.2.1).

4.4.1.3 Zuweisung der einzelnen Beispielpfade in Cluster

Die Auswahl eines für diesen Anwendungsfall geeigneten Clustering-Algorithmus fundiert auf den aufgezeigten Arten von Clustering-Methoden (vgl. Kapitel 2.4.4.3). Da die Anzahl der Cluster vorab nicht feststeht, bieten dichtebasierte Clustering-Methoden diverse Vorteile. Dies ist insbesondere im Zusammenhang mit der virtuellen Verdrahtung nützlich, in der die Bestimmung der Clusteranzahl (vgl. Kapitel 2.4.4.5) in komplexen Geometrien allein auf Basis der Verbindungsliste eine Herausforderung darstellt (vgl. Kapitel 2.3.2). Eine hohe Anzahl elektrischer Verbindungen verschärft diese Problematik zunehmend (vgl. Kapitel 1.2).

Dichtebasierte Algorithmen erlauben die Bildung von Clustern unterschiedlicher Größe, was für die korrekte Identifizierung von Kabelgruppen und -bündeln in verschiedenen Bereichen der Geometrie unerlässlich ist (vgl. Kapitel 2.4.4.3). Wenn beispielsweise nur wenige Kabel in einem Teilbereich der Geometrie verlaufen, sollten sie zusammen gruppiert und nicht einem anderen Cluster hinzugefügt werden. Darüber hinaus ist bei der Auswahl des geeigneten Clustering-Algorithmus zu berücksichtigen, dass die Dichte einzelner Cluster variieren kann. Exemplarisch lassen sich Kabelkanäle mit höherer Dichte nennen. Zur Erkennung von Clustern unterschiedlicher Dichte ist der OPTICS-Algorithmus eine geeignete Lösung. Ein Vorteil dieser Methode besteht darin, dass er nicht zwingend die

Auswahl eines geeigneten ϵ -Wertes wie DBSCAN erfordert. Zwar gibt es Methoden zur Bestimmung des optimalen ϵ -Wertes, wie die Erstellung eines k-Distanzgraphen (vgl. Kapitel 2.4.4.3), diese sind jedoch fehleranfällig. OPTICS nutzt einen optionalen Maximalwert von ϵ , um die Rechenzeit zu reduzieren. Lediglich die minimale Anzahl an elektrischen Verbindungen ist vorzudefinieren. Dieser Parameter definiert die Mindestanzahl an Kabeln für die Bildung eines eigenständigen Clusters. Im Kontext der verwendeten Kommunikationsmethode (vgl. Kapitel 2.5.3) und der Erweiterung der Observation (vgl. Kapitel 4.4.3) ergibt sich eine Mindestanzahl von $N \geq 3$ Agenten je Cluster C_n . Eine Obergrenze ist nicht festgelegt. Als Ergebnis gibt der OPTICS-Algorithmus auf Grundlage der Distanzmatrix ein Label pro Kabel aus. Dieses gibt Auskunft über die Zuordnung zu einem jeweiligen Cluster C_n .

Bei der Verwendung von dichtebasierten Algorithmen wie OPTICS besteht die Möglichkeit, dass einige Kabel keinem Cluster zugeordnet sind. Stattdessen sind sie als Ausreißer (en: *Outlier*) deklariert. Diese Kabel sind elektrische Verbindungen, die keine ausreichende Dichte aufweisen, um als Teil eines Clusters zu gelten (vgl. Kapitel 2.4.4.3). Sie unterscheiden sich in Bezug auf Position und Verlauf signifikant von den anderen Beispielpfaden. Eine Zuordnung zu einem der anderen Cluster C_n würde das Ergebnis der Clusterbildung verfälschen. Aufgrund ihrer hohen Dissimilarität entfallen sie für die Betrachtung im Rahmen der Kommunikationsmethode. Aufgrund der ausbleibenden Zuordnung zu einem Cluster besteht die Prämisse, dass diese Kabel unter keinem wechselseitigen Einfluss mit anderen elektrischen Verbindungen stehen. Ein heuristisch generierter Beispielpfad wird daher als optimal angenommen und über die REST-Schnittstelle der Erzeugung von Kabelpfaden im 3D-CAD-System übergeben (vgl. Kapitel 4.2.1 und 4.2.5). Abbildung 4.13 zeigt die räumliche Trennung der Kabelgruppierungen anhand von Beispielpfaden. Die Überschneidung einzelner Pfade (vgl. Abbildung 2.8b in Kapitel 2.3.3) ist auf die iterative Erzeugung zurückzuführen und wird sich durch die Verwendung der neuen Methode auflösen (vgl. Kapitel 4.4.2).

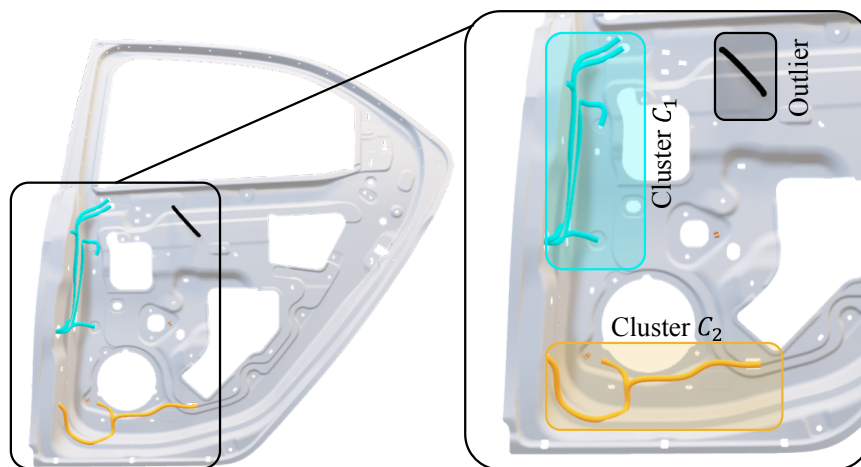


Abbildung 4.13: Clustering von Kabelpfaden anhand der räumlichen Similarität in einer Beispielgeometrie

Die in Abbildung 4.13 blau eingefärbten Pfade sammeln sich im Cluster C_1 , die gelben Kabel gehören dem Cluster C_2 an. Der dunkelgraue Pfad ist aufgrund der Überschreitung des Schwellenwertes der Dichte ϵ als Outlier klassifiziert (vgl. Kapitel 2.4.4) und gehört somit keinem Cluster an. Hierfür wird im Falle von Outliern festgelegt, dass diese elektrische Verbindung keine oder sehr geringe Wechselwirkung zu anderen Kabeln aufweist und daher die Optimalität der Kabelverlegung bereits durch die initial eingesetzte Heuristik gegeben ist (vgl. Kapitel 3.2.1).

4.4.2 Funktionsweise des Communicating Multi-Agent Twin Delayed DDPG

Multi-Agenten-Systeme sind dadurch gekennzeichnet, dass Agenten von einem Austausch ihrer Erfahrungen durch eine gemeinsame Policy π profitieren (vgl. Kapitel 2.5.2). Allerdings ist nicht jede Handlung eines Agenten für alle anderen Agenten relevant, sondern kann sich sogar negativ auf sie auswirken (vgl. Kapitel 2.5.4). Insbesondere wenn die Aktion eines Agenten nur eine Teilmenge aller Agenten betrifft, ist es vorteilhaft, die Policy π aufzuteilen und den Reward \mathcal{R} der einzelnen Agentengruppierungen zu entkoppeln (vgl. Kapitel 2.5.2). Im Anwendungsfall des MAPF ist es nicht sinnvoll, für geometrisch weit entfernte Agenten eine gemeinsame Policy π zu lernen (vgl. Kapitel 2.1.2 und 3.2). Sie agieren in unterschiedlichen Bereichen der Geometrie und beeinflussen sich in ihrer Pfadoptimierung nicht gegenseitig. Zur Umsetzung dieses Konzeptes werden die zuvor erstellten Kabelcluster aufgegriffen, um festzulegen, welche Agenten eine gemeinsame Policy π_{C_n} eines Clusters nutzen. Gemäß der Trainings- und Ausführungsschemata (vgl. Abbildung 2.22 in Kapitel 2.5.2) handelt es sich bei der neuen MARL-Kommunikationsmethode um einen hybriden Ansatz. Innerhalb eines jeden Clusters C_n existiert ein CTCE, während zwischen den Clustern ein DTDE besteht. Abbildung 4.14 zeigt die neue Struktur der Methode auf Grundlage der Gruppierungen.

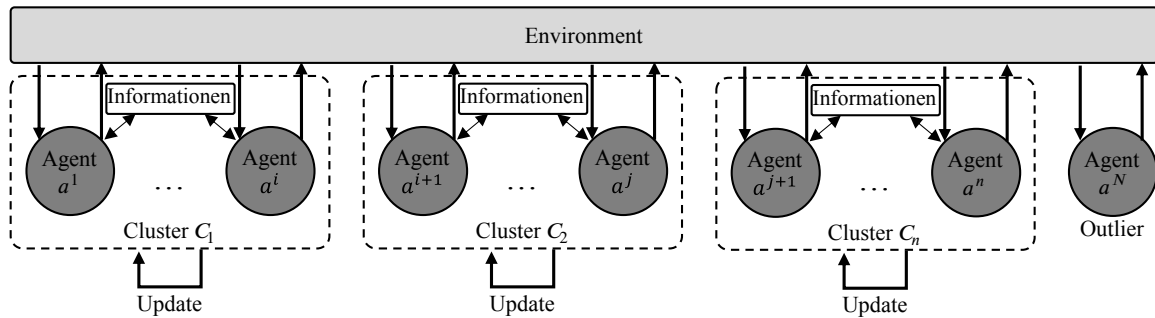


Abbildung 4.14: Schaubild eines hybriden Trainingsschemas mit clusterbasiertem CTCE

In Abbildung 4.14 gehören alle Agenten $a^i \forall i \in \{1, \dots, N\} \geq 3$ einem bestimmten Cluster an. Jedes der Cluster folgt einer individuellen Policy π_{C_n} , die als gemeinsamer zentraler Executor für die Agenten innerhalb des Clusters wirkt. Während des Trainings speisen alle Agenten-Observations O^i gemeinsam das zugrunde liegende neuronale Netz (vgl. Kapitel 2.4.2). In der Ausführungsphase entscheidet hingegen das Netzwerk für jede individuelle Observation $O^i \in \mathcal{O}$, welche Aktion $A^i \in \mathcal{A}$ durchzuführen ist. Innerhalb beider Phasen findet die Übertragung von Informationen zwischen den Agenten eines Clusters statt (vgl.

4.4.3). Folglich besteht für jedes Cluster eine individuelle Policy π_{C_n} , jedoch agieren alle Agenten clusterübergreifend in einem Environment. Das stellt sicher, dass auch geringe Wechselwirkungen, die nicht zur Zuordnung eines Kabels zu einem Cluster geführt haben, dennoch in der Pfadoptimierung berücksichtigt werden. Ein Agent ist in diesem Fall exemplarisch als Outlier klassifiziert (vgl. Abbildung 4.13). Hier besteht eine direkte Interaktion mit dem Environment auf Grundlage des Beispielpfades. Begründet durch die Nutzung des heuristischen Lösungsalgorithmus (vgl. Kapitel 3.2) ist an dieser Stelle keine Policy π im Einsatz. Wird im Laufe des Produktentstehungsprozesses eine neue elektrische Verbindung hinzugefügt, so erfolgt eine Zuordnung zu einem der bestehenden Cluster. Die Verwendung einer bestehenden Policy π_{C_n} stellt sicher, dass das neue Kabel keine neue Policy π erlernen muss, sondern die bestehende Policy des entsprechenden Clusters verwenden kann. Dies ermöglicht eine einfache und schnelle Erweiterung und reduziert den Neuplanungsaufwand. Durch die damit verbundene Analyse der Wechselwirkungen der Kabel zueinander ist lediglich dieses Cluster neu zu planen. Kann die elektrische Verbindung keinem der bestehenden Cluster zugeordnet werden, ist sie als Outlier zu klassifizieren und unterliegt der analogen Vorgehensweise. Der Beispielpfad für die Eingruppierung in ein Cluster entspricht bereits dem optimalen Pfad.

4.4.3 Erweiterung der Observation zur direkten Informationsübertragung der Agenten

Bei der Durchführung des Trainings beobachtet jeder Agent in jedem Schritt seine Umgebung und trifft eine Entscheidung auf der Grundlage des aktuellen Zustands (vgl. Kapitel 2.5 und 4.3.2). Die Beobachtungen O^i der einzelnen Agenten $a^i \forall i \in \{1, \dots, N\}$ können jedoch je nach Umgebung und Aufgabe variieren. Um eine effektive Zusammenarbeit zwischen den Agenten zu ermöglichen, wird festgelegt, dass jeder Agent Informationen über die beabsichtigte Folgeaktion der anderen nächstgelegenen Agenten erhält. Dies hilft ihnen, ihre Aktionen zu koordinieren und fundiertere Entscheidungen in der kooperativen Verhaltensweise zu treffen (vgl. Kapitel 2.5.2). Zum Erreichen eines optimierungszielorientierten Verhaltens (vgl. Kapitel 4.1.4) der Agenten kommen die Kommunikationsmethoden im MARL aus Kapitel 2.5.3 zur Anwendung. Die Prämisse für das Clustering der elektrischen Verbindungen ist hierbei, dass sich nicht alle Agenten gegenseitig beeinflussen. Eine Verwendung des *Broadcasting*-Verfahrens (vgl. Kapitel 2.5.3) ist daher nicht zielführend. Neben der Tatsache, dass der Observation-Space (vgl. Kapitel 2.5) mit zunehmender Anzahl von Agenten stetig wächst, ist durch die Entkopplung der Agenten in Clustern keine Übertragung der Information an alle partizipierenden Agenten notwendig. Um eine einheitliche Beobachtungsgröße O unabhängig von der Anzahl der Agenten im MARL zu erreichen, wird die Kommunikationsmethode der *Target*-Kommunikation verwendet (vgl. Kapitel 2.5.3). Innerhalb des angewendeten Trainingsschemas eines Clusters tauschen die Agenten vordefinierte Informationen zu ihrer Position und ihren Richtungsvektoren aus. Dies dient der abgestimmten Koordination der Aktionen innerhalb des begrenzten Bewegungsraumes (vgl. Kapitel 4.2.3). Das kooperativ zu erreichende Optimierungsziel (vgl. Kapitel 4.1.4 und 2.1.2) der minimalen Gesamtkabellänge soll dadurch unterstützt werden. Dieses Vorgehen wirkt insbesondere der bisher iterativ durchgeführten Verlegung von

elektrischen Verbindungen entgegen und koordiniert die Pfadoptimierung des Kollektivs im MAPF. Im Zusammenhang mit der definierten Mindestgröße eines Clusters von $N \geq 3$ Agenten (vgl. Kapitel 4.4.1.3) ist eine dynamische Kommunikation mit den geometrisch am nächsten liegenden Agenten innerhalb eines Clusters festgelegt. In Abbildung 4.15 ist beispielhaft die Kommunikationseinheit des Agenten a_i dargestellt. Die ausgetauschten Informationen umfassen die Position, den Richtungsvektor zwischen den Agenten und den vorhergesagten Richtungsvektor der anderen Agenten a^{i+1} und a^{i+2} . Agent a^{i+3} ist hier nicht Teil der Kommunikationseinheit.

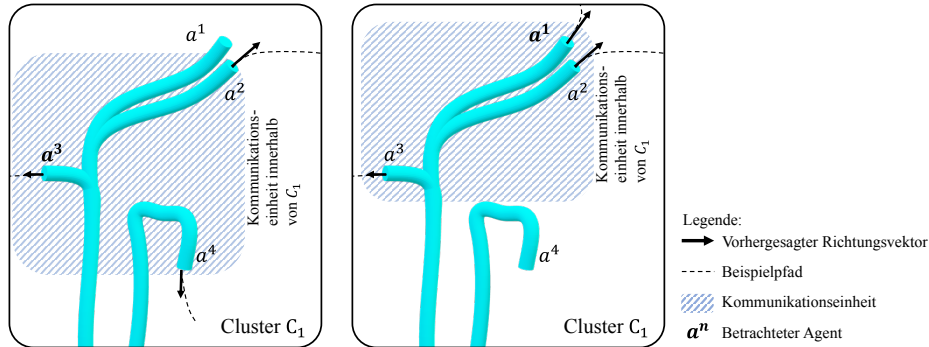


Abbildung 4.15: Darstellung der Kommunikationseinheit innerhalb des Clusters C_1 aus Perspektive von Agent a^3 bzw. a^1

Die erste Erweiterung der Observation (vgl. Abbildung 4.15) besteht darin, die Positionen einer bestimmten Anzahl von Agenten zur Observation hinzuzufügen. Es ist zu beachten, dass sich nahe beieinander liegende Agenten deutlich stärker gegenseitig beeinflussen als weiter entfernte. Daher beschränkt sich die Observation O_{com} auf die Positionen der geometrisch nächstgelegenen Nachbarn. Diese Form der Kommunikation basiert auf der in Kapitel 2.5.3 vorgestellten *Target*-Kommunikation. Auf Grundlage der euklidischen Distanz zu allen anderen Agenten desselben Clusters C_n berechnen sich die k -nächstgelegenen Agenten. Der zugehörige Wert von k ist eng mit der Mindestanzahl von Agenten pro Cluster verknüpft und beträgt folglich $k \geq 3$. Die Position ist mit einer 3D-Koordinate beschrieben und auf die Gesamtgröße des Environments normiert.

Für eine koordinierte Pfadplanung in einem Multi-Agenten-Szenario (vgl. Kapitel 2.1.2) ist eine Abschätzung über die zu erwartenden Richtungsvektoren der anderen Agenten erforderlich. Hierfür finden die ursprünglich für die Clusterbildung genutzten Beispielpfade eine zweite Verwendung (vgl. Kapitel 4.4.1.1). Der exemplarische Pfad gibt eine Vorstellung davon, wie die jeweilige elektrische Verbindung in der dreidimensionalen Geometrie zu verlegen ist. Die Pfade unterliegen der Prämisse, dass sie im Sinne des Optimierungsziels nicht optimal sind (vgl. Kapitel 4.1.4). Jedoch lässt sich mit der Kenntnis der aktuellen Position $P(x, y, z)$ eines jeden Agenten, seines Ziels P_T^i und seines Beispielpfades die erwartete Richtung in jedem Zeitschritt t bestimmen. Die Berechnung des Richtungsvektors kann in drei Szenarien unterteilt werden, die in Abbildung 4.16 dargestellt sind.

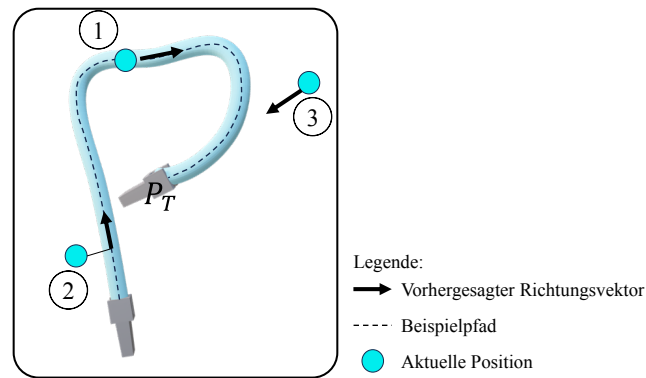


Abbildung 4.16: Beispiel für die Berechnung des erwarteten Richtungsvektors eines Agenten

Die gestrichelte Linie in Abbildung 4.16 deutet den Beispielpfad mit seinen Wegstreckepunkten und der Zielposition P_T an. Die möglichen Varianten der derzeitigen Position P_t^i sind mit blauen Kreisen dargestellt, während die schwarzen Pfeile den korrespondierenden Richtungsvektor anzeigen. Im ersten der drei Fälle (vgl. Abbildung 4.16(1)) befindet sich der hier genauer betrachtete Agent auf einer Position, die Teil des Beispielpfades ist. Der vorhergesagte Richtungsvektor $\vec{\tau}_*^i$ zeigt in diesem Szenario auf die Koordinate, die fünf Schritte weiter auf dem Beispielpfad liegt. Auf diese Weise erhält der Agent nicht nur die Information zur nächsten Aktion, sondern eine allgemeinere Richtungsinformation. Abrupte Richtungswechsel sind damit in der Pfadoptimierung besser zu berücksichtigen. Der Richtungsvektor wird in Gleichung 4.11 beschrieben.

$$\vec{\tau}_*^i = P_{t+5}^i - P_t^i \quad (4.11)$$

Das zweite Szenario (vgl. Abbildung 4.16(2)) ist eine leichte Modifikation des vorangegangenen Falls. Hier weicht der Agent innerhalb einer vordefinierten Toleranz vom Beispielpfad ab. Aufgrund des je nach Geometrie sehr großen Zustandsraumes und der Exploration im RL (vgl. Kapitel 2.5.1) ist diese Abweichung vernachlässigbar. Der resultierende Richtungsvektor $\vec{\tau}_*^i$ wird damit analog zur Gleichung 4.11 bestimmt. Weicht der Agent hingegen über die Toleranz hinaus vom Beispielpfad ab, so ergibt sich das dritte Szenario (vgl. Abbildung 4.16(3)). In diesem Fall hat der ursprünglich berechnete Beispielpfad keine Aussagekraft mehr. Stattdessen findet für die Vorhersage des Richtungsvektors $\vec{\tau}_*^i$ die Orientierung in Richtung des Zielpunktes P_T^i Anwendung. Gleichung 4.12 zeigt die zugehörige Berechnung.

$$\vec{\tau}_*^i = P_T^i - P_t^i \quad (4.12)$$

4.4.4 Anwendungsgebiet des kommunizierenden Ansatzes

Aufgrund der vielfältigen Beschaffenheit von CHRPs ist eine Beschreibung des Einsatzgebiets der neuen Methode erforderlich. Neben dem Fokus auf ein Anwendungsszenario für $N \geq 3$, vorgegeben durch die Kommunikationseinheit, liegt der Schwerpunkt insbesondere auf kompakten Bauräumen und einer Verlegung von Einzeladern. Lediglich innerhalb dieser ergibt sich die Anforderung eines hohen Interaktionsgrades zwischen Agenten aufgrund ihres hohen wechselseitigen Einflusses. Ist das CHRP hingegen durch eine weite räumliche Verteilung der Agenten charakterisiert, so kann der Informationsaustausch mittels Target-Kommunikation (vgl. Kapitel 2.5.3) sogar eine negative Auswirkung aufweisen. Da die Wahrnehmung der anderen Agenten innerhalb des Clusters C_n zu keinem Zeitpunkt Konsequenzen auf die ausgewählten Aktionen hat, kann kein Zusammenhang zwischen der Observation O^i , der Action A^i und dem Reward \mathcal{R}^i hergestellt werden. Eine genaue Einordnung in die Herausforderungen im MARL (vgl. Kapitel 2.5.4) ist zwar nicht möglich, dennoch beeinflusst die Non-Stationarity das Lernverhalten stark. Insbesondere die fehlende Korrelation zwischen Umgebungswahrnehmung, Aktion und Belohnung führt zu Herausforderungen im Erlernen der optimalen Policy π^* .

4.5 Skalierbares Multi-Agent Reinforcement Learning mit Transfer Learning

Die Weiterverwendung von bereits erzeugtem Wissen auf ähnliche Problemstellungen steht im Fokus dieses Kapitels zur Methode des skalierbaren TLTransfer Learnings (vgl. Kapitel 2.6.1). Für die Anwendung der neuen TL-Methode auf großdimensionierte 3D-CAD-Geometrien mit einer hohen Anzahl an elektrischen Einzelverbindungen beschreibt Abschnitt 4.5.1 die notwendigen datentechnischen Anpassungen und die Integration einer Kabelbaummethode. Im nachfolgenden Abschnitt 4.5.1 ist der Aufbau der angewendeten MARL-Methode illustriert. Dieser verdeutlicht die Weiterverarbeitung der Inputdaten in Form der Observation innerhalb des neuronalen Netzes. In Abschnitt 4.5.2 wird das Trainingsframework der MARL-Methode zur Erreichung der Generalisierungsfähigkeit unter Berücksichtigung der geometrischen Adaptionen im Produktentstehungsprozess detailliert (vgl. Kapitel 1.2). Dabei werden die erweiterten Optimierungsziele im Kontext des TL gemäß Abschnitt 2.6.5 integriert.

4.5.1 Skalierung der Methode auf große Geometrien und hohe Verbindungsanzahlen

Die Skalierbarkeit der TL-Methode ist in zwei Teilbereiche untergliedert. Einerseits ist eine hohe Anzahl an Einzelverbindungen zu planen, andererseits stellen großformatige 3D-CAD-Geometrien die bestehenden MAPF-Methoden vor Herausforderungen (vgl. Kapitel 3.2.2). Daraus entsteht der Bedarf einer Weiterentwicklung der aufgezeigten MARL-Methode unter Berücksichtigung des domänenspezifischen Bedarfs einer kabelbaumorientierten Verdrahtung im 3D-CAD-System (vgl. Kapitel 2.3.5). Die neue Methode setzt dafür ein mehrstufiges Verfahren fest, das in den Unterkapiteln 4.5.1.1 zur Modifikation der 3D-CAD-Geometrie, 4.5.1.2 zum Berechnungsverfahren zur Ermittlung von Kabelbündeldurchmessern und 4.5.1.3 zur Reintegration der Modifikationen in die MARL-Methode beschrieben ist.

4.5.1.1 Alpha-Wrap-Methode zur Graphmodifikation

Die neue Methode zur Lösung des Verkabelungsproblems für großdimensionierte 3D-CAD-Geometrien mit einer hohen Anzahl an Einzelverbindungen integriert eine Subgraphstruktur in Form eines Branching Trees H (vgl. Kapitel 2.1.4). Dieser basiert auf einer adaptierten, Mesh-basierten Betrachtungsweise des Environments im POMG (vgl. Abbildung 4.11c in Kapitel 4.3.2). Diese 3D-Modell-Repräsentanz ermöglicht den effizienten Einsatz eines Pfadsuchalgorithmus auf dem Graphen $G(V, E)$ zur Generierung des Branching Trees $H \subset G(V, E)$. Im initialen Verarbeitungsschritt wird dabei der Graph $G(V, E)$ mit der Alpha-Wrap-Methode abgewandelt (vgl. Anhang A.5). Dafür werden die verfügbaren Edges E aus der Oberfläche der 3D-CAD-Geometrie herausbewegt. Abbildung 4.17 stellt die ursprüngliche 3D-CAD-Geometrie in einer Mesh-Repräsentanz einem Mesh gegenüber, das durch die Alpha-Wrap-Methode generiert ist.

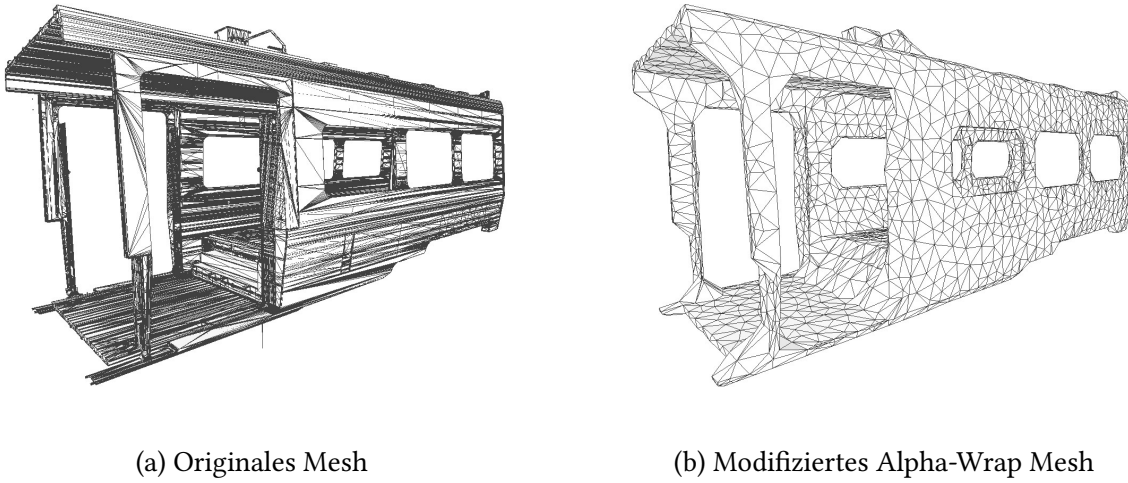


Abbildung 4.17: Grafische Darstellung zur Anwendung der Alpha-Wrap-Methode i. A. a. [133]

Der Gegenüberstellung der beiden 3D-Mesh-Repräsentanten (vgl. Abbildung 4.17) ist zu entnehmen, dass die Nutzung der Alpha-Wrap-Methode nicht nur zur Erzeugung einer Oberflächendistanz, sondern auch zur Simplifizierung des Graphen $G(V, E)$ führt.

Der modifizierte Graph $G_\alpha(V, E)$ erfordert eine weitere Anpassung zur Integration der Anschlusspunkte P_S und P_T aller elektrischen Einzelverbindungen. Andernfalls besteht die Anforderung, dass jeder Anschlusspunkt P_S^i und P_T^i zwangsläufig auf einem bestehenden Knoten E_α liegen muss (vgl. Kapitel 2.1). Abbildung 4.18 zeigt diese Modifikation durch die Integration von elektrischen Anschlusspunkten an einem exemplarischen Mesh.

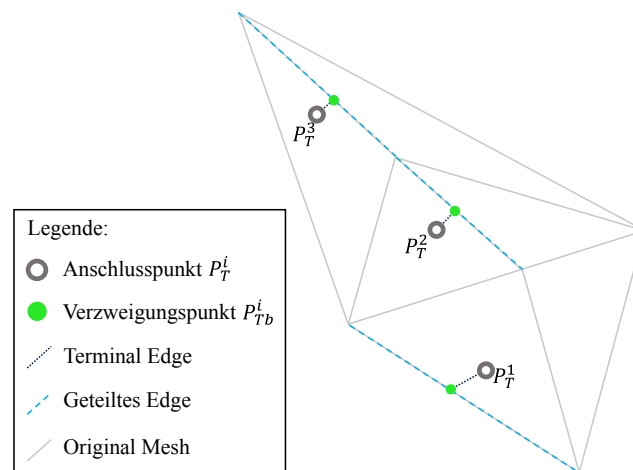


Abbildung 4.18: Exemplarische Erweiterung der 3D-Mesh-Struktur durch elektrische Anschlusspunkte

Die Punkte $P_T^1 - P_T^3$ in Abbildung 4.18 repräsentieren die Anschlusspunkte, die nach der Zusammenführung von ECAD- und 3D-MCAD-Daten (vgl. Kapitel 4.2.2) nicht direkt auf

dem originalen Mesh verortet sind. Im ersten Schritt ist es daher erforderlich, die nächstgelegene Kante E_α und den darauf liegenden nächstgelegenen Punkt des Graphen $G_\alpha(V, E)$ zu identifizieren. Das Bewertungskriterium ist hierfür die euklidische Distanz. Die Verbindung zwischen dem Anschlusspunkt P_T^i und diesem neu erzeugten Verzweigungspunkt P_{Tb}^i wird dem Gesamtgraphen hinzugefügt (vgl. Terminal Edge in Abbildung 4.18). Im zweiten Schritt unterteilen die neu erzeugten Verzweigungspunkte P_{Tb}^i den bestehenden Graphen (vgl. geteiltes Edge in Abbildung 4.18). Somit werden dem Mesh neue Knoten in Form von Verzweigungspunkten hinzugefügt und die bestehenden Edges in zwei separate Kanten aufgeteilt. Innerhalb des modifizierten Graphen $G'_\alpha(V, E)$ können im nachfolgenden Schritt die Verbindungen zwischen Start- und Zielpunkten aller elektrischer Verbindungen berechnet werden. Dafür kommen graphgestützte Pfadoptimierungsalgorithmen wie der A*-Algorithmus (vgl. Kapitel 3.2) zum Einsatz.

Zur Identifikation der Verzweigungspunkte des Kabelbaumes nutzt die neue Methode ein Verfahren zur Adaptierung von Kantengewichten unter Verwendung des Bündelungsfaktors w_B . Dazu wird die Verbindungsliste anhand der maximalen euklidischen Distanz zwischen Start- und Endpunkt sortiert. Mittels A*-Algorithmus (vgl. Kapitel 3.2.1) wird danach innerhalb des Graphen $G'_\alpha(V, E)$ für die erste elektrische Verbindung der kürzeste Pfad ermittelt. In allen darin enthaltenen Kanten $E \subset G'_\alpha(V, E)$ modifiziert sich das assoziierte Kantengewicht $w(u, v)$ (vgl. Kapitel 2.1.1) durch die Multiplikation mit dem Bündelungsfaktor w_B . Die Verwendung der modifizierten Kanten wirkt sich folglich auf nachfolgende Verbindungen positiv aus, da geringere Gesamtkosten für den Pfad im Graphen entstehen (vgl. Kapitel 3.2). Alle elektrischen Verbindungen der Verbindungsliste werden nach dieser Vorgehensweise und der Anpassung der Kantengewichte iterativ durchlaufen (vgl. Kapitel 2.3.2). Abbildung 4.19 zeigt die Modifikation der Kantengewichte exemplarisch auf.

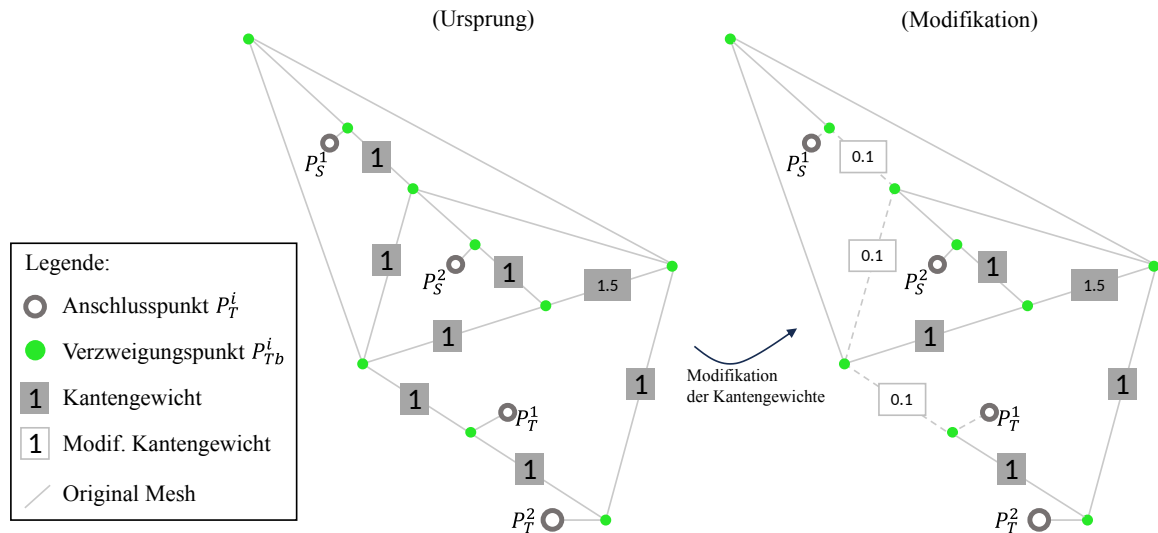


Abbildung 4.19: Exemplarische Darstellung zur Erzeugung einer Subgraphstruktur mit modifizierten Kantengewichten

Im ursprünglichen Zustand des Graphen $G'_\alpha(V, E)$ (vgl. Abbildung 4.19 links) beinhaltet der kürzeste Pfad des Agenten $i = 2$ zwischen dem Startpunkt P_S^2 und dem Zielpunkt P_T^2 ein

minimiertes Gesamtkantengewicht von 3.5 (vgl. Kapitel 2.1.1). Durch die Anpassung der Kantengewichte mittels des Bündelungsfaktors $w_B = 0.1$ passt sich der kürzeste Pfad an und resultiert in einer stärkeren Bündelung beider Pfade. Im Falle von $w_B = 0.9$ verändert sich der Pfad hingegen nicht. Der daraus entstehende Kabelbaum $H_\alpha \subset G'_\alpha(V, E)$ beinhaltet folglich alle elektrischen Verbindungen. Die Information, welche Verbindungen über ein entsprechendes Kantensegment verlaufen, ist für die nachgelagerten Prozessschritte von großer Bedeutung.

Ergibt sich innerhalb des Produktentstehungsprozesses eine Anpassung bzw. Änderung an der elektrischen Verbindungsliste (vgl. Kapitel 2.3.2 und 4.1.1), so wirkt sich diese mitunter auf den gesamten Kabelbaum aus. Unter Abwägung von aufzuwendender Berechnungszeit und der Optimalität des Gesamtergebnisses (vgl. Kapitel 4.1.4) sieht die Methode eine Modifikation des Kabelbaumes anstelle einer Neuberechnung vor. Dabei ist zwischen den Szenarien einer entfallenden (1), einer adaptierten (2) und einer zusätzlichen Verbindung (3) zu unterscheiden. Die drei Fälle sind in den folgenden Punkten 1 bis 3 aufgelistet:

1. **Verbindung entfällt:** Sofern eine elektrische Verbindung aus der Verbindungsliste entfernt wird, sind die davon betroffenen Kantengewichte $w(u, v)$ anzupassen. Dies erfolgt über einen invertierten Prozess der Neuentstehung (vgl. Abbildung 4.19). Bildet die Verbindung einen eigenständigen Branch des Subgraphen $H_\alpha \subset G'_\alpha(V, E)$, so entfällt dieser gänzlich. Andere Verbindungen sind von dieser Anpassung nicht beeinträchtigt.
2. **Modifizierte Verbindung:** Ist der modifizierte Start- oder Zielpunkt der Verbindung nicht auf einem Knoten $V \in G'_\alpha(V, E)$ verortet, wird der Subgraph H_α an der eingangs vorgesehenen Position in seinen Ursprungszustand zurückversetzt. Zudem wird für den neuen Anschlusspunkt eine Knotenerweiterung durchgeführt (vgl. Abbildung 4.18). Die Pfadidentifikation auf dem Subgraphen erfolgt analog zur Vorgehensweise einer Neuerzeugung.
3. **Neue Verbindung:** Ist eine zusätzliche elektrische Verbindung erforderlich, so wird diese analog zur Vorgehensweise in Abbildung 4.19 (rechts) den bestehenden Subgraphen $H_\alpha \subset G'_\alpha(V, E)$ erweitern.

Die assoziierten Durchmesser der jeweiligen Subgraphabschnitte passen sich anhand der Berechnung der Bündeldurchmesser an (vgl. Abschnitt 4.5.1.2). Diese Berechnung liefert Aufschluss darüber, ob ein existentes Planungsergebnis der elektrischen Verdrahtung im 3D-CAD-System nach der Modifikation des Kabelbaumes weiterhin valide ist.

4.5.1.2 Berechnung der Bündeldurchmesser

Die möglichst exakte Berechnung des assoziierten Durchmessers des jeweiligen Kabelbündels ist eine notwendige Voraussetzung für die realitätsgetreue Abbildung der elektrischen Verbindungen innerhalb einzelner Kabelbündel (vgl. Kapitel 2.3.5). Diese Anforderung lässt sich einerseits mit dem Ziel der Vermeidung von Kollisionen (vgl. Kapitel 2.3.3),

andererseits mit der realitätsnahen Darstellungsform von Kabelbäumen in einem 3D-CAD-System begründen (vgl. Kapitel 3.1). Die Ermittlung der tatsächlichen Querschnitte erfolgt in einem dreistufigen Prozess. Der erste Schritt besteht in der Ermittlung der Anzahl an Einzelverbindungen, die mit ihren jeweiligen Kabeldurchmessern ein Edge E des Subgraphen H_α durchlaufen. Im zweiten Schritt erfolgt die Ermittlung des durchschnittlichen Kabeldurchmessers je Edge. Der Durchmesser des jeweiligen Bündels wird folglich unter Berücksichtigung des zuvor ermittelten durchschnittlichen Durchmessers der elektrischen Verbindungen im dritten Schritt berechnet. Die Annäherung der korrekten Bündelform erfolgt in Anlehnung an die Darstellungsform gängiger 3D-CAD-Systeme (vgl. Kapitel 3.1). Die kreisrunde Annäherung des Bündels kann in drei unterschiedlichen Szenarien erfolgen, die in Abbildung 4.20 dargestellt sind.

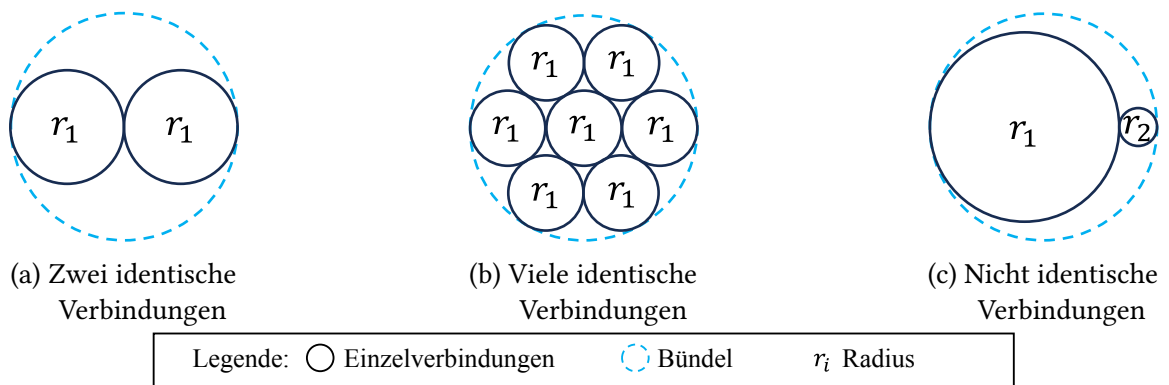


Abbildung 4.20: Exemplarische Berechnung der Bündeldurchmesser in Kabelbäumen i. A. a. [200]

Abbildung 4.20a zeigt, wie der Bündeldurchmesser für zwei Verbindungen mit dem gleichen Radius für r_i als kreisrunde Form angenähert wird. Es ist zu erkennen, dass bei dieser geringen Anzahl an Kabeln bzw. Agenten eine höhere Abweichung zwischen der Annäherung und dem Realwert vorliegt. In Abbildung 4.20b ist dargestellt, wie präzise die Approximierung des Außendurchmessers des Bündels bei einer Vielzahl elektrischer Verbindungen mit gleichem Kabeldurchmesser ist. Es lässt sich erkennen, dass mit steigender Agentenanzahl N (elektrische Verbindungen) die Annäherung eine zunehmende Genauigkeit aufweist. Das Szenario, in dem die Durchmesser der einzelnen elektrischen Verbindungen variieren, ist in Abbildung 4.20c veranschaulicht. Die Berechnung des Bündeldurchmessers d_B auf Grundlage des durchschnittlichen Einzelverbindungsdurchmessers d_{avg} ist in Formel 4.13 beschrieben.

$$d_B = d_{avg} \times N_E \times \sqrt{1.49} \quad (4.13)$$

In Formel 4.13 beschreibt die Variable d_{avg} den durchschnittlichen Durchmesser der Einzelverbindungen und N_E die Anzahl der Agenten, die den entsprechenden Branch durchlaufen. Der Wert $\sqrt{1.49}$ repräsentiert einen konstanten Bündelfaktor, der sich aus der Analyse im Stand der Technik ableitet (vgl. Kapitel 3.2).

4.5.1.3 Reintegration des Kabelbaums in die MARL-Methode

Mit dem Ziel einer skalierbaren Erweiterung der MARL-Methode muss der erzeugte Informationsgehalt der Kapitel 4.5.1.1 und 4.5.1.2 in den grundlegenden Aufbau der neuen Methode integriert werden (vgl. Kapitel 4.1.1). So ändert sich insbesondere die Kardinalität zwischen den elektrischen Verbindungen und den Agenten a^i . Durch die Zusammenführung der Einzelverbindungen in einen Kabelbaum repräsentiert ein einzelner Agent des POMG einen spezifischen Abschnitt des Kabelbaumes. Dies kann einerseits ein vollständiger Branch sein, andererseits auch ein kleines Teilstück des Bündels (vgl. Kapitel 2.3.5). In jedem Fall stellt der Agent eine zusammenhängende Kette von Edges $E \subset G'_\alpha(V, E)$ dar, in der eine gleichbleibende Anzahl von elektrischen Verbindungen enthalten ist. Ausgangspunkt für den Agenten ist damit ein identifizierter Verzweigungspunkt P_b^i oder der Start- bzw. Zielpunkt einer elektrischen Verbindung (vgl. Kapitel 4.5.1.1). Als zugehörigen Durchmesser ersetzt d_B (vgl. Gleichung 4.13) die Kabelquerschnitte der einzelnen heterogenen Verbindungen. Für den zulässigen Biegeradius des gesamten Kabelbündels wird der minimale Biegeradius der enthaltenen Einzelverbindungen festgelegt (vgl. Kapitel 2.3.3). Dies ist durch die nicht näher spezifizierte Position einzelner Kabeltypen innerhalb des Bündels begründet. Andernfalls könnte ein Kabeltyp mit einem sehr kleinen Wert für den minimalen Biegeradius in einer innenliegenden Position des Bündels zu einem nicht validen und somit unbrauchbaren Ergebnis führen.

Diese Herangehensweise greift eine domänenabhängige *Problem-Decomposition* auf (vgl. Kapitel 2.5.4 und 4.4.2). Anstelle einer anwendungsneutralen Aufteilung von Pfadabschnitten des MAPF in Subprobleme steuert die Erzeugung des Kabelbaums die Entscheidung, wie das Ausgangsproblem (CHRP) unterteilt wird. Abbildung 4.21 stellt eine Übersicht der vorgenommenen Modifikationen der Methode bereit und zeigt auf, welchen Einfluss die Kabelbaumerzeugung auf das Routing nimmt.

Abbildung 4.21 verdeutlicht, wie die Alpha-Wrap-Methode (vgl. Kapitel 4.5.1.1 und A.5) nach der Konsolidierung der ECAD- und der 3D-MCAD-Daten (vgl. Kapitel 4.2.2) das ursprüngliche 3D-Mesh-Modell modifiziert (vgl. Abbildung 4.21 (1) bis (3)). Dieses enthält folglich die datentechnischen Grundlagen für die Kabelbaummethode (vgl. Abbildung 4.17 und 4.21 (3)). Daraus ergeben sich modifizierte Start- und Zielpunkte für eine angepasste Kardinalität zwischen den Agenten und den elektrischen Verbindungen (vgl. Abbildung 4.21 (4)). Auswirkungen auf die Beschaffenheit des Environments bestehen abgesehen von einer veränderten Agentenanzahl nicht (vgl. Abbildung 4.21 (5)).

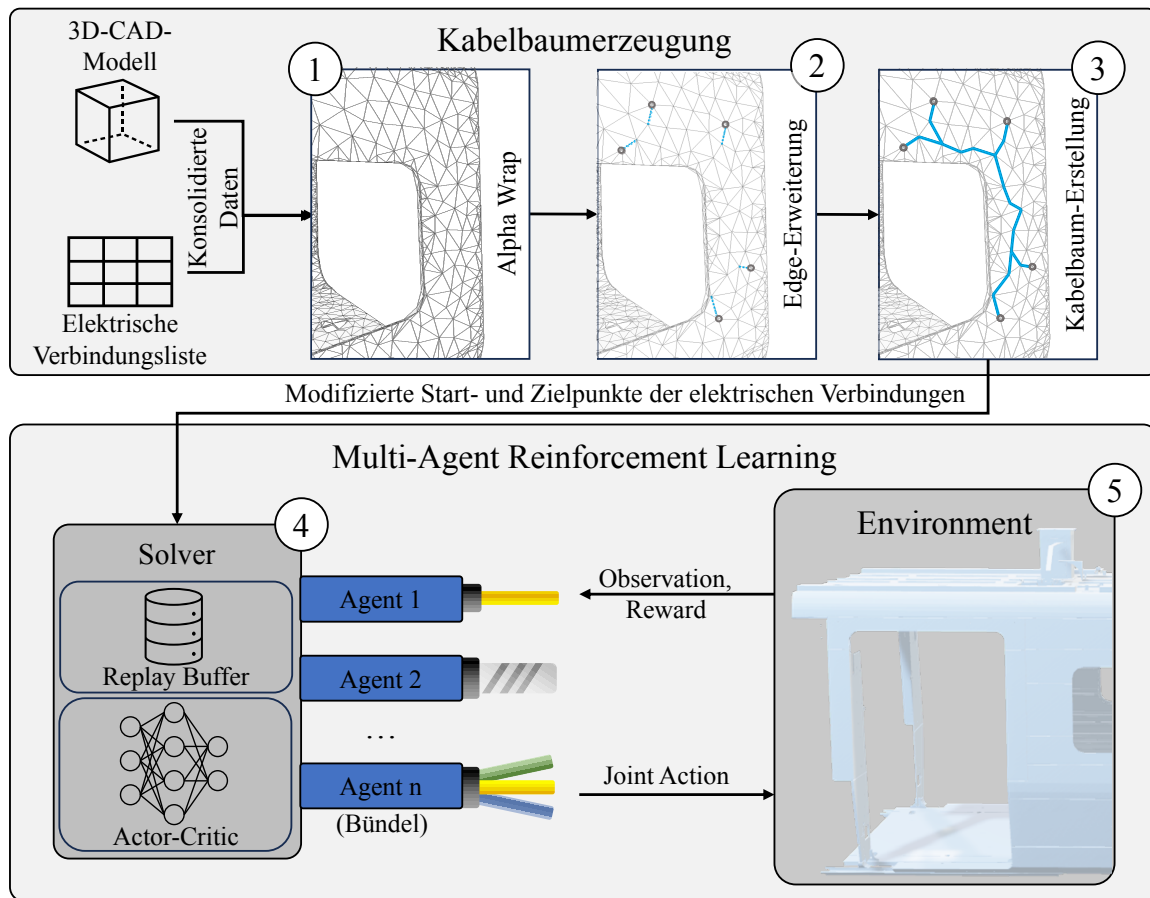


Abbildung 4.21: Verortung der Kabelbaumerstellung im Prozessablauf der Methode in einer exemplarischen 3D-CAD-Geometrie

4.5.2 Trainingsframework zur Verbesserung der Generalisierungsfähigkeit auf unterschiedliche Geometrien

Eines der drei wesentlichen Ziele der neuen Methode ist die Übertragbarkeit der erlernten Lösungsstrategien der Sourcedomäne π_{D_S} auf neue bzw. adaptierte CHRP (vgl. Kapitel 1.3). Um die bisher übliche Kardinalität von 1 : 1 zwischen Lerntask und ML-Methode zu vermeiden (vgl. Kapitel 2.6.1), ist eine spezielle Trainingsmethode erforderlich. Diese unterbindet ein Overfitting an die konkrete Problemstellung des CHRP der Sourcedomäne D_S . Die neue Methode als MARL deckt mit Bezug zur vorliegenden Problemstellung (vgl. Kapitel 1.2) somit nicht nur die Lösung des initialen CHRP ab, sondern ermöglicht auch den Wissenstransfer auf ähnliche Environments. In Kapitel 4.5.2.1 wird das Trainingsframework zur Realisierung einer besseren Übertragbarkeit von erlernten Lösungsstrategien beschrieben. Kapitel 4.5.2.2 zeigt auf, wie die geometrische Ähnlichkeit zwischen zwei 3D-CAD-Geometrien gemessen wird. In Kapitel 4.5.2.3 wird darauf aufbauend festgelegt, wie die berechnete Similarität Einfluss auf den Wissenstransfer nimmt.

4.5.2.1 Trainingsframework für eine erhöhte Generalisierungsfähigkeit

Als Teil der neuen Methode wird ein Trainingsframework festgelegt, das eine Übertragung von generiertem Wissen ermöglicht, ohne eine Überanpassung auf ein konkretes CHRP zu erzeugen. Abbildung 4.22 veranschaulicht den Zusammenhang zwischen dem Trainingsprozess in der Sourcedomäne D_S und der Übertragung auf neue CHRP in der Zieldomäne D_T .

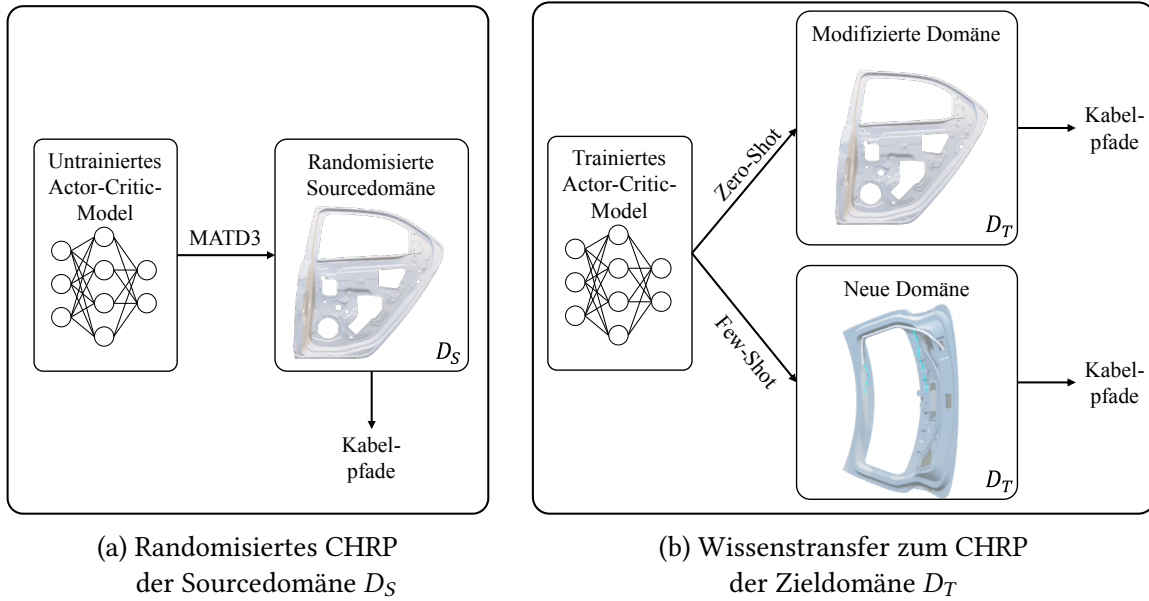


Abbildung 4.22: Schematische Darstellung der Transfer-Learning-Methode in zwei exemplarischen 3D-CAD-Modellen

In Abbildung 4.22 ist der Bezug zur Kategorisierung der TL-Methoden gemäß Aufgaben- und Domänensimilarität dargestellt (vgl. Kapitel 2.6.3). Im initialen Zustand erlernt der in Kapitel A.2 genauer beschriebene MARL-Algorithmus TD3 (vgl. Kapitel 2.5.1) das Lösen der Sourcedomäne D_S (vgl. Abbildung 4.22a). Diese ist assoziiert mit einem CHRP, das durch die 3D-CAD-Geometrie und eine Verbindungsliste als Mindestanforderung beschrieben ist (vgl. Kapitel 4.1.2). Aus dem Trainingsprozess entsteht ein (vor-)trainiertes neuronales Netz, das eine generalisierende Policy π abbildet. Für den Wissenstransfer auf unterschiedliche Zieldomänen (vgl. Abbildung 4.22b) kann dieses neuronale Netz eingesetzt werden, um angepasste bzw. ähnliche CHRP zu lösen.

An dieser Stelle sind zwei wesentliche Trainingsstrategien für das Erlernen der Lösungsfähigkeit eines CHRP zu unterscheiden. Diese beiden Methoden werden nachfolgend in der Validierung in Kapitel 5 voneinander abgegrenzt. Innerhalb des **Standardtrainings** erlernen die Agenten die Lösung des CHRP von Grund auf neu. Währenddessen bleibt die Anzahl der Agenten $N \geq 2$ identisch. Ebenso verändern sich die Koordinaten der Start- (P_S^i) und Zielpunkte (P_T^i) nicht. Das Ziel der Methode besteht folglich darin, das identische POMG fortwährend zu optimieren (vgl. Kapitel 4.3.2). Die Trainingsmethode des **randomisierten Trainings** zielt ebenfalls darauf ab, ein CHRP mit der gleichen Anzahl von Agenten $N \geq 2$ während des gesamten Trainings zu optimieren. Darüber hinaus wird auch hier ein

neuronales Netz von Grund auf neu trainiert. Nach einer bestimmten Anzahl von Trainingsepisoden erfolgt jedoch eine zufällige Verschiebung des Zielpunkts P_T^i jedes Agenten $a^i \forall i \in \{1, \dots, N\}$ innerhalb eines vorgegebenen Entfernungsbereichs. Die letzte Zielposition P_T^i dient dafür als Ausgangspunkt. Dies erfolgt in Anlehnung an die Repositionierung von elektrischen Komponenten innerhalb des Produktentstehungsprozesses (vgl. Kapitel 1.2). Abbildung 4.23 zeigt beispielhaft drei konsekutive Zielpunkt-Randomisierungen eines Agenten innerhalb eines exemplarischen Environment-Ausschnittes.

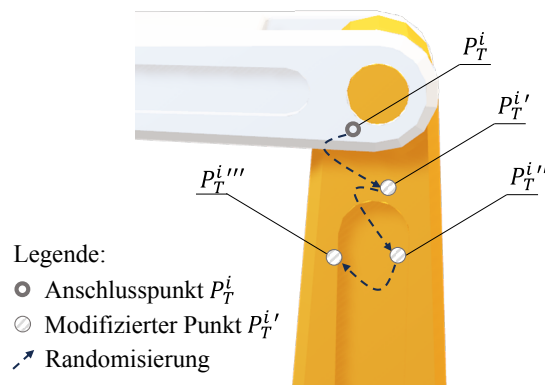


Abbildung 4.23: Exemplarische Zielpunktrandomisierung in einem Beispiel-Environment

In Abbildung 4.23 beschreibt die Variable P_T^i den Zielpunkt des Agenten a^i , wie er initial bei der Zusammenführung von ECAD- und MCAD-Daten festgelegt ist (vgl. Kapitel 4.2.2). Die weiteren Punkte $P_T^{i'}$ bis $P_T^{i'''}$ repräsentieren die jeweiligen Positionen nach den Randomisierungsschritten. Diese haben alle Freiheitsgrade innerhalb des dreidimensionalen Raumes, solange die Position nicht innerhalb der 3D-CAD-Geometrie liegt oder den maximalen Oberflächenabstand d_{max} überschreitet. Der assoziierte Richtungsvektor des Anschlusspunktes bleibt während der Randomisierung identisch. Damit imitiert die Randomisierung mehrere Repositionierungen von elektrischen Komponenten innerhalb der 3D-CAD-Baugruppe als Bestandteil des Produktentstehungsprozesses. Gleichzeitig ist darauf zu achten, den Entfernungsbereich der Randomisierung nicht zu hoch zu setzen, da andernfalls das Risiko des *Catastrophic Forgetting* erhöht wird (vgl. Kapitel 2.5.4).

4.5.2.2 Messung der geometrischen Ähnlichkeit zwischen 3D-CAD-Modellen

Als Indikator für die Übertragbarkeit des bereits generierten Wissens legt die neue Methode einen Grad der geometrischen Ähnlichkeit zwischen den 3D-CAD-Modellen fest. Es ist erforderlich, diese Similarität für eine Differenzierung zwischen unterschiedlichen TL-Strategien zu quantifizieren (vgl. Kapitel 2.2.4). Dabei soll explizit die geometrische Beschaffenheit des Environments selbst Einfluss nehmen, während räumliche Transformationen (z. B. Rotation oder Skalierung) lediglich eine geringe Auswirkung haben. (vgl. Kapitel 2.4.4.2). Die neue Methode ist der Kategorie der graphbasierten Ansätze zuzuordnen (vgl. Kapitel 2.2.4). Die datentechnische Grundlage für die Quantifizierung der Ähnlichkeit ist die Mesh-basierte Betrachtungsweise der 3D-CAD-Geometrie aus dem

MCAD (vgl. Kapitel 4.3.1), die nach der Zusammenführung von ECAD- und MCAD-Daten zur Verfügung steht (vgl. Kapitel 4.2.2). Der Prozess der Ähnlichkeitsquantifizierung ist in Abbildung 4.24 schematisch dargestellt.

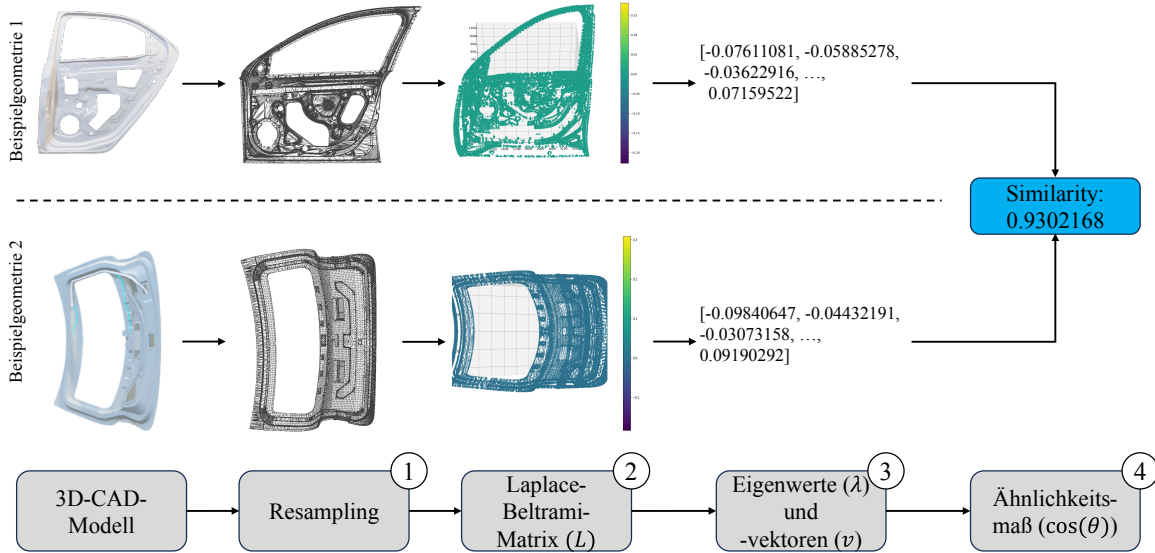


Abbildung 4.24: Schematische Darstellung zum Prozess der Quantifizierung der geometrischen Ähnlichkeit von exemplarischen 3D-CAD-Baugruppen

Die Methode zur Bestimmung des Ähnlichkeitsmaßes von zwei exemplarischen 3D-CAD-Geometrien ist in den Schritten 1. bis 4. in Abbildung 4.24 erläutert.

1. Für die Ermittlung der geometrischen Ähnlichkeit ist im initialen Schritt ein **Resampling** der zu vergleichenden Mesh-Strukturen erforderlich, um eine gleiche Anzahl an Vertices und Edges pro Graph $G(V, E)$ zu erhalten (vgl. Kapitel 2.1). Dieser Prozessschritt reduziert die Anzahl von Flächen, während die grundlegende Form, das Volumen und die räumlichen Grenzen der 3D-CAD-Geometrie erhalten bleiben. Die Methode wendet einen quadratischen Fehler-Metrik-Algorithmus (en: *Quadric Error Metric*) an, um das bereitgestellte Mesh zu simplifizieren.
2. Im nachgelagerten zweiten Schritt wird die **Laplace-Beltrami-Matrix** (auch als Laplace-Matrix bezeichnet) $L \in \mathbb{R}^{n \times n}$ berechnet. Die Basis hierfür ist die Subtraktion der Adjazenzmatrix $A_{i,j}$ von der Gradmatrix $D_{i,j}$ des 3D-Mesh. Die Laplace-Matrix beschreibt die Diffusion auf der Oberfläche des Mesh und erfasst die intrinsische Geometrie und Struktur der Form. Formell ist sie in Gleichung 4.14 abgebildet.

$$L_{i,j} = \begin{cases} w(i, j) = \frac{1}{2}(\cot \alpha_{i,j} + \cot \beta_{i,j}), & \text{wenn } (i, j) \in E \wedge i, j \text{ benachbart,} \\ -\sum_{j \in N(i)} w_{i,j}, & \text{wenn } j = i \\ 0, & \text{sonst} \end{cases} \quad (4.14)$$

In Gleichung 4.14 beschreibt die Variable $L_{i,j}$ ein einzelnes Element der Matrix und $j \in N(i)$ die Liste aller angrenzenden Vertices zum Vertex i (vgl. Kapitel 2.1.1). Die Funktion $\cot \alpha_{i,j}$ repräsentiert die Kotangensfunktion an der Stelle $\alpha_{i,j}$.

3. Zur Ermittlung der geometrischen Struktur werden im nächsten Schritt die **Eigenwerte und -vektoren** beider Laplace-Matrizen ermittelt. Der Eigenwert repräsentiert in diesem Zusammenhang die globalen Eigenschaften des 3D-Mesh, z. B. die Form und Größe. Der Eigenvektor hingegen korrespondiert mit spezifischen Schwingungsmustern auf der Oberfläche des 3D-Mesh. Gleichung 4.15 beschreibt die Basis für die Ermittlung des Eigenvektors v und des zugehörigen Eigenwertes λ auf Basis der Laplace-Matrix L .

$$Lv = \lambda v \quad (4.15)$$

Die Gleichung 4.15 wird mittels Anwendung der Lanczos-Methode gelöst. Dieser Algorithmus approximiert die Matrix durch eine Reihe von kleineren Tridiagonalmatrizen und berechnet deren Eigenwerte und Eigenvektoren. Sie dienen dann als Näherungen für die Eigenwerte und Eigenvektoren der ursprünglichen großen Matrix. Besonders für dünn besetzte Matrizen ist dieses Verfahren sehr recheneffizient.

4. Teilschritt 4 umfasst die **Bestimmung des Ähnlichkeitsmaßes**. Für diesen Teilschritt kommt die Kosinus-Similarität $\cos(\theta)$ zur Anwendung. Sie wird als der Kosinus des Winkels zwischen zwei Vektoren (hier Eigenvektoren) in einem hochdimensionalen Vektorraum berechnet und hat einen Wertebereich von -1 bis 1 . Während ein Wert von $\cos(\theta) = 0$ keine Ähnlichkeit der 3D-CAD-Geometrien impliziert, zeigt der Wert von $\cos(\theta) = 1$ eine vollständige Ähnlichkeit.

4.5.2.3 Einfluss der geometrischen Similarität auf das Transfer Learning

Ausgehend von der berechneten Ähnlichkeit (vgl. Kapitel 4.5.2.2) sind in der neuen Methode unterschiedliche Vorgehensweisen für das TL festgelegt. Hier ist zwischen dem **Zero-Shot-Transfer** und dem **Few-Shot-Transfer** zu unterscheiden (vgl. Kapitel 2.6.1), der sich am Wert der Kosinus-Similarität orientiert.

So ist die Entscheidungsgrundlage, ob in einer Zieldomäne D_T ein Zero-Shot-Transfer oder ein Few-Shot-Transfer angewendet wird, abhängig von der geometrischen Similarität zwischen Source- und Zieldomäne. Prämisse dafür ist, dass eine Policy π_{D_S} durch das randomisierte Training (vgl. Kapitel 4.5.2.1) geometrisch vergleichbare CHRP direkt in adäquater Qualität lösen kann. Unterschreitet der Wert von $\cos(\theta)$ den Schwellenwert von 0.7 , wird angenommen, dass sich ein Zero-Shot-Transfer nicht mehr positiv auf das CHRP im neuen Environment auswirkt und eine Nachtrainingsphase (Few-Shot-Transfer) in der Zieldomäne D_T erforderlich ist. Damit wird eine notwendige Anpassung der Policy π_{D_S} an die neue Domäne D_T erreicht (vgl. Abbildung 4.22b). Liegt die Kosinus-Similarität unter dem Wert von 0.4 , besteht die Annahme, dass sich ein Transfer der Policy π_{D_T} nicht mehr positiv auf das zu lösende CHRP auswirkt und die Integration einer Nachlernphase keinen Mehrwert bringt. Stattdessen wird ein neues POMG mit einer eigenen Policy generiert.

4.6 Zusammenfassung der neuen Methode zur automatisierten virtuellen Verdrahtung im 3D-CAD

Die innerhalb der Kapitel 4.1 bis 4.5 entwickelten Methoden verfolgen das Ziel einer automatisierten, virtuellen Verdrahtung im 3D-CAD-System mittels der MARL-Methode. Dabei sollen insbesondere die identifizierten Anforderungen aus der Analyse des Stands der Technik (vgl. Kapitel 3.3) abgedeckt werden. Darüber hinaus schließen die neuen Methoden partiell die identifizierte Forschungslücke, die in Tabelle 3.1 dargestellt ist.

Als modulares Gesamtkonzept mit einer REST-Schnittstelle zum ECAD- und zum MCAD-System fügt sich die neue Methode aus einer datentechnischen Perspektive nahtlos in die bestehende gängige IT-Systemlandschaft ein. Das übergeordnete Ziel besteht in der optimalen Lösung des CHRP im 3D-CAD-System (vgl. Kapitel 4.1.1). Zu diesem Zweck werden die Daten unterschiedlicher Dimensionalitäten aus 2D-Schaltplan und 3D-CAD-Modell im Format AutomationML konsolidiert (vgl. Kapitel 4.2.2). Die definierte Ausprägung und Qualifikation der mechanischen und elektrischen Eingangsdaten dient dafür als datentechnische Grundlage (vgl. Kapitel 4.1.2). Als Optimierungsziel legt die neue Methode die minimale Gesamtlänge aller elektrischen Einzelverbindungen fest (vgl. Kapitel 4.1.4). Unter Einsatz der MARL-Methode können alle Agenten zeitgleich in einem gemeinsamen Environment agieren (vgl. Kapitel 4.3.2) und überwinden damit die Beschränkung der iterativen Arbeitsweise. Aufgrund der Gestaltung der Reward-Funktion \mathcal{R} wird ein kooperatives Verhalten der Agenten forciert, das eine minimale *Flowtime* realisiert (vgl. Kapitel 2.1.2). Darüber hinaus ist der Aktionsraum \mathcal{A} aller Agenten dahingehend eingeschränkt, dass lediglich Aktionen ausführbar sind, die als wesentliche Randbedingung den kabelspezifischen minimalen Biegeradius einhalten (vgl. Kapitel 2.3.3). Als spezielle Form der Randbedingung berücksichtigt die neue Methode auch dynamische Bauteile innerhalb der 3D-CAD-Geometrie (vgl. Kapitel 4.2.4), sofern die notwendigen Qualifikationen im 3D-CAD-System eingehalten werden (vgl. Kapitel 4.1.2). Mit Blick auf die unterschiedlichen Anwendungsdomänen und deren Vorgehensweise für die virtuelle Verdrahtung kann die neue Methode eine Pfadoptimierung sowohl für Einzelverbindungen in kompakten Bauräumen (vgl. Kapitel 4.4.2) als auch innerhalb von Kabelbäumen realisieren (vgl. Kapitel 4.5.1). Diese Flexibilität geht mit der Übertragbarkeit des generierten Wissens auf artverwandte CHRP mit Hilfe eines Trainingsframeworks durch Zielpunktrandomisierung einher (vgl. Kapitel 4.5.2.1).

Das Zusammenwirken aller einzelnen methodischen Bausteine ermöglicht es schließlich, eine optimierte virtuelle Verdrahtung im 3D-CAD-System zu generieren. In Kapitel 5 erfolgt die Validierung der neuen Methode, wobei auch die Erfüllung der Anforderungen in unterschiedlichen Validierungsszenarien überprüft wird.

5 Validierung der entwickelten Methoden

In den nachfolgenden Kapiteln werden die entwickelten Methoden aus Kapitel 4 validiert. Dies geschieht unter Einbeziehen der in Kapitel 3.3 analysierten Anforderungen an die automatisierte virtuelle Verdrahtung innerhalb eines 3D-CAD-Systems. Eine Überprüfung der Anwendbarkeit erfolgt anhand von verschiedenen Validierungsszenarien unterschiedlicher Komplexitätsstufen. In Kapitel 5.2 sind die geometrischen Validierungsumgebungen definiert und deren Einschränkungen sowie die assoziierten Verbindungslisten beschrieben. Im folgenden Kapitel 5.3 sind die numerischen und visuellen Ergebnisse der Validierungsläufe dargestellt. Das abschließende Kapitel 5.4 fasst die Ergebnisse zusammen und diskutiert diese im Kontext der eingangs aufgestellten Problemstellungen (vgl. Kapitel 1.2) und der Optimierungsziele (vgl. Kapitel 4.1.4).

5.1 Aufbau der Validierung und deren Bewertungskriterien

Der strukturelle Aufbau der Validierung der neuen Methoden orientiert sich an der Kapitelabfolge des Methodenkapitels (vgl. Kapitel 4) und den inhaltlichen Zusammenhängen der jeweiligen Methoden. Abbildung 5.1 zeigt den Aufbau der Validierung.

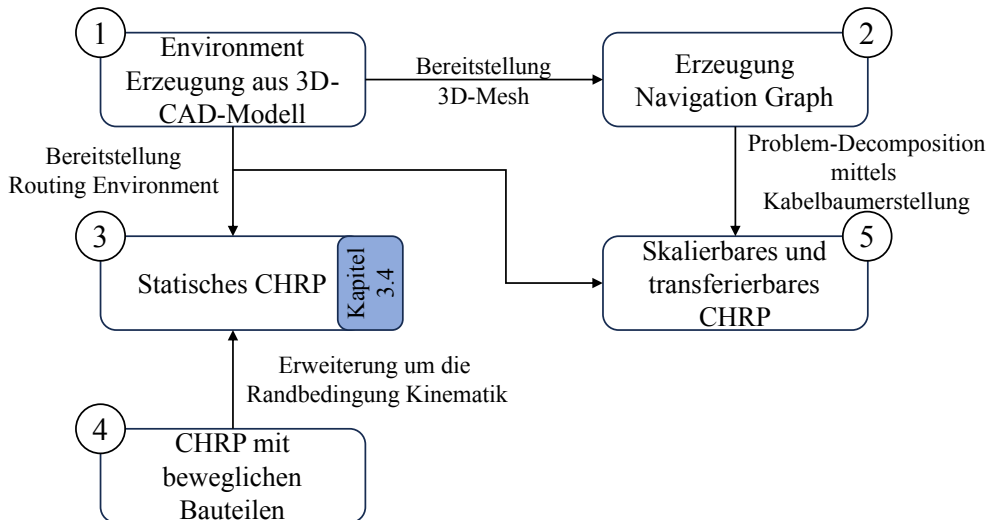


Abbildung 5.1: Überblick über die Schritte der Validierung

In diesem Kontext führt Kapitel 5.2 die beiden Validierungsfälle ein (vgl. Abbildung 5.1 (1)), die Kapitel 5.3.1 für die Validierung des Navigation-Graphs aufgreift (vgl. Abbildung 5.1 (2)). Dieser Abschnitt beabsichtigt die Identifikation der optimalen Verzweigungspunkte und stellt gleichzeitig die datentechnischen Grundlagen für die Validierungen zur Kabelbaumerzeugung dar. Als Bewertungskriterium wird die minimale Gesamtkabellänge $\sum_i^N L(\tau^i)$ festgelegt (vgl. Kapitel 4.1.4) und im Kontext unterschiedlicher Werte von w_B verglichen. Die davon entkoppelten Experimente in statischen 3D-CAD-Geometrien (vgl. Kapitel 5.3.3 und Abbildung 5.1 (3)) untersuchen die Einsetzbarkeit der neuen Methode und deren Vor- und Nachteile gegenüber Methoden des derzeitigen Stands der Technik (vgl. Kapitel 3.2). Ein spezifisches Experiment analysiert die Übertragbarkeit auf bewegliche 3D-CAD-Baugruppen unter Anwendung der Methoden aus Kapitel 4.2.4 (vgl. Abbildung 5.1 (4)). Für diesen Teilbereich wird ebenfalls die minimale Gesamtkabellänge als Bewertungskriterium herangezogen. Im Gegensatz zu den Validierungen der Navigation-Graph-Erzeugung in Kapitel 5.3.1 müssen hier die Randbedingungen z. B. in Form von kabelspezifischen minimalen Biegeradien (vgl. Kapitel 2.3.3) für ein valides Gesamtergebnis berücksichtigt werden. Im Kapitel zur Sicherstellung der Skalierbarkeit der Methode (vgl. Kapitel 5.3.5) stehen erneut die statischen 3D-CAD-Modelle im Mittelpunkt. Mittels Weiterverwendung der Verzweigungspunkte aus Kapitel 5.3.1 löst die neue MARL-Methode ein modifiziertes CHRP im Sinne einer *Sub-Problem-Decomposition*. Im abschließenden Kapitel 5.3.6 werden in Validierungsbeispielen die Metriken um die Quantifizierung des TL erweitert (vgl. Kapitel 2.6.5). In den zugehörigen Experimenten steht die Übertragbarkeit des generierten Wissens im Fokus (vgl. Abbildung 5.1 (5)). Zu diesem Zweck erfolgt in den Experimenten eine Randomisierung der Anschlusspunkte

(vgl. Kapitel 4.5.2). Das nachgelagerte Experiment zeigt dann im Zero-Shot-Transfer, wie die entstandene Policy π auf adaptierte CHRP angewendet werden kann. In den einzelnen Experimenten kommt die Quantifizierung des Rewards, der Anzahl der verursachten Kollisionen, der resultierenden Gesamtkabellänge und der Anzahl der vollständigen Kabelpfade zum Einsatz. Die Ergebnisse werden je Validierungsfall unter Berücksichtigung der initial formulierten Zielsetzung und der Forschungshypothesen analysiert (vgl. Kapitel 1.3).

5.2 Definition der Validierungsszenarien und Überführung in Environments

Zur Validierung der entwickelten Methoden sowie zum Vergleich mit den bestehenden Ansätzen aus dem derzeitigen Stand der Technik (vgl. Kapitel 3.2) kommen zwei unterschiedliche 3D-CAD-Geometrien mit zugehörigen Verbindungslisten der Kabel zum Einsatz. Für beide besteht die Prämisse, dass alle Anforderungen aus Kapitel 4.1.5 erfüllt werden. Beim ersten Validierungsszenario handelt es sich um die mechanische Konstruktion einer Autotür (nachfolgend *Validierungsfall 1* genannt). Abbildung 5.2 zeigt diese aus zwei Perspektiven. Die Tür enthält neben der Schweißbaugruppe auch Dichtungen, Fensterhebermotoren sowie das Verbindungsstück zur Fahrzeugkarosserie. Das darin enthaltene CHRP hat das Ziel einer Verkabelung der enthaltenen elektrischen 3D-CAD-Komponenten und frei positionierten Stecker.

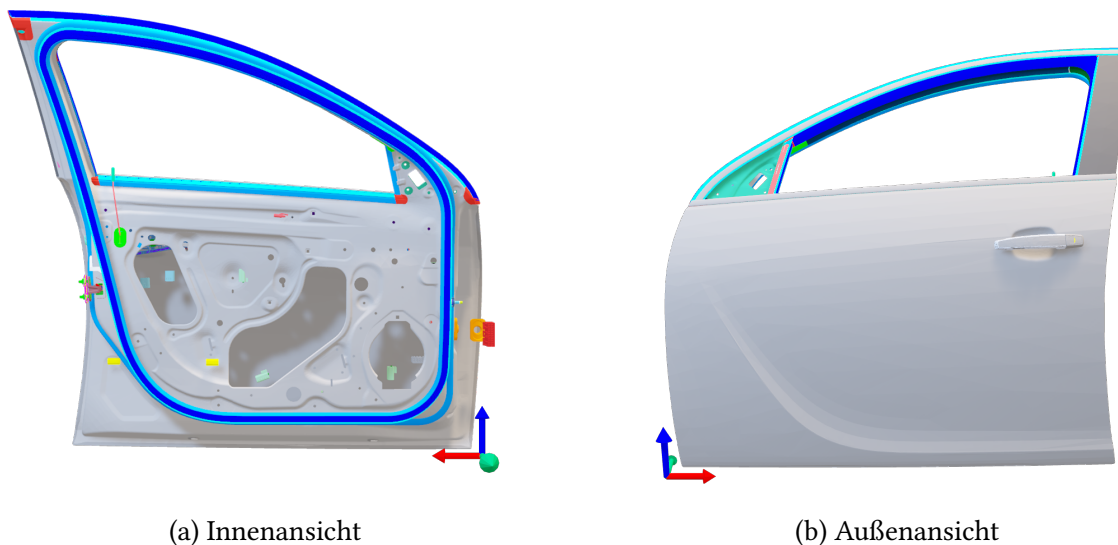


Abbildung 5.2: Validierungsfall zur Verkabelung einer Fahrzeugtür

Die Innenansicht (vgl. Abbildung 5.2a) des *Validierungsfalls 1* bildet insbesondere die Innengeometrie der Tür ab und verdeutlicht die hohe Anzahl an Freiformen und dünnwandigen Kollisionsgeometrien. Ergänzend dazu ist in Abbildung 5.2b die Außenansicht dargestellt. Tabelle 5.1 listet die enthaltenen Anschlusspunkte der elektrischen Komponenten in Form ihrer 3D-CAD-Geometrie auf, welche die Grundlage für das zu lösende CHRP bilden.

Tabelle 5.1: Elektrische Anschlusspunkte der Komponenten in der Validierungsgeometrie
1

Anschluss-ID	Betriebsmittel- kennziffer	Komponente	Anschluss- pin	X-Koord.	Y-Koord.	Z-Koord.
1	-P74	V_LOCKOUT	3	3040.78	-831.06	693.04
2	-ILC1	C-70908	8	2213.06	-727.45	535.82
3	-P68	C-71332	4	2692.32	-852.39	712.03
4	-P64	C-81330	1	2369.46	-832.40	453.0
5	-P72	DOOR_LOCKOUT_PS	A	2907.08	-768.77	456.45
6	-P69	C-81340	6	2656.42	-836.47	392.06
7	-P76	LP-COURTESY_DR	2	3115.54	-888.50	671.77
8	-P67	FUEL_LID	2	2599.17	-837.06	729.44
9	-P77	C-81333	9	3109.90	-860.02	446.37
10	-P70	DOOR_LOCK_DR	4	2775.83	-846.58	411.75
11	-P65	C-61276	7	2479.90	-812.31	669.0
12	-P75	DOOR_LOCK_DRV	2	3192.05	-841.86	675.74
13	-P73	W_MOTOR_DRV	3	2959.34	-816.05	674.15
14	-ILC2	C-70718	B	2344.55	-773.75	681.69
15	-GR2	GR-3543	A	2265.78	-862.08	534.67
16	-P71	C-81329	5	2845.41	-803.99	665.17

Die elektrischen Anschlusspunkte der Komponenten in Tabelle 5.1 stellen gleichzeitig die datentechnische Beschreibung für die elektrische Verbindungsliste dar. Der Ausdruck $[1 \mapsto 2]$ repräsentiert im Folgenden die verkürzte Schreibweise für eine einfache elektrische Verbindung zwischen den Anschlusspunkten der ID 1 und ID 2. Der Ausdruck $[1 \mapsto 3, 4]$ zeigt hingegen eine verzweigte Verbindung ausgehend von Anschluss-ID 1 mit den Zielen ID 3 und ID 4. Die Verbindungsliste ist dementsprechend wie folgt festgelegt: $a_1 = [12 \mapsto 7]$, $a_2 = [7 \mapsto 1]$, $a_3 = [1 \mapsto 13]$, $a_4 = [13 \mapsto 16]$, $a_5 = [16 \mapsto 3]$, $a_6 = [3 \mapsto 8]$, $a_7 = [8 \mapsto 11]$, $a_8 = [11 \mapsto 14]$, $a_9 = [4 \mapsto 6]$, $a_{10} = [6 \mapsto 10]$, $a_{11} = [10 \mapsto 5]$, $a_{12} = [5 \mapsto 9]$, $a_{13} = [9 \mapsto 12]$.

Für eine Verarbeitung und eine Nutzung als Environment im Sinne des POMG ist eine datentechnische Weiterverarbeitung notwendig (vgl. Abbildung 4.11). Sie umfasst neben der Überführung der 3D-CAD-Geometrie in eine Mesh-basierte Darstellung zur Nutzung der Kabelbaummethode (vgl. Kapitel 4.5.1) auch die Erzeugung einer kollisionsorientierten Environment-Repräsentanz in Form eines CoACD-Objektes (vgl. Kapitel 4.3.1). Abbildung 5.3 stellt die beiden Betrachtungen gegenüber und ergänzt diese um eine Überführung der 3D-CAD-Geometrie in eine Voxel-Darstellung (vgl. Kapitel 2.2.3).

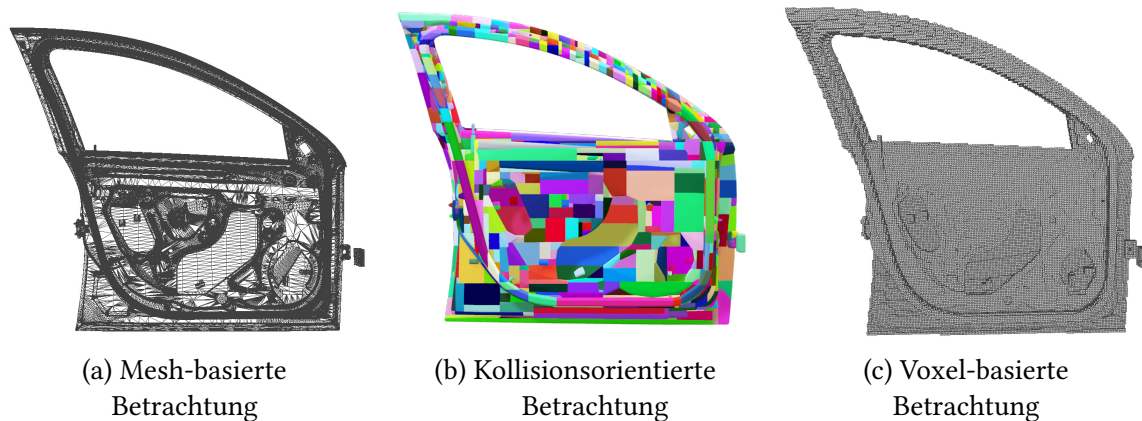


Abbildung 5.3: Environment einer Fahrzeigtür

Abbildung 5.3a zeigt die tesselierte Mesh-Betrachtung für den *Validierungsfall 1*. Darin ist zu erkennen, dass die geometrische Beschaffenheit der 3D-CAD-Geometrie grundlegend erhalten bleibt, wenngleich für eine adäquate Abbildung ein hoher Genauigkeitsgrad erforderlich ist (vgl. Abbildung 2.6). Diese Anforderung gilt insbesondere für die abgerundeten Flächen und Bohrungen (vgl. Kapitel 2.2.3). Die Kollisionsgeometrie in Abbildung 5.3b berücksichtigt die Anforderungen zur exakten und effizienten Identifikation von Kollisionen zwischen dem Kabel und dem mechanischen 3D-CAD-Modell (vgl. Abbildung 4.11b). Die angewendete CoACD-Methode erzeugt aus der 3D-CAD-Geometrie eine approximative konvexe Zerlegung und ermöglicht somit die Anwendung effizienter Algorithmen zur Geometrieverarbeitung und Kollisionsberechnung (vgl. Kapitel 3.2.3). Die Überführung der Validierungsgeometrie in die Darstellungsform eines Voxel-Raumes (vgl. Abbildung 5.3c und Kapitel 2.2.3) zeigt, dass trotz der gewählten Seitenlänge von 2 mm wesentliche geometrische Merkmale der Beispielgeometrie verloren gehen. Es ergibt sich hier eine Voxel-Anzahl von ca. 63 Millionen Datenpunkten für die Repräsentanz des Festkörpers aus den mit Binärwerten belegten Positionen innerhalb der Bounding Box der 3D-CAD-Geometrie. Daraus ist abzuleiten, dass für diese datentechnische Repräsentanz kein kollisionsfreier Kabelpfad mehr gewährleistet werden kann. Insbesondere bei der Überführung der diskreten Wegstreckenpunkte in das 3D-CAD-System (vgl. Kapitel 4.2.1) kann dies zu unzulässigen Verdrahtungsergebnissen führen. Diese Ergebnisse widersprechen nicht nur den gesetzten Randbedingungen in Kapitel 2.3.3, sondern auch der Nebenbedingung $P^i(t) \neq P^j(t), \forall i \neq j, \forall t \in T$ des Optimierungsziels aus Kapitel 4.1.4. Das vorliegende Ergebnis dieser Diskretisierung (vgl. Abbildung 5.3c) führt zu der Schlussfolgerung, dass eine Umgebungsdarstellung mit der Voxel-Methode nicht hinreichend geeignet ist, um beliebige 3D-CAD-Geometrien mit ausreichender Genauigkeit abzubilden.

Die mechanische 3D-CAD-Konstruktion des Wagenkastens eines Zuges repräsentiert den zweiten Validierungsfall (nachfolgend *Validierungsfall 2* genannt). Dieser zeigt eine Vereinfachung eines industriellen Anwendungsfalls mit großen geometrischen Dimensionen. Darin enthalten sind neben den Aussparungen in den Seitenwänden für die Türen und Fenster auch Rohr-, Kabel- und Luftkanäle in Form einer Blechkonstruktion. Abbildung 5.4 zeigt die 3D-CAD-Geometrie des *Validierungsfalls 2* aus zwei Perspektiven.

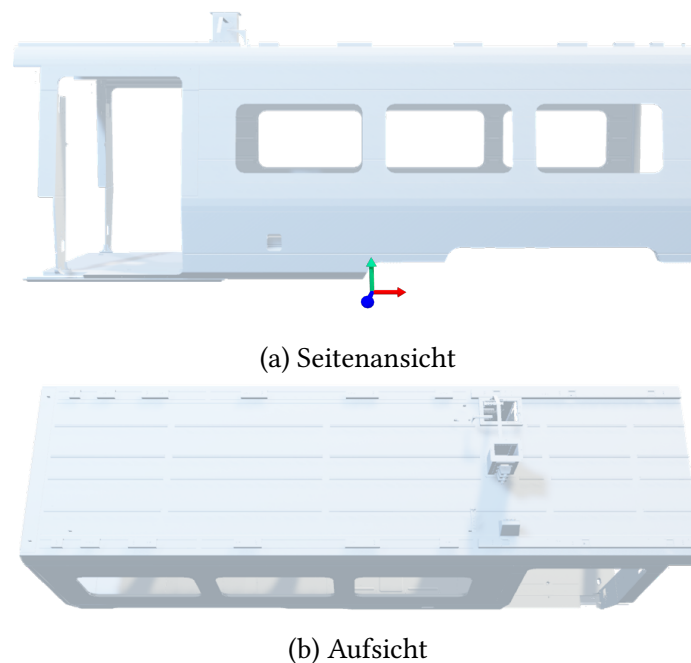


Abbildung 5.4: 3D-CAD-Darstellung des Validierungsfalls 2 für einen Wagenkasten eines Zuges

In Abbildung 5.4a ist die Seitenansicht des 3D-CAD-Modells für *Validierungsfall 2* dargestellt, die vor allem die Aussparungen für Fenster und Türen zeigt. In der Aufsicht in Abbildung 5.4b liegt der Fokus hingegen auf der Dachfläche und den Übergangsstellen zum Wageninnenraum. Diese sind insbesondere in der Verarbeitung der Verbindungsliste als Kabelbaum ein wichtiger Knotenpunkt für elektrische Verbindungen mit Start- und Zielpunkten in unterschiedlichen Ebenen des Wagenkastens. Die elektrische Verbindungsliste zeichnet sich vor allem durch Verbindungen über lange euklidische Distanzen aus und inkludiert in Summe 300 Einzelverbindungen zwischen den elektrischen Anschlusspunkten (vgl. Tabelle 5.2). Aufgrund des daraus resultierenden Umfangs wird die Liste aller Anschlusspunkte und Verbindungen innerhalb dieser Arbeit nicht vollständig abgebildet. Ein Auszug der Geräte und ihrer elektrischen Anschlusspunkte ist in Tabelle 5.2 abgebildet.

Tabelle 5.2: Auszug der elektrischen Geräte und ihrer Anschlusspunkte für den Validierungsfall 2

Anschluss-ID	Betriebsmittelkennziffer	Anschlusspin	X-Koord.	Y-Koord.	Z-Koord.
1	-UD-X405.B	1.1	-2241.61	3823.32	-281.34
2	-UD-X223	1.4	-2241.53	3583.59	-283.08
3	-UD-X408.A	2.3	-2569.49	1523.97	-1244.35
4	-UD-X409.A	2.1	-2554.79	1995.54	-1213.27
5	-FD-T10-X01.A	1.4	-106.68	2584.79	-1226.00
...

Analog zu *Validierungsfall 1* ist eine Vorverarbeitung des 3D-CAD-Modells aus den Eingangsdaten (vgl. Kapitel 4.1.2) erforderlich, um dieses als Environment in einem POMG

einsetzen zu können (vgl. Kapitel 4.3.2). Abbildung 5.5 zeigt neben der Mesh-basierten Darstellung auch das Kollisionsobjekt als CoACD-Objekt aus zwei Perspektiven.

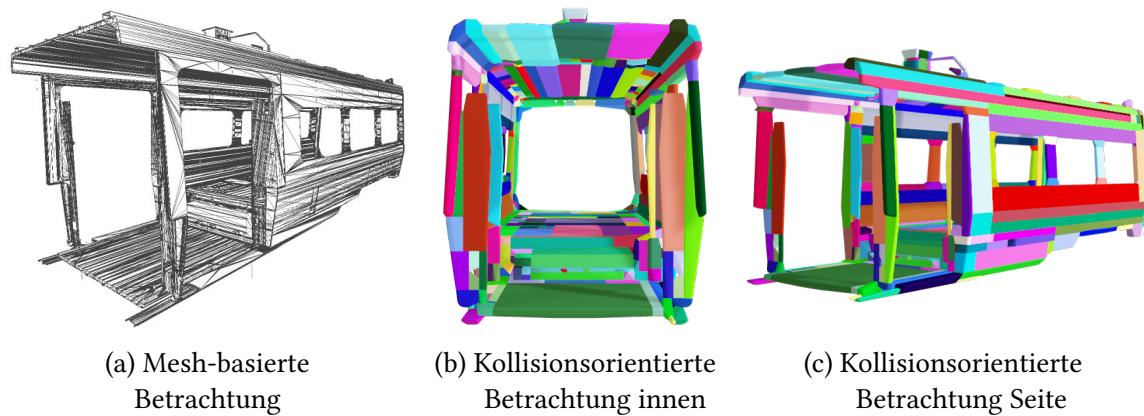


Abbildung 5.5: Environment eines Wagenkastens

Eine Mesh-basierte Darstellungsform der 3D-CAD-Geometrie von *Validierungsfall 2* ist in Abbildung 5.5a zu sehen. Diese bildet die datentechnische Grundlage für das Environment zur Erzeugung des Kabelbaumes und behält alle erforderlichen geometrischen Features in ausreichender Genauigkeit bei (vgl. Abbildung 2.6). Für eine exakte und gleichzeitig recheneffiziente Identifikation von Kollisionen zeigen die Abbildungen 5.5b und 5.5c zwei unterschiedliche Perspektiven des CoACD-Objektes der 3D-CAD-Geometrie (vgl. Kapitel 3.2.3).

Für die durchzuführenden Experimente werden in beiden Validierungsfällen die Mesh-basierte Betrachtungsweise und das Kollisionsobjekt in Form eines CoACD-Objektes verwendet. Diese datentechnischen Repräsentanzen behalten im Gegensatz zu einer Voxel-Darstellung die ursprüngliche 3D-CAD-Geometrie in ausreichender Genauigkeit bei. Gleichzeitig ermöglichen sie durch die graphgestützte Darstellung eine zeiteffiziente Erzeugung von Kabelpfaden (vgl. Kapitel 2.1) und können mit performanten Methoden Kollisionen der elektrischen Verbindungen detektieren (vgl. Kapitel 2.1.2).

5.3 Experimente

Für die in Kapitel 4 entwickelten Methoden zur Verarbeitung der 3D-CAD-Geometrie und zur Lösung des CHRP werden nachfolgend unterschiedliche CHRP im 3D-CAD-System gelöst. Kapitel 5.3.1 zeigt auf, wie innerhalb der Mesh-Darstellung des Environments (vgl. Kapitel 4.3.1) ein Navigation-Graph erzeugt wird. Das nachfolgende Kapitel 5.3.3 dient der Validierung der entwickelten Methoden in statischen 3D-CAD-Geometrien. Die in Kapitel 5.3.4 durchgeführten Experimente erweitern diesen Anwendungsbereich, indem sie den Einfluss der Kinematik innerhalb der 3D-CAD-Geometrie auf das POMG untersuchen. Die Möglichkeiten zur Skalierung der Methoden im Zusammenhang mit einer Kabelbaumerstellung sind Inhalt des Kapitels 5.3.5. Das Kapitel 5.3.6 stellt die Ergebnisse zur Transferierbarkeit der Methoden auf adaptierte CHRP dar.

5.3.1 Erzeugung eines Navigation-Graphs

Die Erzeugung einer Subgraphstruktur in einem Mesh der 3D-CAD-Geometrie bildet die Grundlage für die Skalierbarkeit der entwickelten Methode (vgl. Kapitel 2.1.4 und 4.5.1). Auf Basis der tessellierten Darstellungsform werden im Folgenden recheneffiziente Pfadsuchalgorithmen in Graphen eingesetzt, um die optimalen Verzweigungspunkte der einzelnen Branches zu identifizieren (vgl. Kapitel 3.2). Die Herangehensweise zielt darauf ab, insbesondere in CHRP mit einer hohen Anzahl von elektrischen Verbindungen die notwendige Anzahl der Agenten zu reduzieren (vgl. Kapitel 2.5.2). Zusätzlich kann über den Einsatz der entwickelten Methode die domänenspezifische Planung in Form von Kabelbäumen untersucht werden (vgl. Kapitel 2.3.5). Somit fließt neben der entstehenden Gesamtpfadlänge auch die resultierende Anzahl der Agenten in die Bewertung der Validierungsergebnisse ein.

Abbildung 5.6 zeigt für den *Validierungsfall 1* (vgl. Kapitel 5.2) die Integration der erforderlichen Edges in einem vorgelagerten Schritt (vgl. Kapitel 4.5.1.1). Damit sind die Anschlusspunkte auch im Falle einer Position außerhalb bestehender Edges mit dem Gesamtgraphen verbunden. Die Abbildungen 5.6a und 5.6b stellen dabei den Einfluss des Faktors w_B (vgl. Kapitel 4.5.1.1) auf die Kabelbaummethode gegenüber.

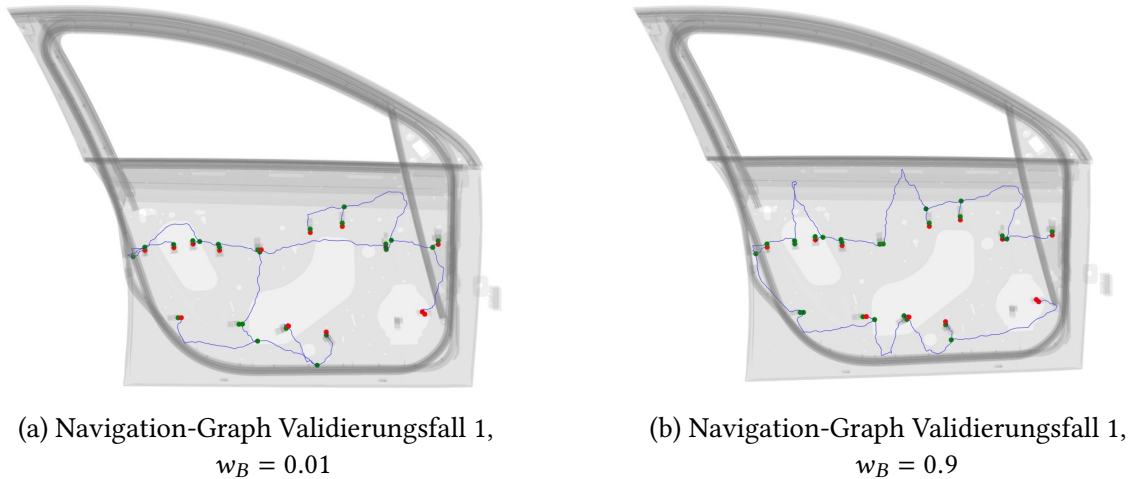


Abbildung 5.6: Gegenüberstellung von Navigation-Graphen in Validierungsfall 1 mit unterschiedlichen w_B

Die Abbildungen 5.6a und 5.6b zeigen, dass sich mit Hilfe der Edge-Erweiterung (vgl. Kapitel 4.5.1) die Anschlusspunkte aus der Tabelle 5.1 zum Gesamtgraphen $G'_\alpha(V, E)$ hinzufügen lassen. Dadurch können einzelne elektrische Anschlusspunkte innerhalb des Graphen verbunden werden. Während Abbildung 5.6a veranschaulicht, dass ein niedriger Wert der Variable w_B zu einer starken Bündelung führt, zeigt Abbildung 5.6b ein konträres Verhalten bei einem hohen Wert für w_B . Basierend auf der entwickelten Methode zur Anpassung der Kantengewichte für eine verstärkte Kabelbündelung (vgl. Kapitel 4.5.1.1) kommt dem initial identifizierten Pfad eine entscheidende Rolle zu. Abbildung 5.6a demonstriert in der exemplarischen Kabelbaumgenerierung, dass die elektrische Verbindung $1 \mapsto 13$ mit ihrem initial definierten kürzesten Pfad das allgemeine Verdrachtungsergebnis prägt. Aufgrund dieses hohen Einflusses auf das Gesamtergebnis wird im Rahmen dieser Validierung die elektrische Verbindung als erste erzeugt, deren Anschlusspunkte den kleinsten euklidischen Abstand aufweisen (vgl. Kapitel 4.5.1.1). Ein vergleichbares Resultat ist in Abbildung 5.6b nicht zu erkennen, da hier die Modifikation der Kantengewichte weniger Relevanz hat. In beiden Szenarien (vgl. Abbildung 5.6a und 5.6b) ist jedoch eindeutig erkennbar, dass für beide w_B -Werte ein valides Ergebnis für die Verkabelung erzeugt werden kann, das alle elektrischen Verbindungen von *Validierungsfall 1* inkludiert (vgl. Tabelle 5.1). Die sich daraus ergebenden, aufsummierten Kantengewichte erlauben eine erste Einschätzung über die notwendigen Verbindungslängen. Tabelle 5.3 stellt die jeweiligen Längen der elektrischen Verbindungen in den Subgraphen (Kabelbäume) numerisch gegenüber (vgl. Kapitel 2.1.4). Zur Einordnung der Ergebnisse werden zudem die Resultate einer Probabilistic Roadmap (PRM) ergänzt. Die Werte der Spalte PRM entsprechen dem Durchschnitt von drei Validierungsläufen mit einem Punkt pro Fläche und 20 verbundenen Nachbarschaftspunkten (vgl. Kapitel A.4).

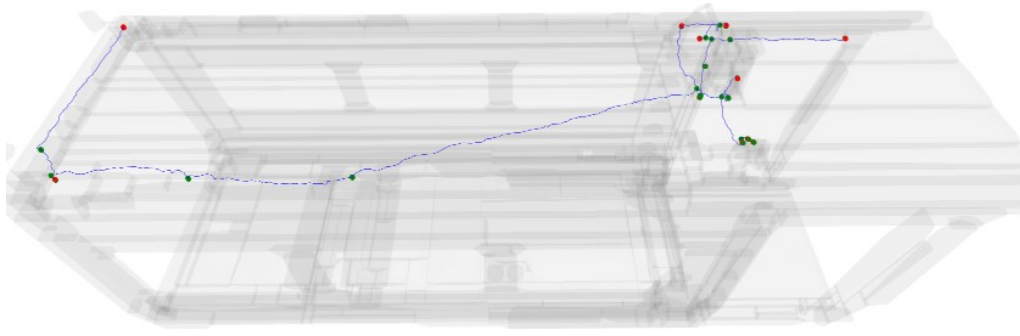
Tabelle 5.3: Gegenüberstellung der Verbindungslängen im Navigation-Graph für den Validierungsfall 1

a^i	$w_B = 0.01$	$w_B = 0.9$	PRM	a^i	$w_B = 0.01$	$w_B = 0.9$	PRM
$12 \mapsto 7$	290.18	290.18	114.49	$11 \mapsto 14$	231.96	216.76	221.32
$7 \mapsto 1$	637.96	536.01	238.31	$4 \mapsto 6$	1517.80	508.42	500.73
$1 \mapsto 13$	134.03	134.03	223.08	$6 \mapsto 10$	408.91	408.91	664.07
$13 \mapsto 16$	168.65	168.65	564.08	$10 \mapsto 5$	558.84	442.21	393.83
$16 \mapsto 3$	981.15	559.48	132.13	$5 \mapsto 9$	436.88	390.99	1172.02
$3 \mapsto 8$	273.45	273.45	362.84	$9 \mapsto 12$	1270.87	335.38	1576.04
$8 \mapsto 11$	539.70	539.70	423.42	$\sum L(\tau^i)$	7450.38	4804.17	6577.42

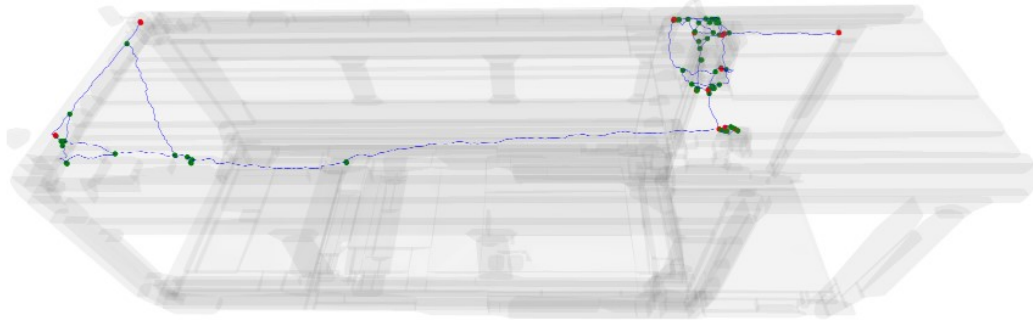
Die in Millimetern angegebenen Längen der einzelnen exemplarischen Verbindungen in Tabelle 5.3 verdeutlichen den Zielkonflikt der Kabelbündelung (vgl. Kapitel 2.3.5). So weist die Konfiguration mit $w_B = 0.01$ zwar einen höheren Grad der Bündelung auf, jedoch auch eine um 45.08% größere Gesamtpfadlänge $\sum_i^N L(\tau^i)$ (vgl. Kapitel 4.1.4). Besonders hervorzuheben sind die Verbindungen $4 \mapsto 6$ und $9 \mapsto 12$, da hier die Divergenz zwischen den Ergebnissen bei der Anwendung unterschiedlicher Bündelungsfaktoren w_B besonders groß ist. Andere Verbindungen weichen in ihrer Pfadlänge nur minimal voneinander ab ($11 \mapsto 14$) oder haben sogar die identische Länge ($6 \mapsto 10$) (vgl. Tabelle 5.3). Verglichen mit den Ergebnissen der PRM sind vor allem die Verbindungen $16 \mapsto 3$ und $5 \mapsto 9$ hervorzuheben, da hier die resultierenden Leitungslängen deutlich von der neuen Methode abweichen. Die Gesamtlänge der Einzelverbindungen ist mit einem Wert von 6577.42mm zwischen den Resultaten der Experimente mit $w_B = 0.01$ und $w_B = 0.9$ einzuordnen.

Als Resultat stellt die Bündelung neben der Kabellänge auch die Positionen der Verzweigungspunkte P_{Tb} bereit. Diese dienen als Grundlage zur Skalierung der neuen Methode (vgl. Kapitel 4.5.1) und fließen somit in die Validierungsschritte in Kapitel 5.3.5 ein. Im Rahmen der Anforderung einer hohen Bündelung ist das Ergebnis dieses Vorverarbeitungsschrittes unter Berücksichtigung der bestehenden Randbedingungen zu bewerten (vgl. Kapitel 2.3.3).

Eine analoge Durchführung der Navigation-Graph-Erzeugung für *Validierungsfall 2* ist in Abbildung 5.7 dargestellt. Hier ist die Validierung der neuen Methode auf 3D-CAD-Geometrien mit großen Abmessungen entscheidend. Gleichmaßen wie in *Validierungsfall 1* werden die Werte $w_B = 0.01$ (vgl. Abbildung 5.7a) und $w_B = 0.9$ (vgl. Abbildung 5.7b) für den Bündelungsfaktor w_B verwendet.



(a) Navigation-Graph Validierungsfall 2,
 $w_B = 0.01$



(b) Navigation-Graph Validierungsfall 2,
 $w_B = 0.9$

Abbildung 5.7: Gegenüberstellung von zwei Navigation-Graphs für große Geometrien am Beispiel von Validierungsfall 2

Auch in diesem Anwendungsbeispiel ist der Effekt der Änderung des Bündelungsfaktors w_B in Abbildung 5.7 zu erkennen. Aufgrund der hohen Anzahl an elektrischen Verbindungen in der Verbindungsliste (vgl. Kapitel 4.1.2) resultiert auch die Anwendung des angepassten Bündelungsfaktors $w_B = 0.9$ (vgl. Abbildung 5.7b) ebenfalls in einem Kabelbaum. Dieses Ergebnis ist damit zu begründen, dass ab einer bestimmten Verbindungsanzahl alle möglichen Start-Zielpunkt-Kombinationen abgedeckt sind. Die zugehörigen Kantengewichte sind über die Vielzahl an Verbindungen so stark modifiziert, dass sie ebenfalls eine Bündelung hervorrufen.

Den dargestellten Ergebnissen in Abbildung 5.7a ist darüber hinaus eine Gesamtlänge des Subgraphen

($\sum_{E \in G'_\alpha(V,E)} w_E(u,v)$, vgl. Kapitel 4.5.1.1) von 15553.09 mm zu entnehmen, wogegen Abbildung 5.7b eine Subgraphenlänge von 25127.68 mm ausweist. Die resultierende Mehrlänge von 61.56% hat insbesondere Auswirkungen auf die daraus abgeleitete Problem-Decomposition aus Kapitel 4.5.1.3. Es ist an dieser Stelle anzumerken, dass analog zu *Validierungsfall 1* die Gesamtheit der individuellen Kabellängen im Szenario $w_B = 0.01$ (731526.73 mm) um 2.87% länger ist als im Szenario $w_B = 0.9$ (711087.27 mm). Ein Vergleich mit der Methode des PRM ist in *Validierungsfall 2* aufgrund der großen Anzahl an Flächen im Mesh nicht möglich (vgl. Kapitel A.4).

5.3.2 Clustering der Kabelpfade auf Grundlage von Beispielpfaden

Die in Kapitel 5.3.1 generierten Pfade für den Gewichtungsfaktor $w_B = 0.9$ können im Folgenden als Beispielpfade für das Clustering verwendet werden (vgl. Kapitel 4.4.1). Auf Grundlage dieser Pfade ist es möglich, eine Distanzmatrix zu erzeugen, die sich auf das Distanzmaß des Fréchet-Abstandes stützt (vgl. Kapitel 2.4.4).

Für *Validierungsfall 1* ergibt sich eine Distanzmatrix der Größe 13×13 , resultierend aus der Agentenanzahl $N = 13$. Tabelle 5.4 zeigt einen Ausschnitt dieser Distanzmatrix.

Tabelle 5.4: Auszug aus der Fréchet-Distanzmatrix für Validierungsfall 1

a^i	1	2	3	4	5	6	7	8	9	...
1	0.00	215.15	388.87	182.53	476.14	135.05	549.50	616.29	379.54	...
2	215.15	0.00	552.75	219.53	628.32	302.20	709.35	779.54	510.17	...
3	388.87	552.75	0.00	445.65	395.25	272.70	189.57	266.68	445.07	...
4	182.53	219.53	445.65	0.00	557.02	239.12	594.48	698.67	456.66	...
5	476.14	628.32	395.25	557.02	0.00	401.09	460.00	499.19	154.72	...
6	135.05	302.20	272.70	239.12	401.09	0.00	421.90	491.83	351.12	...
7	549.50	709.35	189.57	594.48	460.00	421.90	0.00	169.07	511.39	...
8	616.29	779.54	266.68	698.67	499.19	491.83	169.07	0.00	592.99	...
9	379.54	510.17	445.07	456.66	154.72	351.12	511.39	592.99	0.00	...
10	331.95	469.74	160.31	354.21	463.60	246.15	285.28	357.84	462.99	...
11	275.44	387.45	588.16	421.68	337.59	350.20	674.01	792.94	233.07	...
12

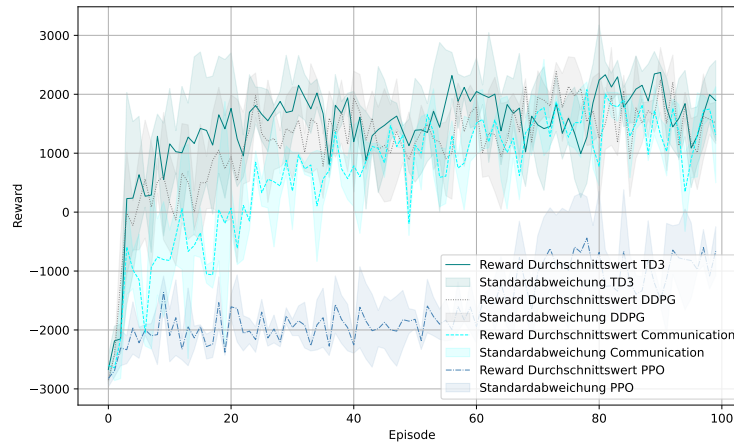
Neben der Diagonale in der Matrix weisen in Tabelle 5.4 keine weiteren Fréchet-Distanzen den Wert 0.0 auf. Dies ist darauf zurückzuführen, dass die Agenten a^i nicht in identischen bzw. nahezu identischen 3D-Koordinaten starten oder enden. Die Matrix (vgl. Tabelle 5.4) ist zugleich Eingangsgröße für die Clustering-Methode (vgl. Kapitel 4.4.1.3). Mit einer Mindestanzahl von $N \geq 3$ Agenten je Cluster C_n und einem Wert von $\epsilon = 0.05$ ergeben sich in *Validierungsfall 1* zwei unterschiedliche Cluster. Cluster C_1 inkludiert dabei die Agenten a^1, a^2, a^4, a^6 und a^{12} , wohingegen die verbleibenden Agenten Cluster C_2 zugeordnet sind. Folglich werden in der Anwendung der Communication-Methode (vgl. Kapitel 4.4.2) zwei unterschiedliche Policies π_{C_n} zeitgleich in einem gemeinsamen Environment trainiert. Zwischen beiden Policies π_{C_1} und π_{C_2} findet keine Form der Informationsübertragung statt, auch wenn beide Agentengruppierungen in der gleichen 3D-CAD-Geometrie verortet sind. Innerhalb der jeweiligen Policy π_{C_n} erfolgt, wie in Kapitel 4.4.2 beschrieben, eine Übertragung von Informationen mittels Observation O .

Eine analoge Ausführung dieser Clustering-Methode wird für *Validierungsfall 2* aufgrund des Umfangs der Verbindungen nicht vollständig in dieser Arbeit dargestellt. Die Zuordnung der 300 Einzelverbindungen führt zu einer Bildung der Cluster C_1 bis C_8 aufgrund der zahlreichen Dopplungen von Start-Ziel-Kombinationen. Outlier existieren nur in Validierungsfall 2 (vgl. Kapitel 2.4.4 und 4.4.2). Für eine bessere Vergleichbarkeit der Experimente werden diese in einem eigenen Cluster gruppiert.

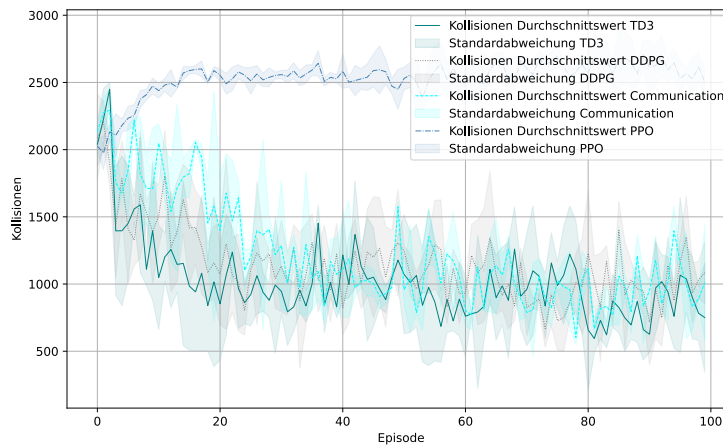
5.3.3 Standardtraining in statischen 3D-CAD-Geometrien

Die Lösung des CHRP unter Berücksichtigung der gesetzten Optimierungsziele (vgl. Kapitel 4.1.4) bildet die grundlegende Problemstellung der neuen Methoden (vgl. Kapitel 1.2). Dieser Abschnitt dient dazu, die neuen Methoden mit bestehenden Ansätzen aus dem Kapitel 3.2 für statische 3D-CAD-Geometrien gegenüberzustellen. Dafür werden die Methoden aus Kapitel 4.4.2 und 4.5 sowohl mit der Anwendung anderer lernbasierter Algorithmen als auch mit klassischen Pfadsuchalgorithmen verglichen. Tabellarisch werden in jedem Experiment die Werte für \mathcal{R}_{max} , \mathcal{R}_{min} , \mathcal{R}_{IQR} , K_{min} und OOS_{min} gegenübergestellt. Ein hervorgehobener Wert (z. B. **3046.72**) in der Tabelle zeigt den erreichten Bestwert innerhalb der Vergleichsgruppe. In den Darstellungen zur Entwicklung der Performance im Lernprozess bildet die x-Achse die Episoden Eps $Eps = 0$ bis $Eps = 100$ ab, die jeweils 250 Zeitschritte ts im POMG umfassen. Auf der y-Achse ist die jeweilige Metrik abgebildet.

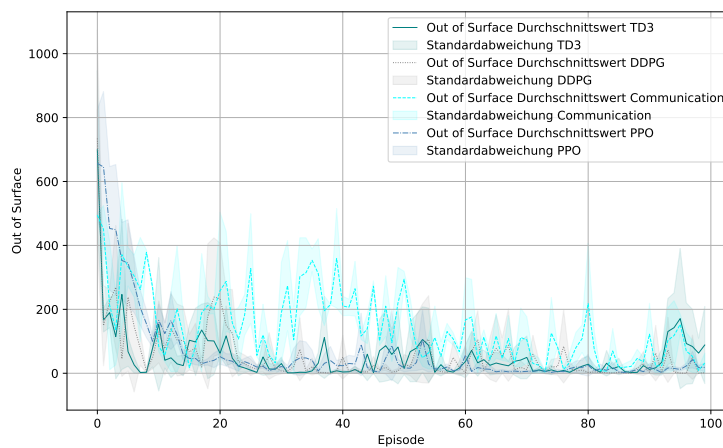
Abbildung 5.8 zeigt die quantitativen Ergebnisse des Lernprozesses der MARL-Methoden im statischen *Validierungsfall 1* ohne Kabelbaumerzeugung. Dabei werden die Algorithmen TD3, Deep Deterministic Policy Gradient (DDPG), PPO und die neue Communication-Methode (nachfolgend in Tabellen mit *COM* abgekürzt) in den Metriken *Gesamt-Reward* (vgl. Abbildung 5.8a), *Anzahl an Kollisionen mit der 3D-CAD-Geometrie* (vgl. Abbildung 5.8b) und *Überschreitung des maximalen Abstands zur Oberfläche* (vgl. Abbildung 5.8c) gegenübergestellt. Sofern möglich, erfolgt eine Gegenüberstellung mit Ergebnissen der Anwendung eines Rapidly-exploring Random Tree (RRT).



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.8: Gegenüberstellung der MARL-Metriken für Validierungsfall 1 ohne Kabelbaumerzeugung

Abbildung 5.8a ist zu entnehmen, dass alle drei Methoden innerhalb der gesetzten Episodenanzahl $Eps = 100$ eine Verbesserung ihrer Policy π im Sinne des Optimierungsziels (vgl. Kapitel 4.1.4) erreichen. Während der Algorithmus TD3 einen minimalen Reward von -2869.98 ausweist, liegt dieser bei der Verwendung des DDPG bei einem vergleichbaren Wert von -2880.18 . Die Communication-Methode (vgl. Kapitel 4.4.2) erreicht einen minimalen Reward \mathcal{R} von -2859.81 und folglich kein signifikant besseres Resultat. Auch das Ergebnis des PPO liegt in dieser Metrik bei einem Wert von 2936.74 . Ein deutlich größerer Unterschied ist beim maximalen Reward \mathcal{R} in Abbildung 5.8a zu erkennen. Hier erreicht der TD3 einen maximalen Wert von 3046.72 , wogegen der DDPG ein Ergebnis von 3004.16 erzielt. Beide Ergebnisse liegen damit über dem maximalen Reward \mathcal{R} der Communication-Methode mit 2665.85 und deutlich über dem maximalen Reward des PPO-basierten Ansatzes mit 301.45 . Ein höherer Wert ist für den Reward als vorteilhaft anzusehen. Die Streuung der Ergebnisse (hier als Interquartilsabstand (IQR) angegeben) zeigt einen stabileren Lernprozess der Communication-Methode ($IQR = 1038.30$) und des PPO ($IQR = 1006.93$) gegenüber dem DDPG ($IQR = 1983.26$) und dem TD3 ($IQR = 2018.38$). Dem Wertebereich der vier Methoden ist zu entnehmen, dass sowohl die TD3- als auch die DDPG-basierte Methode das CHRP mehrfach innerhalb einer Episode vollständig lösen können (vgl. Kapitel 4.3.2).

Für eine genauere Betrachtung des Lernverhaltens und insbesondere der Berücksichtigung von spezifizierten Randbedingungen (vgl. Kapitel 2.3.3) zeigen die Abbildungen 5.8b und 5.8c die Anzahl an Kollisionen und die Häufigkeit der Überschreitungen des maximalen Oberflächenabstandes. Konträr zum Gesamt-Reward (vgl. Abbildung 5.8a) ist hier ein minimaler Wert erstrebenswert. In der Metrik der Kollisionen kann der TD3 eine minimale Anzahl von 239 erreichen, wogegen der Wert beim DDPG mit 284 um 18.82% schlechter ist. Das Ergebnis der Communication-Methode kann mit einem Minimalwert von 296 diesen nicht reduzieren, was nicht auf die stärkere Adaptierung der Policies π_C je Cluster schließen lässt. Der Wert von K_{min} liegt für die Experimente auf Grundlage des PPO bei 1800. Die IQR-Werte von 494.25 (TD3), 461.0 (DDPG), 166.5 (PPO) und 499.25 (Communication-Methode) zeigen vergleichbar große Streuungen der Resultate. Aufgrund des Einflusses der Kollisionen auf den Reward (vgl. Kapitel 4.3.2) ist eine Korrelation der Ergebnisse in den Abbildungen 5.8b und 5.8a zu erkennen. Die Anzahl der Aktionen, die zu einer Überschreitung der maximalen Oberflächendistanz führen (vgl. Abbildung 5.8c), liegt für die analysierten Algorithmen DDPG und TD3 in einem Wertebereich von 0 bis 1831 mit vergleichbaren IQR-Werten von 940.5 für TD3 und 927.5 DDPG. Hier weist die neue Communication-Methode (vgl. Kapitel 4.4.2) ein etwas stabileres Resultat von $IQR = 166.75$ bei einer maximalen Oberflächendistanzüberschreitung von 596 auf. Die numerischen Ergebnisse sind in Tabelle 5.5 zusammengefasst.

Tabelle 5.5: Numerische Auswertung für ein Standardtraining in Validierungsfall 1

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[%] \mathcal{R}_{max}	K_{min}	[%] K_{min}	OOS_{min}	[%] OOS_{min}
TD3	3046.72	-2869.98	2018.38	-	239.0	-	0.0	-
DDPG	3004.16	-2880.18	1983.26	1.39	284.0	18.82	0.0	-
COM	2665.85	-2859.81	1038.30	12.5	296.0	23.84	0.0	-
PPO	301.45	-2936.74	1006.93	90.10	1800	653.13	0.0	-

Für eine Einordnung der Ergebnisse der Experimente werden diese den Resultaten von fünf Durchläufen eines RRT-Algorithmus gegenübergestellt. Auch dieser kann innerhalb der gesetzten Limitation von 10.000 Iterationen je Einzelverbindung nicht alle Pfade der 13 Agenten identifizieren. Im Durchschnitt sind hierbei sechs der Verbindungen vollständig festgelegt. Werden die getätigten Aktionen in das RL-Environment übertragen, lässt sich ein theoretischer Reward von 642.62 ermitteln. Abbildung 5.9 zeigt die Übertragung der besten identifizierten Pfade in das 3D-CAD-System. Zur besseren Veranschaulichung der Resultate wird die automatische Verbindung des letzten Wegstreckenpunktes der MARL-Methoden P_t^i zum jeweiligen Zielpunkt P_T^i nicht dargestellt. Darüber hinaus sind Bestandteile der 3D-CAD-Geometrie in Abbildung 5.9 ausgeblendet, da die Kabelpfade innerhalb der Geometrie verlaufen und andernfalls nicht sichtbar sind. Zur Verbesserung der Sichtbarkeit werden zudem die Kabel schwarz eingefärbt, um sie von der 3D-CAD-Geometrie abzuheben.

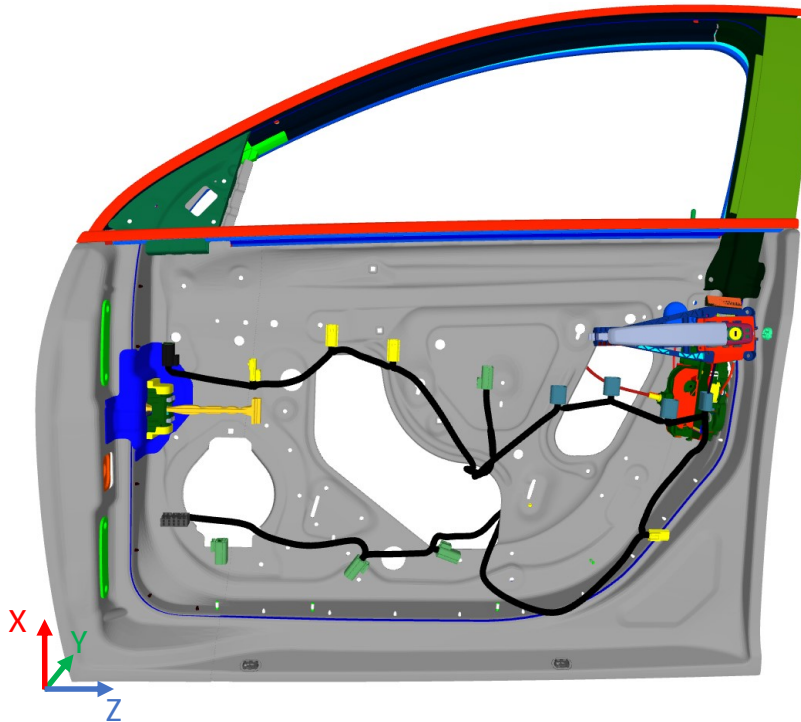
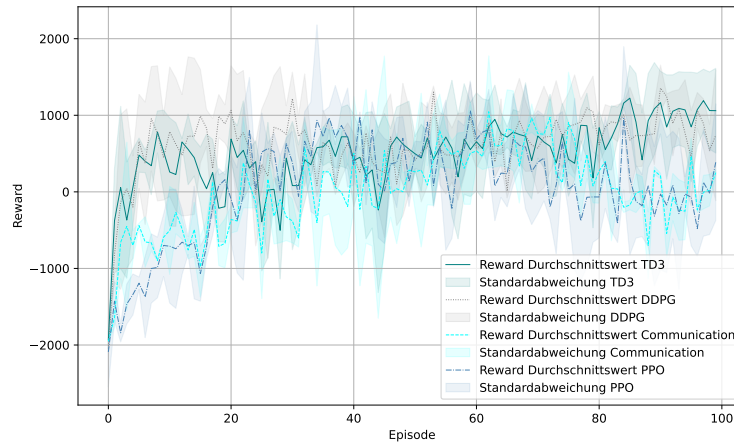


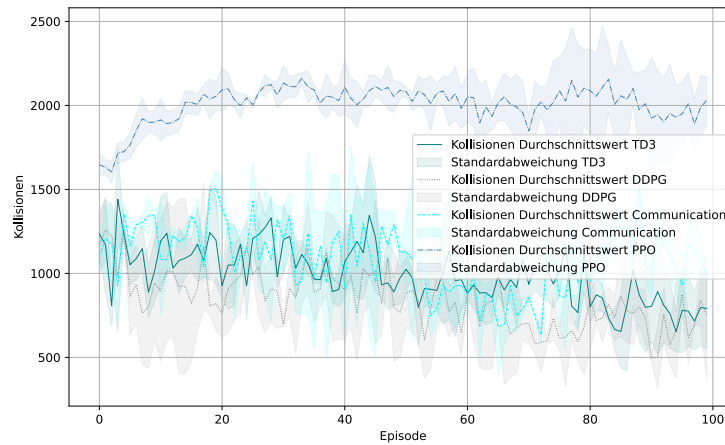
Abbildung 5.9: Darstellung der besten Kabelpfade im 3D-CAD-System für Validierungsfall 1 ohne Kabelbaum

Abbildung 5.9 verdeutlicht, dass die entwickelten Methoden valide Lösungen für das CHRP generieren. Ungeachtet der erheblichen Diskrepanzen in den maximal erzielten Rewards \mathcal{R}_{max} zeigen die Agenten vergleichbare Verhaltensmuster, die auf ihren Policies π basieren. Neben den numerischen Resultaten in Tabelle 5.5 ist auch den visuellen Ergebnissen zu entnehmen, dass die Problem-Decomposition mittels Clustering keinen Vorteil in *Validierungsfall 1* ohne Kabelbaumerstellung mit sich bringt. Vielmehr spiegelt die visuelle Darstellung den geringeren numerischen Wert für \mathcal{R}_{max} wider. Zudem ist ersichtlich, dass die entwickelten Methoden das vorliegende CHRP in keiner der 100 Episoden vollständig lösen können.

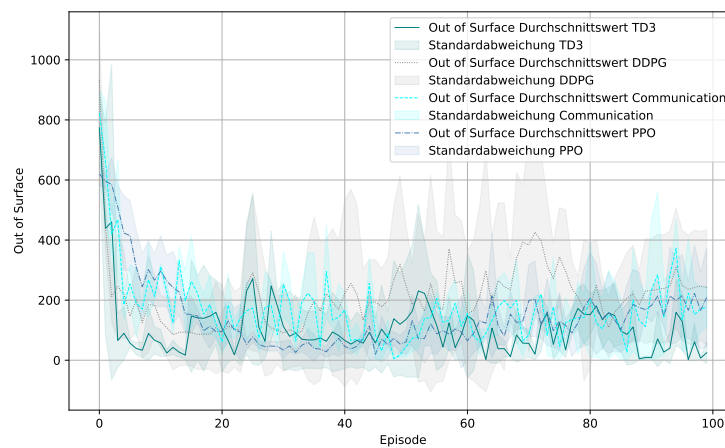
Abbildung 5.10a zeigt die Ergebnisse der Methodenanwendung für *Validierungsfall 2*. Analog zu Abbildung 5.8 ist sie untergliedert in die Abbildungen zum Gesamt-Reward je Episode (vgl. Abbildung 5.10a), die Anzahl an Kollisionen mit der 3D-CAD-Geometrie (vgl. Abbildung 5.10b) und die Überschreitung des maximalen Abstands zur Oberfläche (vgl. Abbildung 5.10c). Auch hier werden die Ergebnisse des TD3 mit denen eines DDPG in einem Trainingsprozess über 100 Episoden mit je 250ts verglichen.



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.10: Gegenüberstellung der MARL-Metriken für Validierungsfall 2 ohne Kabelbaumerzeugung

Im Vergleich zu *Validierungsfall 1* zeigen die Ergebnisse ein verändertes Lernverhalten auf. In der Metrik des Gesamt-Rewards pro Episode (vgl. Abbildung 5.10a) lässt sich ein weniger steiler, jedoch gleichbleibend positiver Anstieg der Ergebnisse beobachten. In den durchschnittlichen Werten erbringen die Methoden DDPG, TD3 und PPO eine vergleichbare Performance. Der maximale Reward \mathcal{R} des DDPG liegt bei 1814.36 und damit 23.84% unter dem des PPO mit 2382.50. Darüber hinaus weist der TD3 in seinen Ergebnissen mit einem minimalen Reward von -2053.38 und einem IQR-Wert von 911.81 etwas bessere Resultate als der DDPG ($\mathcal{R} = -2082.06$, $IQR = 978.47$) zu Beginn des Trainingsverlaufs aus. Die Communication-Methode erreicht innerhalb der 100 Episoden einen maximalen Reward \mathcal{R} von 1775.79 und bleibt damit deutlich unterhalb der Performance der anderen Methoden. Dem steht ein minimaler Reward von -1955.59 und ein IQR-Wert von 943.37 gegenüber, was die etwas konstanteren Ergebnisse in Abbildung 5.10a widerspiegelt.

Insbesondere die verbesserten Werte in den Metriken der Kollisionsanzahl (vgl. Abbildung 5.10b) und der Überschreitung des maximalen Abstands zur Oberfläche (vgl. Abbildung 5.10c) reflektieren sich im Gesamt-Reward je Episode. Die Reduzierung der Kollisionsanzahl von einem maximalen Wert von 1830 (TD3) bzw. 1684 (DDPG) auf 66 und 12 hat einen signifikanten Einfluss auf die Konvergenz der Reward-Kurve in Abbildung 5.10a. Selbiges gilt für die Kollisionswerte der Communication-Methode, die auf einem konstanten Niveau zwischen 1784 und 173 liegen. Dies resultiert in einem IQR-Wert von 469.5. Im PPO-basierten Ansatz liegt der Wert für K_{min} bei 1451 und hat damit deutlichen Einfluss auf den Reward. Die darüber hinaus erlernte Strategie zur Vermeidung von negativen Rewards \mathcal{R} für die Überschreitung der maximalen Oberflächendistanz unterstützt diese Entwicklung. Hier erlernen alle vier Methoden eine gänzliche Vermeidung, ausgehend von Werten im mittleren dreistelligen Bereich. So weisen die Lösungsmethoden einen IQR von 466.25 (DDPG), 261.0 (TD3) 170.5 (PPO) und 181.5 (Communication-Methode) aus und unterstreichen damit die geringe Streuung der Ergebnisse im letzten Drittel des Lernprozesses. Die numerischen Resultate der Lernprozesse für *Validierungsfall 2* sind in Tabelle 5.6 zusammengefasst. Eine Gegenüberstellung mit einem RRT zeigt, dass dieser für sieben der Agenten einen Pfad innerhalb der Limitation von 10.000 Iterationen identifiziert. Der dabei erreichte rechnerische Reward liegt bei 701.43.

Tabelle 5.6: Numerische Auswertung für das Standardtraining in Validierungsfall 2

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[%] \mathcal{R}_{max}	K_{min}	[%] K_{min}	OOS_{min}	[%] OOS_{min}
TD3	2232.05	-2053.38	911.81	6.31	66.0	450.0	0.0	-
DDPG	1814.36	-2082.06	978.47	23.84	12.0	-	0.0	-
COM	1775.79	-1955.59	943.37	25.46	173.0	1341.67	0.0	-
PPO	2382.50	-2457.97	1228.37	-	1451	11991.67	0.0	-

Die sich daraus ergebenden Verdrahtungen im 3D-CAD-System zeigt Abbildung 5.11. Hier sind die besten Ergebnisse der Methoden TD3, DDPG, Communication und RRT kombiniert und farblich differenziert.

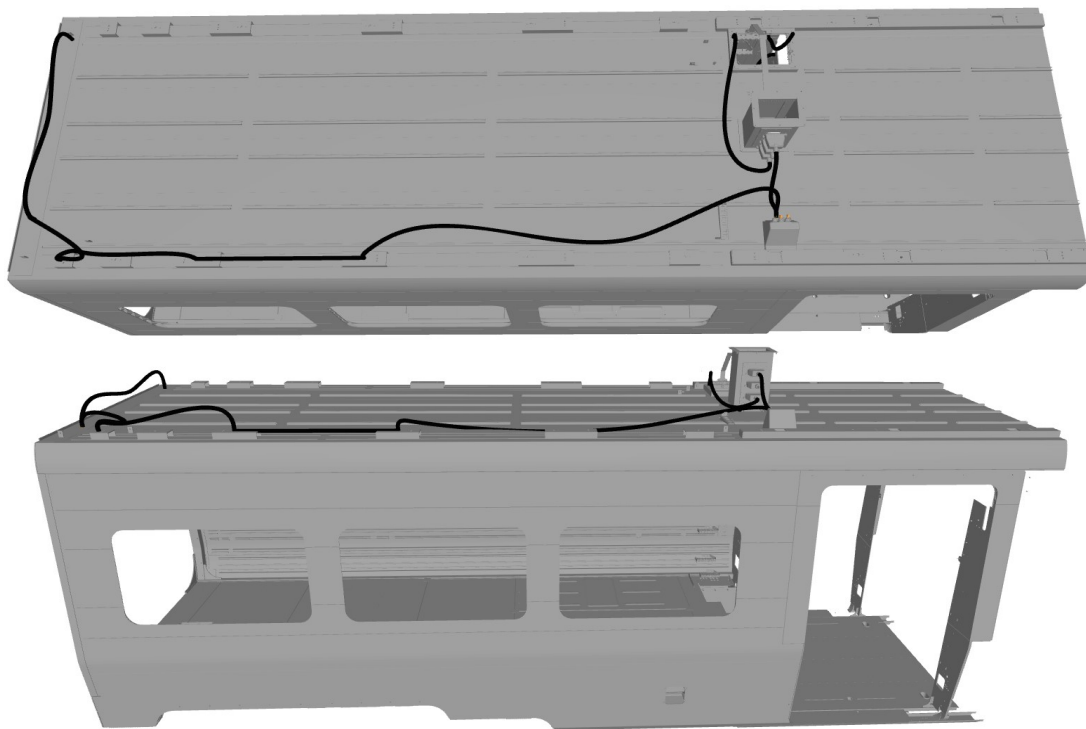


Abbildung 5.11: Darstellung der identifizierten Kabelpfade im 3D-CAD-System für Validierungsfall 2

Abbildung 5.11 ist zu entnehmen, dass die jeweiligen Methoden unterschiedliche Strategien entwickeln, um das CHRP in Validierungsfall 2 zu lösen. Während die MARL-Methoden ähnliche Pfade in der 3D-CAD-Geometrie aufweisen, zeigt der RRT aufgrund seiner Exploration im hochdimensionalen Suchraum stark andersartige Ergebnisse (vgl. Kapitel 3.2.1). Insbesondere unter Berücksichtigung der minimalen Gesamtpfadlänge (vgl. Kapitel 4.1.4) sind die geradlinigen elektrischen Verbindungen der MARL-Methoden als vorteilhaft einzuordnen. Daraus entstehen jedoch an Engstellen Überschneidungen (vgl. zentrale Position auf dem Dach in Abbildung 5.11), die aufgrund des hohen wechselseitigen Einflusses der einzelnen Agenten a^i nicht der geforderten optimalen Lösung des CHRP entsprechen.

Insgesamt lässt sich aus den durchgeführten Experimenten in den *Validierungsfällen 1 und 2* folgern, dass die MARL-Methoden eine grundsätzliche Lösungsfähigkeit für das CHRP aufweisen. Beide betrachteten Metriken für Randbedingungen (vgl. Abbildungen 5.10b und 5.10c) zeigen, dass die Policy π diese anhand des Rewards \mathcal{R} sukzessive reduziert. Gleichzeitig verdeutlichen die Resultate im 3D-CAD-System (vgl. Abbildung 5.11) die Fähigkeit der MARL-Methoden, auch in komplexen CHRP valide Pfade zu ermitteln. Dies ist Grundvoraussetzung für den Einsatz dieser Methoden in einem skalier- und transferierbaren Ansatz (vgl. Kapitel 4.5.1).

5.3.4 Erweiterung der virtuellen Verdrahtung um bewegliche Bauteile

Die Integration dynamischer Bauteile in die automatisierte virtuelle Verdrahtung im 3D-CAD-System ist eine der analysierten Forschungslücken dieser Arbeit (vgl. Tabelle 3.1). Auf Basis der vorgegebenen Qualifizierung der 3D-CAD-Komponenten (vgl. Kapitel 4.1.2) kann die neue Methode die Rotation und die Translation der Bauteile berücksichtigen (vgl. Kapitel 4.2.4). Dabei wird das POMG um einen kinematischen Bereich des Joints erweitert, in dem die Bewegung der Baugruppe Einfluss nimmt. Abbildung 5.12 zeigt den *Validierungsfall Kinematik*, qualifiziert mit vier elektrischen Anschlusspunkten und den erforderlichen Referenzebenen im Abstand $d_{ref} = 90.0mm$ zur Rotationsachse.

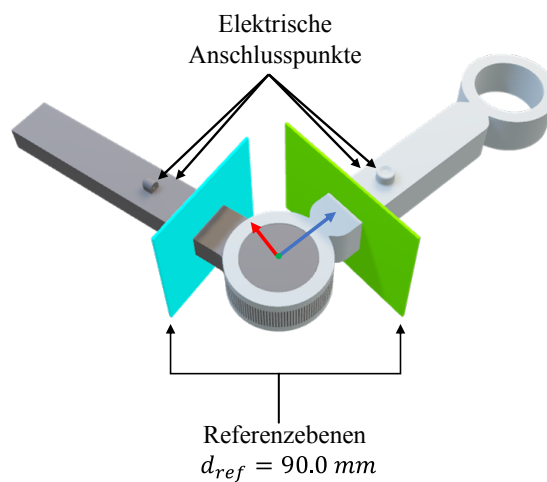
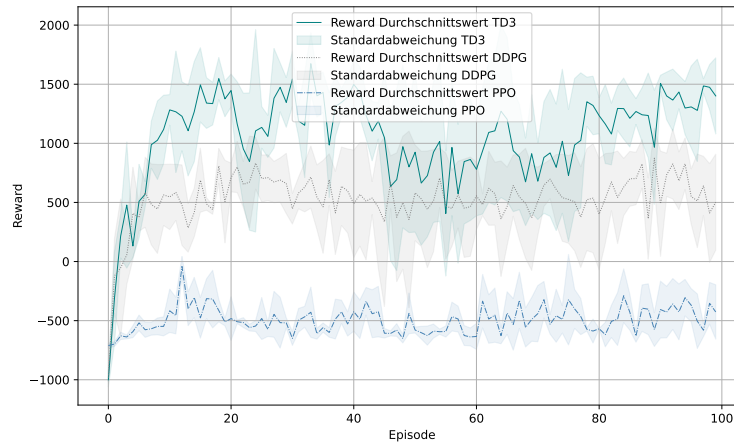


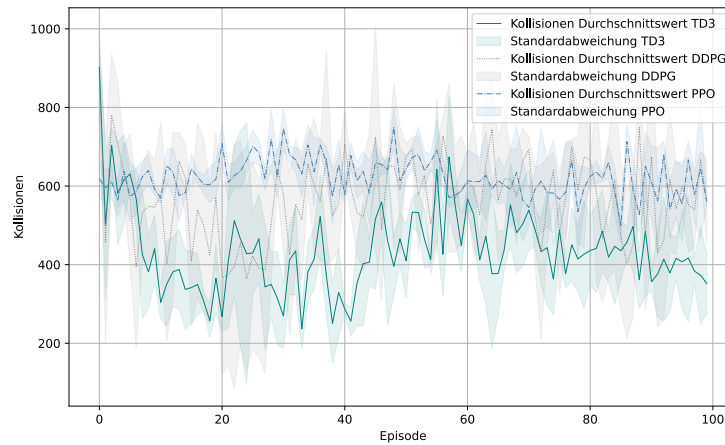
Abbildung 5.12: Validierungsfall Kinematik für ein Rotationsgelenk

Aufgrund der geringeren Agentenanzahl N und der daraus folgenden reduzierten Problemkomplexität werden die Experimente für den Validierungsfall Kinematik mit 100 Episoden durchgeführt. Jede der Episoden Eps umfasst dabei 350ts. Neben der Identifikation von kollisionsfreien Pfaden sieht das CHRP auch die Berücksichtigung des Drehwinkels θ_1 in einem validen Verdrahtungsergebnis vor. Folglich wird ein Ergebnis der MARL-Methode nur als valide eingestuft, wenn die erzeugte Verkabelung im 3D-CAD-System alle mechanisch zulässigen Positionen der Baugruppe ermöglicht. Dafür wird der Reward entsprechend adaptiert (vgl. Kapitel 4.2.4).

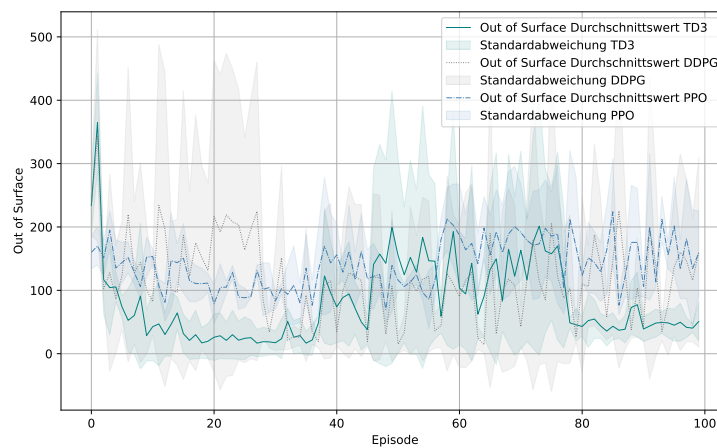
Die nachfolgende Abbildung 5.13 stellt die numerischen Resultate der Methoden Anwendung auf den Validierungsfall Kinematik dar. Analog zur Auswertung im statischen Szenario (vgl. Kapitel 5.3.3) sind diese in die Abbildungen zum Gesamt-Reward (vgl. Abbildung 5.13a), zur Kollisionsanzahl (vgl. Abbildung 5.13b) und zur Überschreitung des Oberflächenabstandes (vgl. Abbildung 5.13c) untergliedert. Die Communication-Methode wird in diesem Validierungsfall nicht angewendet, da eine Agentenanzahl von $N = 4$ keine Bildung von zwei Clustern erlaubt. Damit ist das Ergebnis der Communication-Methode analog zu den Resultaten des DDPG.



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.13: Gegenüberstellung der MARL-Metriken für Validierungsfall Kinematik

In der Betrachtung des Gesamt-Rewards pro Episode (vgl. Abbildung 5.13a) weisen die angewendeten MARL-Methoden signifikante Unterschiede auf. So erreicht der DDPG einen maximalen Reward \mathcal{R} von 1171.93, der TD3 hingegen einen Wert von 2032.69. Im PPO-basierten Ansatz liegt der Wert für \mathcal{R}_{max} bei 208.01. Alle drei Ergebnisse implizieren jedoch, dass das POMG in keiner der 100 Episoden vollständig gelöst werden konnte (vgl. Kapitel 4.3.2). Zudem ist in Abbildung 5.13a dargestellt, dass die Methoden keine klare Konvergenz aufweisen. Diese Annahme wird von den hohen IQR-Werten von 631.97 (DDPG) und 724.38 (TD3) gestützt, die eine große Streuung der erzielten Rewards aufzeigen.

Die Metrik der in der Simulation verursachten Kollisionen je Episode (vgl. Abbildung 5.13b) reflektiert die Reward-Ergebnisse der Trainingsläufe des statischen CHRP. Die Anwendung der Methoden führt zu Resultaten, die nicht vollständig kollisionsfrei sind. Jedoch ist auch hier eine signifikante Schwankung der Ergebnisse zu beobachten. Aus der minimalen Anzahl an Kollisionen von 48 (DDPG), 114 (TD3) und 411 (PPO) im letzten Viertel des Trainingsprozesses resultiert ein zugehöriger IQR zwischen 208.5 und 292.0.

Ein vergleichbares Bild zeigt die Metrik der Distanzüberschreitungen zur Oberfläche des 3D-CAD-Modells (vgl. Abbildung 5.13c). Auch hier finden die Agenten ein Verdrahtungsergebnis, das sich vollständig innerhalb des zulässigen Bereichs befindet. Jedoch bestehen auch Episoden mit maximalen Werten von 502 bis 592 für beide MARL-Lösungsalgorithmen. Zudem ist ersichtlich, dass kein vergleichbares Lernverhalten wie im statischen Szenario eintritt (vgl. Abbildung 5.8c). Die numerischen Resultate der Trainingsläufe sind in Tabelle 5.7 zusammengefasst.

Tabelle 5.7: Numerische Auswertung des Standardtrainings für den Validierungsfall Kinematik

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[$\%$] \mathcal{R}_{max}	K_{min}	[$\%$] K_{min}	OOS_{min}	[$\%$] OOS_{min}
TD3	2032.69	-1079.11	724.38	-	114	137.5	0.0	-
DDPG	1171.93	-1019.49	631.97	42.34	48	-	0.0	-
COM	1171.93	-1019.49	631.97	42.34	48	-	0.0	-
PPO	208.01	-760.19	159.56	89.76	411	756.25	4.0	∞

Auf Grundlage des besten Ergebnisses innerhalb des Lernprozesses der MARL-Methoden zeigt Abbildung 5.14 die zugehörigen Kabelpfade innerhalb der 3D-CAD-Geometrie in den Positionen mit dem maximal zulässigen Drehwinkel θ_1 . Die Kabellängen sind in den beiden Abbildungen 5.14a und 5.14b identisch. Ergebnisse eines RRT sind nicht inkludiert, da dieser innerhalb der gesetzten Grenze von 10.000 Iterationen je Agent keine Lösung identifizieren kann. Dieses Ergebnis unterstreicht die vorliegende Problemkomplexität in diesem Validierungsfall unter Berücksichtigung der Kinematik. Der zugehörige Reward für die Experimente mit einem RRT liegt bei 97.08.

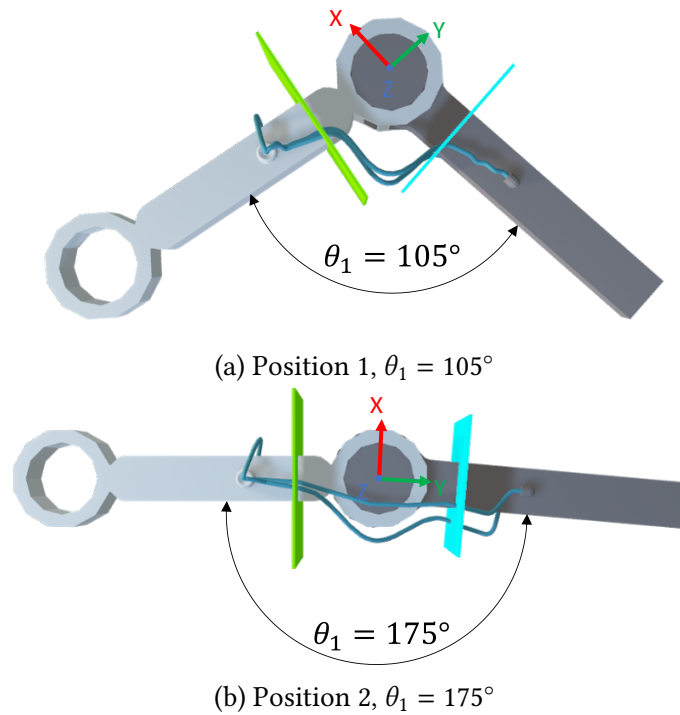


Abbildung 5.14: Darstellung der elektrischen Verbindungen mittels MARL-Methode für den Validierungsfall Kinematik

Die beiden Ebenen in Abbildung 5.14 zeigen die Beschränkung des kinematischen Bereichs (vgl. Kapitel 4.2.4), außerhalb derer sich die Kabelführung durch die Beweglichkeit der 3D-CAD-Bauteile nicht verändert. Abbildung 5.14 ist weiter zu entnehmen, dass die MARL-Methode dazu in der Lage ist, die Modifikation des POMG zu berücksichtigen. Auf Basis der Ergänzung des Rewards \mathcal{R} bzw. der Änderung der Tupel $\langle O_t^i, \mathcal{A}^i, \mathcal{R}^i, O_{t+1}^i \rangle$ im kinematischen Bereich erlernen die Agenten eine Policy π , die valide Kabelpfade generiert. Damit ist es möglich, neben der relevanten Zusatzlänge auch den Biegeradius zum Eintrittspunkt in die kinematische Zone zu berücksichtigen (vgl. Abbildung 2.9). Diese Zusatzlänge ist insbesondere in Abbildung 5.14a zu erkennen.

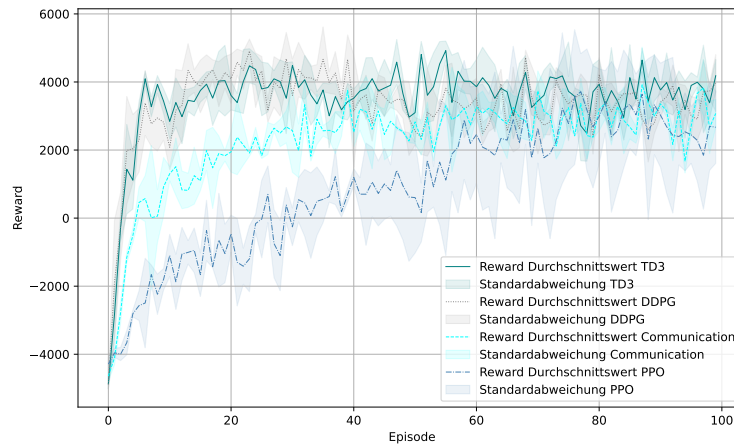
Mit der Integration dieser Randbedingungen geht jedoch auch eine Beeinträchtigung der Optimalität von elektrischen Verbindungen ohne Einfluss der dynamischen Bauteile einher. Dies ergibt sich aus dem Trainings- und Ausführungsschema CTCE, in dem alle Agenten $a^i \forall i \in \{1, \dots, N\}$ die gleiche Policy π mit Tupeln $\langle O_t^i, \mathcal{A}^i, \mathcal{R}^i, O_{t+1}^i \rangle$ speisen. Im Validierungsbeispiel werden vorwiegend elektrische Verbindungen verwendet, die durch die Kinematik in ihrer Wegfindung beeinträchtigt werden. Die damit einhergehende Anpassung der Policy π berücksichtigt diese Charakteristika stärker, was zu ungünstigeren Resultaten der verbleibenden Kabel führt (vgl. Abbildung 5.13a).

5.3.5 Kabelbaumerzeugung und Skalierbarkeit

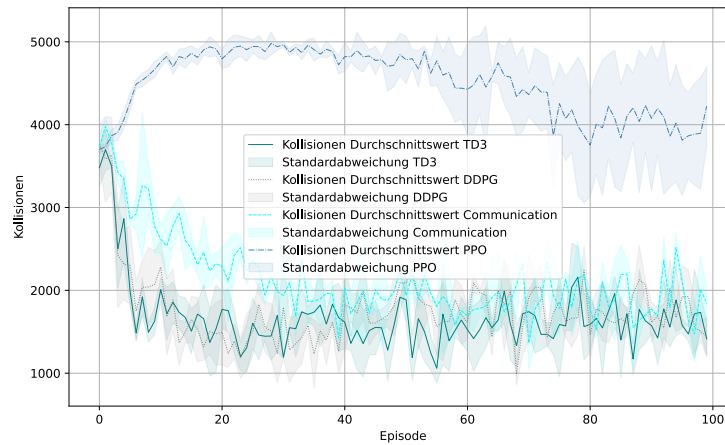
Die Methode zur Skalierung der elektrischen Verdrahtung im 3D-CAD-System enthält eine Integration der identifizierten Verzweigungspunkte P_{Tb} (vgl. Kapitel 4.5.1). In diesem Zusammenhang finden die Ergebnisse aus Kapitel 5.3.1 in diesem Validierungsabschnitt weitere Anwendung. Die entstehenden Zusammenführungspunkte der einzelnen Kabelbündel repräsentieren die modifizierten Start- und Zielpunkte P'_S und P'_T der einzelnen Agenten $a^i \forall i \in \{1, \dots, N\}$. Diese Herangehensweise spiegelt die Intention der Sub-Problem-Decomposition wider und wirkt den Herausforderungen der Verwendung einer MARL-Methoden entgegen (vgl. Kapitel 2.5.4).

Als Eingangsgröße für die Kabelbaumerzeugung werden die Resultate aus Kapitel 5.3.5 verwendet. Diese entsprechen dem ersten Teilschritt der Alpha-Wrap-Methode zur Graphmodifikation in Kapitel 4.5.1.1 und damit den Teilschritten (1) bis (3) in Abbildung 4.21 zur Skalierung der MARL-Methode. Für die Skalierbarkeit und die damit verbundene Sub-Problem-Decomposition ist das Ziel eine besonders starke Bündelung der einzelnen Kabel, was in einer möglichst geringen Agentenanzahl N resultiert. Dafür ist weniger die Gesamtheit der individuellen Kabellängen von Relevanz, als die Gesamtlänge aller Edges E_α des Subgraphen H_α . Ein besonders kleiner Bündelungsfaktor w_B für die Modifikation der Kantengewichte $w(u, v)$ (vgl. Kapitel 2.1 und 4.5.1.1) ist daher als Eingangsgröße gesetzt.

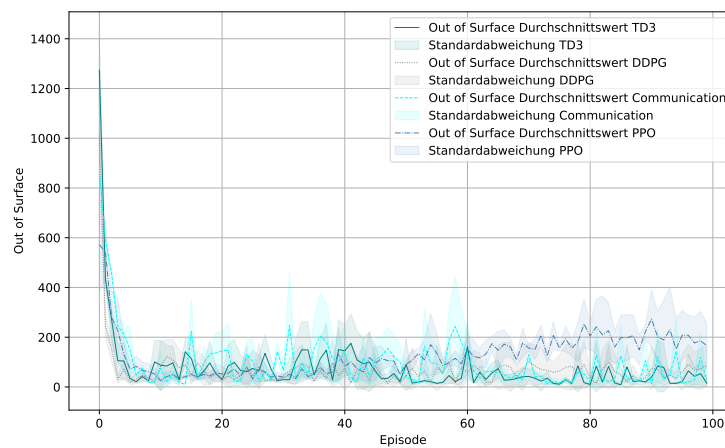
Aufgrund der domänenorientierten Anwendung der automatisierten virtuellen Verdrahtung kommt in *Validierungsfall 1* der Bündelungsfaktor $w_B = 0.01$ zum Einsatz (vgl. Abbildung 5.6a). Damit soll eine kabelbaumorientierte Wegstreckenplanung forciert werden. Infolgedessen steigt die Anzahl der Agenten N in $G'_\alpha(V, E)$ von 13 auf 23. Dies führt zwar in diesem Anwendungsfall nicht zu einer grundsätzlichen Reduzierung der Problemkomplexität, jedoch reduzieren sich die möglichen Wechselwirkungen der Pfade einzelner Agenten zueinander. Dies ergibt sich aus der Tatsache, dass gemeinsame Pfadabschnitte innerhalb eines Bündels verlaufen. Abbildung 5.15 zeigt den Lernprozess der MARL-Methoden mit Fokus auf der Erzeugung eines Kabelbaums für *Validierungsfall 1*.



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.15: Gegenüberstellung der MARL-Metriken für Validierungsfall 1 mit Kabelbaumerzeugung

In Abbildung 5.15a ist dargestellt, welchen maximalen Reward die MARL-Methoden innerhalb der jeweiligen Episode erreichen. Es lässt sich eindeutig feststellen, dass die Methoden des TD3 und des DDPG eine ähnliche Lernkurve aufweisen. Die Lernkurve der PPO-basierten Methode weicht zwar davon ab, erreicht aber ebenfalls einen maximalen Reward \mathcal{R}_{max} von 5543.36. Die Communication-Methode hingegen erzielt mit einem maximalen Wert für \mathcal{R}_{max} von 5204.54 ein schlechteres Ergebnis als TD3 (5448.48) und DDPG (5674.04). Jedoch divergieren auch die Resultate für den maximalen Reward zwischen TD3 und DDPG um den Wert von 3.97 % bzw. 2.30 % zur PPO-basierten Methode.

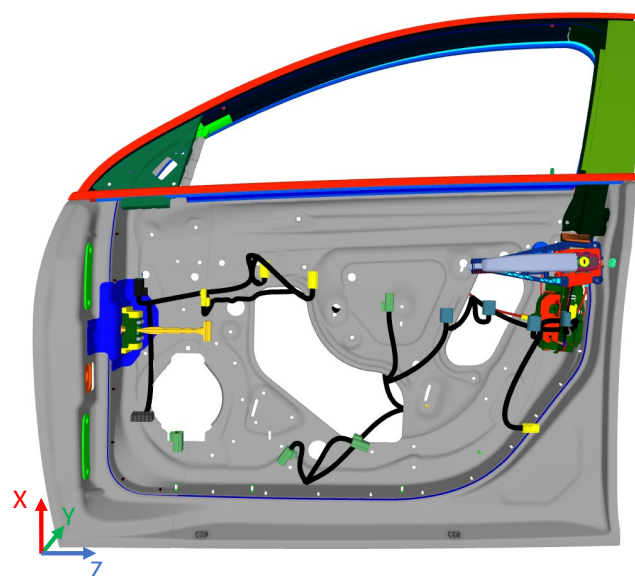
In Bezug auf die zu erlernenden Randbedingungen (vgl. Kapitel 2.3.3) variiert das Lernverhalten der Methoden erheblich. Die in Abbildung 5.15b dargestellten Ergebnisse weisen unterschiedliche Ausprägungen in der Generierung kollisionsfreier Pfade auf. Während die Methode, die auf einem DDPG basiert, einen Minimalwert von 527 erreicht, erzielt der auf TD3 gestützte Ansatz einen Wert von 382. Die Communication-Methode verzeichnet mit einem Wert von 726 weniger zufriedenstellende Ergebnisse hinsichtlich der Vermeidung von Kollisionen. Der PPO-basierte Ansatz erreicht einen Wert von 2716 für K_{min} .

Im Gegensatz dazu sind sämtliche Methoden in der Lage, eine Policy π zu entwickeln, die nahezu keine Überschreitung der maximalen Oberflächendistanz in den Trajektorien τ^i einer Episode aufweist (vgl. Abbildung 5.15c). Die numerischen Ergebnisse der Experimente sind in Tabelle 5.8 zusammengefasst. Zur Einordnung der Ergebnisse erfolgt eine Gegenüberstellung der Resultate mit einem RRT. Dieser erzeugt eine vollständige Lösung für Validierungsfall 1 mit Kabelbaum und ist dabei in der Lage, einen rechnerischen Reward von 3560.71 zu erzielen.

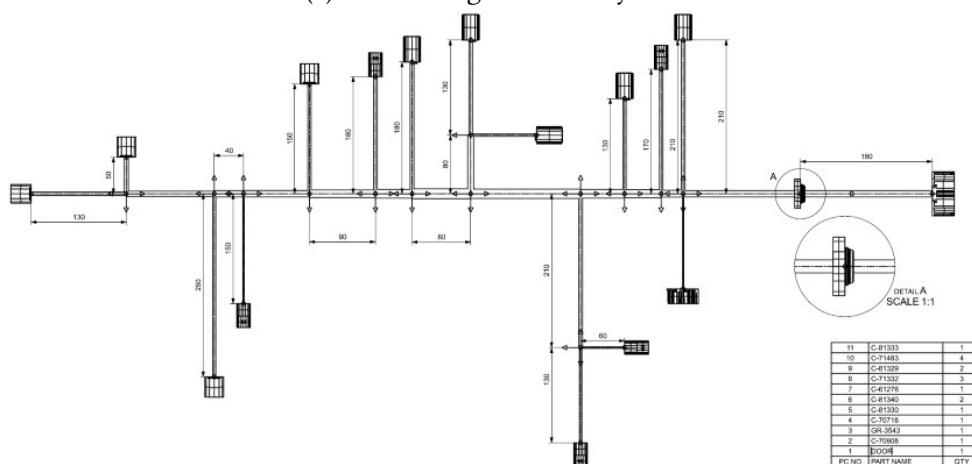
Tabelle 5.8: Numerische Auswertung des Standardtrainings für Validierungsfall 1 mit Kabelbaum

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[%] \mathcal{R}_{max}	K_{min}	[%] K_{min}	OOS_{min}	[%] OOS_{min}
TD3	5448.48	-5066.26	1689.05	3.97	382	-	4	100.0
DDPG	5674.04	-5044.65	1650.71	-	527	37.95	4	100.0
COM	5204.54	-4788.93	1253.55	8.27	726	90.05	2	-
PPO	5543.36	-4558.33	3141.89	2.30	2716	610.99	17	750.0

Zugehörig zu den Resultaten der Experimente in Tabelle 5.8 zeigt Abbildung 5.16 die Ergebnisse nach der Übertragung in das 3D-CAD-System. Dabei ist in Abbildung 5.16a das 3D-CAD-Modell aufgeführt und in Abbildung 5.16b die schematische Darstellung des Kabelbaums.



(a) Darstellung 3D-CAD-System

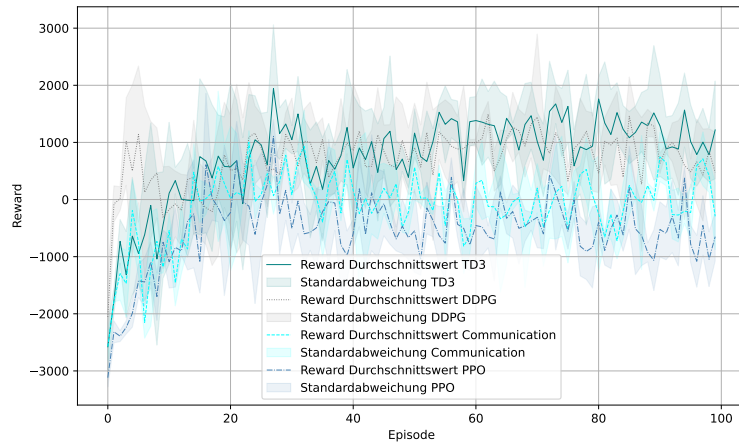


(b) Schematische Darstellung des Kabelbaums

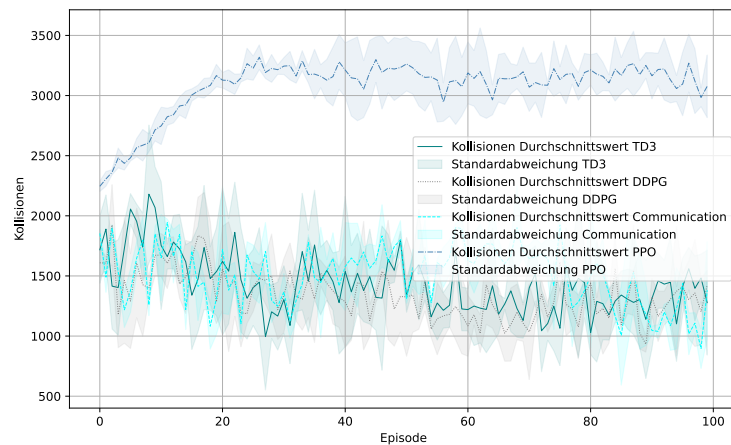
Abbildung 5.16: Darstellung der besten Kabelpfade im 3D-CAD-System für Validierungsfall 1 mit Kabelbaum

In Abbildung 5.16 ist zu erkennen, dass die MARL-Methode einen sehr direkten Weg zwischen den jeweiligen Anschlusspunkten identifizieren kann. Dennoch bestehen keine geradlinigen Pfade, die den gesetzten Randbedingungen widersprechen würden (vgl. Kapitel 2.3.3).

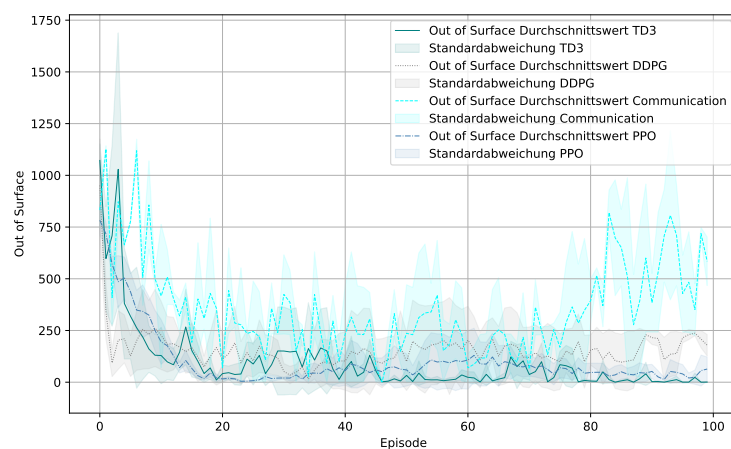
In *Validierungsfall 2* reduziert sich die Anzahl der Agenten N bei $w_B = 0.01$ auf $N = 15$, bei $w_B = 0.9$ auf $N = 71$. Daher kommt in den nachfolgenden Experimenten ein Navigations-Graph mit $w_B = 0.01$ zum Einsatz. Abbildung 5.17 zeigt die quantitativen Resultate des Lernprozesses der MARL-Methoden auf Basis der Problem-Decomposition innerhalb des Lernprozesses von 100 Episoden mit je 250ts.



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.17: Gegenüberstellung der Ergebnisse der MARL-Metriken für Validierungsfall 2 mit Kabelbaumerzeugung

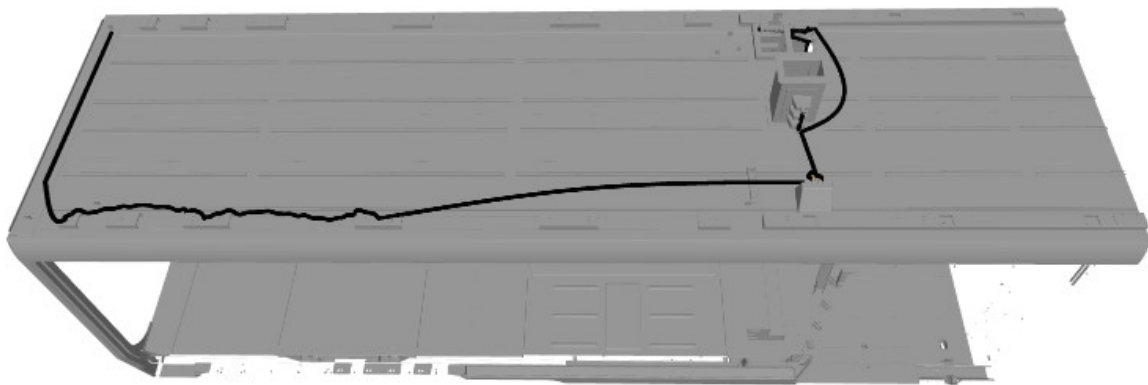
Abbildung 5.17a ist der Lernprozess der Methoden TD3, DDPG, PPO und der Communication-Methode zu entnehmen. Dabei kann die Methode, die den TD3 als MARL-Algorithmus inkludiert, einen Reward \mathcal{R} von 3321.62 innerhalb einer Episode erzielen. Dies übertrifft einerseits die Performance des DDPG-basierten Trainingslaufs um 10.94 % ($\mathcal{R}_{max} = 2958.06$) und zeigt andererseits ein vollständig gelöstes CHRP für *Validierungsfall 2*. Die Communication-Methode erreicht einen maximalen Reward von 1896.47, die PPO-basierte Methode einen Wert von 2349.43. Während der TD3 innerhalb der 100 Episoden das definierte CHRP in 6 Episoden vollständig und in 33 Episoden mit Ausnahme eines einzelnen Kabelbündels (hier äquivalent zu einem Agenten a^i) löst, bewältigen der DDPG und die Communication-Methode die Problemstellung des CHRP jeweils lediglich einmal. In der minimalen Performance zeigen die Methoden vergleichbare Resultate mit einem Reward \mathcal{R} von -2644.23 (DDPG), -3113.23 (TD3), -3300.40 (PPO) und -2824.43 (Communication-Methode) und zugehörigen IQR-Werten von 996.27 (DDPG), 1228.14 (TD3), 1049.15 (PPO) und 937.40 (Communication-Methode). Im Episodenbereich von 60 bis 100 konvergieren die Methoden und erzielen lediglich durch die integrierten Explorationsphasen signifikant bessere Ergebnisse.

Die Metrik der Kollisionen je Episode (vgl. Abbildung 5.17b) spiegelt bei den Methoden des DDPG und TD3 die resultierenden Rewards wider. Mit Werten von 272 (TD3) und 275 (DDPG) für die minimale Anzahl an Kollisionen weichen sie deutlich von den Resultaten der Communication-Methode mit 406 und dem PPO mit 2197 Kollisionen ab. Jedoch zeigen alle vier Lösungsmethoden eine hohe Ergebnisstreuung im Wertebereich zwischen 115.25 und 372.25. Hervorzuheben ist, dass in *Validierungsfall 2* alle Methoden Lösungsstrategien (Policy π) entwickeln konnten, welche keine Überschreitung des maximalen Oberflächenabstandes beinhalten. In Tabelle 5.9 sind die numerischen Ergebnisse der Lösungsmethoden für diese Experimente zusammengefasst. Zur besseren Einordnung der Resultate der MARL-Methoden werden diese mit den Ergebnissen eines RRT verglichen. Dieser ist dazu in der Lage, für alle 15 Agenten einen vollständigen Pfad zu erzeugen. Der zugehörige, rechnerische Reward liegt bei 1340.95.

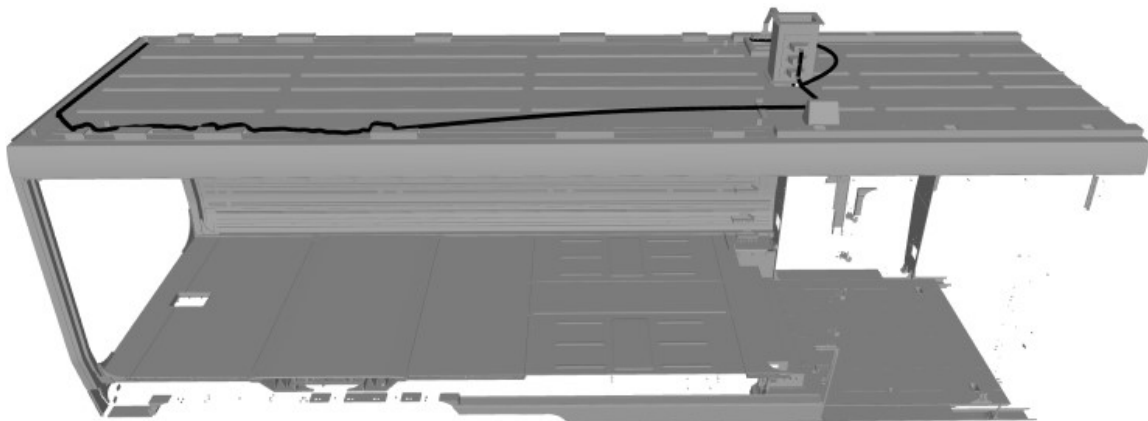
Tabelle 5.9: Numerische Auswertung im Standardtraining für Validierungsfall 2 mit Kabelbaum

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[%] \mathcal{R}_{max}	K_{min}	[%] K_{min}	OOS_{min}	[%] OOS_{min}
TD3	3321.62	-3113.23	1228.14	-	272	-	0.0	-
DDPG	2958.06	-2644.23	996.27	10.94	275	1.10	0.0	-
COM	1896.47	-2824.43	937.40	42.90	406	49.26	0.0	-
PPO	2349.43	-3300.40	1049.15	29.26	2197	707.72	0.0	-

Das beste Verdrahtungsergebnis (vgl. \mathcal{R}_{max} TD3) im 3D-CAD-System ist in Abbildung 5.18 dargestellt. Hier sind die besten Ergebnisse aller eingesetzten Methoden (TD3, DDPG, Communication-Methode und RRT) in zwei Perspektiven gezeigt.



(a) Darstellung 3D-CAD-System Dachfläche



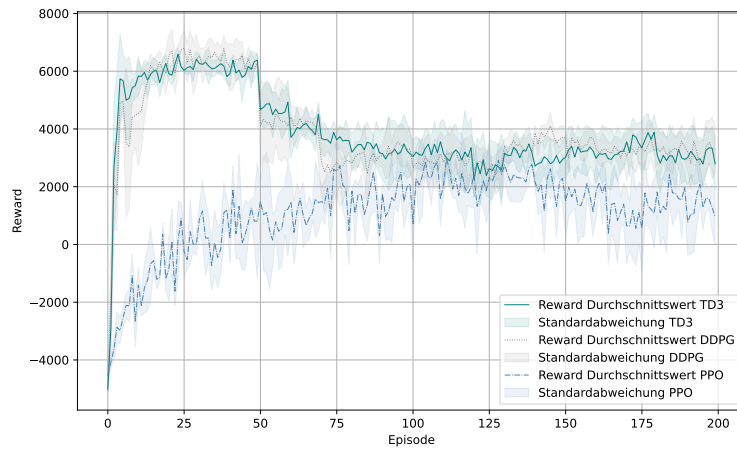
(b) Darstellung 3D-CAD-System Seitenansicht

Abbildung 5.18: Ergebnisse zur Pfadfindung mit Kabelbaum im 3D-CAD-System für Validierungsfall 2 (*ausgeblendete Seitenwand*)

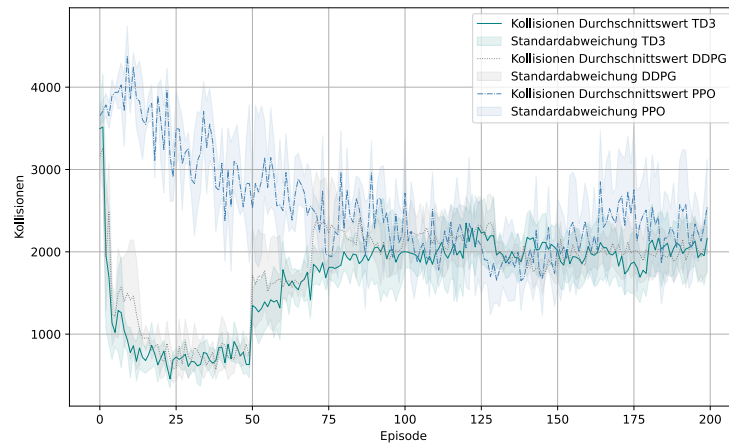
Verglichen mit den Kabelpfaden im Standardtraining (vgl. Abbildung 5.11) weisen die durch Problem-Decomposition und mit Hilfe eines Kabelbaums generierten Wegstreckenkpunkte eine höhere Homogenität auf (vgl. Abbildung 5.18). Insbesondere die Methoden mittels des TD3 und des DDPG zeichnen sich durch ähnliche Trajektorien τ^i aus, auch wenn die prozentuale Abweichung in \mathcal{R}_{max} bei 10.94 % liegt. Obwohl jene Abweichung zur Communication-Methode 42.90 % beträgt, ist das visuelle Ergebnis in Abbildung erkennbar heterogener. Dies verdeutlicht, dass eine Annäherung an die Optimalität des CHRP (vgl. Kapitel 4.1.4) über sehr unterschiedliche Wege erfolgen kann.

5.3.6 Transfer Learning mit randomisierten Anschlusspunkten

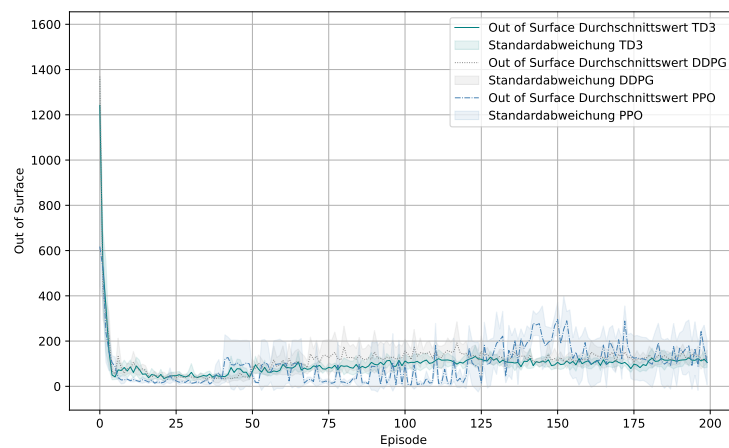
Eine der definierten Zielsetzungen dieser Arbeit (vgl. Kapitel 1.3) ist die Übertragbarkeit von Policies π aus bereits gelösten virtuellen Verdrahtungen auf neue adaptierte CHRP. Die hierfür entwickelte Methode auf Basis von TL zielt darauf ab, die Generalisierungsfähigkeit von MARL-Methoden in geometrisch ähnlichen CHRP zu verbessern (vgl. Kapitel 4.5.2.1). Der initiale Schritt der Methode besteht darin, ein randomisiertes CHRP innerhalb der Sourcedomäne D_S zu lösen. Ausgangspunkt für die Zielpunktrandomisierung sind die Positionen der elektrischen Anschlusspunkte für *Validierungsfall 1*. Diese werden nach einer initialen Anlernphase von 50 Episoden alle 10 Episoden in ihrer Position $[x, y, z]$ verändert. Abbildung 5.19 stellt die Entwicklung der Metriken Reward \mathcal{R} , Anzahl der Kollisionen und Überschreitungen der maximalen Oberflächendistanz über einen Zeitraum von 200 Episoden mit jeweils 250ts dar. Somit lösen die angewendeten Methoden auf Basis von TD3, DDPG und PPO im Laufe des Trainings 16 leicht modifizierte CHRP. Die Anwendung der Communication-Methode im TL wird nicht betrachtet. Ansonsten ist unklar, welche Policy π bei einer neuen elektrischen Verbindung zum Einsatz kommt. Diese Tatsache lässt sich mit dem Trainingsschema der Communication-Methode erklären (vgl. Kapitel 4.4.2).



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.19: Gegenüberstellung der Ergebnisse der MARL-Metriken für Validierungsfall 1 mit Zielpunktrandomisierung

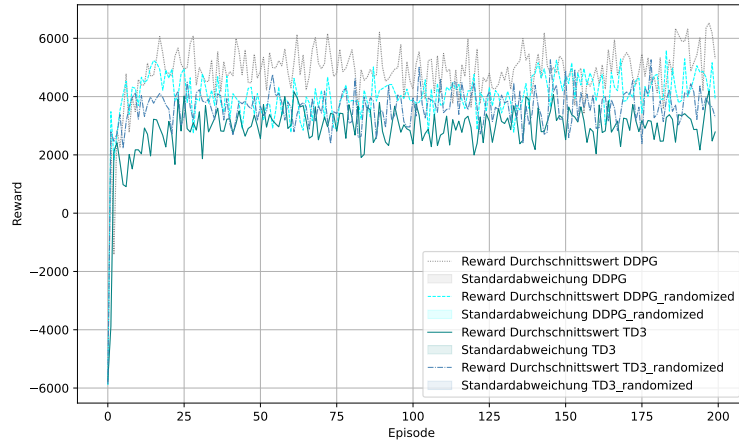
Die in Abbildung 5.19a dargestellte Reward-Kurve zeigt ein Lernverhalten, das mit dem des Standardtrainings in Abbildung 5.17a vergleichbar ist. Beiden Methoden ist es nach einer Anlernphase möglich, das sich stetig anpassende CHRP partiell zu lösen. Nur in wenigen Episoden werden sämtliche elektrischen Verbindungen vollständig geplant. Während die TD3-basierte Methode in 7 Episoden das CHRP ganzheitlich löst, gelingt dies mit der auf einem DDPG beruhenden Methode lediglich in 3 der 200 Episoden. Der maximal erreichte Reward \mathcal{R}_{max} von 8142.16 (TD3) bzw. 7859.09 (DDPG) zeigt auf, dass trotz der eingeschränkten Fähigkeit, das CHRP häufiger zu lösen, das beste numerische Resultat von der DDPG-gestützten Methode erreicht wird. Der PPO-basierte Ansatz erreicht lediglich einen Reward \mathcal{R}_{max} von 4412.18. Die Abweichung in der Betrachtung des minimalen Rewards \mathcal{R}_{min} weist hingegen vergleichbare Werte in den ersten Episoden des Trainings auf. Es ist anzumerken, dass trotz der fortwährenden Modifikation des CHRP durch die Zielpunktrandomisierung die Rewardkurve keinen vollständigen Einbruch der Werte verzeichnet.

Die Analyse der Kollisionen pro Episode (vgl. Abbildung 5.19b) ergibt keine Hinweise auf eine stetige Verbesserung. Im Vergleich zu den Werten aus dem Standardtraining (vgl. Abbildung 5.17b) können keine optimierten Ergebnisse hinsichtlich der Integration der genannten Randbedingung erzielt werden. Dies ist vor allem auf die stetige Veränderung des CHRP zurückzuführen. So zeigt die beste Episode des TD3 einen Wert von 318 für K_{min} , der DDPG kann eine Episode mit nur 377 Kollisionen verzeichnen. Im PPO liegt das beste Ergebnis bei einem Wert von 1093 für K_{min} . Im Kontext der Überschreitung der maximalen Oberflächendistanz erreicht keine der beiden Methoden DDPG und TD3 ein optimales Ergebnis von 0 (vgl. Abbildung 5.19c). Besonders bei signifikanten Verschiebungen der Zielpunkte sind erhebliche Abweichungen nach oben festzustellen. Eine Zusammenfassung der numerischen Ergebnisse des randomisierten Trainings in *Validierungsfall 1* ist in Tabelle 5.10 dargestellt.

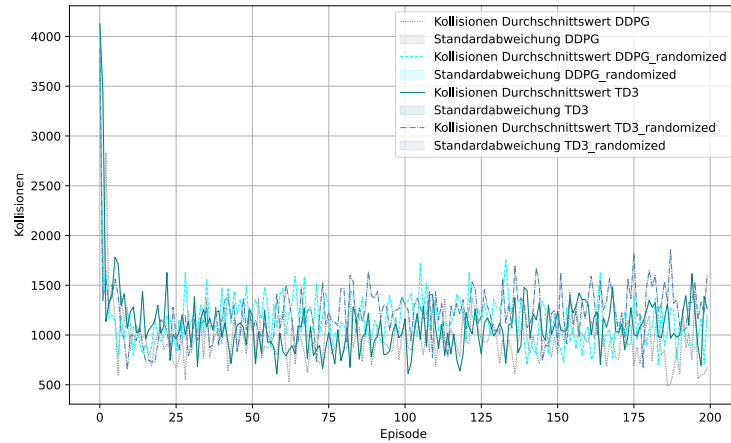
Tabelle 5.10: Numerische Auswertung der Ergebnisse des randomisierten Trainings für Validierungsfall 1

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[%] \mathcal{R}_{max}	K_{min}	[%] K_{min}	OOS_{min}	[%] OOS_{min}
TD3	8142.16	-5143.04	1974.38	-	318	-	18	∞
DDPG	7859.09	-4974.96	1826.95	3.47	377	18.55	14	∞
PPO	4412.18	-4889.92	2097.99	45.81	1093	233.01	0	-

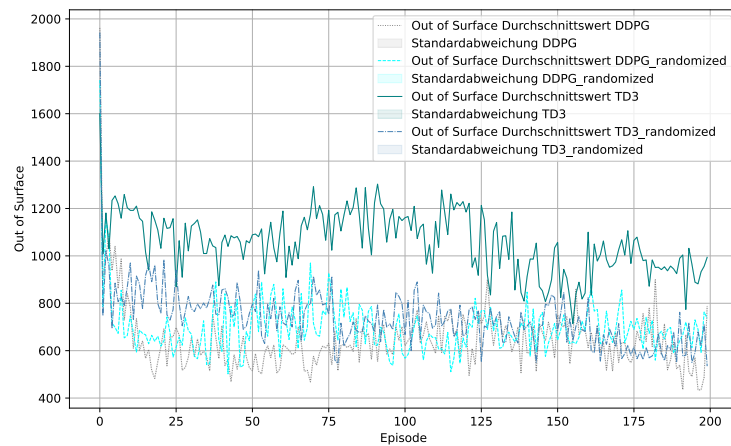
Im zweiten Schritt erfolgt die Übertragung der erlernten Policies π auf das adaptierte CHRP. Der durchgeführte Zero-Shot-Transfer (vgl. Kapitel 4.5) wird im Folgenden für zwei unterschiedliche Kategorien an MARL-Methoden verglichen: einerseits für die Methoden, in deren Lernphase eine Zielpunktrandomisierung erfolgte, andererseits die antrainierten Policies aus dem Standardtraining ohne Randomisierung. Die Resultate werden mit den Ergebnissen aus dem Standardtraining mit Kabelbaumerstellung gegenübergestellt, um die Metriken für das TL zu ermitteln. Abbildung 5.20 zeigt den Gesamt-Reward, die Anzahl der Kollisionen und die Anzahl der Überschreitungen der maximalen Oberflächendistanz. In jeder der Abbildungen sind die randomisierten Methoden (mit Appendix $_{rand}$) mit den nicht randomisierten Methoden dargestellt. Der Fokus dieser Darstellung liegt insbesondere auf der Korrelation der erreichten Ergebnisse zur Lernkurve in D_5 .



(a) Gesamt-Reward je Episode



(b) Anzahl der Kollisionen je Episode



(c) Anzahl der Überschreitungen des max. Oberflächenabstands je Episode

Abbildung 5.20: Gegenüberstellung der Ergebnisse der MARL-Metriken im Zero-Shot-Transfer in Validierungsfall 1

Die Metrik des Gesamt-Rewards je Episode mit 250ts (vgl. Abbildung 5.20a) zeigt für alle Algorithmen eine leicht steigende Tendenz des durchschnittlichen Rewards über die 200 Episoden. Für den maximalen Reward \mathcal{R}_{max} erreicht der DDPG mit 6529.15 ein um 14.51 % besseres Ergebnis, verglichen mit dem randomisierten Pendant ($\mathcal{R}_{max} = 5581.69$). Die Werte für die TD3-basierten Methoden ergeben ein konträres Bild. Hier erreicht der TD3_{rand} mit einem maximalen Reward \mathcal{R}_{max} von 5263.12 nicht nur ein um 25.48 % besseres Ergebnis, sondern löst das CHRP auch vollständig. Es ist anzumerken, dass dieser Wert aber lediglich Outliern in zwei der 200 Episoden entspricht. Die Werte für den minimalen Reward \mathcal{R}_{min} sind für alle vier gegenübergestellten Methoden vergleichbar und sind durch das zufällig initialisierte neuronale Netz in D_S begründet. Auffällig ist darüber hinaus eine vergleichsweise geringere Ergebnisstreuung der Werte für den DDPG.

Ein analoges Resultat ist für die Anzahl der Kollisionen je Episode zu erkennen (vgl. Abbildung 5.20b). Auch hier bewegen sich die Ergebnisse des DDPG auf einem deutlich geringeren Niveau als bei den anderen Methoden. Die Differenz liegt in einem Bereich von 25.00 % bis 43.44 %. Abbildung 5.20b veranschaulicht diesen Unterschied visuell und zeigt, dass selbst die schlechtesten Ergebnisse des DDPG in einigen Fällen besser sind als jene der anderen Methoden. Auch ist für die Methoden DDPG_{rand}, TD3 und TD3_{rand} im zeitlichen Verlauf keine deutliche Verbesserung in dieser Metrik auszumachen. Eine andere Schlussfolgerung lässt Abbildung 5.20c zu. Hier ist ein klarer Lernfortschritt zu erkennen, der in einem Ergebnis von 434 Überschreitungen der zulässigen Oberflächendistanz im DDPG resultiert. Hervorzuheben ist in diesem Kontext der Vergleich mit den Ergebnissen des randomisierten Trainings (vgl. Abbildung 5.19b). Hier erreichen die angewendeten Methoden im Transfer ein niedrigeres Level an Kollisionen als im zugehörigen Training in D_S .

Anders als in den vorangegangenen Metriken (vgl. Abbildung 5.20a und 5.20b) weichen die Ergebnisse der Methoden stärker voneinander ab. Differenzen von 15.66 % bis 64.51 % sind den Resultaten der Experimente in der Metrik OOS_{min} zu entnehmen. Hier können die Methoden das niedrige Niveau an Überschreitungen der maximalen Oberflächendistanz nicht transferieren. Tabelle 5.11 fasst die numerischen Ergebnisse des Zero-Shot-Transfers zusammen.

Tabelle 5.11: Numerische Auswertung der Experimente für Validierungsfall 1 mit Zero-Shot-Transfer

Methode	\mathcal{R}_{max}	\mathcal{R}_{min}	\mathcal{R}_{IQR}	[%] \mathcal{R}_{max}	K_{min}	[%] K_{min}	OOS_{min}	[%] OOS_{min}
TD3	4194.36	-5758.47	688.95	35.75	610	25.0	714	64.51
DDPG	6529.15	-5872.66	796.30	-	488	-	434	-
TD3 _{rand}	5263.12	-5855.88	800.29	19.39	655	34.22	535	23.27
DDPG _{rand}	5581.69	-5894.74	752.94	14.51	700	43.44	502	15.66

In den Experimenten wird die Übertragung der bestehenden Polycys auf ein neues CHRP um die Metriken des TL erweitert (vgl. Kapitel 2.6.5). Bei diesen Experimenten ist die initiale Performance TL_{IP} mit den Resultaten des Zero-Shot-Transfers (vgl. Tabelle 5.11) gleichzusetzen. Auszuklammern ist in dieser Betrachtung die transferierte Policy π der Episode 0, da diese zufällig initialisierte und nicht trainierte neuronale Netze enthält.

In der Metrik der Asymptotic Performance TL_{AP} ist das Ergebnis des DDPG hervorzuheben. Es übertrifft den maximalen Reward \mathcal{R}_{max} des DDPG im Standard-Training bis zur Episode $Eps = 60$. Erst hier wird im Standard-Training ein Reward von \mathcal{R} 6635.65 erreicht. Andernfalls erzielen die randomisierten Methoden lediglich bis zur Episode $Eps = 9$ bessere Ergebnisse als ein neu initialisiertes Training in D_T . Danach führen die randomisierten Methoden im Zero-Shot-Transfer zu keiner Verbesserung der Resultate. Dies verdeutlicht die Notwendigkeit eines Few-Shot-Transfers, der eine Anpassung der Policies an die spezifischen Problemstellungen vorsieht.

Ausgehend von einer Definition des erstmaligen Erreichens einer vollständigen Lösung als Threshold zeigen die transferierten Methoden einen Vorteil auf. In einem neu initialisierten Standard-Training wird das CHRP erst im Bereich von Episode Eps 61 bis 103 erstmals von den MARL-Methoden DDPG und TD3 gelöst. Im TL tritt dieses Ereignis bereits im Abschnitt von Eps 13 bis 27 auf (vgl. Abbildung 5.19a). In der Nutzung eines Zero-Shot-Transfers von randomisierten Methoden sind lediglich die Episoden im $TD3_{rand}$ hervorzuheben, die zu einem Reward \mathcal{R} von 5263.12 führen (vgl. Tabelle 5.11). Auch hier könnte ein Few-Shot-Transfer zu verbesserten Ergebnissen führen.

In Abbildung 5.21 sind die elektrischen Verdrahtungen in der Source-Domäne D_S (vgl. Abbildung 5.21a) und im Transfer-Task der Target-Domäne D_T (vgl. Abbildung 5.21b) gegenübergestellt. Sie zeigen eine exemplarische Änderung im Produktentstehungsprozess am Beispiel einer Replatzierung eines Steckers. Der betroffene Stecker mit der BMKZ –P69 (vgl. Tabelle 5.1) ist in Abbildung 5.21b hervorgehoben und hat die neue Position $[2800.00, -840.00, 453.00]$ mit adaptierter Rotation.

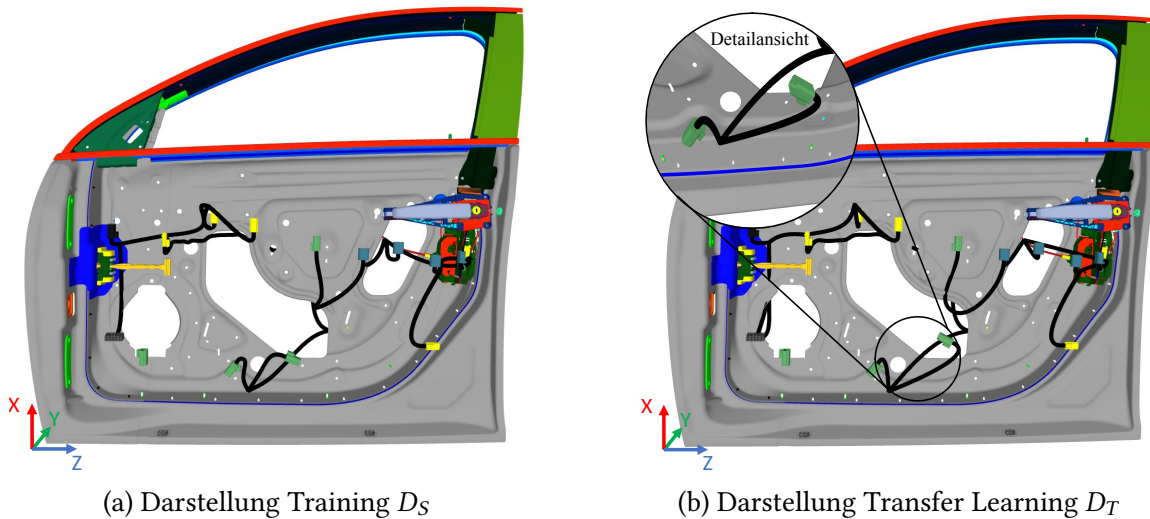


Abbildung 5.21: Gegenüberstellung der Transfer-Learning-Ergebnisse im 3D-CAD-System für Validierungsfall 1 mit Kabelbaum

Abbildung 5.21 zeigt dabei einen hohen Grad an Ähnlichkeit zwischen der Lösungsstrategie in D_S (vgl. Abbildung 5.21a) und dem Resultat im TL (vgl. Abbildung 5.21b). Dies ist darauf zurückzuführen, dass sich der Output des neuronalen Netzes bei einem gleichbleibenden Input ohne die Zugabe von Noise nicht ändert (vgl. Kapitel 2.4.2). Agiert einer der Agenten a^i also in einem vergleichbaren geometrischen 3D-CAD-Raum mit

nahezu identischer Observation O^i , dann wird auch eine analoge Aktion \mathcal{A}^i ausgeführt. Demzufolge sind die meisten der elektrischen Verbindungen in den Abbildungen 5.21a und 5.21b identisch.

5.4 Zusammenfassung und Diskussion der Validierungsergebnisse

Zur Validierung der Einsatzfähigkeit der in Kapitel 4 entwickelten Methoden für die automatisierte virtuelle Verdrahtung im 3D-CAD-System werden in Kapitel 5 verschiedene Experimente in zwei unterschiedlichen Validierungsfällen durchgeführt (vgl. Kapitel 5.2). Ausgehend von unterschiedlichen Metriken für das Lernverhalten und die Einhaltung der definierten Randbedingungen (vgl. Kapitel 5.1) wird überprüft, ob die neuen Methoden für den vorgesehenen Einsatzzweck und dessen Anforderungen tauglich sind (vgl. Kapitel 3.3). Dafür ist im initialen Schritt die Überführung der exemplarischen 3D-CAD-Geometrien in Environments für das definierte POMG erforderlich. Aus dieser Transformation lässt sich ableiten, dass eine Voxel-basierte Darstellungsform der 3D-CAD-Geometrien nicht die notwendige Präzision zur genauen Kollisionsidentifikation bietet. Vielmehr würde eine angemessene Auflösung eine übermäßige Datenmenge verursachen, die den Lernprozess der MARL-Methoden beeinträchtigt. Die Diskretisierung des Bewegungsraums der Agenten a^i belegt diese Schlussfolgerung, indem sie die Optimalität des Verdrahtungsergebnisses an die gewählte Kantenlänge eines Voxels bindet. Diese Erkenntnis trifft nicht nur im ersten Validierungsfall zu, der durch dünnwandige Einzelteile und geometrische Freiformen charakterisiert ist, sondern auch im zweiten Validierungsfall. Dort führt die Nutzung von Voxeldarstellungen als Environment aufgrund der großen geometrischen Abmessungen ebenfalls zu einer Einschränkung der Ergebnisqualität. Die Nutzung von Mesh-basierten und kollisionsorientierten Darstellungsformen der 3D-CAD-Geometrie zeigt einen erheblichen Vorteil. Dieser Mehrwert ergibt sich insbesondere in der Generierung der informationstechnischen Basis für die Konstruktion von Kabelbäumen sowie in der Optimierung der Berechnungszeiten von Kollisionen (vgl. Kapitel 5.3.5).

Für die Validierung des Navigation-Graphs als Grundlage der Kabelbaumerstellung zeigt Kapitel 5.3.1, dass die Wahl des Branching-Faktors w_B großen Einfluss auf das Ergebnis und damit auch auf die spätere Verwendung für die MARL-Methode hat. So ist in einer direkten Gegenüberstellung in *Validierungsfall 1* die kumulierte Gesamtlänge der Einzelverbindungen für $w_B = 0.9$ zwar um 55.08 % kleiner als in Szenario $w_B = 0.01$, dafür aber auch deutlich weniger gebündelt. Unter Berücksichtigung des Optimierungsziels (vgl. Kapitel 4.1.4) und der daraus resultierenden Anforderung an eine **größtmögliche Bündelung** der Kabel (vgl. Kapitel 3.3) ist das Ergebnis für $w_B = 0.01$ als positiver zu bewerten. Diese Schlussfolgerung lässt insbesondere die Analyse der Gesamtlänge des entstehenden Subgraphen H_α zu. Diese Beobachtung wird in *Validierungsfall 2* verstärkt, da hier das Szenario für $w_B = 0.9$ einen um 61.56 % längeren Subgraphen erzeugt als die Validierung mit der Festlegung von $w_B = 0.01$.

Die Validierung der Ergebnisse für das Standardtraining in den *Validierungsfällen 1 und 2* ist der Hauptbestandteil des Kapitels 5.3.3. Sie zeigt neben der grundsätzlichen Fähigkeit der MARL-Methoden zur Lösung des CHRP auch die Integrationsfähigkeit von Randbedingungen wie der Kollisionsfreiheit (vgl. Kapitel 2.3.3) und die Berücksichtigung des minimalen Biegeradius (vgl. Kapitel 3.3). Auch die Zielsetzung der **minimalen Pfadlänge** aller Einzelagenten a^i (vgl. Kapitel 3.3) ist wesentlicher Bestandteil des zu erreichenden Optimierungsziels (vgl. Kapitel 4.1.4). Beim Vergleich der drei Methoden (DDPG, TD3

und Communication) über eine Lerndauer von 100 Episoden mit jeweils 250ts liefern die Experimente in beiden Validierungsfällen hinsichtlich der Einhaltung der Randbedingungen vergleichbare Ergebnisse. Alle drei Methoden sind in der Lage, die Kollisionen pro Episode kontinuierlich zu minimieren, und reduzieren die Überschreitungen der maximalen Oberflächendistanz auf nahezu null. Gleichzeitig zeigen die Abbildungen zum Gesamt-Reward je Episode (vgl. Abbildungen 5.8a und 5.10a), dass die Methoden auf Basis der Rewards \mathcal{R} ihre Policy fortwährend verbessern. Daraus resultiert, dass die Anforderung der **Integration von Randbedingungen**, insbesondere der des minimalen Biegeradius und der Kollisionsfreiheit, als erfüllt angesehen werden kann (vgl. Kapitel 3.3). Gleichwohl offenbaren die Lernkurven sowie die zugehörige numerische Analyse, dass nicht alle angewendeten Methoden das CHRP innerhalb der festgelegten Anzahl an ts je Episode vollständig lösen können (vgl. Tabelle 5.5 und 5.6). Somit lässt sich schlussfolgern, dass die Methoden zwar grundsätzlich dazu in der Lage sind, die Problemstellung des CHRP zu lösen. Jedoch korrespondiert die hohe Anzahl an Agenten mit einer erheblichen Zunahme der Problemkomplexität.

Die Experimente in Kapitel 5.13 erweitern das statische CHRP um die Berücksichtigung von Kinematik innerhalb der 3D-CAD-Baugruppe. Hier nehmen exemplarische **dynamische Einzelteile** (vgl. Kapitel 3.3) direkt bei der automatisierten, virtuellen Verdrahtung Einfluss auf das CHRP. Konträr zu den Resultaten im statischen Anwendungsfall ist für die Metriken innerhalb des Lernprozesses von 200 Episoden keine Konvergenz zu beobachten (vgl. Abbildung 5.13). Obwohl die Reward-Kurve auch in diesem Fall eine Verbesserung erkennen lässt, ist sie gleichzeitig größeren numerischen Schwankungen ausgesetzt (vgl. Tabelle 5.7). Dennoch ist es mittels der eingesetzten MARL-Methoden möglich, nahezu kollisionsfreie Pfade ohne Überschreitung der maximalen Oberflächendistanz zu erlernen. Es lässt sich jedoch feststellen, dass trotz der geringen Anzahl von Kabeln mit $N = 4$ das CHRP in keiner der analysierten Episoden vollständig gelöst wird. Dennoch weisen die identifizierten Wegstrecken eine gültige Integration der kinematischen Einflüsse auf. Daher kann diese Anforderung als teilweise erfüllt betrachtet werden (vgl. Kapitel 3.3).

Kapitel 5.3.5 verbindet die Ergebnisse der Navigation-Graph-Generierung (vgl. Kapitel 5.3.1) mit dem Standardtraining (vgl. Kapitel 5.3.3). Die neue Methode ermöglicht dabei neben der Skalierung der Agentenanzahl N auch die domänenspezifische Anforderung einer Kabelbaumerzeugung. Die Vorgabe einer **minimalen Gesamtpfadlänge** wird in diesen Experimenten geringer gewichtet, da das Hauptaugenmerk auf der maximalen Bündelung liegt, die mit einer Problem-Decomposition in Einklang gebracht wird. So zeigen die Experimente in Kapitel 5.3.5 im ersten Schritt, welchen Einfluss der Faktor w_B auf die adaptierte Kardinalität zwischen Agent und elektrischer Verbindung hat. In direkter Konsequenz der Bündelung erhöht sich zwar die Agentenanzahl N in *Validierungsfall 1* von 13 auf 23 Agenten. Jedoch folgt daraus nicht nur eine reduzierte Problemkomplexität, sondern auch ein verringerter, wechselseitiger Einfluss zwischen den Agenten. Die Ergebnisse zeigen im Vergleich zum Standardtraining in *Validierungsfall 1* nicht nur besseres Lernverhalten für den Gesamt-Reward (vgl. Abbildung 5.15a), sondern gehen auch mit einer höheren Quote an vollständigen Kabelpfaden im CHRP einher. Gleichzeitig sinkt die Anzahl der Kollisionen pro Episode (vgl. K_{min} TD3 in den Tabellen 5.5 und 5.8). Obgleich variiert, welche der MARL-Methoden die beste Performance aufweist, zeigen die TD3- und DDPG-gestützten Methoden die höheren Werte für den maximalen Reward \mathcal{R}_{max} .

Somit kann die Anforderung an ein **Branching** auch im Zusammenhang mit der MARL-Methode als erfüllt angesehen werden (vgl. Kapitel 3.3), da die entwickelten Methoden einen Kabelbaum als Ergebnis für das 3D-CAD-System liefern. Es ist zu erkennen, dass die Communication-Methode in *Validierungsfall 2* neben einem schnelleren Anstieg der Rewards \mathcal{R} (vgl. Abbildung 5.17a) auch ein deutlich niedrigeres Level an Kollisionen K_{min} erreicht (vgl. Abbildung 5.17b). Dieses Ergebnis impliziert, dass die Kombination der beiden Problem-Decompositions in diesen Anwendungsfällen vorteilhaft ist.

Im abschließenden Kapitel 5.3.6 steht der **Wissenstransfer** zwischen den CHRP im Fokus (vgl. Kapitel 3.3). Der erste Schritt der Validierung der neuen TL-Methode beinhaltet die Durchführung eines randomisierten Trainings in der Source-Domäne D_S . Hier weisen die Experimente leichte Abweichungen zum Standardtraining bei der Anwendung von TD3, DDPG und PPO auf. Während sich die Werte für den maximalen Reward \mathcal{R}_{max} von TD3 und DDPG in einem ähnlichen Wertebereich befinden, zeigt insbesondere die Metrik für die minimalen Kollisionen K_{min} je Episode Abweichungen. Auffällig ist dabei, dass die Kollisionsanzahlen für beide Methoden trotzdem ein niedriges Level erreichen. Dies kann einerseits auf vielfältigere Lern-Tupel zurückzuführen sein, andererseits auch auf eine einfachere Erreichbarkeit der Anschlusspunkte P_T hinweisen. Aus den Resultaten der Validierung in Abbildung 5.19 kann gefolgert werden, dass das Erlernen einer sinnvollen Policy π auch in einem randomisierten Environment für beide MARL-Methoden möglich ist. Im zweiten Schritt der Validierung erfolgen eine Übertragung der erlernten Strategien (Policy π) auf die Zieldomäne D_T und eine Gegenüberstellung von randomisierten Trainingsläufen mit Policys aus dem Standardtraining (vgl. Abbildung 5.20). Aus den Übertragungsergebnissen lässt sich ableiten, dass insbesondere die Einhaltung der maximalen Oberflächendistanz keine signifikante Übertragungsfähigkeit darstellt (vgl. Abbildung 5.20c). Die Anzahl der Kollisionen hingegen zeigt im Transfer bessere Werte als im zugehörigen randomisierten Training in D_S . Besonders auffallend ist hierbei, dass im Rahmen des Transfers die DDPG-basierte Methode die Aufgabe der Zieldomäne T_T vollständig erfüllt. Dies erklärt sich durch eine ausgeprägte Ähnlichkeit zwischen dem ursprünglichen CHRP und dem Transfer-Task. Aufgrund der hohen Variabilität in den Ergebnissen und der nur eingeschränkt möglichen Anpassung der MARL-Methoden kann die Anforderung des **Wissenstransfers** (vgl. Kapitel 3.3) folglich nur begrenzt als erfüllt betrachtet werden.

Abschließend zeigen die Ergebnisse, dass sich die in Kapitel 4 entwickelten Methoden für die automatisierte virtuelle Verdrahtung im 3D-CAD-System eignen. Es ist zu berücksichtigen, dass die Validierung den Fokus auf den verbesserten Einsatz von MARL-Methoden in einem CHRP legt. Als Benchmark-Algorithmus für ein MAPF innerhalb eines kontinuierlichen dreidimensionalen Raumes wird in der Fachliteratur die Anwendung eines RRT oder einer PRM vorgeschlagen (vgl. Kapitel 3.2.1). Aufgrund der Tatsache, dass dieser Algorithmus keine optimalen Lösungen gewährleistet und das CHRP als NP-schweres Problem nicht exakt lösbar ist, ist ein direkter Vergleich mit der tatsächlichen optimalen Lösung nicht möglich. Die Verwendung des Branchings als Problem-Decomposition für das CHRP zeigt in allen Validierungsszenarien der MARL-Methoden bessere Ergebnisse als die direkte Verwendung von MARL-Methoden. Dies unterstreicht, dass die in Kapitel 4 entwickelten Methoden eine sinnvolle Erweiterung der Lösungsalgorithmen im Anwen-

dungsfall des CHRP bieten. Aufgrund der modularen Struktur der Methoden (vgl. Kapitel 4.2.1) ist es möglich, einzelne Komponenten individuell zu nutzen oder zu substituieren.

6 Zusammenfassung und Ausblick

Dieses Kapitel fasst die Ergebnisse dieser Arbeit zur Entwicklung einer Methode zum Einsatz von MARL für die virtuelle Verdrahtung im 3D-CAD mit deren zugehörigen Validierungen zusammen. Dabei werden die in Kapitel 1.3 definierten Ziele auf ihre Gültigkeit überprüft und den Ergebnissen aus der Validierung (vgl. Kapitel 5) in Kapitel 6.1 gegenübergestellt. Ein Ausblick auf weitere offene Forschungsansätze im Themenfeld der automatisierten virtuellen Verdrahtung ist in Kapitel 6.2 ausgeführt.

6.1 Zusammenfassung der Ergebnisse und Fazit

Kürzere Lebenszyklen von Produkten führen zu einer erhöhten Anforderung an Flexibilität im Prozess der Produktentwicklung. Die fortschreitende Elektrifizierung dieser individualisierten Produkte erhöht den Druck auf Unternehmen zusätzlich, ihre Planungsprozesse flexibler und effizienter zu gestalten. Es ist zu beachten, dass die manuelle Definition optimaler Kabelverläufe in eingeschränkten mechanischen Räumen mit einem erheblichen zeitlichen Aufwand verbunden ist (vgl. Kapitel 2.3.5). Obwohl 3D-CAD-Systeme den Prozess virtuell bereits unterstützen, ist es dennoch erforderlich, einzelne Wegpunkte für sämtliche elektrischen Verbindungen festzulegen. Insbesondere unter Berücksichtigung aller einzuhaltenden Randbedingungen steigt die dafür erforderliche Zeit überproportional. In Anbetracht der dargestellten Problematik zielt diese Arbeit darauf ab, Methoden zur Automatisierung der virtuellen Verdrahtung innerhalb des 3D-CAD-Systems zu entwickeln. Diese haben auch zum Ziel, die Gesamtkabellänge unter Berücksichtigung der geltenden Randbedingungen zu optimieren (vgl. Kapitel 1.3 und 2.3.3).

Kapitel 3 liefert eine Übersicht und eine Analyse der gegenwärtig existierenden Ansätze zur Lösung des Cable Harness Routing Problem oder artverwandter Problemstellungen. Aus der Analyse des Standes der Forschung und Technik lassen sich die wesentlichen Forschungsbedarfe ableiten (vgl. Kapitel 3.3). Diese bestehen insbesondere im Bereich der Multi-Agenten-Methoden in kontinuierlichen dreidimensionalen Räumen unter Berücksichtigung der Anforderungen an eine Kabelbaumerstellung und die Integration vielseitiger Randbedingungen (vgl. Tabelle 3.1).

Kapitel 4 zeigt auf, wie die bestehenden Forschungslücken mit den neuen Methoden auf Basis von MARL geschlossen werden. Diese Methoden bestehen aus einem dreistufigen Verfahren. Im initialen Schritt erfolgen die Konsolidierung der Eingangsdaten mittels AutomationML und die Überführung in ein Environment für ein POMG (vgl. Kapitel 4.3.2). Im zweiten Schritt kombiniert die Communication-Methode die Verfahren zur Problem-Decomposition mit den MARL-Methoden zur Informationsübertragung zwischen Agenten (vgl. Kapitel 4.4.2). Dafür werden die Agenten auf Basis exemplarischer Pfade unterschiedlichen Clustern zugeordnet, die jeweils eine individuelle Policy π_{C_n} erlernen. Ergänzt wird dies durch die Methode zur domänenspezifischen Problem-Decomposition mittels Kabelbaumgenerierung (vgl. Kapitel 4.5.1). Dieser Ansatz macht sich die Mesh-Strukturen der 3D-CAD-Geometrien aus dem AutomationML-Input zunutze, um eine Subgraphstruktur als Eingangsgröße für die MARL-Algorithmen zu erzeugen (vgl. Kapitel 4.5.1.1). Beide Methoden verfolgen das Ziel einer Zerlegung des CHRP in wechselwirkungsfreie Subprobleme. Mit der Erweiterung der Trainingsmethode um die Zielpunktrandomisierung erfüllt die Methode die Anforderungen an die automatisierte virtuelle Verdrahtung im 3D-CAD (vgl. Kapitel 3.3).

Die Eignung der neuen Methoden wird durch Experimente (vgl. Kapitel 5) unter Einsatz verschiedener Szenarien evaluiert. Der erste Abschnitt der Validierung untersucht die Methoden zur Generierung der Eingangsdaten für die Kabelbaumerstellung auf dem Alpha-Wrap-Mesh der 3D-CAD-Geometrie. Zur Quantifizierung des Lernverhaltens sowohl in Anwendungsfällen mit als auch ohne Kinematik zeigen die Kapitel 5.3.3 bis 5.3.5 die Ergebnisse der entwickelten Methoden aus Kapitel 4.2.4. Die Experimente zur Übertragbarkeit der Policies π ergänzen die Versuchsreihen an den 3D-CAD-Baugruppen (vgl. Kapitel 5.3.6).

Die Resultate der Validierung demonstrieren, dass die neuen Methoden die Anforderungen an eine automatisierte virtuelle Verdrahtung im 3D-CAD-System partiell erfüllen (vgl. Kapitel 1.3).

Basierend auf den Ergebnissen der Validierung werden die drei in Abschnitt 1.3 definierten Zielsetzungen dieser Arbeit bewertet. Das *erste Ziel* kann nur in Teilen als erfüllt bewertet werden. Dabei stellt die Fähigkeit, ein CHRP ohne Einschränkungen hinsichtlich Geometrie oder Komplexitätsgrad zu lösen, eine wesentliche Anforderung an die neuen Methoden dar (vgl. Kapitel 3.3). Durch eine Trennung der automatisierten virtuellen Verdrahtung vom spezifischen 3D-CAD-System (vgl. Kapitel 4.2.1) sowie durch die Überführung dieser Eingangsdaten in ein Environment (vgl. Kapitel 4.3.2) ist die Lösungsfähigkeit der MARL-Methoden nicht an die Geometrie des 3D-CAD-Systems gebunden (vgl. Kapitel 4.2.1). Darüber hinaus zeigen die Resultate der Experimente in Kapitel 5.3.3, dass die neuen Methoden dazu in der Lage sind, den manuellen Prozess der Definition von Wegstreckenpunkten zu automatisieren. Die Optimalität der Verdrahtungslösung korrespondiert mit der Definition des zu erreichenden Optimierungsziels des POMG (vgl. Kapitel 4.1.4). Aufgrund der unzureichenden Bestimmbarkeit einer tatsächlich optimalen Lösung kann das erste Ziel nur teilweise akzeptiert werden (vgl. Kapitel 5.3.3).

Das *zweite Ziel* (vgl. Kapitel 1.3) kann umfänglich angenommen werden. Ein fehlender zentraler und konsolidierter Datenzugriffspunkt für die mechanischen und elektrischen Eingangsdaten der virtuellen Verdrahtung ist ursächlich für zahlreiche Medienbrüche im gesamten Prozess (vgl. Kapitel 2.3.5). Durch die Etablierung eines Single Point of Truth in Form einer AutomationML-Datenstruktur als interdisziplinäres Datenaustauschformat kann die neue Methode auf einem einzigen konsistenten Datenpunkt aufbauen (vgl. Kapitel 4.2). Darüber hinaus entfällt die starre, regelbasierte Einbindung von Randbedingungen, indem die limitierenden Faktoren direkt aus den Eingangsdaten extrahiert werden (vgl. Kapitel 4.2.2). Die von den CAD-Systemen entkoppelte Datenverarbeitung, die Übertragung der Informationen zwischen den beteiligten IT-Systemen mittels AutomationML und die Extraktion der einzuhaltenden Randbedingungen aus den Eingangsdaten sind die wesentlichen Bestandteile zur Erfüllung der Anforderungen aus dem *zweiten Ziel* (vgl. Kapitel 1.3).

Das *dritte Ziel* (vgl. Kapitel 1.3) wird nur teilweise erreicht. Die derzeitige manuelle Durchführung der virtuellen Verdrahtung resultiert in einer unzureichenden Übertragung von Lösungsstrategien zwischen verschiedenen CHRP. Die Automatisierung durch MARL-Methoden führt nicht nur zu einer Verringerung der aufzuwendenden zeitlichen Ressourcen, die für den Prozess erforderlich sind, sondern ermöglicht auch die Übertragung von Lösungsstrategien. Die Anwendung des TL verdeutlicht, wie eine Zielpunkttransdomisierung dafür sorgt, dass Overfitting auf einzelne CHRP vermieden wird (vgl. Kapitel 4.5.2.1). Infolgedessen erlernen die MARL-Methoden eine Policy π , die fähig ist, auch kontinuierlich modifizierte CHRP zu lösen (vgl. Kapitel 4.5). Dies erfüllt die Anforderungen eines agilen Produktentstehungsprozesses. Die Ergebnisse in Kapitel 5.3.6 belegen, dass die MARL-Methoden auch bei einer Veränderung der Zielpunktpositionen im Laufe der Trainingsphase eines CHRP die Problemlösungsfähigkeit erhalten. Aus den Ergebnissen des Zero-Shot-Transfers geht hervor, dass die Übertragbarkeit der Einhaltung von Randbedingungen variiert. Während die Einhaltung der maximalen Oberflächendistanz sehr gut übertragbar ist (vgl. Abbildung 5.20c), zeigt die Qualität des Transfers zur Fähigkeit der

Kollisionsvermeidung Unterschiede (vgl. Abbildung 5.20b). Dies beeinflusst entsprechend den Reward (vgl. Abbildung 5.20a) und die Transfer-Metriken (vgl. Kapitel 2.6.5). Da mittels der neuen Methoden kein durchgehend positiver Transfer erreicht werden kann, ist das dritte Ziel folglich nur teilweise erfüllt.

Insgesamt werden durch diese Arbeit alle gestellten Ziele aus Kapitel 1.3 ganz oder teilweise erfüllt. Auf die offenen Forschungsfragen aus Tabelle 3.1 werden Antworten gegeben. Diese zeigen neben der Konsolidierung von heterogenen Daten auch die Integrationsfähigkeit von MARL-Methoden in bestehende Produktentstehungsprozesse in 3D-CAD-Systemen.

6.2 Ausblick

Die im Rahmen dieser Arbeit entwickelten Methoden erfüllen die theoretischen Anforderungen an eine automatisierte virtuelle Verdrahtung in einem 3D-CAD-System (vgl. Kapitel 3.1). Zur Nutzung der Methode für die Beschleunigung der realen Produktentwicklung muss neben der Integration zusätzlicher Randbedingungen auch die weitere Skalierbarkeit in Betracht gezogen werden.

Im Rahmen der Berücksichtigung oder Erweiterung der integrierten Randbedingungen ist zu erforschen, welche dieser Einflussgrößen in die bestehenden Methoden integrierbar sind (vgl. Kapitel 2.3.3). Dies umfasst allgemeine Randbedingungen wie maximale Füllgrade sowie Restriktionen auf der bauteilspezifischen Ebene. Ein exemplarisches Beispiel ist die Einhaltung eines Mindestabstands zwischen einer Teilbaugruppe und einem bestimmten Kabeltyp aufgrund von EMV-Anforderungen (vgl. Kapitel 2.3.3). Des Weiteren eröffnet die spezifische Randbedingung der Kinematik Möglichkeiten für vertiefte Forschungsaktivitäten (vgl. Kapitel 2.3.4). So sind weitere Gelenktypen in die bestehende Methode zu integrieren oder alternative Übertragungsformen der zugehörigen Bewegungsinformationen in das POMG müssen untersucht werden (vgl. Kapitel 4.2.4). Diese Erweiterungen bieten die Möglichkeit, zusätzliche Anwendungsdomänen abzudecken.

Eine weitere Ausprägung von Weiterentwicklungspotenzial umfasst unterschiedliche Arten von Einflussnahmen seitens der Endanwender auf den automatisierten virtuellen Verdrahtungsprozess im 3D-CAD. Neue methodische Ansätze können das interaktive Feedback der Nutzer in die Anpassung der Optimalitätsdefinition einbeziehen, was eine nutzer- und anwendungsspezifische Modifikation der Ergebnisse in Verbindung mit den entwickelten Methoden ermöglicht (vgl. Kapitel 4.1.4). Eine weitere konkrete Einflussnahme auf die Lösung des CHRP stellt die Vordefinition von Bauraumreservierungen im 3D-CAD-System dar. Diese Praxis entspricht in vielen Bereichen in der Produktentwicklung einer etablierten Vorgehensweise, um bereits in frühen Projektphasen Kabelstränge bestimmten geometrischen Bereichen zuzuordnen. Gleichzeitig wird dadurch eine frühzeitige Reduktion potenzieller wechselseitiger Auswirkungen mit anderen Gewerken erreicht. Die Integration dieser Form der Vordefinition und somit die Einschränkung des Zustandsraums \mathcal{S} könnte daher Gegenstand zukünftiger Forschungen sein (vgl. Kapitel 4.3.2). Dies hat zudem Auswirkungen auf die Skalierung der Anzahl an Agenten und die geometrischen Dimensionen der 3D-CAD-Geometrie (vgl. Kapitel 4.5.1). Darüber hinaus hat die Untersuchung einer automatisierten Gerätepositionierung Potenzial für Weiterentwicklungen im Kontext des CHRP. Aufgrund der hohen Einflussnahme auf die Positionen der Anschlusspunkte im 3D-CAD-System kann dies ein lohnendes Forschungsfeld für künftige Untersuchungen darstellen. Die Qualifizierung der 3D-CAD-Objekte mit Anschlusspunkten sollte in der IEC oder der ISO festgelegt werden.

Im Rahmen der Lösungsmethoden erscheint es sinnvoll, die Integration und den Einfluss von Transformer-Architekturen auf die MARL-Methode zu analysieren [2]. Transformer bieten nicht nur eine verbesserte sequentielle Darstellung der einzelnen Environment-Zustände, sondern auch eine erhöhte Generalisierungsfähigkeit in modifizierten Zieltasks T_T . Hierbei kann es jedoch mitunter erforderlich sein, das Optimierungsproblem entsprechend zu adaptieren.

A Appendix

A.1 UML-Sequenzdiagramm

Die Implementierung der neuen Methoden folgt einer prozessualen Abfolge, die die Interaktion zwischen dem Endanwender, den CAD-Systemen und den KI-Methoden beschreibt. Diese Prozessschritte werden in Form eines UML-Sequenzdiagramms in Abbildung A.1 veranschaulicht. Die Abbildung dient als detaillierte Beschreibung der datentechnischen Integration der neuen Methoden in die vorhandene IT-Systemlandschaft der virtuellen Verdrahtung (vgl. Kapitel 4.2.1).

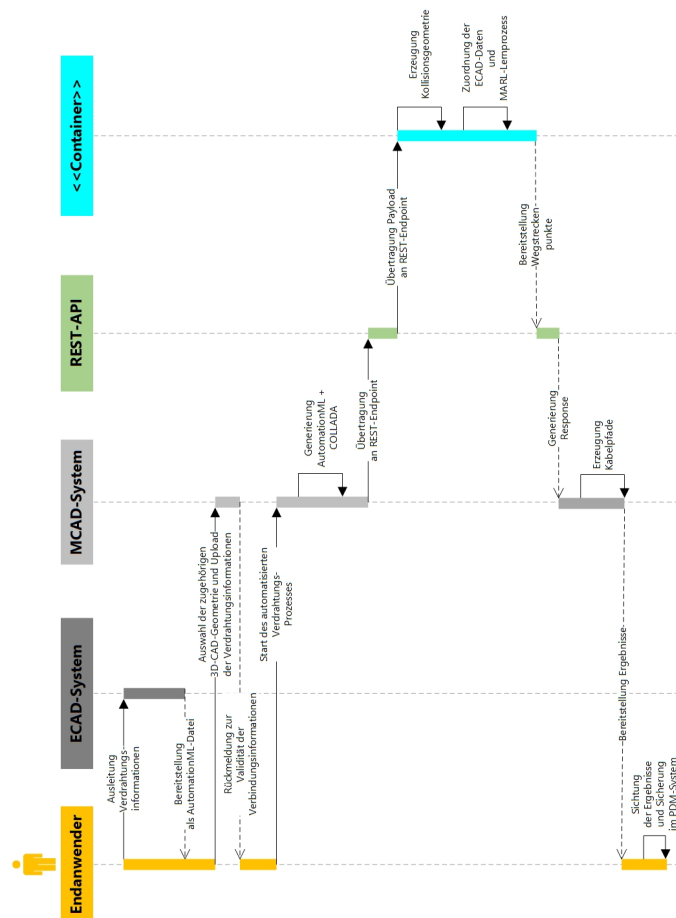


Abbildung A.1: UML-Sequenzdiagramm für die Anwendung der neuen Methode im 3D-CAD-System

A.2 Twin-Delayed Deep Deterministic Policy Gradient

Der Algorithmus TD3 stellt eine Weiterentwicklung der etablierten RL-Methode DDPG dar (vgl. Kapitel 2.5). Die Basis bildet eine Actor-Critic-Architektur in einem Off-Policy-Algorithmus, der ausschließlich für die Anwendung in Environments mit kontinuierlichem Aktionsraum geeignet ist [3]. Ebenso wie im DDPG kommt auch in einem TD3 ein Experience-Replay-Buffer zum Einsatz, der es ermöglicht, dass die Korrelationen zwischen Beispieldupeln (en: *Samples*) reduziert werden [62, 101]. Darüber hinaus weist der TD3 drei wesentliche Differenzierungsmerkmale zu einem DDPG auf [54]:

1. Clipped Double Q-Learning
2. Verzögertes Update der Policy π
3. Integration von Rauschen (en: *Noise*) im Actor-Netzwerk

Das Clipping stellt dabei sicher, dass die Q-Werte nicht überschätzt und innerhalb eines kontrollierbaren Bereichs gehalten werden. Diese Problematik ist als *Overestimation Bias* definiert [54]. Um diesem Problem entgegenzuwirken, beinhaltet der TD3 zwei Q-Funktionen, die gleichzeitig parallel trainiert werden. Nur der jeweils kleinere der beiden Q-Werte nimmt dann im Lernprozess Einfluss. Damit ist die Schätzung stabiler und genauer. [40]

Eine Verzögerung der Actor-Policy-Updates verfolgt das identische Ziel eines stabileren Lernprozesses. Entgegen der Vorgehensweise im DDPG wird die Actor-Policy π nicht wie die Critic-Netzwerke nach jedem Zeitschritt aktualisiert, sondern in der Regel alle 2 oder 3 ts . Dies sorgt für stabilere Updates und verhindert, dass das Lernen zu stark von den neuesten Änderungen abhängt. Gleichzeitig sind damit Exploration und Exploitation besser ausbalanciert. [38]

Die Integration von Noise in der Aktionsauswahl sorgt dafür, dass die Q-Werte nicht zu empfindlich auf kleine Änderungen der Aktionswahl reagieren. Dies stabilisiert das Lernen und verhindert übermäßige Schwankungen in der Policy π [54]

Aufgrund des Anwendungsgebietes von kontinuierlichen Aktionsräumen, des konstanten Lernprozesses und der Mechanismen zum Ausbalancieren von Exploration und Exploitation (vgl. Kapitel 2.5 und 2.5.4) eignet sich dieser Algorithmus besonders für die Integration in ein MAPF.

A.3 Proximal Policy Optimization

Für eine Gegenüberstellung mit anderen MARL-Algorithmen werden in den Experimenten in Kapitel 5.3 die Ergebnisse von TD3, DDPG und der Communication-Methode mit einer PPO verglichen. Alle RL-Methoden werden dabei in einem CTCE-Schema verwendet, in dem jeweils ein Agent ein Einzelkabel (vgl. Kapitel 5.3.3) oder einen Kabelstrang (vgl. Kapitel 5.3.5) repräsentiert.

Die Methode der PPO ist dabei der Gruppe der On-Policy-Algorithmen zuzuordnen (vgl. Kapitel 2.5) und sie stellt eine Weiterentwicklung der Trust Region Policy Optimization dar [53]. Es ist zu berücksichtigen, dass diese Art von Methoden weniger Sample-effizient ist, da neue Daten für jedes Update der Policy benötigt werden. Das bedeutet, im Lernprozess werden nur die neuesten Erfahrungen verwendet, die mit der derzeit gültigen Policy π gesammelt werden. Als Hauptmerkmale sind darüber hinaus die Stabilisierung des Trainingsprozesses mittels Policy-Clipping und die Surrogate-Loss-Funktion (vgl. Gleichung A.1) zu nennen [28]. Letztere ist eine Modifikation des Policy-Gradient-Loss.

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t] \quad (\text{A.1})$$

In Gleichung A.1 repräsentiert $r_t(\theta)$ das Wahrscheinlichkeitsverhältnis zwischen der aktuellen ($\pi_\theta(A_t|S_t)$) und der alten Policy $\pi_{\theta_{old}}(A_t|S_t)$. Für den Fall $r_t(\theta) > 1$ ist die Action A_t wahrscheinlicher Teil der aktuellen Policy. Nimmt $r_t(\theta)$ einen Wert zwischen 0 und 1 an, ist die Aktion entsprechend bei der aktuellen Policy weniger wahrscheinlich. Diese Festlegungen sind die Grundlage der Divergenzberechnung (häufig auch als Likelihood-Ratio bezeichnet) von alter und neuer Policy in Gleichung A.2 [151].

$$r_t(\theta) = \frac{\pi_\theta(A_t|S_t)}{\pi_{\theta_{old}}(A_t|S_t)} \quad (\text{A.2})$$

Darüber hinaus beschreibt der Parameter \hat{A}_t in Gleichung A.1 die Advantage-Funktion, die besagt, wie vorteilhaft eine Aktion war. Die Variable ϵ ist der Clipping-Parameter, der die maximale Änderung des Netzwerkes festlegt. Die *min*-Funktion sorgt dafür, dass zu große Änderungen beschränkt werden. Falls $r_t(\theta)$ im Bereich von $[1 - \epsilon, 1 + \epsilon]$ liegt, bleibt die Änderung unbestraft [38]. Außerhalb dieses Bereichs wird die Verbesserung beschnitten (en: *clipping*), was zu einer regulierten, schrittweisen Aktualisierung der Policy π führt. [151]

A.4 Vergleichsalgorithmen Rapidly-exploring Random Tree und Probabilistic Roadmap

Zur Einordnung der Ergebnisse der MARL-Methoden werden die Resultate der Experimente in Kapitel 5.3 den Metriken eines RRT und einer PRM gegenübergestellt. Diese Methoden sind in die Kategorie der nicht lernenden Heuristiken einzuordnen (vgl. Kapitel 3.2.1). Beide sind dafür konzipiert, in kontinuierlichen 2D- oder 3D-Räumen kollisionsfreie

Pfade zwischen einem Startpunkt (P_S) und einem Zielpunkt (P_T) zu identifizieren, und werden in manchen Methoden auch für die Kabelpfadplanung eingesetzt [196]. Ihr wesentlicher Unterschied liegt in der Raumexploration und der darauf aufbauenden Auswahl des bestmöglichen Pfades. Ein RRT wählt in jedem Schritt einen neuen möglichen Punkt aus (Single-Query), wogegen eine PRM den freien Raum ermittelt und darauf aufbauend die namensgebenden Roadmaps erzeugt (Multi-Query). In Abbildung A.2 werden die Explorationsstrategien der beiden Methoden RRT und PRM in einer exemplarischen 3D-CAD-Umgebung gegenübergestellt.

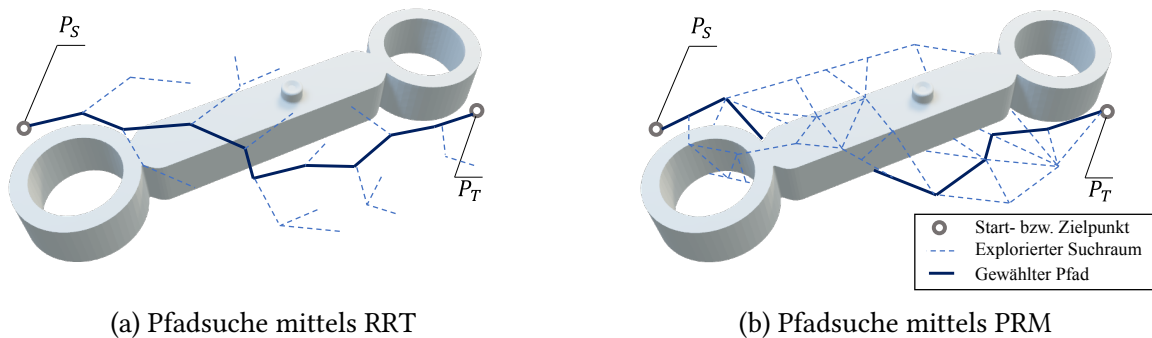


Abbildung A.2: Gegenüberstellung von RRT und PRM in einer exemplarischen 3D-CAD-Umgebung für eine Einzelverbindung

Abbildung A.2a zeigt die Explorationsstrategie des RRT, die darauf abzielt, hochdimensionale Suchräume durch eine zufällige Auswahl von Edges E nach potenziellen Pfaden zu untersuchen [149]. Aufgrund der zufälligen Selektion des nächsten Knotens kann schnell ein valider Pfad identifiziert werden, dieser ist jedoch nicht optimal. Der Parameter der maximalen Iterationsanzahl limitiert neben der maximalen Schrittgröße die Suche des RRT. In den Experimenten (vgl. Kapitel 5.3) ist die Schrittgröße analog zum Action Space der MARL-Methoden festgelegt.

Abbildung A.2b stellt dem Ansatz des RRT eine PRM gegenüber. Dem explorierten Suchraum ist zu entnehmen, dass eine festgelegte Anzahl an Punkten innerhalb des dreidimensionalen Raumes generiert wird [67]. Diese bilden im zweiten Schritt einen Graphen, in dem die jeweils nächstgelegenen Knoten miteinander verbunden sind. Bestehende Pfadsuchalgorithmen, z. B. ein A^* , können dann auf diesen Graphen angewendet werden [206]. Die einflussnehmenden Parameter einer PRM umfassen in den Experimenten neben der Anzahl der generierten Knoten E auch die Anzahl der nächstgelegenen Knoten k sowie die Menge der Punkte pro Flächeneinheit. Letzteres ist eine notwendige Anpassung für eine Vergleichbarkeit. Über diesen Parameter wird festgelegt, wie viele Punkte je Fläche des Alpha-Wrap-Mesh (vgl. Kapitel A.5) erzeugt werden. Diese Vorgehensweise ermöglicht eine Integration der maximalen Oberflächendistanz in die Pfadsuche mittels PRM (vgl. Kapitel 5.3). Bei einer zu hohen Anzahl an Flächen des Alpha-Wrap-Mesh ist die Anwendung der PRM auf diese Art und Weise jedoch nicht mehr möglich (vgl. Validierungsfall 2 in Kapitel 5.3.3 und 5.3.5).

A.5 Methode des Alpha Wrappings mit Offset

Die Anwendung der Alpha-Wrap-Methode auf Grundlage eines bestehenden Mesh zielt darauf ab, eine neue, glatte und vereinfachte Hülle (en: *Wrapper-Mesh*) zu erzeugen (vgl. Kapitel 4.3.1). Ein Einflussfaktor auf die Generierung des Wrapper-Mesh ist der namensgebende Kontrollparameter α_W . Dieser definiert den Einflussbereich und bestimmt folglich, wie weit das erzeugte Mesh von der ursprünglichen Geometrie abweichen darf. Ein kleiner Wert von α_W resultiert in einer Hülle, die nahe am Original-Mesh liegt, wogegen ein großes α_W eine glattere, abstrahiertere und ungenauere Darstellung erzeugt. Abbildung A.3 zeigt den Einfluss von α an einer exemplarischen 3D-CAD-Geometrie. [133]

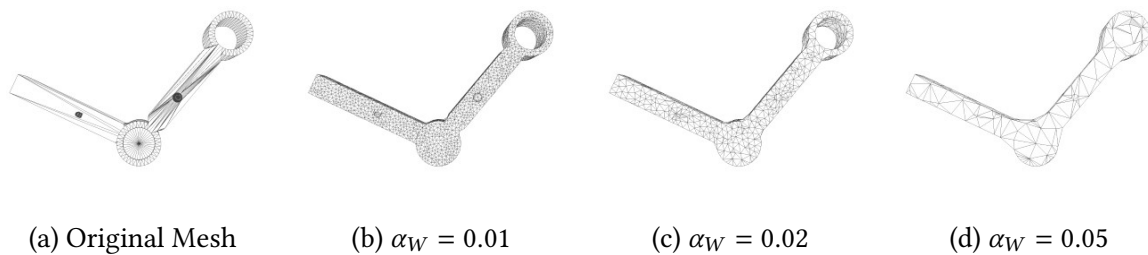


Abbildung A.3: Gegenüberstellung von Meshes auf Basis unterschiedlicher Werte von α_W i.A.a. [133]

Abbildung A.3a zeigt das ursprüngliche Mesh, das in Kapitel 5.3.4 als Validierungsgeometrie für den Einflussfaktor der Kinematik dient. In den Abbildungen A.3b bis A.3d sind die resultierenden Wrapper-Meshes mit unterschiedlichen Werten für α_W dargestellt. Diese verdeutlichen, dass mit einem zunehmenden Wert für α_W das ursprüngliche Mesh mit einer geringeren Anzahl an Vertices V und Edges E approximiert wird (vgl. Kapitel 2.1). Die Methode bietet insbesondere mit der Erweiterung um den Offset-Faktor δ_W eine Möglichkeit, bestehende Meshes als Eingangsgröße für das Branching-Verfahren zu verwenden (vgl. Kapitel 4.5.1.1). Dabei löst der Offset-Faktor δ_W die Problematik, dass berechnete Kabelpfade des Kabelbaums zum Teil innerhalb der 3D-CAD-Geometrie liegen (vgl. Kapitel 3.2.1 und 3.2.3).

A.6 Publikationsliste

Tabelle A.1: Publikationsliste

Titel	Autoren	Konferenz, Ort	DOI
Enhancing Monte-Carlo Tree Search with Multi-Agent Deep Q-Network in Open Shop Scheduling	O. Lohse, A. Haag, T. Dagner	WCMEIM 2022, Ma'anshan, China	10.1109/WCMEIM56910. 2022.10021570
Optimized planning of the integration of a reference plant into existing brownfield environments based on an entity model	J. Leidich, T. Dagner , P. Robl	DfX 2022, Hamburg, Deutschland	10.35199/dfx2022.14
Multi-Agent Deep Q-Network in Voxel-based Automated Electrical Routing	T. Dagner , R. Parzeller, S. Kesler	ICAISC 2023, Zakopane, Polen	10.1007/978-3-031- 42505-9_4
Individualized Clustered Cooperative Communication Units in Automated Electrical Routing in 3D CAD	T. Dagner , S. Kesler	INDIN 2023, Lemgo, Deutschland	10.1109/INDIN51400.2023. 10218004
A Dataset for Machining Feature Detection in Rotation Symmetric Components	S. Kesler, O. Lohse, T. Dagner	ICAC 2023, Birmingham, England	10.1109/ICAC57885.2023. 10275265
Automating the assembly planning process to enable design for assembly using reinforcement learning	R. Parzeller, D. Koziol, T. Dagner , D. Gerhard	DESIGN 2024, Dubrovnik, Kroatien	10.1017/pds.2024.220
Transfer-Learning in Multi-Agent Path-finding in Discrete 3D Environments	T. Dagner , M. Krähschütz, R. Parzeller	ICIEA 2024, Kristiansand, Norwegen	10.1109/ICIEA61579.2024. 10665019
Integrating Assembly Sequence Planning into Product Lifecycle Management	R. Parzeller, T. Dagner , O. Lohse, D. Gerhard	PLM 2024, Bangkok, Thailand	tbd
A Deep Reinforcement Learning Approach for Real-World 3D Facility Layout Problems	M. Krähschütz, T. Dagner , J. Leidich	SSCI 2025, Trondheim, Norwegen	tbd

A.7 Patente

Tabelle A.2: Patentanmeldungen

Titel	Anmelde- jahr	Anmelde- land	Amtliches Kennzeichen
Verfahren zum Betreiben einer Industrieumgebung mit zumindest zwei unterschiedlichen Anlagen mittels einer elektronischen Recheneinrichtung, Computerprogrammprodukt, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2023	WO, DE	PCT/EP2024/085274, 10 2023 212 481.5
Verfahren zum Bestimmen einer Bauteilstückliste zum Herstellen eines Produktes mittels einer elektronischen Recheneinrichtung, Computerprogrammprodukt, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2023	WO	PCT/EP2024/082196
Verfahren zum Erzeugen eines Rohrleitungs- und Instrumentenfließschemas, Computerprogrammprodukt, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2024	DE	10 2024 206 458.0
Verfahren zum Bestimmen einer Kabellänge eines Kabels für einen beweglichen Bereich für eine dynamische Maschine mittels einer elektronischen Recheneinrichtung, Computerprogrammprodukte, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2024	DE	10 2024 205 336.8
Verfahren zum Anlernen eines maschinellen Lernmodells zum Bestimmen eines Kabelverlaufs, Computerprogrammprodukt, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2024	DE	10 2024 204 989.1
Verfahren zum Erzeugen einer Rohrleitungs- und Instrumentenfließschemata-Stückliste für eine spezifische Maschine mittels einer elektronischen Recheneinrichtung, Computerprogrammprodukt, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2024	DE	10 2024 209 583.4

Titel	Anmelde- jahr	Anmelde- land	Amtliches Kennzeichen
Verfahren zum Erzeugen eines Anlagenaufbauplans für eine Anlage mittels einer elektronischen Recheneinrichtung, Computerprogrammprodukt, computerlesbares Speichermedium sowie elektronische Recheneinrichtung	2024	DE	10 2024 209 581.8
Verfahren und Vorrichtung zur Vereinheitlichung heterogener technischer Angaben	2024	DE	10 2024 209 579.6

Literatur

- [1] Akshat Agarwal, Sumit Kumar und Katia Sycara. *Learning Transferable Cooperative Behavior in Multi-Agent Teams*. 2019. DOI: 10.48550/arXiv.1906.01202.
- [2] Pranav Agarwal u. a. *Transformers in Reinforcement Learning: A Survey*. 2023.
- [3] Charu C. Aggarwal. *Artificial Intelligence*. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-72356-9. DOI: 10.1007/978-3-030-72357-6.
- [4] Charu C. Aggarwal und Chandan K. Reddy, Hrsg. *Data clustering: Algorithms and applications*. Chapman & Hall/CRC data mining and knowledge discovery series. Boca Raton, Fla. u.a.: CRC Press, 2014. ISBN: 978-1466558212.
- [5] A. Albers u. a. "Evaluation of Engineering Changes Based on Variations from the Model of PGE – Product Generation Engineering in an Automotive Wiring Harness". In: *Proceedings of the Design Society 2* (2022), S. 303–312. DOI: 10.1017/pds.2022.32.
- [6] Jean-Marc Alkazzi und Keisuke Okumura. "A Comprehensive Review on Leveraging Machine Learning for Multi-Agent Path Finding". In: *IEEE Access* 12 (2024), S. 57390–57409. DOI: 10.1109/ACCESS.2024.3392305.
- [7] Moritz Altner u. a. "Improving engineering change management by introducing a standardised description for engineering changes for the automotive wiring harness". In: *Procedia CIRP* 109 (2022), S. 538–543. ISSN: 22128271. DOI: 10.1016/j.procir.2022.05.291.
- [8] American National Standards Institute. *ANSI Y14.26M: Digital Representation for Communication of Product Definition Data (IGES)*. New York, USA, 1981.
- [9] Andi Asmara. *Pipe routing framework for detailed ship design*. Delft: VSSD, 2013. ISBN: 9789065623263.
- [10] Klaus Backhaus u. a. "Clusteranalyse". In: *Multivariate Analysemethoden*. Hrsg. von Klaus Backhaus u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, S. 453–516. ISBN: 978-3-662-46075-7. DOI: 10.1007/978-3-662-46076-4_9.
- [11] Ettore Bartolucci und George Matto. "Cable Able". In: *IEEE Industry Applications Magazine* 13.2 (2007), S. 24–35. ISSN: 1077-2618. DOI: 10.1109/MIA.2007.322272.
- [12] Jacob Beck u. a. *A Survey of Meta-Reinforcement Learning*. 2023. DOI: 10.48550/arXiv.2301.08028.
- [13] Eseoghene Ben-Iwhiwhu u. a. "Lifelong Reinforcement Learning with Modulating Masks". In: (2022). DOI: 10.48550/arXiv.2212.11110.
- [14] M. Blokland u. a. "Literature Survey on Automatic Pipe Routing". In: *Operations Research Forum* 4.2 (2023). DOI: 10.1007/s43069-023-00208-5.

- [15] Guillaume Bono u. a. “Cooperative Multi-agent Policy Gradient”. In: *Machine Learning and Knowledge Discovery in Databases*. Hrsg. von Michele Berlingerio u. a. Bd. 11051. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, S. 459–476. ISBN: 978-3-030-10924-0. DOI: 10.1007/978-3-030-10925-7_28.
- [16] Mario Botsch u. a. *Polygon mesh processing*. Natick, Mass.: A K Peters, 2010. ISBN: 978-1568814261. URL: <https://permalink.obvsg.at/AC08232403>.
- [17] Ulisses Braga-Neto. *Fundamentals of Pattern Recognition and Machine Learning*. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-27655-3. DOI: 10.1007/978-3-030-27656-0.
- [18] Daniel Brewer und Nathan Sturtevant. “Benchmarks for Pathfinding in 3D Voxel Space”. In: *Proceedings of the International Symposium on Combinatorial Search 9.1* (2018), S. 143–147. ISSN: 2832-9171. DOI: 10.1609/socs.v9i1.18464.
- [19] Kevin Buchin, Tim Ophelders und Bettina Speckmann. *Computing the Similarity Between Moving Curves*. 2015. DOI: 10.48550/arXiv.1507.03819.
- [20] Kevin Buchin u. a. “On Length-Sensitive Fréchet Similarity”. In: *Algorithms and Data Structures*. Hrsg. von Pat Morin und Subhash Suri. Bd. 14079. Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2023, S. 208–231. ISBN: 978-3-031-38905-4. DOI: 10.1007/978-3-031-38906-1_15.
- [21] Lucian Busoniu, Robert Babuska und Bart de Schutter. “A Comprehensive Survey of Multiagent Reinforcement Learning”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38.2 (2008), S. 156–172. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.913919.
- [22] Alberto Castagna und Ivana Dusparic. *Multi-Agent Transfer Learning in Reinforcement Learning-Based Ride-Sharing Systems*. 2021. DOI: 10.48550/arXiv.2112.00424.
- [23] Kenneth W. Chase u. a. “Including Geometric Feature Variations in Tolerance Analysis of Mechanical Assemblies”. In: *IEEE Transactions* 28.10 (1996), S. 795–807. ISSN: 0740-817X. DOI: 10.1080/15458830.1996.11770732.
- [24] Baiming Chen u. a. *Delay-Aware Multi-Agent Reinforcement Learning for Cooperative and Competitive Environments*. 2020. DOI: 10.48550/arXiv.2005.05441.
- [25] Lin Chen u. a. “Multiagent Path Finding Using Deep Reinforcement Learning Coupled With Hot Supervision Contrastive Loss”. In: *IEEE Transactions on Industrial Electronics* 70.7 (2023), S. 7032–7040. ISSN: 0278-0046. DOI: 10.1109/TIE.2022.3206745.
- [26] Han-Chung Cheng u. a. “Shape similarity measurement for 3D mechanical part using D2 shape distribution and negative feature decomposition”. In: *Computers in Industry* 62.3 (2011), S. 269–280. ISSN: 01663615. DOI: 10.1016/j.compind.2010.09.001.
- [27] Tianshu Chu, Sandeep Chinchali und Sachin Katti. *Multi-agent Reinforcement Learning for Networked System Control*. 2020. DOI: 10.48550/arXiv.2004.01339.

- [28] Jaehoon Chung u. a. *Learning to Team-Based Navigation: A Review of Deep Reinforcement Learning Techniques for Multi-Agent Pathfinding*. 2023. DOI: 10.48550/arXiv.2308.05893.
- [29] Jason Cong und Bingjun Xiao. “Minimizing Computation in Convolutional Neural Networks”. In: *Artificial Neural Networks and Machine Learning – ICANN 2014*. Hrsg. von Stefan Wermter u. a. Bd. 8681. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, S. 281–290. ISBN: 978-3-319-11178-0. DOI: 10.1007/978-3-319-11179-7_36.
- [30] A. B. Conru. “A genetic approach to the cable harness routing problem”. In: *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*. IEEE, 1994, S. 200–205. ISBN: 0-7803-1899-4. DOI: 10.1109/ICEC.1994.350016.
- [31] Felipe Leno Da Silva und Anna Helena Reali Costa. “A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems”. In: *Journal of Artificial Intelligence Research* 64 (2019), S. 645–703. DOI: 10.1613/jair.1.11396.
- [32] Felipe Leno Da Silva u. a. “Agents teaching agents: a survey on inter-agent transfer learning”. In: *Autonomous Agents and Multi-Agent Systems* 34.1 (2020). ISSN: 1387-2532. DOI: 10.1007/s10458-019-09430-0.
- [33] Abhishek Das u. a. *TarMAC: Targeted Multi-Agent Communication*. 2018. DOI: 10.48550/arXiv.1810.11187.
- [34] Martin Denk u. a. “Generating Digital Twins for Path-Planning of Autonomous Robots and Drones Using Constrained Homotopic Shrinking for 2D and 3D Environment Modeling”. In: *Applied Sciences* 13.1 (2023), S. 105. DOI: 10.3390/app13010105.
- [35] Di Wang, Hongbin Deng und Zhenhua Pan. “MRCDRL: Multi-robot coordination with deep reinforcement learning”. In: *Neurocomputing* 406 (2020), S. 68–76. ISSN: 09252312. DOI: 10.1016/j.neucom.2020.04.028.
- [36] *DIN ISO 16792:2022-12, Technische Produktdokumentation_ - Praktische Anwendungen von digitalen Produktdefinitionsdaten (ISO_16792:2021); Text Deutsch und Englisch*. Berlin. DOI: 10.31030/3389769.
- [37] J. Dinkelacker u. a. “System integration based on packing, piping and harness routing automation using graph-based design languages”. In: *CEAS Aeronautical Journal* 14.2 (2023), S. 591–606. ISSN: 1869-5582. DOI: 10.1007/s13272-023-00642-5.
- [38] Hao Dong, Zihan Ding und Shanghang Zhang. *Deep Reinforcement Learning*. Singapore: Springer Singapore, 2020. ISBN: 978-981-15-4094-3. DOI: 10.1007/978-981-15-4095-0.
- [39] Rainer Drath. *AutomationML: Das Lehrbuch für Studium und Praxis*. De Gruyter Studium. Berlin: De Gruyter Oldenbourg, 2022. ISBN: 978-3-11-078293-6. DOI: 10.1515/9783110782998.

-
- [40] Wei Du und Shifei Ding. “A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications”. In: *Artificial Intelligence Review* 54.5 (2021), S. 3215–3238. ISSN: 0269-2821. DOI: 10.1007/s10462-020-09938-y.
- [41] Sebastian Durst und Christian Dülme. “Digitalisierung von Wertschöpfungsketten end-to-end am Beispiel der Klemmleistenbestückung”. In: *Geschäftsmodelle in der digitalen Welt*. Hrsg. von Wolfgang Becker u. a. Wiesbaden: Springer Fachmedien Wiesbaden, 2019, S. 685–693. ISBN: 978-3-658-22128-7. DOI: 10.1007/978-3-658-22129-4_36.
- [42] K. Eder u. a. “Knowledge Documentation Based on Automatic Identification and Clustering of Change Intentions in CAD Data of Wiring Harnesses”. In: *Proceedings of the Design Society 2* (2022), S. 683–692. DOI: 10.1017/pds.2022.70.
- [43] Kevin Eder u. a. “Knowledge Reuse of CAD Data in Parallel Development of Multiple Wiring Harness Variants”. In: *Product Lifecycle Management. PLM in Transition Times: The Place of Humans and Transformative Technologies*. Hrsg. von Frédéric Noël u. a. Bd. 667. IFIP Advances in Information and Communication Technology. Cham: Springer Nature Switzerland, 2023, S. 196–205. ISBN: 978-3-031-25181-8. DOI: 10.1007/978-3-031-25182-5_20.
- [44] Kevin Eder u. a. “Product Data Management for Continuously Consistent CAD Data in Wiring Harness Development”. In: *Design in the Era of Industry 4.0, Volume 3*. Hrsg. von Amaresh Chakrabarti und Vishal Singh. Bd. 346. Smart Innovation, Systems and Technologies. Singapore: Springer Nature Singapore, 2023, S. 1175–1187. ISBN: 978-981-99-0427-3. DOI: 10.1007/978-981-99-0428-0_96.
- [45] Marc Eheim, Dennis Kaiser und Roland Weil. “On Automation Along the Automotive Wire Harness Value Chain”. In: *Advances in Automotive Production Technology – Theory and Application*. Hrsg. von Philipp Weißgraeber, Frieder Heieck und Clemens Ackermann. ARENA2036. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, S. 178–186. ISBN: 978-3-662-62961-1. DOI: 10.1007/978-3-662-62962-8_21.
- [46] Absalom E. Ezugwu u. a. “A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects”. In: *Engineering Applications of Artificial Intelligence* 110 (2022), S. 104743. ISSN: 09521976. DOI: 10.1016/j.engappai.2022.104743.
- [47] Tingxiang Fan u. a. *Fully Distributed Multi-Robot Collision Avoidance via Deep Reinforcement Learning for Safe and Efficient Navigation in Complex Scenarios*. 2018. DOI: 10.48550/arXiv.1808.03841.
- [48] Gerald Farin. *Kurven und Flächen im Computer Aided Geometric Design*. Wiesbaden: Vieweg+Teubner Verlag, 1994. ISBN: 978-3-663-10603-6. DOI: 10.1007/978-3-663-10602-9.
- [49] Ariel Felner u. a. “Search-Based Optimal Solvers for the Multi-Agent Pathfinding Problem: Summary and Challenges”. In: *Proceedings of the International Symposium on Combinatorial Search 8.1* (2017), S. 29–37. ISSN: 2832-9171. DOI: 10.1609/socs.v8i1.18423.

- [50] Rodrigo Fernandes de Mello und Moacir Antonelli Ponti. *Machine Learning*. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-94988-8. DOI: 10.1007/978-3-319-94989-5.
- [51] Jakob Foerster u. a. “Counterfactual Multi-Agent Policy Gradients”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018). ISSN: 2159-5399. DOI: 10.1609/aaai.v32i1.11794.
- [52] Jakob N. Foerster u. a. *Learning to Communicate with Deep Multi-Agent Reinforcement Learning*. 2016. DOI: 10.48550/arXiv.1605.06676.
- [53] Vincent Francois-Lavet u. a. “An Introduction to Deep Reinforcement Learning”. In: (2018). DOI: 10.48550/arXiv.1811.12560.
- [54] Scott Fujimoto, Herke van Hoof und David Meger. *Addressing Function Approximation Error in Actor-Critic Methods*. 2018. DOI: 10.48550/arXiv.1802.09477.
- [55] Shani Gamrian und Yoav Goldberg. “Transfer Learning for Related Reinforcement Learning Tasks via Image-to-Image Translation”. In: (2018). DOI: 10.48550/arXiv.1806.07377.
- [56] E. G. Gilbert, D. W. Johnson und S. S. Keerthi. “A fast procedure for computing the distance between complex objects in three-dimensional space”. In: *IEEE Journal on Robotics and Automation* 4.2 (1988), S. 193–203. ISSN: 08824967. DOI: 10.1109/56.2083.
- [57] E. N. Gilbert und H. O. Pollak. “Steiner Minimal Trees”. In: *SIAM Journal on Applied Mathematics* 16.1 (1968), S. 1–29. ISSN: 0036-1399. DOI: 10.1137/0116001.
- [58] Steven J. Gortler. *Foundations of 3D computer graphics*. The MIT Press, 2012. ISBN: 978-0-262-01735-0. URL: <https://permalink.obvsg.at/>.
- [59] Günther Görz, Ute Schmid und Tanya Braun. *Handbuch der Künstlichen Intelligenz*. De Gruyter, 2020. ISBN: 9783110659948. DOI: 10.1515/9783110659948.
- [60] O. Graeser und J. Schmelter. “Durchgängiges Engineering auf Basis von AutomationML und eCl@ss”. In: *Konstruktion* 67.07-08 (2015), S. 50–52. ISSN: 0720-5953. DOI: 10.37544/0720-5953-2015-07-08-50.
- [61] Sven Gronauer und Klaus Diepold. “Multi-agent deep reinforcement learning: a survey”. In: *Artificial Intelligence Review* 55.2 (2022), S. 895–943. ISSN: 0269-2821. DOI: 10.1007/s10462-021-09996-w.
- [62] Siyu Guo u. a. “An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning”. In: *Sensors (Basel, Switzerland)* 20.2 (2020). DOI: 10.3390/s20020426.
- [63] Jayesh K. Gupta, Maxim Egorov und Mykel Kochenderfer. “Cooperative Multi-agent Control Using Deep Reinforcement Learning”. In: *Autonomous Agents and Multi-agent Systems*. Hrsg. von Gita Sukthankar und Juan A. Rodriguez-Aguilar. Bd. 10642. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, S. 66–83. ISBN: 978-3-319-71681-7. DOI: 10.1007/978-3-319-71682-4_5.

-
- [64] Vratika Gupta u. a. “An Overview of Supervised Machine Learning Algorithm”. In: *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*. IEEE, 2022, S. 87–92. ISBN: 978-1-6654-8732-0. DOI: 10.1109/SMART55829.2022.10047618.
 - [65] Victor Häfner. *PolyVR - A Virtual Reality Authoring Framework for Engineering Applications*. 2019. DOI: 10.5445/IR/1000098349.
 - [66] Jiawei Han, Micheline Kamber und Jian Pei. “Cluster Analysis”. In: *Data Mining*. Elsevier, 2012, S. 443–495. ISBN: 9780123814791. DOI: 10.1016/B978-0-12-381479-1.00010-1.
 - [67] Jihee Han. “An efficient approach to 3D path planning”. In: *Information Sciences* 478 (2019), S. 318–330. ISSN: 00200255. DOI: 10.1016/j.ins.2018.11.045.
 - [68] Wenjian Hao u. a. *Adaptive Policy Learning to Additional Tasks*. 2023. DOI: 10.48550/arXiv.2305.15193.
 - [69] Ali Hashemian und Seyed Farhad Hosseini. “An integrated fitting and fairing approach for object reconstruction using smooth NURBS curves and surfaces”. In: *Computers & Mathematics with Applications* 76.7 (2018), S. 1555–1575. ISSN: 08981221. DOI: 10.1016/j.camwa.2018.07.007.
 - [70] Zichen He u. a. “Asynchronous Multithreading Reinforcement-Learning-Based Path Planning and Tracking for Unmanned Underwater Vehicle”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52.5 (2022), S. 2757–2769. ISSN: 2168-2216. DOI: 10.1109/TSMC.2021.3050960.
 - [71] Peter Hehenberger. *Computerunterstützte Produktion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020. ISBN: 978-3-662-60875-3. DOI: 10.1007/978-3-662-60876-0.
 - [72] T. Hermansson und E. Åblad. “Routing of curves with piecewise constant curvature applied to routing of preformed hoses”. In: *Computer-Aided Design* 139 (2021), S. 103067. ISSN: 00104485. DOI: 10.1016/j.cad.2021.103067.
 - [73] Tomas Hermansson u. a. “Automatic assembly path planning for wiring harness installations”. In: *Journal of Manufacturing Systems* 32.3 (2013), S. 417–422. ISSN: 02786125. DOI: 10.1016/j.jmsy.2013.04.006.
 - [74] Irina Higgins u. a. *DARLA: Improving Zero-Shot Transfer in Reinforcement Learning*. 2017. DOI: 10.48550/arXiv.1707.08475.
 - [75] Asmaul Hosna u. a. “Transfer learning: a friendly introduction”. In: *Journal of big data* 9.1 (2022), S. 102. ISSN: 2196-1115. DOI: 10.1186/s40537-022-00652-w.
 - [76] Hongtao Hu u. a. “Anti-conflict AGV path planning in automated container terminals based on multi-agent reinforcement learning”. In: *International Journal of Production Research* 61.1 (2023), S. 65–80. ISSN: 0020-7543. DOI: 10.1080/00207543.2021.1998695.
 - [77] Peng Huang, Adrian Hilton und Jonathan Starck. “Shape Similarity for 3D Video Sequences of People”. In: *International Journal of Computer Vision* 89.2-3 (2010), S. 362–381. ISSN: 0920-5691. DOI: 10.1007/s11263-010-0319-9.

- [78] Kashif Hussain u. a. “Metaheuristic research: a comprehensive survey”. In: *Artificial Intelligence Review* 52.4 (2019), S. 2191–2233. ISSN: 0269-2821. DOI: 10.1007/s10462-017-9605-z.
- [79] International Organization for Standardization. *ISO 10303-242:2022-12, Industrial automation systems and integration - Product data representation and exchange: Part 242: Application protocol: Managed model-based 3D engineering*. Genf, 2022.
- [80] International Organization for Standardization. *ISO 14306:2017, Industrial automation systems and integration - JT file format specification for 3D visualization*. Genf, 2017.
- [81] Natraj Iyer u. a. “Three-dimensional shape searching: state-of-the-art review and future trends”. In: *Computer-Aided Design* 37.5 (2005), S. 509–530. ISSN: 00104485. DOI: 10.1016/j.cad.2004.07.002.
- [82] Jiechuan Jiang und Zongqing Lu. *Learning Attentional Communication for Multi-Agent Cooperation*. 2018. DOI: 10.48550/arXiv.1805.07733.
- [83] Jiechuan Jiang u. a. *Graph Convolutional Reinforcement Learning*. 2018. DOI: 10.48550/arXiv.1810.09202.
- [84] Shuai Jiang. “Branch Cable Path Planning Method Based on “Genetic-A*” Fusion Algorithm”. In: *2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS)*. IEEE, 2023, S. 1179–1182. ISBN: 979-8-3503-3344-2. DOI: 10.1109/ICPICS58376.2023.10235407.
- [85] Xin Jin und Jiawei Han. “Partitional Clustering”. In: *Encyclopedia of Machine Learning*. Hrsg. von Claude Sammut und Geoffrey I. Webb. Boston, MA: Springer US, 2010, S. 766. ISBN: 978-0-387-30768-8. DOI: 10.1007/978-0-387-30164-8_631.
- [86] Taeho Jo. *Machine Learning Foundations*. Cham: Springer International Publishing, 2021. ISBN: 978-3-030-65899-1. DOI: 10.1007/978-3-030-65900-4.
- [87] Ameet V. Joshi. *Machine Learning and Artificial Intelligence*. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-26621-9. DOI: 10.1007/978-3-030-26622-6.
- [88] T. Karlsson u. a. “Automatic Cable Harness Layout Routing in a Customizable 3D Environment”. In: *Computer-Aided Design* 169 (2024), S. 103671. ISSN: 00104485. DOI: 10.1016/j.cad.2023.103671.
- [89] Tatsuya Kasai, Hiroshi Tenmoto und Akimoto Kamiya. “Learning of communication codes in multi-agent reinforcement learning problem”. In: *2008 IEEE Conference on Soft Computing in Industrial Applications*. IEEE, 2008, S. 1–6. ISBN: 978-1-4244-3782-5. DOI: 10.1109/SMCIA.2008.5045926.
- [90] Saekyeol Kim u. a. “Automatic design system for generating routing layout of tubes, hoses, and cable harnesses in a commercial truck”. In: *Journal of Computational Design and Engineering* 8.4 (2021), S. 1098–1114. DOI: 10.1093/jcde/qwab034.
- [91] Saekyeol Kim u. a. “Sequential graph-based routing algorithm for electrical harnesses, tubes, and hoses in a commercial vehicle”. In: *Journal of Intelligent Manufacturing* 32.4 (2021), S. 917–933. ISSN: 0956-5515. DOI: 10.1007/s10845-020-01596-9.

-
- [92] Daniil Kirilenko u. a. *TransPath: Learning Heuristics For Grid-Based Pathfinding via Transformers*. 2022. DOI: 10.48550/arXiv.2212.11730.
 - [93] James Kirkpatrick u. a. “Overcoming catastrophic forgetting in neural networks”. In: *Proceedings of the National Academy of Sciences of the United States of America* 114.13 (2017), S. 3521–3526. DOI: 10.1073/pnas.1611835114.
 - [94] J. T. Klosowski u. a. “Efficient collision detection using bounding volume hierarchies of k-DOPs”. In: *IEEE Transactions on Visualization and Computer Graphics* 4.1 (1998), S. 21–36. ISSN: 10772626. DOI: 10.1109/2945.675649.
 - [95] L. Kou, G. Markowsky und L. Berman. “A fast algorithm for Steiner trees”. In: *Acta Informatica* 15.2 (1981), S. 141–145. ISSN: 0001-5903. DOI: 10.1007/BF00288961.
 - [96] Joseph B. Kruskal. “On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem”. In: *Proceedings of the American Mathematical Society* 7.1 (1956), S. 48. ISSN: 00029939. DOI: 10.2307/2033241.
 - [97] Takahiro Kunii, Miho Asano und Hiroshi Noborio. “Voxel-Based Route-Search Algorithm for Tumor Navigation and Blood Vessel Avoidance”. In: *Human-Computer Interaction. Interaction Techniques and Novel Applications*. Hrsg. von Masaaki Kurosu. Bd. 12763. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, S. 566–581. ISBN: 978-3-030-78464-5. DOI: 10.1007/978-3-030-78465-2_41.
 - [98] Thanh Le Nguyen-Meidine u. a. “Knowledge distillation methods for efficient unsupervised adaptation across multiple domains”. In: *Image and Vision Computing* 108 (2021), S. 104096. ISSN: 02628856. DOI: 10.1016/j.imavis.2021.104096.
 - [99] Yann LeCun, Yoshua Bengio und Geoffrey Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), S. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539.
 - [100] Zewen Li u. a. “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects”. In: *IEEE transactions on neural networks and learning systems* 33.12 (2022), S. 6999–7019. DOI: 10.1109/TNNLS.2021.3084827.
 - [101] Timothy P. Lillicrap u. a. *Continuous control with deep reinforcement learning*. 2015. DOI: 10.48550/arXiv.1509.02971.
 - [102] Michael L. Littman. “Markov games as a framework for multi-agent reinforcement learning”. In: *Machine Learning Proceedings 1994*. Elsevier, 1994, S. 157–163. ISBN: 9781558603356. DOI: 10.1016/B978-1-55860-335-6.50027-1.
 - [103] Yanchi Liu u. a. “Understanding of Internal Clustering Validation Measures”. In: *2010 IEEE International Conference on Data Mining*. IEEE, 2010, S. 911–916. ISBN: 978-1-4244-9131-5. DOI: 10.1109/ICDM.2010.35.
 - [104] Oliver Lohse. *Entwicklung einer Methode zum Einsatz von Reinforcement Learning für die dynamische Fertigungsdurchlaufsteuerung*. 2023. DOI: 10.5445/KSP/1000156002.
 - [105] Uwe Lorenz. *Reinforcement Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020. ISBN: 978-3-662-61650-5. DOI: 10.1007/978-3-662-61651-2.

- [106] Ryan Lowe u. a. *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*. 2017. DOI: 10.48550/arXiv.1706.02275.
- [107] Kevin M. Lynch und Frank C. Park. *Modern robotics: Mechanics, planning, and control*. Cambridge: University Press, 2017. ISBN: 978-1107156302. URL: <https://permalink.obvsg.at/AC15560713>.
- [108] Adil M. Bagirov, Napsu Karmita und Sona Taheri. *Partitional Clustering via Nonsmooth Optimization*. Cham: Springer International Publishing, 2020. ISBN: 978-3-030-37825-7. DOI: 10.1007/978-3-030-37826-4.
- [109] Hang Ma u. a. *Multi-Agent Path Finding with Deadlines*. 2018. DOI: 10.48550/arXiv.1806.04216.
- [110] Ziyuan Ma, Yudong Luo und Hang Ma. *Distributed Heuristic Multi-Agent Path Finding with Communication*. 2021. DOI: 10.48550/arXiv.2106.11365.
- [111] Ziyuan Ma, Yudong Luo und Jia Pan. *Learning Selective Communication for Multi-Agent Path Finding*. 2021. DOI: 10.48550/arXiv.2109.05413.
- [112] Nafiseh Masoudi und Georges Fadel. “An Optimization Framework for the Design of Cable Harness Layouts in Planar Interconnected Systems”. In: *Journal of Mechanical Design* 144.1 (2022). ISSN: 1050-0472. DOI: 10.1115/1.4051685.
- [113] Daniel Maturana und Sebastian Scherer. “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, S. 922–928. ISBN: 978-1-4799-9994-1. DOI: 10.1109/IR0S.2015.7353481.
- [114] U. Maulik und S. Bandyopadhyay. “Performance evaluation of some clustering algorithms and validity indices”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.12 (2002), S. 1650–1654. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2002.1114856.
- [115] Bhairav Mehta u. a. *Active Domain Randomization*. 2019. DOI: 10.48550/arXiv.1904.04762.
- [116] Hideo Miyachi und Koshiro Murakami. “A Study of 3D Shape Similarity Search in Point Representation by Using Machine Learning”. In: *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*. Hrsg. von Leonard Barolli, Peter Hellinckx und Juggapong Natwichai. Bd. 96. Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2020, S. 265–274. ISBN: 978-3-030-33508-3. DOI: 10.1007/978-3-030-33509-0_24.
- [117] Siddharth Nayak u. a. *Scalable Multi-Agent Reinforcement Learning through Intelligent Information Aggregation*. 2022. DOI: 10.48550/arXiv.2211.02127.
- [118] Jonas Neckenich u. a. “3D-Master-Leitungssatz – Virtuelle Produkt-entwicklung von Automobilleitungssätzen/3D-Master Wiring Harness – Virtual Product Design of Automotive Wiring Harness as Combination of Installation Space- and Product Orientation”. In: *Konstruktion* 70.05 (2018), S. 71–75. ISSN: 0720-5953. DOI: 10.37544/0720-5953-2018-05-71.

-
- [119] Angelia Nedic und Asuman Ozdaglar. “Distributed Subgradient Methods for Multi-Agent Optimization”. In: *IEEE Transactions on Automatic Control* 54.1 (2009), S. 48–61. ISSN: 0018-9286. DOI: 10.1109/TAC.2008.2009515.
 - [120] Moritz Neumaier u. a. “Automated Piping in an Airbus A320 Landing Gear Bay Using Graph-Based Design Languages”. In: *Aerospace* 9.3 (2022), S. 140. DOI: 10.3390/aerospace9030140.
 - [121] Thanh Thi Nguyen, Ngoc Duy Nguyen und Saeid Nahavandi. “Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications”. In: *IEEE transactions on cybernetics* 50.9 (2020), S. 3826–3839. DOI: 10.1109/TCYB.2020.2977374.
 - [122] Nasimul Noman. “A Shallow Introduction to Deep Neural Networks”. In: *Deep Neural Evolution*. Hrsg. von Hitoshi Iba und Nasimul Noman. Natural Computing Series. Singapore: Springer Singapore, 2020, S. 35–63. ISBN: 978-981-15-3684-7. DOI: 10.1007/978-981-15-3685-4_2.
 - [123] Chigozie Nwankpa u. a. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018. DOI: 10.48550/arXiv.1811.03378.
 - [124] Keiron O’Shea und Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. DOI: 10.48550/arXiv.1511.08458.
 - [125] R. Ohbuchi, T. Minamitani und T. Takei. “Shape-similarity search of 3D models by using enhanced shape functions”. In: *Proceedings of Theory and Practice of Computer Graphics, 2003*. IEEE Comput. Soc, 2003, S. 97–104. ISBN: 0-7695-1942-3. DOI: 10.1109/TPCG.2003.1206936.
 - [126] Frans A. Oliehoek und Christopher Amato. “Multiagent Systems Under Uncertainty”. In: *A Concise Introduction to Decentralized POMDPs*. Hrsg. von Frans A. Oliehoek und Christopher Amato. SpringerBriefs in Intelligent Systems. Cham: Springer International Publishing, 2016, S. 1–9. ISBN: 978-3-319-28927-4. DOI: 10.1007/978-3-319-28929-8_1.
 - [127] OpenAI. *GPT-4 Technical Report*. 2023. DOI: 10.48550/arXiv.2303.08774.
 - [128] Afshin Oroojlooy und Davood Hajinezhad. “A review of cooperative multi-agent deep reinforcement learning”. In: *Applied Intelligence* 53.11 (2023), S. 13677–13722. ISSN: 0924-669X. DOI: 10.1007/s10489-022-04105-y.
 - [129] Sinno Jialin Pan und Qiang Yang. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), S. 1345–1359. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191.
 - [130] W.P.J. Pamarathne und T.G.I. Fernando. “Wire and cable routings and harness designing systems with AI, a review”. In: *2016 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*. IEEE, 2016, S. 1–6. ISBN: 978-1-5090-6132-7. DOI: 10.1109/ICIAfS.2016.7946575.

- [131] D. T. Pham und A. A. Afify. "Machine-learning techniques and their applications in manufacturing". In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 219.5 (2005), S. 395–412. ISSN: 0954-4054. DOI: 10.1243/095440505X32274.
- [132] L. Piegl. "On NURBS: a survey". In: *IEEE Computer Graphics and Applications* 11.1 (1991), S. 55–71. ISSN: 0272-1716. DOI: 10.1109/38.67702.
- [133] Cédric Portaneri u. a. "Alpha wrapping with an offset". In: *ACM Transactions on Graphics* 41.4 (2022), S. 1–22. ISSN: 0730-0301. DOI: 10.1145/3528223.3530152.
- [134] R. C. Prim. "Shortest Connection Networks And Some Generalizations". In: *Bell System Technical Journal* 36.6 (1957), S. 1389–1401. ISSN: 00058580. DOI: 10.1002/j.1538-7305.1957.tb01515.x.
- [135] Zhixin Qi, Hongzhi Wang und Zejiao Dong. "Density-Based Clustering for Incomplete Data". In: *Dirty Data Processing for Machine Learning*. Hrsg. von Zhixin Qi, Hongzhi Wang und Zejiao Dong. Singapore: Springer Nature Singapore, 2024, S. 73–92. ISBN: 978-981-99-7656-0. DOI: 10.1007/978-981-99-7657-7_5.
- [136] Han Qie u. a. "Joint Optimization of Multi-UAV Target Assignment and Path Planning Based on Multi-Agent Reinforcement Learning". In: *IEEE Access* 7 (2019), S. 146264–146272. DOI: 10.1109/ACCESS.2019.2943253.
- [137] Yanfeng Qu, Dan Jiang und Qingyan Yang. "Branch pipe routing based on 3D connection graph and concurrent ant colony optimization algorithm". In: *Journal of Intelligent Manufacturing* 29.7 (2018), S. 1647–1657. ISSN: 0956-5515. DOI: 10.1007/s10845-016-1203-4.
- [138] Yanfeng Qu u. a. "Pipe Routing Approach for Aircraft Engines Based on Ant Colony Optimization". In: *Journal of Aerospace Engineering* 29.3 (2016). ISSN: 0893-1321. DOI: 10.1061/(ASCE)AS.1943-5525.0000543.
- [139] Nadia Rahmah und Imas Sukaesih Sitanggang. "Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra". In: *IOP Conference Series: Earth and Environmental Science* 31 (2016), S. 012012. ISSN: 1755-1307. DOI: 10.1088/1755-1315/31/1/012012.
- [140] Bonsa Regassa Hunde und Abraham Debebe Woldeyohannes. "Future prospects of computer-aided design (CAD) – A review from the perspective of artificial intelligence (AI), extended reality, and 3D printing". In: *Results in Engineering* 14 (2022), S. 100478. ISSN: 25901230. DOI: 10.1016/j.rineng.2022.100478.
- [141] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), S. 53–65. ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7.
- [142] Julian Ryde und Huosheng Hu. "3D mapping with multi-resolution occupied voxel lists". In: *Autonomous Robots* 28.2 (2010), S. 169–185. ISSN: 0929-5593. DOI: 10.1007/s10514-009-9158-3.

-
- [143] Christopher Saal u. a. “3D Model-Based Product Definition and Production – a Mind Change with Technical Hurdles”. In: *2020 3rd International Symposium on Small-scale Intelligent Manufacturing Systems (SIMS)*. IEEE, 2020, S. 1–4. ISBN: 978-1-7281-6419-9. DOI: 10.1109/SIMS49386.2020.9121460.
- [144] Albert Salim u. a. “General pattern recognition using machine learning in the cloud”. In: *Procedia Computer Science* 216 (2023), S. 565–570. ISSN: 18770509. DOI: 10.1016/j.procs.2022.12.170.
- [145] Joerg Sander. “Density-Based Clustering”. In: *Encyclopedia of Machine Learning and Data Mining*. Hrsg. von Claude Sammut und Geoffrey I. Webb. Boston, MA: Springer US, 2016, S. 1–5. ISBN: 978-1-4899-7502-7. DOI: 10.1007/978-1-4899-7502-7_70-1.
- [146] Poornachandra Sarang. “Density-Based Clustering”. In: *Thinking Data Science*. Hrsg. von Poornachandra Sarang. The Springer Series in Applied Machine Learning. Cham: Springer International Publishing, 2023, S. 209–228. ISBN: 978-3-031-02362-0. DOI: 10.1007/978-3-031-02363-7_12.
- [147] Guillaume Sartoretti u. a. “PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning”. In: (2018). DOI: 10.48550/arXiv.1809.03531.
- [148] Mike Schäfer und Thomas Lengauer. “Automated Layout Generation and Wiring Area Estimation for 3D Electronic Modules”. In: *Journal of Mechanical Design* 123.3 (2001), S. 330–336. ISSN: 1050-0472. DOI: 10.1115/1.1371478.
- [149] Jonatan Scharff Willners u. a. “Online 3-Dimensional Path Planning with Kinematic Constraints in Unknown Environments Using Hybrid A* with Tree Pruning”. In: *Sensors (Basel, Switzerland)* 21.4 (2021). DOI: 10.3390/s21041152.
- [150] Günther Schuh. “Lebenszyklusorientierte Produktentwicklung”. In: *Neue Entwicklungen in der Unternehmensorganisation*. Hrsg. von Dieter Spath u. a. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, S. 195–210. ISBN: 978-3-662-55425-8. DOI: 10.1007/978-3-662-55426-5_23.
- [151] John Schulman u. a. *Proximal Policy Optimization Algorithms*. 2017. DOI: 10.48550/arXiv.1707.06347.
- [152] Fatih Semiz, Mücahit Alkan Yorgancı und Faruk Polat. “Solving an industry-inspired generalization of lifelong MAPF problem including multiple delivery locations”. In: *Advanced Engineering Informatics* 57 (2023), S. 102026. ISSN: 14740346. DOI: 10.1016/j.aei.2023.102026.
- [153] Samaneh Hosseini Semnani u. a. *Multi-agent Motion Planning for Dense and Dynamic Environments via Deep Reinforcement Learning*. 2020. DOI: 10.48550/arXiv.2001.06627.
- [154] Mohit Sewak. *Deep Reinforcement Learning*. Singapore: Springer Singapore, 2019. ISBN: 978-981-13-8284-0. DOI: 10.1007/978-981-13-8285-7.
- [155] Guni Sharon u. a. “Conflict-based search for optimal multi-agent pathfinding”. In: *Artificial Intelligence* 219 (2015), S. 40–66. ISSN: 00043702. DOI: 10.1016/j.artint.2014.11.006.

- [156] Peizhi Shi u. a. “A novel learning-based feature recognition method using multiple sectional view representation”. In: *Journal of Intelligent Manufacturing* 31.5 (2020), S. 1291–1309. ISSN: 0956-5515. DOI: 10.1007/s10845-020-01533-w.
- [157] David Silver u. a. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (2016), S. 484–489. ISSN: 0028-0836. DOI: 10.1038/nature16961.
- [158] Rahul Dev Singh, Ajay Mittal und Rajesh K. Bhatia. “3D convolutional neural network for object recognition: a review”. In: *Multimedia Tools and Applications* 78.12 (2019), S. 15951–15995. ISSN: 1380-7501. DOI: 10.1007/s11042-018-6912-6.
- [159] Nick Smith. “Enhancing Automotive Wire Harness Manufacturing Through Digital Continuity”. In: *SAE Technical Paper Series*. SAE Technical Paper Series. SAE International 400 Commonwealth Drive, Warrendale, PA, United States, 2015. DOI: 10.4271/2015-01-0238.
- [160] Roni Stern. “Multi-Agent Path Finding – An Overview”. In: *Artificial Intelligence*. Hrsg. von Gennady S. Osipov, Aleksandr I. Panov und Konstantin S. Yakovlev. Bd. 11866. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, S. 96–115. ISBN: 978-3-030-33273-0. DOI: 10.1007/978-3-030-33274-7_6.
- [161] Roni Stern u. a. *Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks*. 2019. DOI: 10.48550/arXiv.1906.08291.
- [162] Josip Stjepandic u. a. “Standardized Approach to ECAD/MCAD Collaboration”. In: *Moving Integrated Product Development to Service Clouds in the Global Economy* (2014). DOI: 10.3233/978-1-61499-440-4-587.
- [163] Peter Sunehag u. a. *Value-Decomposition Networks For Cooperative Multi-Agent Learning*. 2017. DOI: 10.48550/arXiv.1706.05296.
- [164] Richard S. Sutton und Andrew Barto. *Reinforcement learning: An introduction*. Second edition. Adaptive computation and machine learning. Cambridge, Massachusetts, London, England und Ipswich: The MIT Press und EBSCO, 2018. ISBN: 978-0262039246.
- [165] Adam Taylor u. a. “Parallel Transfer Learning in Multi-Agent Systems: What, when and how to transfer?” In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, S. 1–8. ISBN: 978-1-7281-1985-4. DOI: 10.1109/IJCNN.2019.8851784.
- [166] Matthew E. Taylor und Peter Stone. “Transfer Learning for Reinforcement Learning Domains: A Survey”. In: *J. Mach. Learn. Res.* 10 (2009), S. 1633–1685. ISSN: 1532-4435. DOI: 10.5555/1577069.1755839.
- [167] Daniel Thul u. a. “Approximate convex decomposition and transfer for animated meshes”. In: *ACM Transactions on Graphics* 37.6 (2018), S. 1–10. ISSN: 0730-0301. DOI: 10.1145/3272127.3275029.

-
- [168] Semuil Tjiharjadi, Szalinsyah Razali und Hamzah Asyrani Sulaiman. “A Systematic Literature Review of Multi-agent Pathfinding for Maze Research”. In: *Journal of Advances in Information Technology* 13.4 (2022). ISSN: 17982340. DOI: 10.12720/jait.13.4.358-367.
 - [169] Josh Tobin u. a. *Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World*. 2017. DOI: 10.48550/arXiv.1703.06907.
 - [170] Jerome Trommnau u. a. “Overview of the State of the Art in the Production Process of Automotive Wire Harnesses, Current Research and Future Trends”. In: *Procedia CIRP* 81 (2019), S. 387–392. ISSN: 22128271. DOI: 10.1016/j.procir.2019.03.067.
 - [171] Sándor Vajna u. a. *CAX für Ingenieure*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. ISBN: 978-3-662-54623-9. DOI: 10.1007/978-3-662-54624-6.
 - [172] T. van den Berg. “Harnessing the potential of Knowledge Based Engineering in manufacturing design”. Diss. Delft University of Technology, 2013. DOI: 10.4233/uuid:d44876fe-45dd-44ae-8947-622613eb963c.
 - [173] Gino van den Bergen. “Convex Objects”. In: *Collision Detection in Interactive 3D Environments*. Elsevier, 2003, S. 105–169. ISBN: 9781558608016. DOI: 10.1016/B978-1-55860-801-6.50006-4.
 - [174] Christian van der Velden u. a. “An intelligent system for automatic layout routing in aerospace design”. In: *Innovations in Systems and Software Engineering* 3.2 (2007), S. 117–128. ISSN: 1614-5046. DOI: 10.1007/s11334-007-0021-4.
 - [175] Ashish Vaswani u. a. *Attention Is All You Need*. 2017. DOI: 10.48550/arXiv.1706.03762.
 - [176] Verein Deutscher Ingenieure. *3-D-Produktmodellierung - Technische und organisatorische Voraussetzungen - Verfahren, Werkzeuge und Anwendungen - Wirtschaftlicher Einsatz in der Praxis*. 2009.
 - [177] Quan Vuong u. a. *How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?* 2019. DOI: 10.48550/arXiv.1903.11774.
 - [178] Silke Wagner und Dorothea Wagner. *Comparing Clusterings - An Overview*. 2007. DOI: 10.5445/IR/1000011477.
 - [179] Xiang Wan, Philip T. Evers und Martin E. Dresner. “Too much of a good thing: The impact of product variety on operations and sales performance”. In: *Journal of Operations Management* 30.4 (2012), S. 316–324. ISSN: 02726963. DOI: 10.1016/j.jom.2011.12.002.
 - [180] Binyu Wang u. a. *Mobile Robot Path Planning in Dynamic Environments through Globally Guided Reinforcement Learning*. 2020. DOI: 10.48550/arXiv.2005.05420.
 - [181] Ning Wang und Dan Zhang. “3D Shape Similarity Measurement Based on Scale Invariant Functional Maps”. In: *Image and Graphics Technologies and Applications*. Hrsg. von Wang Yongtian und Wu Lifang. Bd. 1910. Communications in Computer and Information Science. Singapore: Springer Nature Singapore, 2023, S. 95–109. ISBN: 978-981-99-7548-8. DOI: 10.1007/978-981-99-7549-5_8.

- [182] Peng-Shuai Wang u. a. “O-CNN”. In: *ACM Transactions on Graphics* 36.4 (2017), S. 1–11. ISSN: 0730-0301. DOI: 10.1145/3072959.3073608.
- [183] Zhechao Wang u. a. “Reinforcement Learning in Few-Shot Scenarios: A Survey”. In: *Journal of Grid Computing* 21.2 (2023). ISSN: 1570-7873. DOI: 10.1007/s10723-023-09663-0.
- [184] Zhiyong Wang und Sisi Zlatanova. “Multi-agent based path planning for first responders among moving obstacles”. In: *Computers, Environment and Urban Systems* 56 (2016), S. 48–58. ISSN: 01989715. DOI: 10.1016/j.compenvurbsys.2015.11.001.
- [185] Zhou Wang u. a. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 13.4 (2004), S. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.
- [186] Xinyue Wei u. a. “Approximate Convex Decomposition for 3D Meshes with Collision-Aware Concavity and Tree Search”. In: *ACM Transactions on Graphics* 41.4 (2022), S. 1–18. ISSN: 0730-0301. DOI: 10.1145/3528223.3530103.
- [187] Jens Weise, Steven Benkhardt und Sanaz Mostaghim. “Graph-based multi-objective generation of customised wiring harnesses”. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. Hrsg. von Manuel López-Ibáñez, Anne Auger und Thomas Stützle. New York, NY, USA: ACM, 2019, S. 407–408. ISBN: 9781450367486. DOI: 10.1145/3319619.3321908.
- [188] Karl Weiss, Taghi M. Khoshgoftaar und DingDing Wang. “A survey of transfer learning”. In: *Journal of big data* 3.1 (2016). ISSN: 2196-1115. DOI: 10.1186/s40537-016-0043-6.
- [189] Muning Wen u. a. *Multi-Agent Reinforcement Learning is a Sequence Modeling Problem*. 2022. DOI: 10.48550/arXiv.2205.14953.
- [190] Shuhuan Wen u. a. “A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning”. In: *Applied Soft Computing* 110 (2021), S. 107605. ISSN: 15684946. DOI: 10.1016/j.asoc.2021.107605.
- [191] Slawomir Wierzchoń und Mieczysław Kłopotek. *Modern Algorithms of Cluster Analysis*. Bd. 34. Cham: Springer International Publishing, 2018. ISBN: 978-3-319-69307-1. DOI: 10.1007/978-3-319-69308-8.
- [192] Lauren J. Wong und Alan J. Michaels. “Transfer Learning for Radio Frequency Machine Learning: A Taxonomy and Survey”. In: *Sensors (Basel, Switzerland)* 22.4 (2022). DOI: 10.3390/s22041416.
- [193] Dongkuan Xu und Yingjie Tian. “A Comprehensive Survey of Clustering Algorithms”. In: *Annals of Data Science* 2.2 (2015), S. 165–193. ISSN: 2198-5804. DOI: 10.1007/s40745-015-0040-1.

-
- [194] Yusheng Xu, Xiaohua Tong und Uwe Stilla. "Voxel-based representation of 3D point clouds: Methods, applications, and its potential use in the construction industry". In: *Automation in Construction* 126 (2021), S. 103675. ISSN: 09265805. DOI: 10.1016/j.autcon.2021.103675.
- [195] Rikiya Yamashita u. a. "Convolutional neural networks: an overview and application in radiology". In: *Insights into imaging* 9.4 (2018), S. 611–629. ISSN: 1869-4101. DOI: 10.1007/s13244-018-0639-9.
- [196] Xiaodong Yang u. a. "A review of cable layout design and assembly simulation in virtual environments". In: *Virtual Reality & Intelligent Hardware* 1.6 (2019), S. 543–557. ISSN: 20965796. DOI: 10.1016/j.vrih.2019.11.001.
- [197] Yang Yang, Li Juntao und Peng Lingling. "Multi-robot path planning based on a deep reinforcement learning DQN algorithm". In: *CAAI Transactions on Intelligence Technology* 5.3 (2020), S. 177–183. ISSN: 2468-2322. DOI: 10.1049/trit.2020.0024.
- [198] Shen Ying u. a. "3D building similarity for a random single-view-image pair based on a local 3D shape". In: *International Journal of Applied Earth Observation and Geoinformation* 123 (2023), S. 103467. ISSN: 15698432. DOI: 10.1016/j.jag.2023.103467.
- [199] Hexiang Yuan u. a. "Group-based multiple pipe routing method for aero-engine focusing on parallel layout". In: *Frontiers of Mechanical Engineering* 16.4 (2021), S. 798–813. ISSN: 2095-0233. DOI: 10.1007/s11465-021-0645-3.
- [200] Z. Zhu. *Automatic 3D Routing for the Physical Design of Electrical Wiring Interconnection Systems for Aircraft*. 2016. DOI: 10.4233/uuid:2ca107b4-202d-4638-a044-d45649b89275.
- [201] Dan Zhang u. a. "Multi-branch Cable Harness Layout Optimization Based on Pattern Search Improved Particle Swarm Optimization Algorithm". In: *Advances in Mechanical Design*. Hrsg. von Jianrong Tan. Bd. 111. Mechanisms and Machine Science. Singapore: Springer Nature Singapore, 2022, S. 1255–1273. ISBN: 978-981-16-7380-1. DOI: 10.1007/978-981-16-7381-8_78.
- [202] Dan Zhang u. a. "Multi-objective Layout Optimization of Aircraft Multi-branch Cable Harness Based on MOPSO/D". In: *2021 12th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE, 2021, S. 139–144. ISBN: 978-1-6654-3321-1. DOI: 10.1109/ICMAE52228.2021.9522568.
- [203] Kaiqing Zhang, Zhuoran Yang und Tamer Basar. "Networked Multi-Agent Reinforcement Learning in Continuous Spaces". In: *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, S. 2771–2776. ISBN: 978-1-5386-1395-5. DOI: 10.1109/CDC.2018.8619581.
- [204] Kaiqing Zhang, Zhuoran Yang und Tamer Başar. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*. 2019. DOI: 10.48550/arXiv.1911.10635.
- [205] Kaiqing Zhang u. a. *Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents*. 2018. DOI: 10.48550/arXiv.1802.08757.

- [206] Yingfeng Zhao u. a. “Multi-Branch Cable Harness Layout Design Based on Genetic Algorithm with Probabilistic Roadmap Method”. In: *Chinese Journal of Mechanical Engineering* 34.1 (2021). ISSN: 1000-9345. DOI: 10.1186/s10033-021-00544-9.
- [207] Zhi-Hua Zhou. *Machine Learning*. Singapore: Springer Singapore, 2021. ISBN: 978-981-15-1966-6. DOI: 10.1007/978-981-15-1967-3.
- [208] Changxi Zhu, Mehdi Dastani und Shihan Wang. *A Survey of Multi-Agent Reinforcement Learning with Communication*. 2022. DOI: 10.48550/arXiv.2203.08975.
- [209] Z. Zhu, G. La Rocca und M. J. L. van Tooren. “A methodology to enable automatic 3D routing of aircraft Electrical Wiring Interconnection System”. In: *CEAS Aeronautical Journal* 8.2 (2017), S. 287–302. ISSN: 1869-5582. DOI: 10.1007/s13272-017-0238-3.
- [210] Zhuangdi Zhu u. a. *Transfer Learning in Deep Reinforcement Learning: A Survey*. 2020. DOI: 10.48550/arXiv.2009.07888.
- [211] Fuzhen Zhuang u. a. *A Comprehensive Survey on Transfer Learning*. 2019. DOI: 10.48550/arXiv.1911.02685.